Constrained Clustering and Cognitive Decline Detection

Zhengdong Lu

A dissertation submitted to the faculty of the
OGI School of Science & Engineering
at Oregon Health & Sciences University
in partial fulfillment of the
requirements for the degree
Doctor of Philosophy
in
Computer Science and Engineering

June 2008

The dissertation "Constrained Clustering and Cognitive Decline Detection" by Zhengdong Lu has been examined and approved by the following Examination Committee:

Todd K. Leen
Professor
Thesis Research Adviser

Eric A. Wan
Associate Professor

Deniz Erdogmus
Assistant Professor

Andrew M. Fraser
Los Alamos National Lab

Dedication

My parents and my dear wife Haiming.

Acknowledgements

This thesis contains work I did during the years 2002-2007 at the OGI school of science and engineering, OHSU. During that period I received a lot of help from faculty, students and friends. However, I feel that I should start by thanking my parents Chuanzhong Lu and Huiyun Zhang, and my wife Haiming Zheng, for their support during my time at OGI. Similarly, I thank my master thesis advisor Pinfan Yan from Tsinghua University for helping me getting started with research in machine learning.

At OGI I received help from a number of people, most importantly my advisor Todd K. Leen. Todd helped me in many ways. He provided excellent technical hands-on assistance, as well as help on high-level and strategic issues. Working with Todd was a very inspirational and educational experience. It fundamentally changed the way I do research and turned me into a better researcher. I also thank Todd for giving me the freedom of wandering around many topics I am interested.

I also would like to thank Professor Miguel Á. Carreira-Perpiñán for being such a good teacher and friend to me. His generous help and constant encouragement have helped me put through the most difficult time of mine at OGI. His joke about deep-belief nets is priceless.

I thank my thesis committee members Eric A. Wan, Deniz Erdogmus and Andrew M. Fraser for their helpful comments. I benefited from interactions with several professors and graduate students at OGI. Rudolph van der Merwe, Xubo Song, Sergy Frolov, and Alex T. Nelson have helped me during different phases of my study in OGI. Despite not helping directly on topics related to the thesis, I benefited from discussion with Xiaochuan Niu, Tian Lan, Houwu Bai, Catherine Huang, Umut Ozertem, Andriy Myronenk, and Wensheng Guo. I would like to thank them all for helping to improve my understanding of machine learning and to write a better thesis.

Contents

List of Tables

List of Figures

Abstract

Constrained Clustering and Cognitive Decline Detection

Zhengdong Lu

Supervising Professor: Todd K. Leen

My thesis consists of two independents parts: (1) constrained clustering and (2) cognitive decline.

Constrained Clustering: While clustering is usually executed completely unsupervised, there are circumstances in which we have prior belief (with varying degrees of certainty) that pairs of samples should (or should not) be assigned to the same cluster. These pairwise constraints are less informative than direct labeling of the samples, but are often considerably easier to obtain. We proposed two probabilistic clustering algorithms to make use of this kind of pairwise constraints. Our first algorithm, called Penalized Probabilistic Clustering (PPC), is based on Gaussian mixture models (GMMs), where our belief on the pairwise constraints are expressed as a prior probability on the assignments of data points to clusters. Unlike previous effort in this direction, this clustering model naturally accommodates both hard constraints and soft preferences in a framework. Although PPC and its follow-up models are successful in many applications, they also suffer from their limited modeling capability and inefficiency on using the pairwise constraints. Our second clustering algorithm is specifically designed to address these two limitations. Instead of adapting a traditional clustering model, we started from the Gaussian process classifiers

(GPCs), a type of discriminative models carefully chosen for our specific constrained clustering requirement, and treated the pairwise relations as a special form of observation. The prior probability of the latent process is controlled with a kernel designed using the graph Laplacian of all the available data, thus we can make use of the samples that are not involved in pairwise relations.

Cognitive Decline Detection: We studied the approaches to the detection of decline in people's cognitive ability based on the longitudinal clinical observations. The ultimate goal is to evaluate a subject's risk of becoming cognitively impaired at different age, given his or her past clinical observations including motor ability and neuro-psychological test score. Our work consists of two strongly related parts. In the first part, we studied modeling a population of similar time series with mixed-effect models. This mixed-effect model does not only capture the group characteristic of different population, but also provides a means to learn an effective prior for individual time series modeling. The second part of our project is a cross-sectional study, where we try to predict whether a cognitively healthy subject will later develop into cognitive impairment. Towards this end, we first constructed a probabilistic classifier based on mixed-effect models trained separately on healthy and impaired populations, and demonstrated the gain of discriminative power by modeling the individual-specific random effects. To circumvent the shortcomings of the generative model-based classifier, we also considered discriminative approaches. We adopted the support vector machine (SVM) with kernels specially developed for longitudinal time series. We extended the design of Fisher kernel to take mixed-effect models as the generative model based on its hierarchical structure. In addition, we proposed a non-parametric distance measure for time series based on Gaussian processes (GPs) and reproducing kernel Hilbert space (RKHS). A Gaussian kernel based on this distance measure were also used in SVM. Experiments show that the discriminative approaches yield improved classification accuracy over generative models on four motor ability observations, while on neuro-psychological test scores the two schools of methods have comparably good classification performances.

Part I
Clustering with Pairwise Constraints

Chapter 1

Introduction to Constrained Clustering

Clustering is an unsupervised learning problem, which tries to group a set of points into clusters such that points in the same cluster are more similar to each other than points in different clusters, under a particular clustering distortion or distance measure [43]. Here, the learning algorithm just observes a set of points without observing any corresponding class/category labels. Clustering problems can also be categorized as generative or discriminative. In the generative clustering model, a parametric form of data generation is assumed, and the goal in the maximum likelihood formulation is to find the parameters that maximize the probability (likelihood) of generation of the data given the model. In the discriminative clustering setting (e.g., graph-theoretic clustering), the clustering algorithm tries to cluster the data so as to maximize within-cluster similarity and minimize between-cluster similarity, based on a similarity matrix defined over the input data set, it is not necessary to consider an underlying parametric data generation model. In both the generative and discriminative models, clustering algorithms are generally posed as optimization problems and solved by iterative methods like EM [24] or its approximation variants algorithms [44].

While clustering is usually executed completely unsupervised, there are circumstances in which we have prior belief (with varying degrees of certainty) that pairs of samples should (or should not) be assigned to the same cluster. More specifically, we define the following *pairwise relations*[1]:

[1] They are almost equivalent to the more popular "must-link" and "cannot-link" coined by Wagstaff et al. We deliberately choose a softer name for them since allowing the violation of the pairwise relations and thus incorporating the certainty information associated with them are a very important part of our work.

link: two sample should be assigned into one cluster

do-not-link: two samples should be assigned into different clusters,

with a pictorial demonstration in Figure 1.1. These pairwise relations are less informative than direct labeling of the samples, but are often considerably easier to obtain. Indeed, there are many occasions where the pairwise relations can be naturally derived from expert knowledge or common sense.

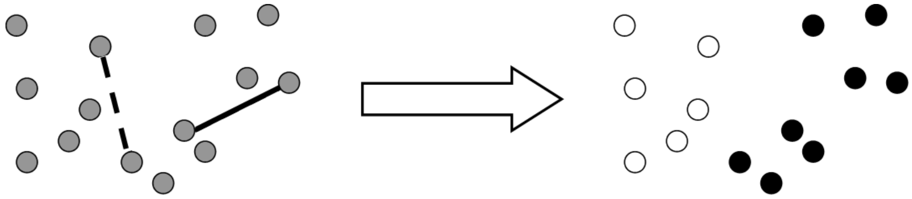


Figure 1.1: Example of clustering with pairwise relations. The panel left to the arrow shows the 2-dimensional data points, denoted as gray dots. The solid line is the a link constraint between the two samples connected by it and the dashed line is the do-not-link. The panel right to the arrow shows the data grouped into two classes after the two constraints are incorporated, with the group membership denoted by the color.

Our interest in such problems was kindled when we tried to manually segment a satellite image by grouping small image clips from the image. One finds that it is often hard to assign the image clips to different "groups" since we do not know clearly the characteristic of each group, or even how many classes we should have. In contrast, it is much easier to compare two image clips and to decide how much they look alike and thus how likely they should be in one cluster. Another interesting example is word sense disambiguation. Ambiguous words like 'plant', tend to exhibit only one meaning in one discourse [87]. In other words, two 'plant's in the same discourse probably should be assigned to the same class of sense. This fact is very useful in training unsupervised word sense disambiguation models. The third example is in information retrieval. Cohn et al. [19] suggested that in creating a document taxonomy, the expert critique is often in the form "these two documents shouldn't be in the same cluster". The last example is continuity, which suggests that neighboring pairs of samples in a time series or in an image are likely to belong to the same class of object, is also a source of clustering preferences [76, 2]. We would like these preferences to be incorporated into the cluster structure so that the assignment of out-of-sample data to clusters captures the concept(s) that give rise to the preferences expressed in the training data.

The effort of integrating pairwise relation information into the clustering algorithm fall into three general categories: the constraint-based, distance-based, and discrimination-based methods.

- The constraint-based methods, starting from the seminal work by Wagstaff et al. [82], typically adapt an existed clustering methods,such as K-means, to satisfy the constraints posed by the pairwise constraints. Based on the adapted clustering methods, there are constrained Kmeans [82, 81, 6, 4], constrained agglomerative clustering [46], and constrained Gaussian mixture model [70, 53, 89, 54, 51]. Depending on the way the constraints are incorporated, there are algorithms based on hard constraints [82, 6, 70], where the constraints must be satisfied by the clustering result, or the soft constraints [54, 51]. Among them, our work [53, 54] provides a flexible framework that can encompass many of the constrained clustering methods, including constrained Kmeans and Gaussian mixture model, with soft and hard constraints, as special cases.

- In distance-based approaches, an existing clustering algorithm that uses a particular distance to measure clustering distortion is employed; however, the distance measure is trained to satisfy the labels or constraints in the supervised data. Several adaptive distance measures have been used for semi-supervised clustering, including string-edit distance trained using Expectation Maximization (EM) [10], KL divergence trained using gradient descent [19], Mahalanobis distances trained using convex optimization [85, 22] or more efficient LDA-based techniques [8], etc. Many of the distance-based semi-supervised clustering formulations are based on a partitional clustering algorithm, e.g., K-Means, except some that study semi-supervised hierarchical clustering [46, 21] or spectral clustering [45].

- Quite recently, two discriminative model-based methods have been proposed. The general idea is to treat the pairwise relation as some kind of observation, and thus incorporate them as empirical error in the cost function for the discriminative model. Among them, Yan et al. [86] use support vector machine as the discriminative method and solve the whole problem with a convex optimization. Independently we [55] use Gaussian process classifiers and thus give a probabilistic model that can also handle the pairwise relations with significant uncertainty.

1.1 Roadmap

In this thesis, we give a detailed exposition of the two approaches we proposed for clustering with pairwise constraints. The first one, which we named as Penalized Probabilistic Clustering (PPC), is a constraint-based model adapted from Gaussian mixture model (GMM). The PPC model will be discussed in Chapter 2 and its connection to other constraint-based clustering models will be given in Chapter 3. Our second approach [55], named Semi-supervised Pairwise Gaussian Process classifier (SPGP), is specifically designed to address the limitations of the PPC model. Instead of adapting a traditional clustering model, we start from the Gaussian process classifiers (GPCs), a type of discriminative models carefully chosen for our specific constrained clustering requirement, and treat the pairwise relations as a special form of observation. This approach will be discussed in Chapter 4.

Notation

Following are some notation convention we will adopt through this part of thesis.

- We use $P(A)$ to denote the probability of discrete event A, and $p(x)$ as the continuous probability density at x.

- We use $P(A;\Theta)$ (or $p(x;\Theta)$) to denote the probability of A (density of x) with model parameter Θ. We use $P(A|B;\Theta)$ (or $p(x|B;\Theta)$) to denote the probability of A (density of x) conditioned on event B with model parameter Θ.

Chapter 2

Penalized Probabilistic Clustering

In this chapter we will describe our work on adapting the Gaussian mixture model (GMM) to integrate pairwise constraints [53, 54]. This model, named Penalized Probabilistic Clustering (PPC), can handle both hard and soft constraints in a unified model, and provides a fairly flexible framework that can encompass several other constrained clustering model as its particular cases.

2.1 Clustering with Gaussian Mixture Model

A Gaussian mixture model (GMM) with M components consists of the set of parameters $\Theta = \{\pi_1, \cdots, \pi_M, \theta_1, \cdots, \theta_M\}$. Here π_m gives the prior probability (mixing proportion) for the M Gaussian component, with

$$\pi_m \geq 0, \; \sum_{m=1}^{M} \pi_m = 1 \tag{2.1}$$

and $\theta_i = \{\mu_i, \Sigma_i\}$ is the parameters for the m^{th} Gaussian component. The generative process of any sample $\mathbf{x}$ from a Gaussian mixture model is the following: we first pick up a Gaussian component θ_m with prior probability π_m; then we draw $\mathbf{x}$ from the distribution $\mathcal{N}(\mu_i, \Sigma_i)$. Given Θ, the probability of $\mathbf{x}$ is

$$p(\mathbf{x};\, \Theta) \;\; = \;\; \sum_{m=1}^{M} \pi_m \, p(\mathbf{x};\, \theta_m). \tag{2.2}$$

Assuming the data set $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ is independently drawn from Θ-parameterized Gaussian mixture model, the distribution of X can be written as

$$\mathcal{L}(\Theta) \equiv p(X;\, \Theta) = \prod_{i=1}^{N} p(\mathbf{x}_i;\, \Theta) = \prod_{i=1}^{N} (\sum_{m=1}^{M} \pi_m \, p(\mathbf{x}_i;\, \theta_m)). \tag{2.3}$$

Here we introduce the latent variable $Z = \{z_1, z_2, \cdots, z_N\}$, with z_i indexing the Gaussian component $\mathbf{x}_i$ is generated from. We have the *complete* data likelihood

$$\begin{aligned} p(X, Z; \Theta) &= p(X\,|Z; \Theta)P(Z; \Theta) && (2.4) \\ &= \prod_{i=1}^{N} p(\mathbf{x}_i; \theta_{z_i}) \prod_{i=1}^{N} \pi_{z_i} && (2.5) \end{aligned}$$

where $P(Z; \Theta) = \prod_{i=1}^{N} \pi_{z_i}$ is the prior probability of latent variable Z. The data likelihood, as expressed in Equation (2.2), can also be obtained through the following marginalization

$$p(X; \Theta) = \sum_{Z} p(X, Z; \Theta), \tag{2.6}$$

where the summation is over all possible combination of Z.

We can fit a Gaussian mixture model to the data X by maximizing the data likelihood

$$\Theta^* = \arg\max_{\Theta} \mathcal{L}(\Theta) \tag{2.7}$$

This fitting process can be done with the expectation-maximization (EM) algorithm [24], which iterates between following two steps

$$\begin{aligned} &\texttt{E-step:} \quad Q(\Theta, \Theta^{(t-1)}) = E_{Z|X}(\log p(X, Z; \Theta)|X; \Theta^{(t-1)}) \\ &\texttt{M-step:} \quad \Theta^{(t)} = \arg\max_{\Theta} Q(\Theta, \Theta^{(t-1)}). \end{aligned}$$

until convergence. We can get a clustering out of a fit Gaussian mixture model by treating each Gaussian component as a cluster, and using the posterior probability

$$P(z_i = m\,|\mathbf{x}_i; \Theta) = \frac{p(\mathbf{x}_i, z_i = m\,|\mathbf{x}_i; \Theta)}{p(\mathbf{x}_i; \Theta)}, \tag{2.8}$$

as the *soft* membership of sample $\mathbf{x}_i$ to m^{th} cluster.

2.2 Penalized Data Likelihood

We start describing our model by revisiting the generative process of data set X, as in Section 2.1. Instead of sequentially choosing the generating component for each $\mathbf{x}_i$, we pick up a Z with probability

$$P(Z; \Theta) = \prod_{i=1}^{N} \pi_{z_i},$$

and set the generative Gaussian components for all the samples in X. We notice that each Z gives a unique realization of the cluster assignments for all samples. Base on this view, we can alter the generative process to make it closer to our prior belief the cluster assignments. We choose to do so by multiplying it with a weighting factor $g(Z)$, which is large when Z conforms all (or most of) pairwise constraints and small otherwise. After Z is chosen, the generative process is the same with the standard Gaussian mixture model.

In our work, we choose to let $g(Z)$ take the following form

$$g(Z) = \prod_{i \neq j} e^{\mathbf{W}_{ij}\ \delta(z_i, z_j)}, \tag{2.9}$$

where $\delta(\cdot, \cdot)$ is the Kronecker delta-function and $\mathbf{W}_{ij}$ is the weight associated with sample pair $(\mathbf{x}_i, \mathbf{x}_j)$. This weight satisfies

$$\mathbf{W}_{ij} \in [-\infty, \infty],\ \mathbf{W}_{ij} = \mathbf{W}_{ji}.$$

The weight $\mathbf{W}_{ij}$ reflects our preference for assigning $\mathbf{x}_i$ and $\mathbf{x}_j$ into one cluster.

We use positive $\mathbf{W}_{ij}$ when we prefer to assign $\mathbf{x}_i$ and $\mathbf{x}_j$ into one cluster (link), and negative $\mathbf{W}_{ij}$ when we prefer to assign them into different clusters (do-not-link). The absolute value $|\mathbf{W}_{ij}|$ reflects the strength of the preference. The prior probability with the pairwise relations is

$$P(Z; \Theta, \mathbf{W}) = \frac{1}{\Omega} \prod_{i} \pi_{z_i} \prod_{i \neq j} e^{\mathbf{W}_{ij}\ \delta(z_i, z_j)}, \tag{2.10}$$

where *Omega* is the normalization constant:

$$\Omega = \sum_{Z} \{ \prod_{i} \pi_{z_i} \prod_{i \neq j} e^{\mathbf{W}_{ij}\ \delta(z_i, z_j)} \}. \tag{2.11}$$

It appears that the $g(Z)$ in Equation (2.9) changes asymmetrically with the violation of link and do-not-link: When a link is *conformed*, $g(Z)$ increases; when a do-not-link is violated, $g(Z)$ decreases. Nevertheless, the prior probability given by Equation (2.10) decreases under both types of violations. We assume that the likelihood of the data, *given a specific cluster assignment* Z, is independent of the cluster assignment preferences:

$$p(X|Z;\ \Theta, \mathbf{W}) = p(X|Z;\ \Theta). \tag{2.12}$$

The complete likelihood can be defined as

$$p(X, Z;\, \Theta, \mathbf{W}) = P(Z;\, \Theta, \mathbf{W}) p(X|Z;\, \Theta) \tag{2.13}$$

and the data likelihood follows from

$$p(X;\, \Theta, \mathbf{W}) = \sum_{Z} p(X, Z;\, \Theta, \mathbf{W}). \tag{2.14}$$

The PPC model is clearly connected to the standard GMM and the constrained clustering model proposed by Shental et al [69]. We shall show that both models can be viewed as special cases of PPC with particular $\mathbf{W}$. The connection between PPC and other semi-supervised clustering models is less straightforward, and will be discussed in Chapter 3. If $\mathbf{W}_{ij} = 0$, we have no prior knowledge on the assignment relevancy of $\mathbf{x}_i$ and $\mathbf{x}_j$. When $\mathbf{W}_{ij} = 0$ for all pairs (i, j), we have $g(Z) = 1$, hence the complete likelihood reduces to the standard one:

$$p(X, Z;\, \Theta, \mathbf{W}) = \frac{1}{\Omega} p(X, Z; \Theta) g(Z) = p(X, Z; \Theta). \tag{2.15}$$

In the other extreme with $|\mathbf{W}_{ij}| \to \infty$, assignments Z that violate the pairwise relations between $\mathbf{x}_i$ and $\mathbf{x}_j$ have zero prior probability, since for those assignments

$$P(Z; \Theta, G) = \frac{\prod_k \pi_{z_k} \prod_{i \neq j} e^{\mathbf{W}_{ij}\ \delta(z_i, z_j)}}{\sum_Z \prod_l \pi_{z_l} \prod_{m \neq n} e^{\mathbf{W}^p_{mn}\ \delta(z_m, z_n)}} \to 0.$$

Then the relations become *hard constraints*, while the relations with $|\mathbf{W}_{ij}| < \infty$ are called *soft preferences*. When all the specified pairwise relations are hard constraints, the data likelihood becomes

$$p(X, Z; \Theta, G) = \frac{1}{\Omega} \prod_{ij \in \mathcal{L}} \delta(z_i, z_j) \prod_{ij \in \mathcal{N}} (1 - \delta(z_i, z_j)) \prod_{i=1}^{N} \pi_{z_i} p(\mathbf{x}_i |\, \theta_{z_i}), \tag{2.16}$$

where $\mathcal{L}$ is the set of linked sample pairs, and $\mathcal{N}$ is the set of do-not-link sample pairs. It is straightforward to verify that Equation (2.16) is essentially the same with the complete data likelihood given by [69]. Therefore the model proposed by Shental et al. [69] is equivalent to PPC with hard constraints. In Chapter 3, we give a detailed derivation of Equation (2.16) and the equivalence of two models. When only hard constraints are available, we simply implement PPC based on Equation (2.16). In the remainder of this chapter as well as throughout Chapter 3, we will use $\mathbf{W}$ to denote the prior knowledge on pairwise relations, that is

$$p(X, Z; \Theta, G) \equiv p(X, Z; \Theta, \mathbf{W}) = \frac{1}{\Omega} p(X, Z; \Theta) \prod_{i \neq j} e^{\mathbf{W}_{ij}\ \delta(z_i, z_j)} \tag{2.17}$$

2.3 Model Fitting

We use the EM algorithm [24] to fit the model parameters Θ:

$$\Theta^* = \arg\max_{\Theta} L(X; \Theta, \mathbf{W})$$

The expectation step (E-step) and maximization step (M-step) are

$$\begin{aligned} &\texttt{E-step:} \quad Q(\Theta, \Theta^{(t-1)}) = E_{Z|X}(\log p(X, Z; \Theta, \mathbf{W})|X; \Theta^{(t-1)}, \mathbf{W}) \\ &\texttt{M-step:} \quad \Theta^{(t)} = \arg\max_{\Theta} Q(\Theta, \Theta^{(t-1)}). \end{aligned}$$

In the M-step, the optimal mean and covariance matrix of each component is:

$$\mu_k = \frac{\sum_{j=1}^{N} \mathbf{x}_j P(k|\mathbf{x}_j; \Theta^{(t-1)}, \mathbf{W})}{\sum_{j=1}^{N} P(k|\mathbf{x}_j; \Theta^{(t-1)}, \mathbf{W})} \tag{2.18}$$

$$\Sigma_k = \frac{\sum_{j=1}^{N} P(k|\mathbf{x}_j; \Theta^{(t-1)}, \mathbf{W})(\mathbf{x}_j - \mu_k)(\mathbf{x}_j - \mu_k)^T}{\sum_{j=1}^{N} P(k|\mathbf{x}_j; \Theta^{(t-1)}, \mathbf{W})} . \tag{2.19}$$

The update of the prior probability of each component is more difficult due to the normalizing constant Ω in the data likelihood

$$\Omega = \sum_{Z} \{ \prod_{k=1}^{N} \pi_{z_k} \prod_{i \neq j} e^{\mathbf{W}_{ij}\ \delta(z_i, z_j)} \}. \tag{2.20}$$

We need to find

$$\pi \equiv \{\pi_1, \ldots, \pi_m\} = \arg\max_{\pi} \sum_{l=1}^{M} \sum_{i=1}^{N} \log \pi_l P(l|\mathbf{x}_i; \Theta^{(t-1)}, \mathbf{W}) - \log \Omega(\pi), \tag{2.21}$$

which, unfortunately, does not have a closed-form solution in general [1]. In this chapter, we use a rather crude approximation of the optimal π instead. First, we estimate the

[1] Shental et al. [69] pointed out that with a different sampling assumption, a closed-form solution for Equation (2.21) exists when only hard links are available. See Chapter 3.

values of $\log \Omega(\pi)$ on a grid $H = \{\hat{\pi}^n\}$ on the simplex defined by

$$\sum_{k=1}^{M} \pi_k = 1, \; \pi_k \geq 0.$$

Then in each M-step, we calculate the value of $\sum_{l=1}^{M} \sum_{i=1}^{N} \log \hat{\pi}_l^n P(l|\mathbf{x}_i; \Theta^{(t-1)}, \mathbf{W})$ for each node $\hat{\pi}^n \in H$ and find the node $\hat{\pi}^*$ that maximizes the function defined in Equation (2.21):

$$\hat{\pi}^* = \arg \max_{\hat{\pi}^n \in H} \sum_{l=1}^{M} \sum_{i=1}^{N} \log \hat{\pi}_l^n P(l|\mathbf{x}_i; \Theta^{(t-1)}, \mathbf{W}) - \log \Omega(\hat{\pi}^n). \tag{2.22}$$

We use $\hat{\pi}^*$ as the approximative solution of Equation (2.21). In this chapter, the resolution of the grid is set to be 0.01. Although it works very well for all experiments in this chapter, we notice that the search over grid will be fairly slow for $M > 5$. Shental et al. [69] proposed to find optimal π using gradient descent and approximate $\Omega(\pi)$ by pretending all specified relations are non-overlapping (see Section 2.6). Although this method is originally designed for hard constraints, it can be easily adapted for PPC.

It is critically important to note that with a non-trivial $\mathbf{W}$, the assignment independence is broken:

$$P(z_i, z_j|\mathbf{x}_i, \mathbf{x}_j; \Theta, \mathbf{W}) \neq P(z_i|\mathbf{x}_i; \Theta, \mathbf{W}) P(z_j|\mathbf{x}_j; \Theta, \mathbf{W}),$$

which means that the posterior estimation of each sample can not be done separately. This fact brings extra computational problem and will be discussed later in Section 2.6.

2.4 Penalized Probabilistic Clustering

Like for the standard Gaussian mixture model, we can get a clustering by using the posterior probability of samples

$$P(z_i = m\,|\mathbf{x}_i; \Theta, \mathbf{W}) = \frac{p(\mathbf{x}_i, z_i = m; \Theta, \mathbf{W})}{p(\mathbf{x}_i; \Theta, \mathbf{W})}, \tag{2.23}$$

as the soft membership to each cluster. Note that if $\mathbf{x}_i$ is linked or do-not-linked to any other sample, the posterior probability $P(z_i = m\,|\mathbf{x}_i; \Theta, \mathbf{W})$ will be decided by both the weight matrix $\mathbf{W}$ and the Gaussian mixture model parameters Θ. The calculation of

$P(z_i = m\,|\mathbf{x}_i;\,\Theta,\mathbf{W})$ can be quite expensive, as will be elucidated in Section 2.6. For any unconstrained sample $\mathbf{x}_i$, or equivalently $\mathbf{W}_{ij} = 0, \forall j$, we have

$$P(z_i = m\,|\mathbf{x}_i;\,\Theta,\mathbf{W}) \;=\; P(z_i = m\,|\mathbf{x}_i;\,\Theta) \tag{2.24}$$

$$= \frac{p(\mathbf{x}_i, z_i = m\,|\mathbf{x}_i;\,\Theta)}{p(\mathbf{x}_i;\Theta)} \tag{2.25}$$

which is essentially a inference with Gaussian mixture model and is solely decided by the Gaussian parameter Θ. To achieve a good clustering result on the unconstrained samples as well, we need the information from the pairwise constraints to be expressed in the Gaussian mixture model parameters. In PPC, this is done through the model fitting Equation (2.18), (2.19), and (2.21). As we will show in Section 2.5 and Section 2.7, PPC can successfully integrate the pairwise constraints information into the Gaussian mixture model parameters. The algorithm of PPC can be summarized as follows:

Penalized Probabilistic Clustering

Input: data set $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ where $\mathbf{x}_i \in \mathbb{R}^d$, $(i = 1, 2, \cdots, N)$, the pairwise constraints $\mathbf{W}\,(\in \mathbb{R}^{N\times N})$, and the number of cluster M.

step 1 Initialize the Gaussian mixture model parameters $\Theta = \{\pi_1, \mu_1, \Sigma_1, \cdots, \pi_M, \mu_M, \Sigma_M\}$, where π_k, μ_k and Σ_k are respectively the prior, mean and covariance of the k^{th} Gaussian component.

step 2 Let $\Theta^g = \Theta$ and calculate the $Q(\Theta, \Theta^g) = E_{Z|X}(\log p(X, Z; \Theta, \mathbf{W})|X; \Theta^g, \mathbf{W})$.

step 3 Find $\Theta = \arg\max_\Theta Q(\Theta, \Theta^g)$. The optimization of μ_k and Σ_k can be given in closed form

$$\mu_k = \frac{\sum_{j=1}^N \mathbf{x}_j P(k|\mathbf{x}_j; \Theta^{(t-1)}, \mathbf{W})}{\sum_{j=1}^N P(k|\mathbf{x}_j; \Theta^{(t-1)}, \mathbf{W})}$$

$$\Sigma_k = \frac{\sum_{j=1}^N P(k|\mathbf{x}_j; \Theta^{(t-1)}, \mathbf{W})(\mathbf{x}_j - \mu_k)(\mathbf{x}_j - \mu_k)^T}{\sum_{j=1}^N P(k|\mathbf{x}_j; \Theta^{(t-1)}, \mathbf{W})}.$$

For π, find the node $\hat{\pi}^*$ in the grid $H = \{\hat{\pi}^n\}$ on the simplex: $\sum_{k=1}^M \pi_k = 1,\ \pi_k \geq 0$, so that

$$\hat{\pi}^* = \arg\max_{\hat{\pi}^n \in H} \sum_{l=1}^M \sum_{i=1}^N \log \hat{\pi}_l^n P(l|\mathbf{x}_i; \Theta^{(t-1)}, \mathbf{W}) - \log \Omega(\hat{\pi}^n).$$

where $\Omega = \sum_Z \{\prod_{k=1}^N \pi_{z_k} \prod_{i\neq j} e^{\mathbf{W}_{ij}\ \delta(z_i,z_j)}\}$. Let $\pi = \hat{\pi}^*$.

step 4 Go back to step 2 until convergence.

step 5 For $i = 1, 2, \cdots, N$, estimate the posterior probability of label

$$P(z_i = m\,|\mathbf{x}_i;\,\Theta, \mathbf{W}) = \frac{p(\mathbf{x}_i, z_i = m;\,\Theta, \mathbf{W})}{p(\mathbf{x}_i;\Theta, \mathbf{W})},$$

and assign the sample to the most probable cluster.

2.5 Setting the Weight for Pairwise Constraints

The weight matrix $\mathbf{W}$ is crucial to the performance of the PPC. As we have elucidated in Section 2.2, we can move from standard unsupervised Gaussian mixture model to hard-constrained PPC only by changing the weight matrix. In this section we will fist demonstrate the transformation from Gaussian mixture model to PPC when we gradually increase the weight on constraints. Then in the rest of this section, we will discuss a heuristic of deriving the weight from prior knowledge.

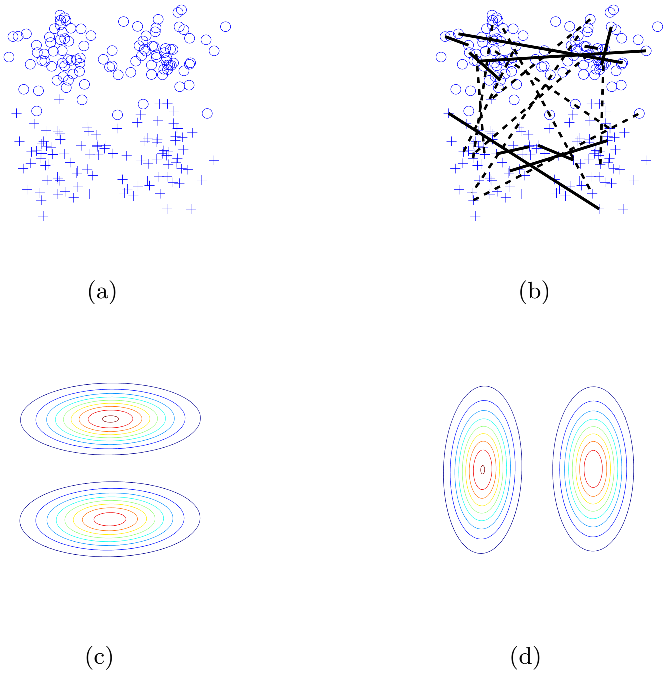
2.5.1 Example: How the Weight W Affects Clustering

Here we give an example demonstrating how the weight of pairwise relations affects the clustering process. Figure 2.1 (a) shows the 2-dimensional data from *two* classes, as indicated by the symbols. Besides the data set, we also have 20 pairs correctly labeled as links and do-not-links, as shown in Figure 2.1 (b). We try to fit the data set with a two-component GMM. Figure 2.1 (c) and (d) give the density contour of the two possible models on the data. Without any pairwise relations specified, we have essentially equal chance to get each. After incorporating pairwise relations, the EM optimization process is biased to the correct one. The weights of pairwise relations are given as follows

$$\mathbf{W}_{ij} = \begin{cases} w & \text{if } (\mathbf{x}_i, \mathbf{x}_j)\text{is linked} \\ -w & \text{if}(\mathbf{x}_i, \mathbf{x}_j)\text{is do-not-linked} \\ 0 & \text{otherwise,} \end{cases}$$

where $w \geq 0$ measures the certainty of all specified pairwise relations. In Figure 2.2, we give three runs with *same* initial model parameters but different weight given to the specified pairwise relations.

For each run, we give snapshots of the model after 1, 3, 5 and 20 EM iterations. The first row is the run with $w = 0$ (standard GMM). The search ends up with a model that violates our prior knowledge of class membership. The middle row is the run with w set to 1.3, with the same poor initial condition, the model fitting process still goes to the wrong one again, although at a slower pace. In the bottom row, we increase w to 3, this time the model converges to the one we intend.


(a)

(b)

(c)

(d)

Figure 2.1: The influence of constraint weight on model fitting. (a) artificial data set. (b) links (solid lines) and do-not-links (dotted line). (c) and (d): the probability density contour of two possible fitted models.

2.5.2 Choosing Weight W Based on Prior Knowledge

There are some occasions where we can translate our prior belief on the relations into the weight **W**. Here we assume that the pairwise relations are labeled by an oracle but

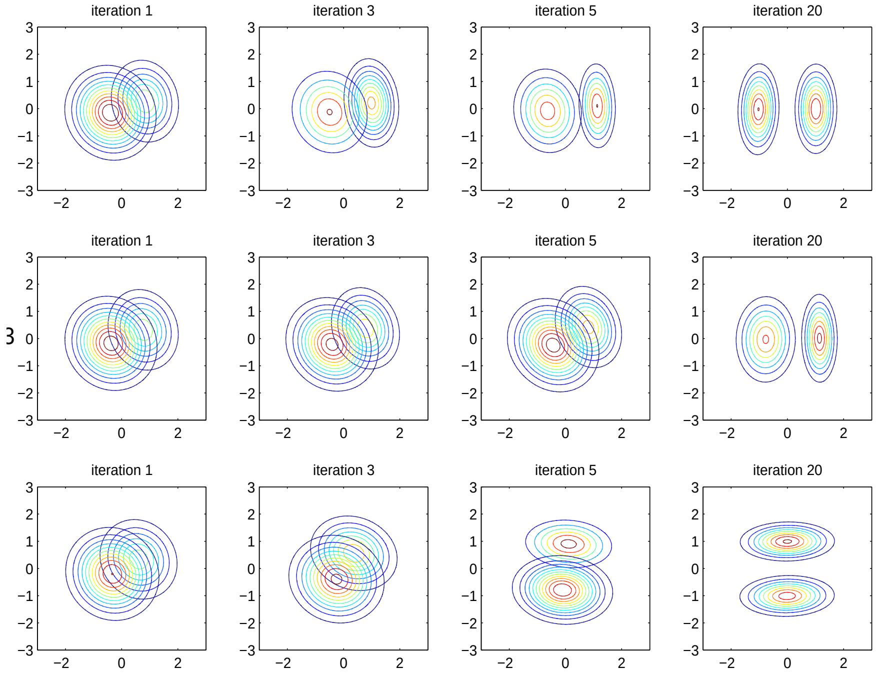


Figure 2.2: The contour of probability density fit on data with different weight given to pairwise relations. Top row: $w = 0$; Middle row: w=1.3; Bottom row: $w = 3$.

contaminated by flipping noise before they are delivered to us. For each labeled pair $(\mathbf{x}_i, \mathbf{x}_j)$, there is thus a certainty value $0.5 \leq \gamma_{ij} \leq 1$ equal to the probability that pairwise relation is *not* flipped, i.e. that label is correct [2]. Our prior knowledge would include those specified pairwise relations and their certainty values $\Gamma = \{\gamma_{ij}\}$.

This prior knowledge can be *approximately* encoded into the weight $\mathbf{W}$ by letting

$$\mathbf{W}_{ij} = \begin{cases} \frac{1}{2}\log(\frac{\gamma_{ij}}{1-\gamma_{ij}}) & (\mathbf{x}_i, \mathbf{x}_j) \text{ is specified as linked} \\ -\frac{1}{2}\log(\frac{\gamma_{ij}}{1-\gamma_{ij}}) & (\mathbf{x}_i, \mathbf{x}_j) \text{ is specified as do-not-linked} \\ 0 & \text{otherwise.} \end{cases} \tag{2.26}$$

The details of the derivation are in Appendix A1. It is obvious from Equation (2.26) that for a specified pairwise relation $(\mathbf{x}_i, \mathbf{x}_j)$, the greater the certainty value γ_{ij}, the greater

[2] We only consider the certainty value > 0.5, because a pairwise relation with certainty $\gamma_{ij} < 0.5$ can be equivalently treated as its opposite relation with certainty $1 - \gamma_{ij}$.

the absolute value of weight $\mathbf{W}_{ij}$.

Note that the weight designed this way is not necessarily optimal in terms of classification accuracy, as will be demonstrated by experiment in Section 2.7.1. The reason is twofold. First, Equation (2.26) is derived based on a (possibly crude) approximation. Second, Gaussian mixture models, as classifiers, are often considerably biased from true class distribution of data. As a result, even if the PPC prior $P(Z;\,\Theta,\mathbf{W})$ faithfully reflects the truth, it does not necessarily lead to the best classification accuracy. Nevertheless, Equation (2.26) gives a good initial guidance for choosing the weight. Our experiments in Section 2.7.1 show that this design often yields superior classification accuracy than simply using the hard constraints or ignoring the pairwise relations (standard GMM).

One use for this scheme of weight is when pairwise relations are labeled by domain experts and the certainty values are given at the same time. We might also *estimate* the flipping noise parameters from historical data or from available statistics. For example, we can derive soft pairwise relations based on spatial or temporal continuity among samples. That is, we add soft links to all adjacent pair of samples, assuming the flipping noise explaining all the adjacent pairs that are actually *not* in one class. We further assume that the flipping noise each pair follows the same distribution. Accordingly we assign uniform weight $w > 0$ to all adjacent pairs. Let q denote the probability that the label on a adjacent pair is flipped. We might be able to estimate q from labeled instances of a similar problem, for example, segmented images or time series. The maximum likelihood (ML) estimation of q is given by simple statistics:

$$\tilde{q} = \frac{\text{the number of adjacent pairs that are not in the same class}}{\text{the number of all adjacent pairs}}.$$

We give an application of this idea in Section 2.7.2.

2.6 Computational Issue

The M-step requires the cluster membership posterior. Computing this posterior is simple for the standard GMM since each data point $\mathbf{x}_i$ can be assigned to a cluster independent of the other data points and we have the familiar cluster origin posterior $P(z_i = k|\mathbf{x}_i;\Theta)$. The pairwise constraints bring extra relevancy in assignment among samples involved. For example, from Equation (2.17), if $\mathbf{W}_{ij} \neq 0$, $P(z_i, z_j|\mathbf{x}_i, \mathbf{x}_j; \Theta, \mathbf{W}) \neq P(z_i|\mathbf{x}_i; \Theta, \mathbf{W})P(z_j|\mathbf{x}_j; \Theta, \mathbf{W})$. Consequently, the posterior probability of $\mathbf{x}_i$ and $\mathbf{x}_j$ cannot be estimated separately. To model the relevancy caused indirectly by the pairwise constraints, we define the following relation:

Definition If $\mathbf{W}_{ij} \neq 0$, we say there is *direct assignment relevancy* between $\mathbf{x}_i$ and $\mathbf{x}_j$, denoted by $\mathbf{x}_i R_d \mathbf{x}_j$. If $P(z_i, z_j|\mathbf{x}_i, \mathbf{x}_j; \Theta, \mathbf{W}) \neq P(z_i|\mathbf{x}_i; \Theta, \mathbf{W})P(z_j|\mathbf{x}_j; \Theta, \mathbf{W})$, we say there is *assignment relevancy* between $\mathbf{x}_i$ and $\mathbf{x}_j$, denoted by $\mathbf{x}_i R_a \mathbf{x}_j$.

It is clear that R_a is reflexive, symmetric and transitive. Hence, R_a is an equivalence relation. It can be shown that R_a is the transitive closure of R_d. In other words, two samples have assignment relevancy relation R_a if they can be connected by a path consisting of R_a relations, as illustrated in Figure 2.3. We call each equivalence class associated with R_a a clique. It is clear that cliques are the smallest sets of samples whose posterior probabilities can be calculated independently. When calculating posterior probabilities, all samples within a clique need to be considered together. In a clique T with size $|T|$, the posterior probability of a given sample $\mathbf{x}_i \in T$ is calculated by marginalizing the posterior over the entire clique

$$P(z_i = k|X; \Theta, \mathbf{W}) = \sum_{Z_T|z_i=k} P(Z_T|X_T; \Theta, \mathbf{W}),$$

with the posterior on the clique given by

$$P(Z_T|X_T; \Theta, \mathbf{W}) = \frac{p(Z_T, X_T; \Theta, \mathbf{W})}{p(X_T; \Theta, \mathbf{W})} = \frac{p(Z_T, X_T; \Theta, \mathbf{W})}{\sum_{Z'_T} p(Z'_T, X_T; \Theta, \mathbf{W})}.$$

Exact calculation of the posterior probability of a sample in clique T requires time complexity $O(M^{|T|})$, where M is the number of components in the mixture model. This calculation can get prohibitively expensive if $|T|$ is very big (e.g., 50) for any model size $M \geq 2$. Hence small size cliques are required to make the marginalization computationally reasonable.

2.6.1 Two Special Cases with Easy Inference

Apparently the inference is easy when we limit ourselves to small cliques. Specifically, when $|T| \leq 2$, the pairwise relations are *non-overlapping*, as illustrated in Figure 2.4 (a)

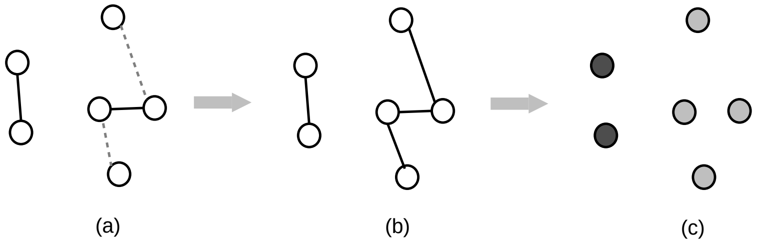


Figure 2.3: The example of assignment relevancy. (a) Links (solid line) and do-not-links (dotted line) among six samples; (b) Direct assignment relevancy R_d (solid line) translated from links in (a); (c) Equivalence classes defined by assignment relevancy R_a, denoted by color.

and (b). With non-overlapping constraints, the posterior probability for the whole data set can be given in closed-form with $O(N)$ time complexity. Moreover, the evaluation of the normalization factor $\Omega(\pi)$ is simple:

$$\Omega(\pi) = (1 - \sum_{k=1}^{M} \pi_k^2)^{N_L} (\sum_{k=1}^{M} \pi_k^2)^{N_N},$$

where N_L and N_N are respectively the number of links and do-not-links. The optimization of π in M-step can thus be achieved with little cost. Sometimes non-overlapping relations are a natural choice: they can be generated by picking up sample pairs from sample set and labeling the relations *without replacement.* More generally, we can avoid the expensive computation in posterior inference by breaking large clique into small ones. To do this, we need to deliberately ignore some links or do-not-links. In Section 2.7.2, Experiment 3 is an application of this idea.

The second simplifying situation is we have only hard links ($\mathbf{W}_{ij} = +\infty$ or 0), as illustrated in Figure 2.4 (c). In this case, the posterior probability for each sample must be exactly the same with the others in the same clique, so a clique can be treated as a single sample. That is, assume $\mathbf{x}_i$ is in clique T, we then have

$$\begin{aligned} P(z_i = k|\mathbf{x}_i; \Theta, \mathbf{W}) &= P(Z_T = k|X_T; \Theta, \mathbf{W}) \\ &= \frac{p(X_T, Z_T = k; \Theta, \mathbf{W})}{\sum_k' p(X_T, Z_T = k'; \Theta, \mathbf{W})} \\ &= \frac{p(X_T, Z_T = k; \Theta)}{\sum_k' p(X_T, Z_T = k'; \Theta)} \\ &= \frac{\prod_{j \in T} \pi_k p(\mathbf{x}_j; \theta_k)}{\sum_{k'} (\prod_{j \in T} \pi_{k'} p(\mathbf{x}_j; \theta_{k'}))}. \end{aligned}$$

Similar ideas have been proposed independently by several authors [82, 69, 9]. This case is useful when we are sure that a group of samples are from one source [69].

For more general cases, where the exact inference is computationally prohibitive, we propose to use Gibbs sampling [62] and the mean field approximation [38] to estimate the posterior probability. This will be discussed in Section 2.6.2 and 2.6.3.

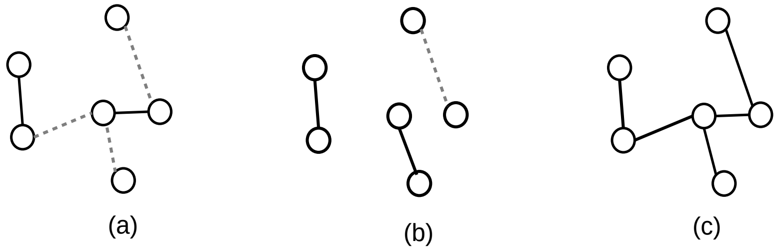


Figure 2.4: Examples of overlapping relations and non-overlapping relations.(a) Overlapping pairwise relations, with links (solid line) and do-not-links (dotted line); (b) Non-overlapping pairwise relations; (c) Only hard links

2.6.2 Estimation with Gibbs Sampling

For fixed Θ, finding $P(Z; \Theta, \mathbf{W})$ is a typical inference problem for graphical models. Techniques for approximate inference developed for graphical models can also be used here. In this section, we use Gibbs sampling to estimate the posterior probability in each EM iteration.

In Gibbs sampling, we estimate $P(z_i|X; \Theta, \mathbf{W})$ as a sample mean

$$P(z_i = k|X; \Theta, \mathbf{W}) = E(\delta(z_i, k)|X; \Theta, \mathbf{W}) \approx \frac{1}{S} \sum_{t=1}^{S} \delta(z_i^{(t)}, k),$$

where the sum is over a sequence of S samples from $P(Z|X;\Theta,G)$ generated by the Gibbs MCMC. The t^{th} sample in the sequence is generated by the usual Gibbs sampling technique:

- Pick $z_1^{(t)}$ from distribution $P(z_1|z_2^{(t-1)}, z_3^{(t-1)}, ..., z_N^{(t-1)}, X; \mathbf{W}, \Theta)$
- Pick $z_2^{(t)}$ from distribution $P(z_2|z_1^{(t)}, z_3^{(t-1)}, ..., z_N^{(t-1)}, X; \mathbf{W}, \Theta)$
  ...
- Pick $z_N^{(t)}$ from distribution $P(z_N|z_1^{(t)}, z_2^{(t)}, ..., z_{N-1}^{(t)}, X; \mathbf{W}, \Theta)$

For pairwise relations it is helpful to introduce some notation. Let Z_{-i} denote an assignment of data points to clusters that leaves out the assignment of $\mathbf{x}_i$. Let $U(i)$ be the indices of the set of samples that participate in a pairwise relation with sample $\mathbf{x}_i$, $U(i) = \{j : \mathbf{W}_{ij} \neq 0\}$. Then we have

$$P(z_i|Z_{-i}, X; \Theta, \mathbf{W}) \propto p(\mathbf{x}_i, z_i; \Theta) \prod_{j \in U(i)} e^{2\mathbf{W}_{ij}\, \delta(z_i, z_j)}. \tag{2.27}$$

The time complexity of each Gibbs sampling pass is $O(NnM)$, where n is the maximum number of pairwise relations a sample can be involved in. When $\mathbf{W}$ is sparse, the size of $U(i)$ is small, thus calculating $P(z_i|Z_{-i}, X; \Theta, \mathbf{W})$ is fairly cheap and Gibbs sampling can effectively estimate the posterior probability.

2.6.3 Estimation with Mean Field Approximation

Another approach to posterior estimation is to use mean field theory [38, 50]. Instead of directly evaluating the intractable $P(Z|X;\Theta,\mathbf{W})$, we try to find a tractable mean field approximation $Q(Z)$. To find a $Q(Z)$ close to the true posterior probability $P(Z|X;\Theta,\mathbf{W})$, we minimize the Kullback-Leibler divergence between them, i.e.

$$\min_Q \mathrm{KL}(Q(Z)|P(Z|X;\Theta,\mathbf{W})), \tag{2.28}$$

which can be recasted into:

$$\max_Q [H(Q) + E_Q\{\log P(Z|X;\Theta,\mathbf{W})\}], \tag{2.29}$$

where $E_Q\{\cdot\}$ denotes the expectation with respect to Q. The simplest family of variational distribution is one where all the latent variables $\{z_i\}$ are independent of each other:

$$Q(Z) = \prod_{i=1}^{N} Q_i(z_i). \tag{2.30}$$

With this $Q(Z)$, the optimization problem in Equation (2.29) does not have a closed-form solution, nor is it a convex problem. Instead, a locally optimal Q can be found iteratively with the following update equations

$$Q_i(z_i) \leftarrow \frac{1}{\Omega_i} e^{E_Q\{\log P(Z|X;\Theta,\mathbf{W})|z_i\}} \tag{2.31}$$

for all i and $z_i \in \{1, 2, \cdots, M\}$. Here $\Omega_i = \sum_{z_i} e^{E_Q\{\log P(Z|X,\Theta,\mathbf{W})|z_i\}}$ is the local normalization constant. For the PPC model, we have

$$e^{E_Q\{\log P(Z|X;\Theta,\mathbf{W})|z_i\}} = P(z_i|\mathbf{x}_i;\Theta)e^{\sum_{j\neq i}\mathbf{W}_{ij}Q_j(z_i)}.$$

Equation (2.31), collectively for all i, are the *mean field equations.* Evaluation of mean field equations requires at most $O(NnM)$ time complexity, which is same as the time complexity of one Gibbs sampling pass. Successive updates of Equation (2.31) will converge to a local optimum of Equation (2.29). In our experiments, the convergence usually occurs after about 20 iterations, which is much less than the number of passes required for Gibbs sampling.

2.7 Experiments

The experiments section consists of two parts. In Section 2.7.1, we examine the way the number of constraints affects the clustering results. For each clustering task in this section, we generate artificial pairwise relations based on class labels. In Section 2.7.2, we address real-world problems, where the constraints are derived from our prior knowledge. Also in this section, we demonstrate the approaches to reduce computational complexity, as described in Section 2.6.

Following are some abbreviations we will use throughout this section: *soft-PPC* is PPC with soft constraints, *hard-PPC* is PPC with hard constraints (implemented based on Equation (2.16)), *soft-CKmeans* is the K-means with soft constraints [6] and *hard-CKmeans* is the K-means with hard constraints [82]. The Gaussian mixture model with hard constraints [69, 70] will be referred to as constrained-EM.

2.7.1 Artificial Constraints

In this section, we discuss the influence of pairwise relations on PPC's clustering, and compare the result to other semi-supervised clustering models. This section includes two experiments. In Experiment 1, we consider only correct pairwise relations, as an example of authentic knowledge. Accordingly, we use hard constraints in clustering. In Experiment 2, we discuss the situation where pairwise relations contain significant error. We evaluate the performance of soft-PPC and test the weight design strategy described in Section 2.5. The result is compared to hard-PPC and other semi-supervised clustering models.

Constraint Selection: To avoid the computational burden, we will limit our discussion to the non-overlapping pairwise relations in Experiment 1 and 2. As discussed in Section 2.6, the non-overlapping pairwise relations, hard or soft, allows fast solution in the maximization step in each EM iteration. The pairwise relations are generated as follows: we randomly pick two samples from the *training* set without replacement. If the two have the same class label, we then add a link constraint between them; otherwise, we add a do-not-link constraint. Note that the application of PPC is not limited to the non-overlapping cases. In Section 2.7.2, we will discuss more complicated real-world problems where overlapping constraints are necessary, and we also will present approaches to solve the computational problems.

Performance Evaluation: We try PPC (with the number of components equal to the number of classes) with various numbers of pairwise relations. For each clustering result, a confusion matrix is built to compare it to true labeling. The classification accuracy is calculated as the ratio of the sum of diagonal elements to the number of all samples.

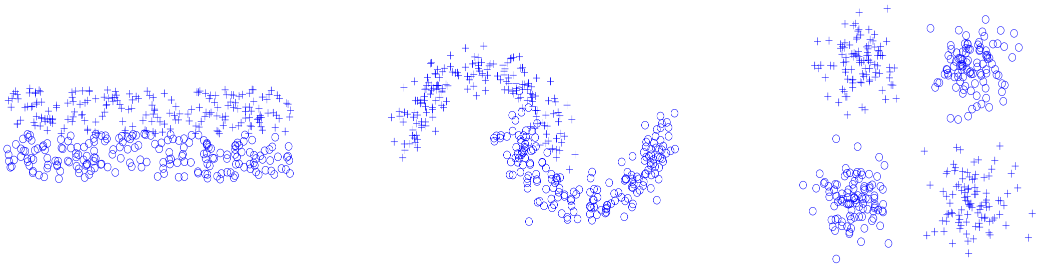
Experiment 1: Artificial Hard Constraints

This experiment is designed to answer two questions: How do the number of pairwise relations affect the clustering result, and whether has the information in the relations been successfully encoded into the trained model. To answer the second question, we examine the out-of-sample classification of the Gaussian mixture model fit with the aid of the pairwise relations. Towards this end, we divide each data set into training set (90% of data) and a held-out test set (10% of data). Pairwise relations are generated among samples in the training set. After the density model is fit on the training set and the pairwise relations, it will be applied to the test set. Since the classification on test set is merely decided by the fit Gaussian mixture model, it will reflect the influence of pairwise relations on the trained model. For comparison, we also give results of two other constrained clustering methods: (1) the hard-CKmeans [82], for which the accuracy on test set is given by the nearest-neighbor classification with the cluster centers fit on training set, and (2) constrained-EM [70]. As we will show in Chapter 3 that constrained-EM can come with two different sampling assumptions, and when using sampling Assumption 2 constrained-EM is equivalent to hard-PPC. Therefore we only need to consider the constrained-EM with sampling Assumption 1. The reported classification accuracy is averaged over 100 different realizations of pairwise relations.

The three two-dimensional artificial data sets shown in Figure 2.5 are designed to highlight PPC's superior modeling flexibility over constrained K-means[3]. In each example, there are 200 samples in each class. It is clear from Figure 2.5 that for all three problems, data in each class are non-Gaussian. So not surprisingly, standard K-means and GMM do not return satisfactory clustering results. Figure 2.6 compares the clustering result of hard-PPC and hard-CKmeans with various number of pairwise relations. As shown in Figure 2.6, the accuracy of hard-PPC improves significantly when pairwise relations are incorporated. After enough number of pairwise relations are added in, we can finally reach close to 100% accuracy on the training data. On the test set, although no pairwise relation is available, we observe significantly improved accuracy as well. For the hard-CKmeans,

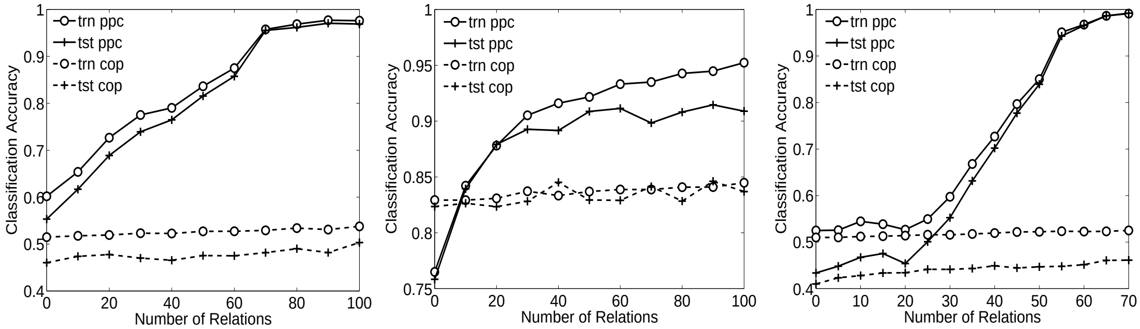
[3] Basu et al. [9] combined standard or constrained K-means with metric learning based on pairwise relations, and reported improvement on classification accuracy. This will not be discussed in this thesis.

we do not observe any substantial accuracy improvement on both training set and test set. The classification task of data set 1 is relatively easy for Gaussian mixture and difficult for K-means. The classification accuracy of hard-PPC climbs from 60% to close 100% (on both training and test set) after 70 pairwise relations, whereas the accuracy of hard-CKmeans remains less than 60% even with 100 relations. On data set 2, the hard-PPC accuracy is improved from 75% to close 95% on training set and stops at around 90% on test set. This divergence happens because the two classes in data set 2 are overlapped, and thus defies a perfect GMM classifier. Data set 3 is the most difficult since it is highly non-Gaussian. It takes over 100 pairs for the hard-PPC to reach 95% accuracy, whereas hard-CKmeans never reaches 55%.



(a) data set 1 (b) data set 2 (c) data set 3

Figure 2.5: The artificial data sets. Classes are denoted by symbols.



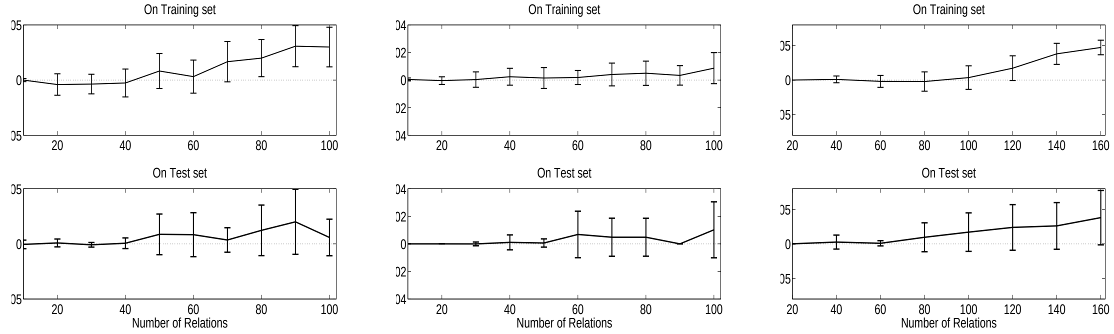
(a) On data set 1 (b) On data set 2 (c) On data set 3

Figure 2.6: The performance of hard-PPC and hard-CKmeans with various number of relations. **trn ppc**: accuracy on training set with hard-PPC; **tst ppc**: accuracy on test set from the GMM trained by hard-PPC; **trn cop**: accuracy on training set with hard-CKmeans; **tst cop**: accuracy of K-means on test set;

The comparison of PPC and constrained-EM is presented in a way to highlight the difference between the classification accuracy of the two methods. Basically, we record the classification from PPC and constrained-EM with same pairwise relations and initial condition, and then calculate

$$\Delta\text{Accuracy} = \text{classification accuracy by PPC} - \text{classification accuracy by CEM}$$

on both training and test set. In Figure 2.7, we report the mean and standard deviation of ΔAccuracy estimated over 100 different realizations of pairwise relations. From Figure 2.7, the difference between PPC and constrained-EM is indistinguishable when the number of relations is small, while PPC is slightly but consistently better than constrained-EM when the relations are abundant.



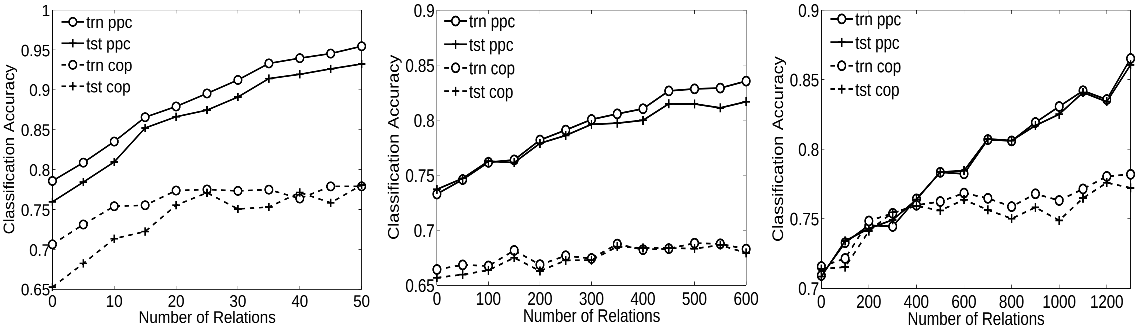
(a) data set 1

(b) data set 2

(c) data set 3

Figure 2.7: Comparison of PPC and constrained EM on artificial data. With each number of pairwise relations, we show the mean of ΔAccuracy ± standard deviation estimated over 100 random realization of pairwise relations.

We perform the same experiments on three UCI data sets: the Iris data set has 150 samples and three classes, 50 samples in each class; the Waveform data set has 5000 samples and three classes, around 1700 samples in each class; the Pendigits data set includes four classes (digits 0, 6, 8, 9), each with 750 samples. The results are summarized in Figure 2.8 and Figure 2.9. As indicated by Figure 2.8, hard-PPC can consistently improve its clustering accuracy on the training set when more pairwise constraints are added; also, the effect brought by constraints generalizes to the test set. In contrast, as in the artificial data set case, the increase of accuracy from hard-CKmeans is much less salient than that of hard-PPC. Figure 2.9 shows that hard-PPC is slightly better than constrained-EM, especially when the number of constraints is large.

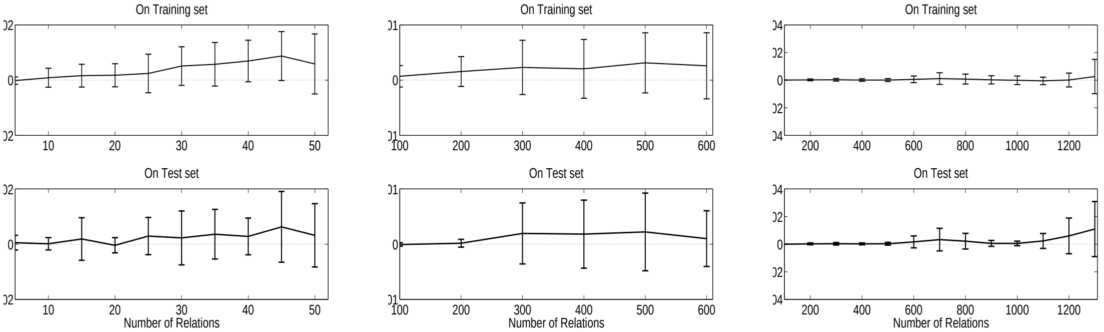


(a) Iris data

(b) Waveform data

(c) Pendigits data

Figure 2.8: The performance of PPC on UCI data sets with various number of relations.



(a) Iris

(b) Waveform

(c) Pendigits

Figure 2.9: Comparison of PPC and constrained EM on UCI data sets. With each number of pairwise relations, we show the mean of ΔAccuracy ± standard deviation estimated over 100 random realization of pairwise relations.

Experiment 2: Artificial Soft Constraints

In this experiment, we evaluate the performance of soft-PPC when the specified pairwise relations contain substantial error. The results are compared to hard-PPC, soft-CKmeans, and hard-CKmeans.

The artificial constraints are generated the same way as in the previous experiment. The flipping noise is realized by randomly flipping each pairwise relation with a certain probability $q \leq 0.5$. For the soft-PPC model, the weight $\mathbf{W}_{ij}$ to each specified pairwise relation is given as follows:

$$\mathbf{W}_{ij} = \begin{cases} \frac{1}{2}\log(\frac{1-q}{q}) & (z_i, z_j) \text{ specified as link} \\ -\frac{1}{2}\log(\frac{1-q}{q}) & (z_i, z_j) \text{ specified as do-not-link.} \end{cases} \tag{2.32}$$

We use w to denote the absolute value of the weight for non-trivial pairs. For soft-PPC, we have $w = \frac{1}{2}\log(\frac{1-q}{q})$. For soft-CKmeans, we give equal weights to all the specified constraints. Because there is no guiding rule in literature on how to choose weight for soft-CKmeans model, we simply use the weight that yields the highest classification accuracy.

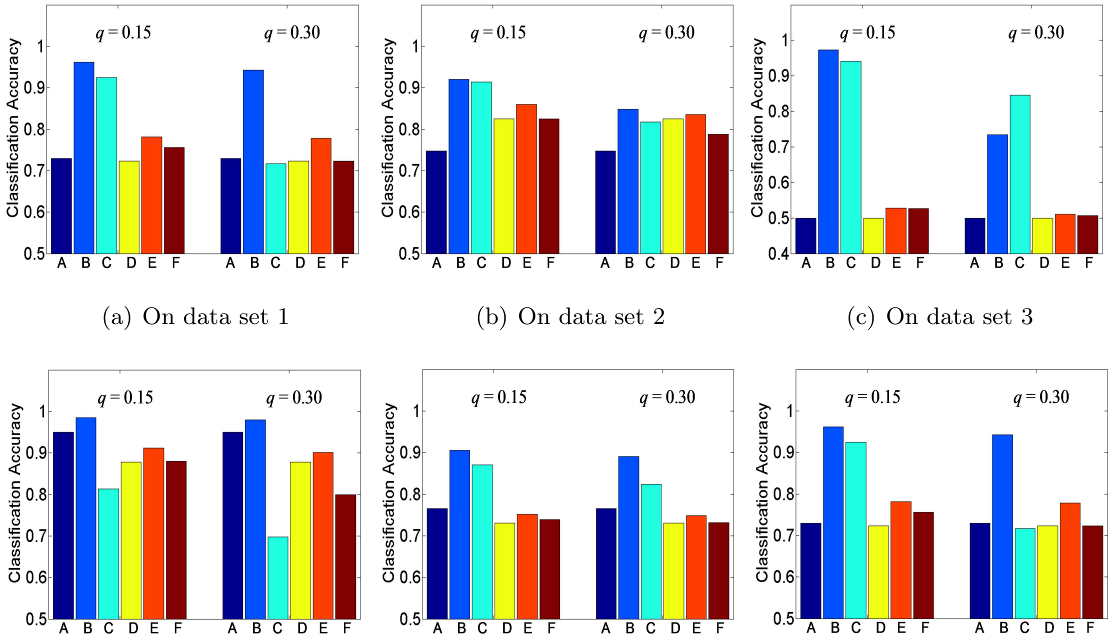
We present the results on the three artificial data sets and three UCI data sets used in the Experiment 1. Unlike Experiment 1, we use all the available in clustering. On each data set, we randomly generate a number of non-overlapping pairwise relations to have 50% of the data involved. In this experiment, we try two different noise levels with q set to 0.15 and 0.3. Figure 2.10 compares the classification accuracies given by the maximum likelihood (ML) solutions[4] of different models. The accuracy for each model is averaged over 20 random realizations of pairwise relations. On all data sets except artificial data set 3, soft-PPC with the designed weight gives higher accuracy than hard-PPC ($w = \infty$) and standard GMM ($w = 0$) on both noise levels. On artificial data set 3 , when $q = 0.3$ hard-PPC gives the best classification accuracy [5]. Soft-PPC apparently gives superior classification accuracy to the K-means models on all six data sets, even though the weight of soft-CKmeans is optimized. Figure 2.10 also shows that it can be harmful to use hard constraints when pairwise relations are noisy, especially when the noise is significant. Indeed, as shown by Figure 2.10 (d) and (f), hard-PPC can yield accuracy even worse than standard GMM.

2.7.2 Real World Problems

In this section, we present two examples where pairwise constraints are from domain experts or common sense. Both examples are about image segmentation based on Gaussian

[4] We choose the one with the highest data likelihood among 100 runs with different random initialization. For K-means models, including soft-CKmeans and hard-CKmeans, we use the solutions with the smallest value of cost function.

[5] Further experiment shows that on this data, soft-PPC with the optimal w ($>$ the one suggested by Equation (2.32)) is still slightly better than hard-PPC.



(a) On data set 1

(b) On data set 2

(c) On data set 3

(d) On Iris data

(e) On Waveform data

(f) On Pendigits data

Figure 2.10: Classification accuracy with noisy pairwise relations. We use all the data in clustering. In each panel, **A**: standard GMM; **B**: soft-PPC; **C**: hard-PPC; **D**: standard K-means; **E**: soft-CKmeans with optimal weight; **F**: hard-CKmeans.

mixture models. In the first problem (Experiment 3), hard pairwise relations are derived from image labeling done by a domain expert. In the second problem, soft pairwise relations are generated based on spatial continuity.

Experiment 3: Hard Do-not-links from Partial Class Information

The experiment in this subsection shows the application of pairwise constraints on partial class information. For example, consider a problem with six classes $A, B, ..., F$. The classes are grouped into several class-sets $C_1 = \{A, B, C\}, C_2 = \{D, E\}, C_3 = \{F\}$. The samples are partially labeled in the sense that we are told which class-set a sample is from, but not which specific class it is from. We can logically derive a do-not-link constraint between any pair of samples known to belong to different class-sets, while no link constraint can be derived if each class-set has more than one class in it.

Figure 2.11 (a) is a 120 × 400 (pixels) region from Greenland ice sheet from NASA

Langley DAAC [6] [74]. Each pixel has intensities from seven spectrum bands. This region is labeled into snow area and non-snow area, as indicated in Figure 2.11 (b). The snow area may contain samples from several classes of interest: ice, melting snow and dry snow, while the non-snow area can be bare land, water or cloud. The labeling from expert contains incomplete but useful information for further segmentation of the image. To segment the image, we first divide it into 5x5x7 blocks (175 dim vectors). We use the first 50 principal components as feature vectors. Our goal is then to segment the image into (typically > 2) areas by clustering those feature vectors. With PPC, we can encode the partial class information into do-not-links.

For hard-PPC, we use half of the data samples for training, and the rest for test. Hard do-not-link constraints (only on training set) are generated as follows: for each block in the non-snow area, we randomly choose (without replacement) six blocks from the snow area to build do-not-link constraints. By doing this, we achieve cliques with size seven (1 non-snow block + 6 snow blocks). As in Section 2.7.1, we apply the model fit with hard-PPC to the test set and combine the clustering results on both data sets into a complete picture. Clearly, the clustering task is non-trivial for any $M > 2$. A typical clustering result of 3-component standard GMM and 3-component PPC are shown as Figure 2.11 (c) and (d) respectively. Standard GMM gives a clustering that is clearly in disagreement with the human labeling in Figure 2.11 (b). The hard-PPC segmentation makes far fewer mis-assignments of snow areas (tagged white and gray) to non-snow (black) than does the GMM. The hard-PPC segmentation properly labels almost all of the non-snow regions as non-snow. Furthermore, the segmentation of the snow areas into the two classes (not labeled) tagged white and gray in Figure 2.11 (d) reflects subtle differences in the snow regions captured by the gray-scale image from spectral channel 1, as shown in Figure 2.11 (a).

[6] We use the first seven MoDerate Resolution Imaging Spectroradiometer (MODIS) Channels with bandwidths as follows (in nm): Channel 1: 620-670, Channel 2: 841-876, Channel 3: 459-479, Channel 4: 545-565, Channel 5: 1230-1250, Channel 6: 1628-1652, Channel 7:2105-2155

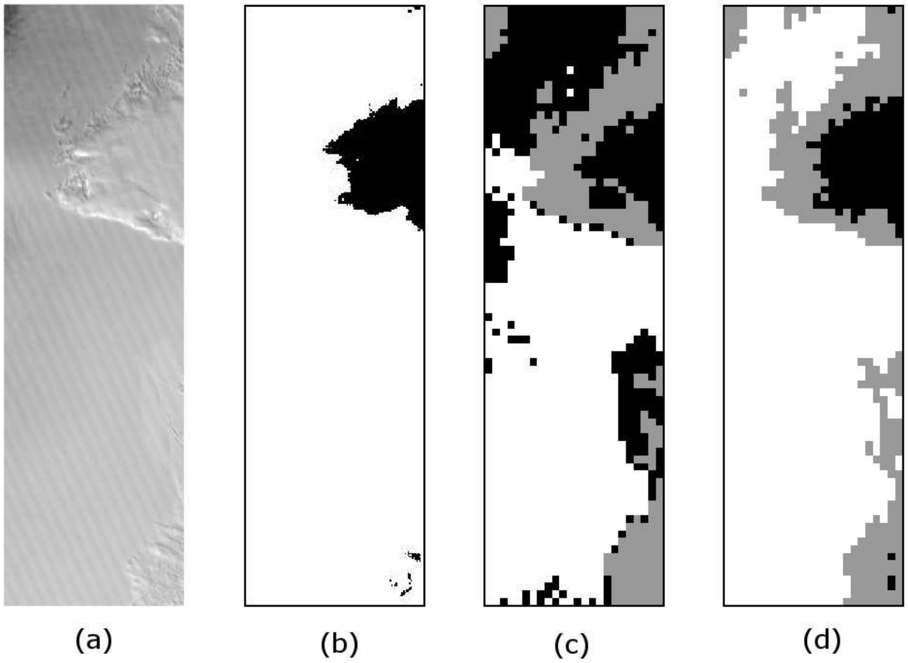


Figure 2.11: Clustering with hard do-not-links derived from partial labeling. (a) Gray-scale image from the first spectral channel 1. (b) Partial label given by expert, black pixels denote non-snow area and white pixels denote snow area. Clustering result of standard GMM (c) and PPC (d). (c) and (d) are colored according to image blocks' assignment.

Experiment 4: Soft Links from Continuity

In this subsection, we will present an example where soft constraints come from continuity. As in the previous experiment, we try to do image segmentation based on clustering. The image is divided into blocks and rearranged into feature vectors. We use a GMM to model those feature vectors, with each Gaussian component representing one texture. However, standard GMM often fails to give good segmentations because it cannot make use of the spatial continuity of image, which is essential in many image segmentation models, such as random field [13]. In our algorithm, the spatial continuity is incorporated as the soft link preferences with uniform weight between each block and its neighbors. As described in Section 2.5, the weight w of the soft link can be given as

$$w = \frac{1}{2}\log(\frac{1-q}{q}), \tag{2.33}$$

where q is the ratio of softly-linked adjacent pairs that are not in the same class. Usually q is given by an expert or estimated from segmentation result of similar images. In this experiment, we assume we already know the ratio q, which is calculated from the label of the image.

The *complete* data likelihood is

$$p(X, Z; \Theta, \mathbf{W}) = \frac{1}{\Omega} p(X, Z; \Theta) \prod_{i} \prod_{j \in U(i)} e^{w\, \delta(z_i, z_j)}, \tag{2.34}$$

where $U(i)$ means the neighbors of the i^{th} block. The EM algorithm can be roughly interpreted as iterating on two steps: (1) estimating the texture description (parameters of mixture model) based on segmentation, and (2) segmenting the image based on the texture description given by step 1. Since exact calculation of the posterior probability is intractable due to the large clique containing all samples, we have to resort to approximation methods. In this experiment, both the Gibbs sampling (see Section 2.6.2) and the mean field approximation (see Section 2.6.3) are used for posterior estimation. For Gibbs sampling, Equation (2.27) is reduced to

$$P(z_i | Z_{-i}, X; \Theta, \mathbf{W}) \propto p(\mathbf{x}_i, z_i; \Theta) \prod_{j \in U(i)} e^{2w\, \delta(z_i, z_j)}.$$

The mean field Equation (2.31) is reduced to

$$Q_i(z_i) \leftarrow \frac{1}{\Omega_i} p(\mathbf{x}_i, z_i; \Theta) \prod_{j \in U(i)} e^{2w\, Q_j(z_i)}.$$

The image shown in Figure 2.12 (a) is built from four Brodatz textures [7] . This image is divided into 7x7 blocks and then rearranged to 49-dim vectors. We use those vectors' first five principal components as the associated feature vectors. A typical clustering result of 4-component standard GMM is shown in Figure 2.12 (b). For soft-PPC, the soft links with weight w calculated from Equation (2.33) are added between each block and its four neighbors. Figure 2.12 (c) and (d) are the clustering result of 4-component soft-PPC with respectively Gibbs sampling and mean field approximation. One run with Gibbs sampling takes around 160 minutes on a PC with Pentium 4, 2.0 GHZ processor whereas

[7] Downloaded from http://sipi.usc.edu/services/database/Database.html, April, 2004.

the algorithm using the mean field approximation takes only 3.1 minutes. Although mean field approximation is about 50 times faster than Gibbs sampling, the clustering result are comparable according to Figure 2.12. Comparing to the result given by standard GMM, soft-PPC with both approximation methods achieves significantly better segmentation after incorporating spatial continuity.

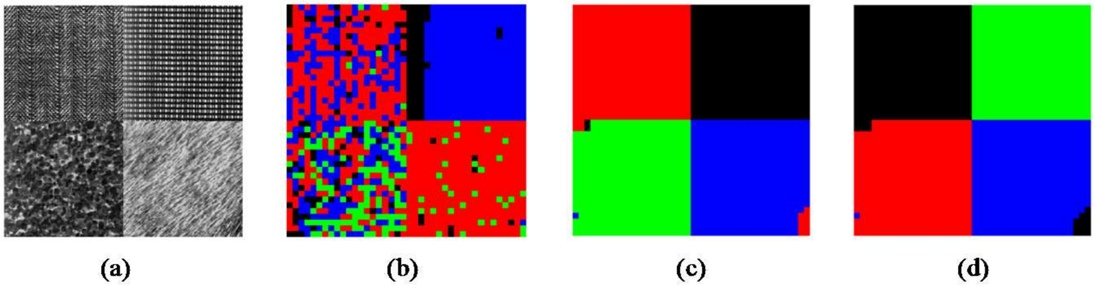


Figure 2.12: Clustering of texture data with soft links derived from the spatial continuity. (a) Texture combination. (b) Clustering result of standard GMM. (c) Clustering result of soft-PPC with Gibbs sampling. (d) Clustering result of soft-PPC with mean field approximation. (b)-(d) are shaded according to the blocks assignments to clusters.

2.8 Discussion

We have proposed a probabilistic clustering model that incorporates prior knowledge in the form of pairwise relations between samples. Unlike previous work in semi-supervised clustering, our model formulates clustering preferences as a Bayesian prior over the assignment of data points to clusters, and so naturally accommodates both hard constraints and soft preferences. Unlike many semi-supervised learning methods [75, 90, 92] addressing labeled subset, PPC returns a fitted parametric density model and thus can deal with unseen data. Experiments on different data sets have shown that pairwise relations can consistently improve the performance of the clustering process.

Despite its success, PPC has its own limitations. First, PPC often needs a substantial *proportion* of samples involved in pairwise relations to give good results. Indeed, if we have the number of relations fixed and keep adding samples without any new relations, the algorithm will finally degenerate into unsupervised learning (clustering). To overcome this, one can instead build semi-supervised model based on discriminative models such as neural network or Gaussian process classifier, and use the pairwise relations in the form of hint [71] or observation [92]. Second, since PPC is based on the Gaussian mixture model, it works well in the situation where the data in each class can be approximated by a Gaussian distribution. When this condition is not satisfied, PPC could lead to poor results. One way to alleviate this situation is to use multiple clusters to model one class, an interesting direction for future exploration. Third, in choosing the weight matrix $\mathbf{W}$, although our design works well on some data sets, it is not clear how to set the weight for a more general situation.

In the work discussed in this thesis, we implement hard constraints using Equation (2.16). Alternatively, we can approximate hard constraints by using large $|\mathbf{W}_{ij}|$ for every constrained pair $(\mathbf{x}_i, \mathbf{x}_j)$. Indeed, from Equation (2.10), when a constraint with large weight is violated in assignments Z, the prior probability $P(Z; \Theta, \mathbf{W})$ will be close to zero. The value of $P(Z; \Theta, \mathbf{W})$ with such a Z can be made arbitrarily small by increasing the corresponding weight. This is convenient when we want to model soft and hard relations at the same time. This situation is not covered here, but remains an interesting direction for future exploration.

To address the computational difficulty caused by large cliques, we propose two approximation methods: Gibbs sampling and mean field approximation. We also observe Gibbs sampling can be fairly slow for large cliques. One way to address this problem is to use fewer sampling rounds (and thus a cruder approximate inference) in the early phase of EM training, and gradually increase the number of sampling rounds (and a finer approximation) when EM is close to convergence. By doing this, we may be able to achieve a much faster algorithm without sacrificing too much precision. For the mean field approximation, the bias brought by the independence assumption among $Q_i(\cdot)$ could be severe for some problems. We can ameliorate this, as suggested by Jaakkola [38], by retaining more sub-structure of the original graphical model (for PPC, it is expressed in $\mathbf{W}$), while still keeping the computation tractable.

Chapter 3

From PPC to Other Constrained Clustering Models

Prior to our work on penalized probabilistic clustering, different authors have proposed several constrained clustering models based on K-means, including the seminal work by Wagstaff and colleagues [82, 81], and its successor [5, 6, 9]. These models generally fall into two classes. The first class of algorithms [82, 5] keep the original K-means cost function (reconstruction error) while confining the cluster assignments to be consistent with the specified pairwise relations. The problem can be cast into the following constrained optimization problem

$$\begin{aligned} \min_{Z,\mu} \quad & \sum_{i=1}^{N} ||\mathbf{x}_i - \mu_{z_i}||^2 \\ \text{subject to} \quad & z_i = z_j, \text{ if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{L} \\ & z_i \neq z_j, \text{ if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{N}, \end{aligned}$$

where $\mu = \{\mu_1, \cdots, \mu_M\}$ is the cluster centers. In the second class of algorithms, cluster assignments that violate the pairwise relations are allowed, but will be penalized. They employ a modified cost function [6]:

$$J(\mu, Z) = \frac{1}{2}\sum_{i=1}^{N} ||\mathbf{x}_i - \mu_{z_i}||^2 + \sum_{(i,j)\in\mathcal{L}} a_{ij}(z_i \neq z_j) + \sum_{(i,j)\in\mathcal{N}} b_{ij}(z_i = z_j), \qquad (3.1)$$

where a_{ij} is the penalty for violating the link between $(\mathbf{x}_i, \mathbf{x}_j)$ and b_{ij} is the penalty when the violated pairwise relation is a do-not-link. It can be shown that both classes of algorithms are encompassed by PPC as special cases. The particular PPC model we consider has spherical Gaussian components with radius shrunk to zero and the weight matrix **W** expands properly. The detail are in Section 3.2 and Section 3.3.

One weakness shared by the semi-supervised K-means algorithms is the limited capability of K-means to model complex data distribution. If data in one class is far from being spherical, it may take a great number of pairwise relations to achieve reasonable classification accuracy [81]. Another serious problem lies in the optimization strategy employed by those algorithms to find the optimal assignment within each EM iteration. Due to the extra dependency brought by the pairwise relations, finding the optimal assignment of samples to clusters is not trivial. Evaluating every potential assignments requires $O(M^{|T|})$ time complexity where $|T|$ denotes the size of the biggest clique, which is prohibitively expensive when $|T|$ is big. The greedy search used by these algorithms can only return local optima [5, 6], and the sequential assignment strategy employed by Wagstaff et al. [82] may lead to the situation where one cannot assign a sample to any cluster because of the conflict with some assigned samples.

To remedy the limited capability of constrained K-means, several authors proposed probabilistic models based on Gaussian mixture models. The models proposed by Shental et al. [69, 70] address the situation where pairwise relations are hard constraints. The authors partition the whole data set into a number of (maximal) "chunklets" consisting of samples that are (hard) linked to each other[1]. Shental et al. [69, 70] discuss two sampling assumptions:

- Assumption 1: chunklet X_i is generated i.i.d from component k with prior π_k [70], and the complete data likelihood is

$$p(X, Y; \Theta, E_\Omega) = \frac{1}{\Omega} \prod_{i \neq j \in \mathcal{N}} (1 - \delta(z_i, z_j)) \cdot \prod_{l=1}^{L} \{\pi_{z_l} \prod_{\mathbf{x}_i \in X_l} p(\mathbf{x}_i | \theta_{z_l})\}, \tag{3.2}$$

  where E_Ω denotes the specified constraints.

- Assumption 2: chunklet X_i generated from component k with prior $\propto \pi_k^{|X_i|}$, where

[1] If a sample is not linked to any other samples, it comprises a chunklet by itself.

$|X_i|$ is the number of samples in X_i [70]. The complete data likelihood is:

$$\begin{aligned} p(X,Y;\Theta,E_\Omega) &= \frac{1}{\Omega}\prod_{ij\in\mathcal{N}}(1-\delta(z_i,z_j))\cdot\prod_{l=1}^{L}\{\pi_{z_l}^{|X_l|}\prod_{\mathbf{x}_i\in X_l}p(\mathbf{x}_i|\theta_{z_l})\} \qquad (3.3)\\ &= \frac{1}{\Omega}\prod_{ij\in\mathcal{L}}\delta(z_i,z_j)\prod_{ij\in\mathcal{N}}(1-\delta(z_i,z_j))\prod_{i=1}^{N}\pi_{z_i}p(\mathbf{x}_i|\,\theta_{z_i}). \qquad (3.4)\end{aligned}$$

In Section 3.1 we show that when using Assumption 2, this model (as expressed in Equation (3.3)-(3.4)) is equivalent to the PPC with only hard constraints (as expressed in Equation (2.16)). It is suggested [70] that Assumption 1 might be appropriate, for example, when chunklets are generated from temporal continuity. When pairwise relations are generated by labeling sample pairs picked from data set, Assumption 2 might be more reasonable. Assumption 1 allows a closed-form solution in the M-step (including solution for π) in each EM iteration [70]. The empirical comparison of the two sampling assumptions is discussed in Section 2.7.1.

To incorporate the uncertainty associated with pairwise relations, Law et al. [51, 52] proposed to use soft group constraints. To model a link between any sample pair $(\mathbf{x}_i, \mathbf{x}_j)$, they create a group l and express the strength of the link as the membership of $\mathbf{x}_i$ and $\mathbf{x}_j$ to group l. This strategy works well for some simple situations, for example, when the pairwise relations are non-overlapping (as defined in Section 2.6.1). However, it is awkward if samples are shared by multiple groups, which is unavoidable when samples are commonly involved in multiple relations. Another serious drawback of the group constraints model is its inability to model do-not-links.

3.1 From PPC to Constrained GMM

In this section, we prove that when $|\mathbf{W}_{ij}| \rightarrow \infty$ for each specified pair $(\mathbf{x}_i, \mathbf{x}_j)$, the complete likelihood of PPC can written as in Equation (2.16), and thus equivalent to the model proposed by Shental et al. [69].

In the model proposed by Shental et al. [69], the complete likelihood is written as :

$$\begin{aligned} p(X,Z;\Theta,E_\Omega) &= \frac{1}{\Omega}\prod_{c_i}\delta_{y_{c_i}}\prod_{a_i^1\neq a_i^2}(1-\delta(y_{a_i^1},y_{a_i^2}))\prod_{i=1}^{N}P(z_i;\Theta)p(\mathbf{x}_i|z_i;\Theta) \\ &= \frac{1}{\Omega}\prod_{c_i}\delta_{y_{c_i}}\prod_{a_i^1\neq a_i^2}(1-\delta(y_{a_i^1},y_{a_i^2}))p(X,Z;\Theta) \end{aligned}$$

where E_Ω stands for the pairwise constraints, $\delta_{y_{c_i}}$ is 1 iff all the points in the chunklet (the clique of samples connected with only hard links) c_i have the same label, (a_i^1,a_i^2) is the index of the sample pair with hard do-not-link between them. This is equivalent to

$$p(X,Z;\Theta,E_\Omega) = \begin{cases} \frac{1}{\Omega}p(X,Z;\Theta) & Z \text{ satisfies all the constraints;} \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

In the corresponding PPC model with hard constraints, we have

$$\mathbf{W}_{ij} = \begin{cases} +\infty & i \text{ and } j \text{ is linked} \\ -\infty & i \text{ and } j \text{ is do-not-linked} \\ 0 & \text{no relation} \end{cases} \tag{3.6}$$

According to Equation (2.12) and (3.5), to prove

$$p(X,Z;\Theta,E_\Omega) = p(X,Z;\Theta,\mathbf{W})$$

we only need to prove

$$P(Z;\Theta,\mathbf{W}) = 0,$$

for all the Z that violate the constraints, that is

$$P(Z;\Theta,\mathbf{W}) = \frac{\prod_k \pi_{z_k}\prod_{i\neq j}e^{\mathbf{W}_{ij}\;\delta(z_i,z_j)}}{\sum_Z\prod_l \pi_{z_l}\prod_{m\neq n}e^{W_{mn}^p\;\delta(z_m,z_n)}} = 0.$$

First let us assume Z violates one link between pair (α,β) ($\mathbf{W}_{\alpha\beta}=+\infty$), we have

$$z_\alpha \neq z_\beta \Rightarrow \delta(z_\alpha,z_\beta)=0 \Rightarrow e^{W_{\alpha\beta}^p\;\delta(z_\alpha,z_\beta)} = 1.$$

We assume the constraints are consistent. In other words, there is at least one Z that satisfies all the constraints. We can denote one such Z by Z^*. We also assume each component has a positive prior probability. It is straightforward to show that

$$P(Z^*;\Theta,\mathbf{W}) > 0.$$

Then it is easy to show

$$\begin{aligned}
P(Z;\Theta,\mathbf{W}) &= \frac{\prod_k \pi_{z_k} \prod_{i\neq j} e^{\mathbf{W}_{ij}\ \delta(z_i,z_j)}}{\sum_Z \prod_l \pi_{z_l} \prod_{m\neq n} e^{\mathbf{W}_{mn}\ \delta(z_m,z_n)}} \\
&\leq \frac{\prod_k \pi_{z_k} \prod_{i\neq j} e^{\mathbf{W}_{ij}\ \delta(z_i,z_j)}}{\prod_k \pi_{z_k^*} \prod_{i\neq j} e^{\mathbf{W}_{mn}\ \delta(z_i^*,z_j^*)}} \\
&= (\prod_k \frac{\pi_{z_k}}{\pi_{z_k^*}} \prod_{(i,j)\neq(\alpha,\beta)} \frac{e^{\mathbf{W}_{ij}\ \delta(z_i,z_j)}}{e^{\mathbf{W}_{ij}\ \delta(z_i^*,z_j^*)}}) \frac{e^{2\mathbf{W}_{\alpha\beta}\ \delta(z_\alpha,z_\beta)}}{e^{2\mathbf{W}_{\alpha\beta}\ \delta(z_\alpha^*,z_\beta^*)}} \\
&= (\prod_k \frac{\pi_{z_k}}{\pi_{z_k^*}} \prod_{(i,j)\neq(\alpha,\beta)} \frac{e^{\mathbf{W}_{ij}\ \delta(z_i,z_j)}}{e^{\mathbf{W}_{ij}\ \delta(z_i^*,z_j^*)}}) \frac{1}{e^{2\mathbf{W}_{\alpha\beta}\ \delta(z_\alpha^*,z_\beta^*)}}
\end{aligned}$$

Since Z^* satisfies all the constraints, we must have

$$\prod_{(i,j)\neq(\alpha,\beta)} \frac{e^{\mathbf{W}_{ij}\ \delta(z_i,z_j)}}{e^{\mathbf{W}_{ij}\ \delta(z_i^*,z_j^*)}} \leq 1.$$

So we have

$$P(Z;\Theta,\mathbf{W}) \quad \leq \quad (\prod_k \frac{\pi_{z_k}}{\pi_{z_k^*}}) \frac{1}{e^{2\mathbf{W}_{\alpha\beta}\ \delta(z_\alpha^*,z_\beta^*)}}.$$

When

$$\mathbf{W}_{\alpha\beta} \rightarrow +\infty,$$

we have

$$\frac{1}{e^{2\mathbf{W}_{\alpha\beta}\ \delta(z_\alpha^*,z_\beta^*)}} \rightarrow 0$$

and then

$$P(Z;\Theta,\mathbf{W}) \quad \leq \quad (\prod_k \frac{\pi_{z_k}}{\pi_{z_k^*}}) \frac{1}{e^{2\mathbf{W}_{\alpha\beta}\ \delta(z_\alpha^*,z_\beta^*)}} \rightarrow 0. \tag{3.7}$$

The do-not-link case can be prove in similar way. ■

3.2 From PPC to K-means with soft constraints

The adopted cost function for K-means with soft constraints is:

$$J(\mu, Z) \quad = \quad \frac{1}{2}\sum_{i=1}^{N} ||\mathbf{x}_i - \mu_{z_i}||^2 + \sum_{(i,j)\in\mathcal{L}} a_{ij}(z_i \neq z_j) + \sum_{(i,j)\in\mathcal{N}} b_{ij}(z_i = z_j) \tag{3.8}$$

where μ_k is the center of the k^{th} cluster. Equation (3.1) can be rewritten as

$$J(\mu, Z) \quad = \quad \frac{1}{2}\sum_{i=1}^{N} ||\mathbf{x}_i - \mu_{z_i}||^2 - \sum_{ij} \mathbf{W}_{ij}\delta(z_i, z_j) + C, \tag{3.9}$$

with $C = -\sum_{(i,j)\in\mathcal{L}} a_{ij}$ is a constant and

$$\mathbf{W}_{ij} = \begin{cases} a_{ij} & (i,j) \in \mathcal{L} \\ -b_{ij} & (i,j) \in \mathcal{N} \\ 0 & \text{otherwise.} \end{cases} \tag{3.10}$$

The clustering process includes minimizing the cost function $J(\mu, Z)$ over both the model parameters $\mu = \{\mu_1, \mu_2, ..., \mu_M\}$ and cluster assignment $Z = \{z_1, z_2, ..., z_N\}$. The optimization is usually done iteratively with modified Linde-Buzo-Gray (LBG) algorithm. Assume we have the PPC model with the matrix $\mathbf{W}$ same as in Equation (3.9). We further constrain each Gaussian component to be spherical with radius σ. The complete data likelihood for PPC model is

$$p(X, Z; \Theta, \mathbf{W}) = \frac{1}{\Omega}\prod_{i=1}^{N}\{\pi_{z_i} e^{-\sum_{i=1}^{N}\frac{||\mathbf{x}_i - \mu_{z_i}||^2}{2\sigma^2}}\}\prod_{mn} e^{\mathbf{W}_{mn}\delta(z_m, z_n)}, \tag{3.11}$$

where Ω is the normalizing constant and μ_k is the mean of the k^{th} Gaussian component. To build its connection to the cost function in Equation (3.9), we consider the following scaling:

$$\sigma \rightarrow \alpha\sigma, \quad \mathbf{W}_{ij} \rightarrow \mathbf{W}_{ij}/\alpha^2. \tag{3.12}$$

The complete data likelihood with the scaling parameters α is

$$p(X, Z; \Theta, \mathbf{W}, \alpha) = \frac{1}{\Omega(\alpha)}\prod_{i=1}^{N}\{\pi_{z_i} e^{-\sum_{i=1}^{N}\frac{||\mathbf{x}_i - \mu_{z_i}||^2}{2\alpha^2\sigma^2}}\}\prod_{mn} e^{\frac{\mathbf{W}_{mn}}{\alpha^2}\delta(z_m, z_n)}. \tag{3.13}$$

It can be shown that when $\alpha \rightarrow 0$, the maximum data likelihood will dominate the data likelihood

$$\lim_{\alpha\rightarrow 0}\frac{\max_Z p(X, Z; \Theta, \mathbf{W}, \alpha)}{\sum_Z p(X, Z; \Theta, \mathbf{W}, \alpha)} = 1. \tag{3.14}$$

To prove Equation (3.14), we first show that when α is small enough, we have

$$\arg\max_{Z} p(X, Z; \Theta, \mathbf{W}, \alpha) = Z^* \equiv \arg\min_{Z}\{\sum_{i=1}^{N}\frac{||\mathbf{x}_i - \mu_{z_i^*}||^2}{2} - \sum_{mn}\mathbf{W}_{mn}\delta(z_m^*, z_n^*)\}. \tag{3.15}$$

Proof of Equation (3.15): Assume Z' is any cluster assignment different than Z^*. We only need to show that when α is small enough,

$$p(X, Z^*; \Theta, \mathbf{W}, \alpha) > p(X, Z'; \Theta, \mathbf{W}, \alpha). \tag{3.16}$$

To prove Equation (3.16), we notice that

$$\begin{aligned}&\log p(X, Z^*; \Theta, \mathbf{W}, \alpha) - \log p(X, Z'; \Theta, \mathbf{W}, \alpha)\\ &\quad = \sum_{i=1}^{N}(\log \pi_{z_i^*} - \log \pi_{z_i'}) + \frac{1}{\alpha^2}\{\sum_{i=1}^{N}(\frac{||\mathbf{x}_i - \mu_{z_i'}||^2}{2} - \frac{||\mathbf{x}_i - \mu_{z_i^*}||^2}{2}) - \\ &\qquad \sum_{mn} \mathbf{W}_{mn}(\delta(z_m', z_n') - \delta(z_m^*, z_n^*))\}. \end{aligned} \tag{3.17}$$

Since $Z^* = \arg\min_Z \{\sum_{i=1}^{N} \frac{||\mathbf{x}_i - \mu_{z_i^*}||^2}{2} - \sum_{mn} \mathbf{W}_{mn}\delta(z_m^*, z_n^*)\}$, we have

$$\sum_{i=1}^{N}(\frac{||\mathbf{x}_i - \mu_{z_i'}||^2}{2} - \frac{||\mathbf{x}_i - \mu_{z_i^*}||^2}{2}) - \sum_{mn} \mathbf{W}_{mn}(\delta(z_m', z_n') - \delta(z_m^*, z_n^*)) > 0. \tag{3.18}$$

Let $\varepsilon = \sum_{i=1}^{N}(\frac{||\mathbf{x}_i - \mu_{z_i'}||^2}{2} - \frac{||\mathbf{x}_i - \mu_{z_i^*}||^2}{2}) - \sum_{mn} \mathbf{W}_{mn}(\delta(z_m', z_n') - \delta(z_m^*, z_n^*))$, we can see that when α is small enough

$$\log p(X, Z^*; \Theta, \mathbf{W}, \alpha) - \log p(X, Z'; \Theta, \mathbf{W}, \alpha) = \sum_{i=1}^{N}(\log \pi_{z_i^*} - \log \pi_{z_i'}) + \frac{\varepsilon}{\alpha^2} > 0. \tag{3.19}$$

■

It is obvious from Equation (3.19) that for any Z' different than Z^*

$$\begin{aligned}&\lim_{\alpha \to 0} \log p(X, Z^*; \Theta, \mathbf{W}, \alpha) - \log p(X, Z'; \Theta, \mathbf{W}, \alpha)\\ =\;& \lim_{\alpha \to 0} \sum_{i=1}^{N}(\log \pi_{z_i^*} - \log \pi_{z_i'}) + \frac{\varepsilon}{\alpha^2}\\ =\;& +\infty,\end{aligned}$$

or equivalently

$$\lim_{\alpha \to 0} \frac{p(X, Z'; \Theta, \mathbf{W}, \alpha)}{p(X, Z^*; \Theta, \mathbf{W}, \alpha)} = 0, \tag{3.20}$$

which proves Equation (3.14). As the result of Equation (3.14), when optimizing the model parameters we can equivalently maximize $\max_Z p(X, Z; \Theta, \mathbf{W}, \alpha)$ over Θ. It is then a joint optimization problem

$$\max_{\Theta,Z} p(X,Z;\Theta,\mathbf{W},\alpha).$$

Following the same thought, we find the soft posterior probability of each sample (as in conventional mixture model) becomes hard membership (as in K-means). This fact can be simply proved as follows. The posterior probability of sample $\mathbf{x}_i$ to component k is

$$P(z_i = k|X;\Theta,\mathbf{W},\alpha) \quad = \quad \frac{\sum_{Z|z_i=k} p(X,Z;\Theta,\mathbf{W},\alpha)}{\sum_Z p(X,Z;\Theta,\mathbf{W},\alpha)}.$$

From Equation (3.14)), it is easy to see

$$\lim_{\alpha\to 0} P(z_i = k|X;\Theta,\mathbf{W},\alpha) = \begin{cases} 1 & z_i^* = k \\ 0 & \text{otherwise.} \end{cases} \tag{3.21}$$

The negative logarithm of the complete likelihood $p(X,Z;\Theta,\mathbf{W},\alpha)$ is then:

$$\begin{aligned} J_\alpha(\Theta,Z) &= -\log p(X,Z;\Theta,\mathbf{W},\alpha) \\ &= -\sum_{i=1}^{N}\log\pi_{z_i} + \sum_{i=1}^{N}\frac{||\mathbf{x}_i-\mu_{z_i}||^2}{2\alpha^2} - \sum_{mn}\frac{\mathbf{W}_{mn}}{\alpha^2}\delta(z_m,z_n) + \log(\Omega(\alpha)) \\ &= -\sum_{i=1}^{N}\log\pi_{z_i} + \frac{1}{\alpha^2}(\sum_{i=1}^{N}\frac{||\mathbf{x}_i-\mu_{z_i}||^2}{2} - \sum_{mn}\mathbf{W}_{mn}\delta(z_m,z_n)) + C, \end{aligned}$$

where $C = \log\Omega(\alpha)$ is a constant. It is obvious that when $\alpha \to 0$, we can neglect the term $-\sum_{i=1}^{N}\log\pi_{z_i}$. Hence the only model parameters left for adjusting are the Gaussian means μ. We only have to consider the new cost function

$$\tilde{J}_\alpha(\mu,Z) = \frac{1}{\alpha^2}(\sum_{i=1}^{N}\frac{||\mathbf{x}_i-\mu_{z_i}||^2}{2} - \sum_{mn}\mathbf{W}_{mn}\delta(z_m,z_n)), \tag{3.22}$$

the optimization of which is obviously equivalent to Equation (3.8). So we can conclude that when $\alpha \to 0$ in Equation (3.12), the PPC model shown in Equation (3.11) becomes a K-means model with soft constraints.

3.3 From PPC to K-means with hard constraints (COPK-means)

COPK-means is a hard clustering algorithm with hard constraints. The goal is to find a set of cluster centers μ and clustering result Z that minimizes the cost function

$$\sum_{i=1}^{N} ||\mathbf{x}_i - \mu_{z_i}||^2, \tag{3.23}$$

while subject to the constraints

$$z_i = z_j, \text{ if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{L} \tag{3.24}$$

$$z_i \neq z_j, \text{ if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{N}. \tag{3.25}$$

Assume we have the PPC model with soft relations represented with the matrix $\mathbf{W}$ such that:

$$\mathbf{W}_{ij} = \begin{cases} w & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{L} \\ -w & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases} \tag{3.26}$$

where $w > 0$. We further constrain each Gaussian component to be spherical with radius σ. The complete data likelihood for PPC model is

$$P(X, Z; \Theta, \mathbf{W}) = \frac{1}{\Omega} \prod_{i=1}^{N} \{\pi_{z_i} e^{-\sum_{i=1}^{N} \frac{||\mathbf{x}_i - \mu_{z_i}||^2}{2\sigma^2}}\} \prod_{(m,n)\in\mathcal{L}} e^{w\delta(z_m, z_n)} \prod_{(m',n')\in\mathcal{N}} \exp(-w\delta(z_{m'}, z_{n'})), \tag{3.27}$$

where μ_k is the mean of the k^{th} Gaussian component. There are infinite ways to get Equation (3.23)-(3.25) from Equation (3.27), but we consider the following scaling with factor β:

$$\sigma \to \beta\sigma, \;\; \mathbf{W}_{ij} \to \mathbf{W}_{ij}/\beta^3. \tag{3.28}$$

The complete data likelihood with the scaled parameters is

$$p(X, Z; \Theta, \mathbf{W}, \beta) = \frac{1}{\Omega(\beta)} \prod_{i=1}^{N} \{\pi_{z_i} e^{-\sum_{i=1}^{N} \frac{||\mathbf{x}_i - \mu_{z_i}||^2}{2\beta^2\sigma^2}}\} \prod_{(m,n)\in\mathcal{L}} e^{\frac{w}{\beta^3}\delta(z_m, z_n)} \prod_{(m',n')\in\mathcal{N}} e^{-\frac{w}{\beta^3}\delta(z_{m'}, z_{n'})}, \tag{3.29}$$

As established in previous section, when $\beta \rightarrow 0$, the maximum data likelihood will dominate the data likelihood

$$\lim_{\beta \rightarrow 0} \frac{\max_Z p(X, Z; \Theta, \mathbf{W}, \beta)}{\sum_Z p(X, Z; \Theta, \mathbf{W}, \beta)} = 1.$$

As a result, when optimizing the model parameters Θ we can equivalently maximize $\max_Z p(X, Z; \Theta, \mathbf{W}, \beta)$. Also, the soft posterior probability (as in conventional mixture model) become hard membership (as in K-means).

The negative logarithm of the complete likelihood $p(X, Z; \Theta, \mathbf{W}, \beta)$ is then:

$$\begin{aligned} J_\beta(\Theta, Z) = &-\sum_{i=1}^{N} \log \pi_{z_i} + C + \\ &\frac{1}{\beta^2}(\sum_{i=1}^{N} \frac{||\mathbf{x}_i - \mu_{z_i}||^2}{2} + \frac{1}{\beta}(\sum_{(m',n') \in \mathcal{N}} w\delta(z_{m'}, z_{n'}) - \sum_{(m,n) \in \mathcal{L}} w\delta(z_m, z_n))), \end{aligned} \tag{3.30}$$

where $C = \log \Omega(\beta)$ is a constant. It is obvious that when $\beta \rightarrow 0$, we can neglect the term $-\sum_{i=1}^{N} \log \pi_{z_i}$. Hence we only have to consider the new cost function

$$\tilde{J}_\beta(\mu, Z) = \frac{1}{\beta^2}(\sum_{i=1}^{N} \frac{||\mathbf{x}_i - \mu_{z_i}||^2}{2} + \frac{1}{\beta}(\sum_{(m',n') \in \mathcal{N}} w\delta(z_{m'}, z_{n'}) - \sum_{(m,n) \in \mathcal{L}} w\delta(z_j, z_k))), \tag{3.31}$$

the minimization of which is obviously equivalent to the following equation since we can neglect the constant factor $\frac{1}{\beta^2}$:

$$\tilde{\tilde{J}}_\beta(\mu, Z) = \sum_{i=1}^{N} \frac{||\mathbf{x}_i - \mu_{z_i}||^2}{2} + \frac{w}{\beta} J_c(Z). \tag{3.32}$$

where $J_c(Z) = \sum_{(m',n') \in \mathcal{N}} \delta(z_{m'}, z_{n'}) - \sum_{(m,n) \in \mathcal{L}} \delta(z_m, z_n)$ is the cost function term from pairwise constraints.

Let $S_Z = \{Z | z_i = z_j \text{ if } \mathbf{W}_{ij} > 0; z_i \neq z_j \text{ if } \mathbf{W}_{ij} < 0;\}$. We assume the pairwise relations are consistent, that is, $S_Z \neq \emptyset$. Obviously, all Z in S_Z achieve the same minimum value of the term $J_c(Z)$. That is

$$\begin{aligned} &\forall Z \in S_Z, Z' \in S_Z \qquad J_c(Z) = J_c(Z') \\ &\forall Z \in S_Z, Z'' \notin S_Z \qquad J_c(Z) < J_c(Z''). \end{aligned}$$

It is obvious that when $\beta \to 0$, any Z that minimizes $\tilde{\tilde{J}}_\beta(\mu, Z)$ must be in S_Z. So the minimization of Equation (3.29) can be finally casted into the following form:

$$\begin{array}{rl} \min\limits_{Z,\mu} & \sum\limits_{i=1}^{N} ||\mathbf{x}_i - \mu_{z_i}||^2 \\ \text{subject to} & Z \in S_Z, \end{array}$$

which is apparently equivalent to Equation (3.23)-(3.25). So we can conclude that $\beta \to 0$ in Equation (3.28), the PPC model shown in Equation (3.27) becomes a K-means model with hard constraints.

Chapter 4

Constrained Clustering with Gaussian Process Classifiers

In §1 − §3 we have discussed constrained clustering adapted from existing clustering algorithms and their limitations. In this chapter we will try to solve the same problem from a new perspective. To avoid the limitations inherited from generative models, we based our algorithm on discriminative models, more specifically, Gaussian process classifiers. The major difficulty in marrying pairwise constraints and discriminative models is finding a principled way to inject the pairwise constraints into the Bayes formulation for probabilistic classifiers

$$p(\text{parameters}|\text{observation}) = \frac{p(\text{observation}|\text{parameters})p(\text{parameters})}{p(\text{observation})}$$

or the cost function for the non-probabilistic classifiers

$$C = \text{empirical error} + \text{regularization term.}$$

In the discriminative models, both empirical error and the likelihood $p(\text{observation})$ measures the difference between observations and model prediction, and that is where the pairwise constraints should enter. Using a generalized observation form that pairwise constraints as special case, we are able to cast the constrained clustering task into a form similar to the conventional semi-supervised learning, where a subset of samples are labeled. As an advantage of this, we are able to use the technique newly developed for semi-supervised learning, such as graphical kernels.

4.1 Gaussian Process Classifiers

4.1.1 Supervised Learning with Gaussian Process Classifiers

In this section, we will briefly introduce the Gaussian process classifiers (GPCs). Suppose we have data set $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ and its labels $Y = \{y_1, y_2, \cdots, y_N\}$. For simplicity, we only consider the binary classification problem, that is, $y_i \in \{-1, +1\}$. The Gaussian process classifiers assume a latent Gaussian process f with zero mean and covariance function K, and at each point $\mathbf{x}$, we observe label y with the probability

$$P(y|f(\mathbf{x})) = \frac{e^{f(\mathbf{x})\delta(y,+1)}}{1+e^{f(\mathbf{x})}} = \begin{cases} \frac{e^{f(\mathbf{x})}}{1+e^{f(\mathbf{x})}} & y = +1, \\ \frac{1}{1+e^{f(\mathbf{x})}} & y = -1 \end{cases} \tag{4.1}$$

where $\delta(\cdot,\cdot)$ is the Kronecker delta-function. Let $\mathbf{f}$ be the values of f at X

$$\mathbf{f} = [f(\mathbf{x}_1), f(\mathbf{x}_2), ..., f(\mathbf{x}_N)]^T,$$

which follows a N-dimensional Gaussian distribution:

$$p(\mathbf{f}) = (2\pi)^{-\frac{N}{2}}|\mathbf{K}|^{-\frac{1}{2}}e^{-\frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f}} \tag{4.2}$$

where $\mathbf{K} \in \mathbb{R}^{N\times N}$ is the covariance matrix (kernel) with $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$.

We assume the correlation between samples are fully modeled by the latent variable f and assume that the observation made at any point $\mathbf{x}$, given the latent field value $f(\mathbf{x})$, is independent of observations made at any other point. It then follows that probability of observing Y factorizes

$$P(Y|\mathbf{f}) = \prod_{i=1}^{N} P(y_i|f(\mathbf{x}_i)) = \prod_{i=1}^{N} \frac{e^{f(\mathbf{x}_i)\delta(y_i,+1)}}{1+e^{f(\mathbf{x}_i)}}. \tag{4.3}$$

In supervised learning, both X and Y are given as the training data, and the probability of observing label y at any novel sample $\mathbf{x}$ is

$$P(y|\mathbf{x}, X, Y) = \int P(y|f(\mathbf{x}))p(f(\mathbf{x})|\mathbf{f})p(\mathbf{f}|Y)d\mathbf{f} \tag{4.4}$$

where $p(f(\mathbf{x})|\mathbf{f})$ is simply a conditional Gaussian given by $p(f(\mathbf{x})|\mathbf{f}) = p(\mathbf{f}, f(\mathbf{x}))/p(\mathbf{f})$, and $p(\mathbf{f}|Y)$ can be calculated from Bayes' rule:

$$p(\mathbf{f}|Y) = \frac{p(\mathbf{f}, Y)}{P(Y)} \propto P(Y|\mathbf{f})p(\mathbf{f}).$$

If we also parameterize the covariance function with θ and assume a prior probability for it, we need to further marginalize out the θ as follows

$$P(y|\mathbf{x}, X, Y) = \int_{\theta} \int_{\mathbf{f}} P(y|f(\mathbf{x}))p(f(\mathbf{x})|\mathbf{f}, \theta)p(\mathbf{f}, \theta|Y) d\mathbf{f} d\theta, \tag{4.5}$$

where $p(\mathbf{f}, \theta|Y) \propto p(\theta)p(\mathbf{f}|\theta)P(Y|f)$. Due to the nonlinearity introduced in the softmax observation function, the integral in Equation (4.4) and (4.5) generally cannot be given in closed form. We typically need to resort to approximate inference, such as variational approximation or sampling.

4.1.2 Semi-supervised Learning with Gaussian Process Classifiers

In semi-supervised learning, we only have label for a subset of training samples. Without loss of generality, we assume the first l samples (denoted X_l) are labeled as $Y_l = \{y_1, y_2, \cdots, y_l\}$, and rest samples denoted as X_u with unknown label Y_u. We further extend this notation convention to the latent field and let

$$\begin{aligned} \mathbf{f}_l &= [f(\mathbf{x}_1), f(\mathbf{x}_2,) \cdots, f(\mathbf{x}_l)]^T \\ \mathbf{f}_u &= [f(\mathbf{x}_{l+1}), f(\mathbf{x}_{l+2}) \cdots, f(\mathbf{x}_N)]^T. \end{aligned}$$

The goal in semi-supervised learning is twofold. First, we need to predict the label for all the unlabeled data in the training set. Second, we may also want to extend this prediction to samples outside of training set. Here, we only consider the fist goal and the research on the second goal can be found at [72][1]. We need to rewrite Equation (4.4) as follows

$$P(Y|X, Y_l) = \int P(Y_u|\mathbf{f}_u)\, p(\mathbf{f}_u|\mathbf{f}_l)\, p(\mathbf{f}_l|Y_l) df(X_u) \tag{4.6}$$

Clearly, the label information in Y_l propagates to the unlabeled samples X_u through the conditional probability $p(\mathbf{f}_u|\mathbf{f}_l)$, which is fully specified by the covariance matrix $\mathbf{K}$ for the joint variable $[\mathbf{f}_l; \mathbf{f}_u]^T$. For any $\mathbf{K}$ that is derived from a covariance function $K(\cdot, \cdot)$

$$\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j),$$

[1] Of course, one obvious choice is we extend our training set to include all the new samples, and perform semi-supervised learning again.

Equation (4.6) factorizes as follows

$$P(Y_u|X, Y_l) = \prod_{i=l+1}^{N} \int P(y_i|f(\mathbf{x}_i))\, p(f(\mathbf{x}_i)|\mathbf{f}_l)\, p(\mathbf{f}_l|Y_l) df(\mathbf{x}_i) \tag{4.7}$$

where $\int P(y_i|f(\mathbf{x}_i))\, p(f(\mathbf{x}_i)|\mathbf{f}_l)\, p(\mathbf{f}_l|Y_l) df(\mathbf{x}_i)$ is exactly the classification as expressed in Equation (4.4). To prevent the semi-supervised learning reducing to $N - l$ *independent* supervised learning tasks, we need to weave the information from the unlabeled data set into the covariance matrix $\mathbf{K}$, and consequently $p(f(\mathbf{x}_i)|\mathbf{f}_l)$ and $p(\mathbf{f}_l|Y_l)$, and thus achieve a input-dependent regularization [68]. The covariance matrix designed with unlabeled information is usually called semi-supervised kernel.

4.1.3 Towards Constrained Clustering

Our effort to incorporate the pairwise relations into clustering consist of two parts. First, in Section 4.2, we introduce the generalized observation form for pairwise relations, and the corresponding likelihood of $\mathbf{f}$ can be given through manipulating Equation (4.2) and (4.3). Second, in Section 4.5, we discuss the prior form of $\mathbf{f}$ that can exploit the samples not involved in any pairwise relations. This design of prior is realized by using so-called semi-supervised kernels as the $\mathbf{K}$ in Equation (4.2). In Section 4.6, we propose the Semi-supervised Pairwise Gaussian Process Classifier (SPGP) by combining our work on the prior (Section 4.5) and on the likelihood (Section 4.2) .

4.2 Pairwise Constraints as Observation

In a semi-supervised scenario, we have incomplete knowledge about the class label of samples: it can be a labeled subset [16, 75, 90, 92, 7] (called partial labeling) or some pairwise relations. Consequently, instead of one, we have a *set* of class assignments Y consistent with our knowledge. Taking each Y as an atomic event, our knowledge can be equivalently expressed as a union of all feasible events Y, denoted as Ω. For pairwise relations, we have

$$\Omega = \{Y|(y_i = y_j, \forall\ (i,j) \in \mathcal{M}) \wedge (y_i \neq y_j, \forall\ (i,j) \in \mathcal{C}\}, \tag{4.8}$$

where $\mathcal{M}$ and $\mathcal{C}$ are respectively the set of links and do-not-links. Since

$$P(Y, Y') = 0, \;\; Y \neq Y'$$

the probability of Ω can then written as the sum of probability of all the component Ys:

$$P(\Omega) = \sum_{Y \in \Omega} P(Y). \tag{4.9}$$

Also it is straightforward to see that

$$P(\Omega|Y) = \begin{cases} 1 & Y \in \Omega \\ 0 & \text{otherwise} \end{cases}, \tag{4.10}$$

which should be interpreted as Ω happens if any of component Y happens. Given the latent field $\mathbf{f}$, the likelihood of Ω is:

$$P(\Omega|\mathbf{f}) = \sum_{Y} P(\Omega, Y|\mathbf{f}) = \sum_{Y} P(\Omega|Y, \mathbf{f}) P(Y|\mathbf{f}) = \sum_{Y} \{P(\Omega|Y) \prod_{i=1}^{N} \frac{e^{f(\mathbf{x}_i)\delta(y_i, +1)}}{1 + e^{f(\mathbf{x}_i)}}\}, \tag{4.11}$$

which makes use of Equation (4.3) and the fact $P(\Omega|Y, \mathbf{f}) = P(\Omega|Y)$. Note Equation (4.11) is a natural generalization of full labeling (supervised learning) and partial labeling (semi-supervised learning): For a full labeling

$$\Omega = \{\text{the given } Y\}, \tag{4.12}$$

and Equation (4.11) becomes

$$P(\Omega|\mathbf{f}) = \sum_{Y} P(\Omega, Y|\mathbf{f}) = \sum_{Y} P(\Omega|Y, \mathbf{f}) P(Y|\mathbf{f}) = P(Y|\mathbf{f}); \tag{4.13}$$

For a partial labeling described in Section (4.1.2) we have

$$\Omega = \{Y'|y'_1 = y_1, y'_2 = y_2, \cdots, y'_l = y_l\}, \tag{4.14}$$

and Equation (4.11) becomes

$$P(\Omega|\mathbf{f}) = \sum_{Y} P(\Omega|Y, \mathbf{f}) P(Y|\mathbf{f}) = \prod_{i=1}^{l} \frac{e^{f(\mathbf{x}_i)\delta(y_i, +1)}}{1 + e^{f(\mathbf{x}_i)}}\}, \tag{4.15}$$

which is equivalent to the likelihood formula given in [92].

In reality, pairwise relations often come with significant uncertainty, so it is desired for $P(\Omega|Y)$ to be a *soft* membership that reflects our confidence. We start with modeling the conditional probability $P(Y|\Omega)$ via the following Gibbs distribution:

$$P(Y|\Omega) = \frac{1}{Z_1} e^{\sum_{i<j} \mathbf{W}_{ij}\delta(y_i,y_j)} = \frac{1}{Z_1} \prod_{i<j} e^{\mathbf{W}_{ij}\delta(y_i,y_j)}, \tag{4.16}$$

where $\mathbf{W}_{ij}$ is the weight for pair $(\mathbf{x}_i, \mathbf{x}_j)$ and Z_1 is the partition function. Like in PPC (see Chapter 2), we use $\mathbf{W}_{ij}$ to express both the type of pairwise relation between $(\mathbf{x}_i, \mathbf{x}_j)$ and its confidence value γ_{ij} (> 0.5) through

$$\frac{e^{\mathbf{W}_{ij}}}{1 + e^{\mathbf{W}_{ij}}} = \gamma_{ij}^{L_{ij}} (1 - \gamma_{ij})^{1-L_{ij}}, \tag{4.17}$$

where $L_{ij} = 1$ if $(\mathbf{x}_i, \mathbf{x}_j)$ is specified to be linked, and $L_{ij} = 0$ for a do-not-link. It follows from Equation (4.17) that $\mathbf{W}_{ij} > 0$ for a link between $(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{W}_{ij} < 0$ for a do-not-link. We set $\mathbf{W}_{ij} = 0$ if no prior knowledge is available on pair $(\mathbf{x}_i, \mathbf{x}_j)$. Clearly, $|\mathbf{W}_{ij}|$ reflects our confidence since

$$\frac{e^{|\mathbf{W}_{ij}|}}{1 + e^{|\mathbf{W}_{ij}|}} = \gamma_{ij}.$$

Using the Bayes' rule, we can get $P(\Omega|Y)$ as follows

$$P(\Omega|Y) = \frac{P(Y|\Omega)P(\Omega)}{P(Y)} = \frac{1}{Z_2} \prod_{i<j} e^{\mathbf{W}_{ij}\delta(y_i,y_j)}. \tag{4.18}$$

Here we assume a uniform $P(\mathbf{Y}) = 2^{-N}$, which is the prior probability before any information on $\mathbf{X}$ or Ω is known [2]. As will be shown in Section 4.4, Z_2 does not affect the final clustering result. From Equation (4.18), $P(\Omega|Y)$ is larger if Y satisfies the specified pairwise relations and vice versa. When $|\mathbf{W}_{ij}| \to \infty$, we have $P(\Omega|Y) = 0$ if (y_i, y_j) violates the specified relation. This fact can be briefly proven as follows.

Proof: (sketch)

Let's assume that there exists $Y^* = \{y_1^*, \cdots, y_N^*\}$ that satisfies all the pairwise relations with infinity weight (otherwise the there must be some contradiction in our prior knowledge). It is easy to see that for any $Y = \{y_1, \cdots, y_N\}$

$$P(\Omega|Y) = \frac{1}{Z_2} \prod_{i<j} e^{\mathbf{W}_{ij}\delta(y_i,y_j)} \leq \frac{\prod_{i<j} e^{\mathbf{W}_{ij}\delta(y_i,y_j)}}{\prod_{i<j} e^{\mathbf{W}_{ij}\delta(y_i^*,y_j^*)}}.$$

[2] Do not confuse this assumption with the situation when covariance matrix for $\mathbf{f}$ is known. In that case, $P(\mathbf{Y})$ is generally not uniform from Equation (4.2) and (4.3).

Suppose $|\mathbf{W}_{i'j'}| \to \infty$ and $(y_{i'}, y_{j'})$ violates the specified relation. We first notice that

$$\frac{\prod_{i<j-(i',j')} e^{\mathbf{W}_{ij}\delta(y_i,y_j)}}{\prod_{i<j-(i',j')} e^{\mathbf{W}_{ij}\delta(y_i^*,y_j^*)}}$$

is upper bounded by a constant $C > 0$, where $\prod_{i<j-(i',j')}$ stands for the product for all the index pair (i,j) with $i<j$ except (i',j'). Then it is easy to see

$$\begin{aligned}\frac{\prod_{i<j} e^{\mathbf{W}_{ij}\delta(y_i,y_j)}}{\prod_{i<j} e^{\mathbf{W}_{ij}\delta(y_i^*,y_j^*)}} &= \frac{e^{\mathbf{W}_{i'j'}\delta(y_{i'},y_{j'})}}{e^{\mathbf{W}_{i'j'}\delta(y_{i'}^*,y_{j'}^*)}} \frac{\prod_{i<fj-(i',j')} e^{\mathbf{W}_{ij}\delta(y_i,y_j)}}{\prod_{i<j-(i',j')} e^{\mathbf{W}_{ij}\delta(y_i^*,y_j^*)}} \\ &\leq C\frac{e^{\mathbf{W}_{i'j'}\delta(y_{i'},y_{j'})}}{e^{\mathbf{W}_{i'j'}\delta(y_{i'}^*,y_{j'}^*)}} = C\frac{1}{e^{\mathbf{W}_{i'j'}\delta(y_{i'}^*,y_{j'}^*)}} \to 0\end{aligned}$$

which leads to

$$\frac{\prod_{i<j} e^{\mathbf{W}_{ij}\delta(y_i,y_j)}}{\prod_{i<j} e^{\mathbf{W}_{ij}\delta(y_i^*,y_j^*)}} \to 0,$$

which completes the proof. ■

When $|\mathbf{W}_{ij}|$, we have *hard constraints* between $(\mathbf{x}_i, \mathbf{x}_j)$; otherwise, the relation is *soft*. When all specified pairwise relations are hard, $P(\Omega|Y)$ degenerates to the extreme case described in Equation (4.8). Based on Equation (4.18), the likelihood of $\mathbf{f}$ defined in Equation (4.11) can be written as:

$$P(\Omega|\mathbf{f}) = \frac{1}{Z_2}\sum_{Y}\{\prod_{i<j} e^{\mathbf{W}_{ij}\ \delta(y_i,y_j)} \cdot \prod_{k=1}^{N} \frac{e^{\delta(y_k,+1)f(x_k)}}{e^{f(\mathbf{x}_k)}+1}\}. \tag{4.19}$$

4.2.1 Illustrative Example

We give here one example to illustrate the design of hard constraints. For simplicity, we consider the situation where there is only a constraint between sample $\mathbf{x}_1$ and $\mathbf{x}_2$, or equivalently, for any $i \neq 1$ or $j \neq 2$, we have $\mathbf{W}_{ij} = 0$. As the result, $P(\Omega|\mathbf{f})$ can be simplified as $P(\Omega|f(\mathbf{x}_1), f(\mathbf{x}_2))$. The experiment shown in Figure 4.1 is designed to study the interaction between the values of the weight $\mathbf{W}_{12}$ and $(f(\mathbf{x}_1), f(\mathbf{x}_2))$ in deciding the likelihood $P(\Omega|f(\mathbf{x}_1), f(\mathbf{x}_2))$. Our observation can be summarized as follows:

- When $\mathbf{W}_{12} > 0$ (link), $P(\Omega|f(\mathbf{x}_1), f(\mathbf{x}_2))$ is large when $f(\mathbf{x}_1)$ and $f(\mathbf{x}_2)$ are both positive or both negative, and $P(\Omega|f(\mathbf{x}_1), f(\mathbf{x}_2))$ is small when $f(\mathbf{x}_1)$ and $f(\mathbf{x}_2)$ have different sign. This effect becomes more salient when $\mathbf{W}_{12}$ increases. Actually when $\mathbf{W}_{12} \rightarrow +\infty$, we can see that

$$P(\Omega|5,-5) = P(\Omega|-5,5) \ll P(\Omega|5,5) = P(\Omega|-5,-5),$$

which indicates, as is consistent with our intuition, that the weight of pairwise constraints decides how one set of latent variables are favored over the other. The case of do-not-link ($\mathbf{W}_{ij} < 0$) can be analyzed similarly.

- The $P(\Omega|f(\mathbf{x}_1), f(\mathbf{x}_2))$ always favors $f(\mathbf{x}_1)$ and $f(\mathbf{x}_2)$ with large absolute values when the sign of them are consistent with the pairwise constraints. This is again understandable since when $|f(\mathbf{x})|$ increases, we will get a classification y with more confidence based on the equation

  $$P(y) = \frac{e^{f(\mathbf{x})\delta(y,+1)}}{1+e^{f(\mathbf{x})}},$$

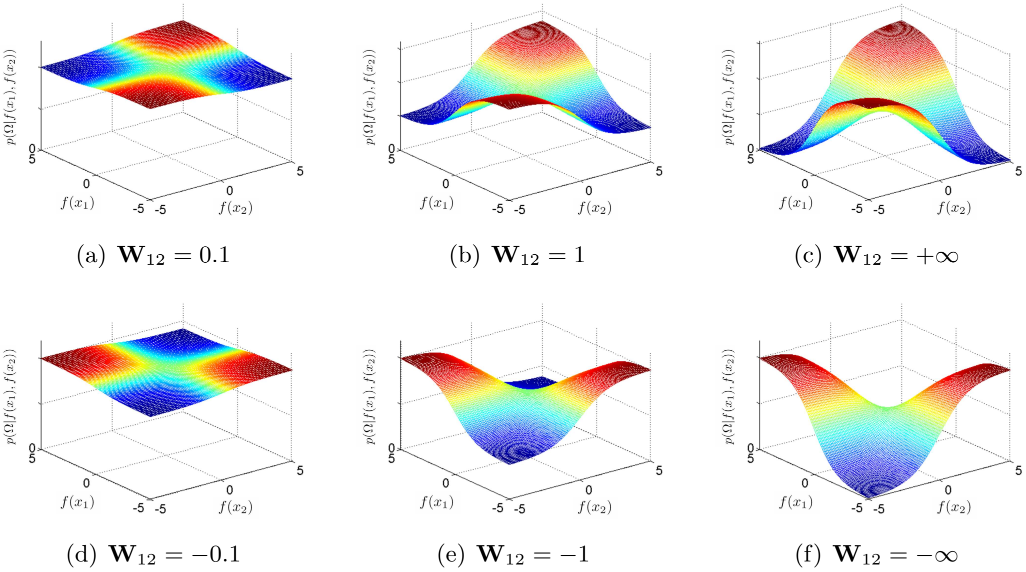
  and thus a high likelihood of Ω if y is consistent with it. On the other hand we need to consider the smoothness of $\mathbf{f}$ [3], which favors a $\mathbf{f}$ with less difference among the component values and directly opposes to set values in $\mathbf{f}$ to be too large or too small.

4.3 Approximation of Likelihood

One major difficulty of our method would be effectively estimating $P(\Omega|\mathbf{f})$, since direct calculation is generally intractable due to the summarization over all Y. We first notice that

$$P(\Omega|\mathbf{f}) = \frac{1}{Z_2} E_Y \{\prod_{i<j} e^{\mathbf{W}_{ij}\delta(y_i,y_j)}\}, \tag{4.20}$$

[3] Here, the smoothness requirement may depends on the distribution of data, as will be clear in Section 4.5



(a) $\mathbf{W}_{12} = 0.1$ (b) $\mathbf{W}_{12} = 1$ (c) $\mathbf{W}_{12} = +\infty$

(d) $\mathbf{W}_{12} = -0.1$ (e) $\mathbf{W}_{12} = -1$ (f) $\mathbf{W}_{12} = -\infty$

Figure 4.1: The probability of Ω with different $\mathbf{W}$.

here $E_Y\{\cdot\}$ stands for the expectation under distribution $P(Y|\mathbf{f})$. We get a approximation of $P(\Omega|\mathbf{f})$, denoted as $J(\mathbf{f},\Omega)$, by exchanging the order of $\prod$ and E_Y in Equation (4.20):

$$J(\mathbf{f},\Omega) = \frac{1}{Z_2}\prod_{i<j} E_Y\{e^{\mathbf{W}_{ij}\delta(y_i,y_j)}\} \tag{4.21}$$

$$= \frac{1}{Z_2}\prod_{\substack{i<j\\ \mathbf{W}_{ij}\neq 0}} \frac{e^{\mathbf{W}_{ij}}\{e^{f(\mathbf{x}_i)+f(\mathbf{x}_j)}+1\} + e^{f(\mathbf{x}_i)}+e^{f(\mathbf{x}_j)}}{(e^{f(\mathbf{x}_i)}+1)(e^{f(\mathbf{x}_j)}+1)}. \tag{4.22}$$

It is easy to verify that $J(\mathbf{f},\Omega) = P(\Omega|\mathbf{f})$ when pairwise relations are disjoint: each sample is involved in *at most* one pairwise relation. (probably some more details here)

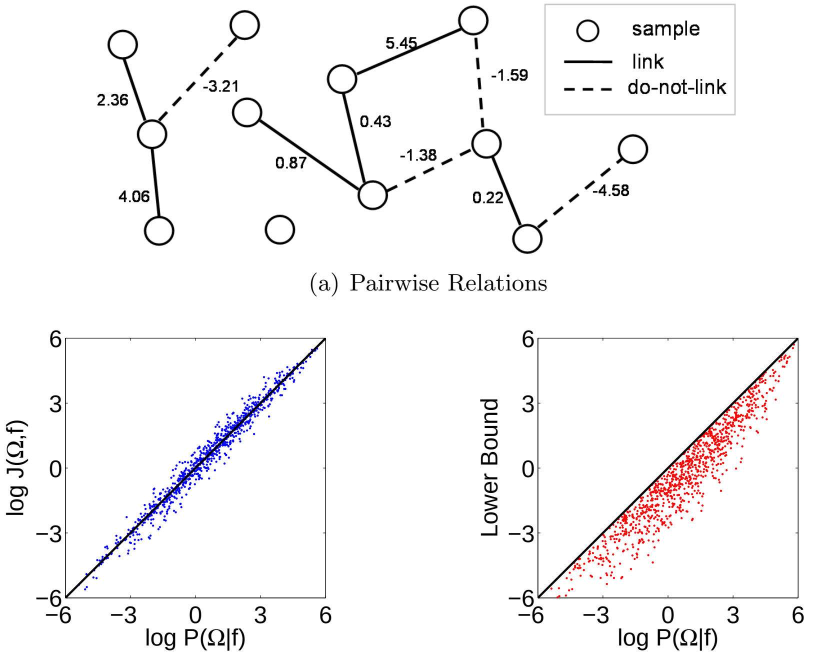
In practice, $J(\mathbf{f},\Omega)$ yields a good approximation when pairwise relations are scarce. For comparison, we also consider another approximation of $\log P(\Omega|\mathbf{f})$ given by the Jensen's inequality

$$\log E_Y\{\prod_{i<j} e^{\mathbf{W}_{ij}\delta(y_i,y_j)}\} \geq E_Y\{\log \prod_{i<j} e^{\mathbf{W}_{ij}\delta(y_i,y_j)}\}.$$

In this case we get a lower bound of $\log P(\Omega|\mathbf{f})$:

$$\log P(\Omega|\mathbf{f}) \geq -\log Z_2 + \sum_{i<j} \mathbf{W}_{ij}\frac{e^{f(\mathbf{x}_i)+f(\mathbf{x}_j)}+1}{(e^{f(\mathbf{x}_i)}+1)(e^{f(\mathbf{x}_j)}+1)}. \tag{4.23}$$

Figure 4.2 compares $\log J(\mathbf{f},\Omega)$ with the lower bound given in Equation (4.23) on a toy



(a) Pairwise Relations

(b) $\log P(\Omega|\mathbf{f})$ vs. $\log J(\mathbf{f},\Omega)$

(c) $\log P(\Omega|\mathbf{f})$ vs. lower bound

Figure 4.2: Comparison between two approximations of $\log P(\Omega|\mathbf{f})$. In the toy problem, we randomly assign 10 pairwise relations (with weight $\sim N(0,100)$) among 12 samples. The field value $\mathbf{f} \in \mathbb{R}^{12}$ is randomly chosen from $N(0, 25\mathbb{I}_{12})$. (a): A typical example of pairwise relations that will be used in (b) and (c); (b): Scatter plot of $\log P(\Omega|\mathbf{f})$ vs. $\log J(\mathbf{f},\Omega)$ with 1000 random $\mathbf{f}$; (c): Scatter plot of $\log P(\Omega|\mathbf{f})$ vs. lower bound given in Equation (4.23) with 1000 random $\mathbf{f}$.

problem. It is clear from Figure 4.2 (b) and (c) that $\log J(\mathbf{f},\Omega)$ gives more accurate approximation.

4.4 Why Use Maximum a Posteriori (MAP) GPC

The principle Bayesian solution used for standard (supervised) GPC [83] marginalizes out the latent functions $\mathbf{f}$. However, this solution does not work for GPC when only pairwise relations are available, as elucidated by the following proposition.

Proposition 1. *With Ω defined in Equation (4.16), for* $i = 1, 2, \cdots, N$,

$$P(y_i = +1|\Omega) = \int_{\mathbb{R}^N} P(y_i = +1|\mathbf{f},\Omega) p(\mathbf{f}|\Omega) \mathrm{d}\mathbf{f} = 0.5.$$

Proof: From Equation (4.2) and (4.19), we have

$$p(\mathbf{f}|\Omega) = p(-\mathbf{f}|\Omega).$$

From symmetry, we have

$$\begin{aligned}
P(y_i = +1|\Omega) &= \frac{1}{2}\{\int_{\mathbb{R}^N} P(y_i = +1|\mathbf{f}, \Omega)p(\mathbf{f}|\Omega)\mathrm{d}\mathbf{f} + \int_{\mathbb{R}^N} P(y_i = +1| - \mathbf{f}, \Omega)p(-\mathbf{f}|\Omega)\mathrm{d}\mathbf{f}\} \\
&= \frac{1}{2}\{\int_{\mathbb{R}^N} P(y_i = +1|\mathbf{f}, \Omega)p(\mathbf{f}|\Omega)\mathrm{d}\mathbf{f} + \int_{\mathbb{R}^N} P(y_i = +1| - \mathbf{f}, \Omega)p(\mathbf{f}|\Omega)\mathrm{d}\mathbf{f}\} \\
&= \frac{1}{2}\int_{\mathbb{R}^N} (P(y_i = +1|\mathbf{f}, \Omega) + P(y_i = +1| - \mathbf{f}, \Omega))P(\mathbf{f}|\Omega)\mathrm{d}\mathbf{f} \\
&= \frac{1}{2}\int_{\mathbb{R}^N} p(\mathbf{f}|\Omega)\mathrm{d}\mathbf{f} = \frac{1}{2}
\end{aligned}$$

The proof makes use of the fact that $P(y_i = +1|\mathbf{f}, \Omega) + P(y_i = +1| - \mathbf{f}, \Omega) = 1$, which is easy to verify. ■

There are two ways to stay in the standard GPC framework. For a two-class problem[4], we can break the symmetry by assigning an arbitrary sample to class +1 (or −1), but this strategy does not work for a multi-class situation. Another choice is to calculate the probability $P(y_i = y_j|X, \Omega)$ for all pair $(\mathbf{x}_i, \mathbf{x}_j)$, and lately use this as new measure of similarity. However, this requires $O(N^2)$ inferences with GPC, and is therefore computationally undesirable. Moreover, one has to use another similarity-based clustering algorithm to get the cluster assignments for samples. Instead, we find the maximum a posteriori (MAP) solution of $\mathbf{f}$ [5], or equivalently the solution that minimizes $L(\mathbf{f}) = -\log p(\mathbf{f}|\Omega)$. In practice we use $J(\mathbf{f}, \Omega)$ in place of $P(\Omega|\mathbf{f})$, and optimize the following objective function:

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}}\{-\log J(\mathbf{f}, \Omega) + 1/2\, \mathbf{f}^T \mathbf{K}^{-1}\mathbf{f}\}. \tag{4.24}$$

We know from the form of $J(\mathbf{f}, \Omega)$ that Z_2 only appears in a constant term $\log Z_2$ in this objective function, and therefore will not affect the optimal solution $\hat{\mathbf{f}}$. In Section 4.5, we shall show that the optimization in Equation (4.24) can be simplified. Once $\hat{\mathbf{f}}$ is determined, the classification of X is carried out with Equation (4.1).

[4] Although the model we proposed is also binary classification problem, it can be readily extended to multi-class cases.

[5] Clearly MAP solution appears in pairs, since $P(\mathbf{f}|\Omega) = P(-\mathbf{f}|\Omega)$

4.5 Design of Kernel

4.5.1 The Role of the Unconstrained Samples

We divide the data set X into the constrained set $X_c = \{\mathbf{x}_i| \ \exists j \ \mathbf{W}_{ij} \neq 0\}$ and unconstrained set $X_u = \{\mathbf{x}_i|\forall j \ \mathbf{W}_{ij} = 0\}$. We want the unconstrained set to effectively influence the resulted classifier, much the same role played by the unlabeled set in the more familiar partial labeling scenario. Not surprisingly, this intention can not be realized with a conventional covariance matrix, as elucidated by the proposition we will give presently. Without loss of generality, we assume $X_c = \{\mathbf{x}_1, \cdots, x_L\}$. Accordingly, we can decompose the field $\mathbf{f}$ as follows:

$$\mathbf{f} = \begin{pmatrix} \mathbf{f}_c \\ \mathbf{f}_u \end{pmatrix}, \tag{4.25}$$

with $\mathbf{f}_c$ corresponding to the field values X_c and $\mathbf{f}_u$ on X_u. The covariance matrix $\mathbf{K}$ can also be decomposed accordingly into four sub-matrices

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_c & \mathbf{K}_{uc}^T \\ \mathbf{K}_{uc} & \mathbf{K}_u \end{pmatrix}.$$

It can be shown both $P(\Omega|\mathbf{f})$ and $J(\mathbf{f}, \Omega)$ depend *only* on $\mathbf{f}_c$. The following proposition can be easily verified using the conditional property of Gaussian variables.

Proposition 2: *The solution of the problem*

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}}\{R(\mathbf{f}_c) + \frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f}\}$$

for any function R bounded from below can be written as $\hat{\mathbf{f}} = \begin{pmatrix} \hat{\mathbf{f}}_c \\ \hat{\mathbf{f}}_u \end{pmatrix}$, *where*

$$\hat{\mathbf{f}}_c = \arg\min_{\mathbf{f}_c}\{R(\mathbf{f}_c) + \frac{1}{2}\mathbf{f}_c^T\mathbf{K}_c^{-1}\mathbf{f}_c\} \tag{4.26}$$

$$\hat{\mathbf{f}}_u = \mathbf{K}_{uc}\mathbf{K}_c^{-1}\hat{\mathbf{f}}_c. \tag{4.27}$$

Proof: Let

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}}\{R(\mathbf{f}_c) + \frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f}\} \tag{4.28}$$

and let $\hat{\mathbf{f}}'_c$ and $\hat{\mathbf{f}}'_y$ be the set of components corresponding to X_c and X_u. It is easy to see that

$$\begin{aligned}
\hat{\mathbf{f}}'_u &= \arg\min_{\mathbf{f}_u}\{R(\mathbf{f}'_c) + \begin{pmatrix}\hat{\mathbf{f}}'_c \\ \hat{\mathbf{f}}_u\end{pmatrix}^T \mathbf{K}^{-1} \begin{pmatrix}\hat{\mathbf{f}}'_c \\ \hat{\mathbf{f}}_u\end{pmatrix}\} && (4.29)\\
&= \arg\min_{\mathbf{f}_u} \begin{pmatrix}\hat{\mathbf{f}}'_c \\ \hat{\mathbf{f}}_u\end{pmatrix}^T \mathbf{K}^{-1} \begin{pmatrix}\hat{\mathbf{f}}'_c \\ \hat{\mathbf{f}}_u\end{pmatrix} && (4.30)\\
&= \mathbf{K}_{uc}\mathbf{K}_c^{-1}\hat{\mathbf{f}}'_c && (4.31)
\end{aligned}$$

We can simplify the optimization in Equation (4.28) by substituting $\mathbf{f}_u$ with $\mathbf{K}_{uc}\mathbf{K}_c^{-1}\mathbf{f}_c$

$$\begin{aligned}
\hat{\mathbf{f}} &= \arg\min_{\mathbf{f}_c}\{R(\mathbf{f}_c) + \frac{1}{2}\begin{pmatrix}\mathbf{f}_c \\ \mathbf{K}_{uc}\mathbf{K}_c^{-1}\mathbf{f}_c\end{pmatrix}^T \mathbf{K}^{-1} \begin{pmatrix}\mathbf{f}_c \\ \mathbf{K}_{uc}\mathbf{K}_c^{-1}\mathbf{f}_c\end{pmatrix}\} && (4.32)\\
&= \arg\min_{\mathbf{f}_c}\{R(\mathbf{f}_c) + \frac{1}{2}\mathbf{f}_c^T\mathbf{K}_c^{-1}\mathbf{f}_c\}, && (4.33)
\end{aligned}$$

which completes the proof. ■

For a “local” kernel $\mathbf{K}$ [92], e.g. RBF kernel, the entry $\mathbf{K}_{ij}$ only depends on $\mathbf{x}_i$ and $\mathbf{x}_j$ and *not any other samples*. Proposition 2 tells us that with such a local kernel $\mathbf{K}$, the unconstrained set X_u is useless for the classification based on Equation (4.24). Indeed, X_u does not affect the optimization in Equation (4.26) (with $R(\mathbf{f}_c)$ set to be $-\log J(\mathbf{f}, \Omega)$), while in Equation (4.27), $\hat{\mathbf{f}}_u$ is simply interpolated from $\hat{\mathbf{f}}_c$. To overcome this problem, we need a $\mathbf{K}$ with information of X_u encoded in the entries of $\mathbf{K}_c$. Such kernels will be referred to as semi-supervised kernels since they are typically designed to use samples bearing no label information.

4.5.2 Semi-supervised Kernels

Our kernel design strategy largely follows previous work on graph kernel [92, 73]. The key difference is that we fit the kernel to the pairwise relations, instead of some labeled samples as in [91]. Let S be the affinity matrix of X with $S_{ij} = e^{-||\mathbf{x}_i - \mathbf{x}_j||^2/s^2}$. The normalized graph Laplacian is defined as $\Delta = \mathbb{I} - D^{-\frac{1}{2}}SD^{-\frac{1}{2}}$, where D is a diagonal matrix with entry $D_{ii} = \sum_j S_{ij}$. Suppose the eigen-decomposition of Δ is:

$$\Delta = \sum_{i=1}^{N} \mu_i \phi_i \phi_i^T.$$

We know from [7] that the eigenvectors $\{\phi_i\}$ provide the harmonic basis with frequency indicated by the eigenvalues $\{\mu_i\}$. Roughly speaking, the higher frequency component has bigger eigenvalue, and vice versa. We build a semi-supervised kernel $\mathbf{K}$ based on a transform of $\{\mu_i\}$:

$$\mathbf{K} = \sum_{i=1}^{N} g(\mu_i) \phi_i \phi_i^T, \ g(\mu_i) \geq 0.$$

The regularizer $\mathbf{f}^T \mathbf{K}^{-1} \mathbf{f} = \sum_{i=1}^{N} \frac{<\mathbf{f}, \phi_i>^2}{g(\mu_i)}$ should restrain the high frequency part and encourage low frequency part, which leads to $g(\mu_i) \geq g(\mu_j)$ for $\mu_i \leq \mu_j$. Different parametric forms of g give different kernels. We study the following three types of kernels that have been proposed in literature [16, 91, 90]:

- Step function kernel [6]: $\begin{cases} \lambda & \mu_i \leq \mu_{cut} \\ 0 & \text{otherwise;} \end{cases}$
- Heat Diffusion kernel: $g(\mu_i) = \lambda e^{-t\mu_i}, \ \ t > 0;$
- Lazy-Random-Walk kernel:[7], $g(\mu_i) = \lambda(\mu_i + \sigma^2)^{-1}$.

For each chosen kernel, there are three parameters to be decided: (1) the radius s in the affinity matrix S; (2) the μ_{cut}, t or σ as parameter in $g(\cdot)$; and (3) the scaling factor λ. The first two parameters, denoted as Θ, can be fit to the pairwise relations Ω with a modified kernel-target alignment. In the original kernel-target alignment [20], we find $\mathbf{K}$ (or equivalently Θ) that maximizes the alignment score:

$$A(\mathbf{K}, \mathbf{T}) = \frac{\langle \mathbf{K}, \mathbf{T} \rangle_F}{\sqrt{\langle \mathbf{K}, \mathbf{K} \rangle_F \ \langle \mathbf{T}, \mathbf{T} \rangle_F}}, \tag{4.34}$$

where $\mathbf{T} \in \mathbb{R}^{N \times N}$ is the target matrix with entry $\mathbf{T}_{ij} = 1$ if $y_i = y_j$, and -1 otherwise. For binary class labels $\{+1, -1\}$, we have $\mathbf{T}_{ij} = y_i y_j$. Unlike class labels, pairwise relations

[6] In practice we use $\mathbf{K} + \epsilon \mathbb{I}$ as the kernel to make it positive definite, here $\epsilon = 0.001\lambda$.

[7] It is also known as Gaussian field kernel [92].

generally do not contain enough information for deciding $\mathbf{T}$. Instead, we try to maximize the expectation of $A(\mathbf{K},\mathbf{T})$ with respect to Y: $\bar{A}(\mathbf{K},\Omega) \doteq \sum_Y P(Y|\Omega)A(\mathbf{K},\mathbf{T})$. It is straightforward to verify that

$$\bar{A}(\mathbf{K},\Omega) = \frac{1}{N}\left\langle \mathbf{K},\bar{\mathbf{T}}\right\rangle_F \left\langle \mathbf{K},\mathbf{K}\right\rangle_F^{-1/2},$$

where $\bar{\mathbf{T}}$ is a $N \times N$ matrix with $\bar{\mathbf{T}}_{ij} = \sum_{y_i,y_j} y_i y_j P(y_i, y_j|\Omega)$. Direct evaluation of $\bar{\mathbf{T}}_{ij}$ can be expensive due to the marginalization in calculating $P(y_i, y_j|\Omega)$. To circumvent, we use a simple approximation for $\bar{\mathbf{T}}_{ij}$ by ignoring the pairwise relations that do not involve $\mathbf{x}_i$ or $\mathbf{x}_j$:

$$\bar{\mathbf{T}}_{ij} \approx \sum_{y_i,y_j} y_i y_j \{\frac{e^{\mathbf{W}_{ij}\delta(y_i y_j)}}{1+e^{\mathbf{W}_{ij}}} \prod_{\substack{k\,:\,\mathbf{W}_{ik}\neq 0\\ \mathbf{W}_{jk}\neq 0}} \sum_{y_k} \frac{e^{\mathbf{W}_{ik}\delta(y_i,y_k)+\mathbf{W}_{jk}\delta(y_j,y_k)}}{(1+e^{\mathbf{W}_{ik}})(1+e^{\mathbf{W}_{jk}})}\}. \tag{4.35}$$

From Equation (4.35), the approximation of $\bar{\mathbf{T}}_{ij}$ is non-zero only if $\mathbf{W}_{ij} \neq 0$ or both $\mathbf{x}_i$ and $\mathbf{x}_j$ connected to some sample x_k. Performing the approximation for entire $\bar{\mathbf{T}}$ requires $O(n^2)$ time, where n is the number of specified pairwise relations. This approximation is cheap since we are particularly interested in the situation where n is small. The scaling factor λ can not be fit this way since it does not affect kernel-target alignment score. In our experiment, we use an empirical λ. More systemic methods, like cross validation, are expected to yield better results.

4.6 Semi-supervised Pairwise Gaussian Process Classifier

We can now combine the likelihood (and its approximation) formulated in Equation (4.15) and (4.22), and a Gaussian prior based on the semi-supervised kernel. As mentioned in Section 4.4, the classification is given by the MAP solution of $\mathbf{f}$. According to Proposition 2, the optimization in Equation (4.24) can be divided into the following two steps:

$$\text{step 1:}\quad \hat{\mathbf{f}}_c = \arg\min_{\mathbf{f}_c}\{\frac{1}{2}\mathbf{f}_c^T\mathbf{K}_c^{-1}\mathbf{f}_c - \sum_{\mathbf{W}_{ij}\neq 0}\log\frac{e^{\mathbf{W}_{ij}}\{e^{f(\mathbf{x}_i)+f(\mathbf{x}_j)}+1\}+e^{f(\mathbf{x}_i)}+e^{f(\mathbf{x}_j)}}{(e^{f(\mathbf{x}_i)}+1)(e^{f(\mathbf{x}_j)}+1)}\} \tag{4.36}$$

$$\text{step 2:}\quad \hat{\mathbf{f}}_u = \mathbf{K}_{uc}\mathbf{K}_c^{-1}\hat{\mathbf{f}}_c. \tag{4.37}$$

Here $\mathbf{K}$ is one of the graph kernels, and both $\mathbf{K}$ and $\mathbf{f}$ are decomposed as in section 4.1. The decomposition (step 1-step 2) effectively reduces the optimization over $\mathbf{f}$ to a subset

$\mathbf{f}_c$, which is substantially cheaper when only a small portion of samples are constrained.

The objective function in step 1 consists of two terms: the empirical error

$$-\sum_{\mathbf{W}_{ij}\neq 0} \log \frac{e^{\mathbf{W}_{ij}}\{e^{f(\mathbf{x}_i)+f(\mathbf{x}_j)}+1\}+e^{f(\mathbf{x}_i)}+e^{f(\mathbf{x}_j)}}{(e^{f(\mathbf{x}_i)}+1)(e^{f(\mathbf{x}_j)}+1)},$$

and regularizer $\frac{1}{2}\mathbf{f}_c^T\mathbf{K}_c^{-1}\mathbf{f}_c$. A closer look at the two terms reveals that the empirical error term favors those $\mathbf{f}$ that are consistent with the pairwise relations. Indeed, if $\mathbf{W}_{ij} > 0$ (link), we tend to have bigger $J(\mathbf{f},\Omega)$ if $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$ are both big (positive) or both small (negative); if $\mathbf{W}_{ij} < 0$ (do-not-link), $J(\mathbf{f},\Omega)$ is bigger when one of $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$ is small (negative) and the other is big (positive). The regularizer term enforces a smooth $\mathbf{f}$. Since $\mathbf{K}$ is non-local, X_u enters into $\mathbf{K}_c$ and therefore affects the optimal $\hat{\mathbf{f}}$. We solve the optimization in step 1 with the quasi-Newton method (Matlab function `fminunc`). To find a good local optimum, we usually try multiple runs with different initial $\mathbf{f}_c$. We name the algorithm Semi-supervised Pairwise Gaussian Process classifier (SPGP). The step-by-step algorithm of SPGP is given as follows:

Semi-supervised Pairwise Gaussian Process Classifier

Input: data set $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ ($x_i \in \mathbb{R}^d$)and pairwise constraints $\mathbf{W}$ ($\in \mathbb{R}^{N\times N}$).

step 1 Construct the affinity matrix S with $S_{ij} = e^{-||\mathbf{x}_i-\mathbf{x}_j||^2/s^2}$, and calculate the normalized graph Laplacian Δ

$$\Delta = \mathbb{I} - D^{-\frac{1}{2}}SD^{-\frac{1}{2}},$$

with D being a diagonal matrix with entry $D_{ii} = \sum_j S_{ij}$.

step 2 Design a semi-supervised kernel $\mathbf{K}$ based on Δ and $\mathbf{W}$ through

$$\mathbf{K}^* = \arg\max_{\mathbf{K}} \frac{\langle \mathbf{K}, \mathbf{T}\rangle_F}{\sqrt{\langle \mathbf{K}, \mathbf{K}\rangle_F \ \langle \mathbf{T}, \mathbf{T}\rangle_F}},$$

where $\mathbf{T}$ is the target matrix approximated from the pairwise relations $\mathbf{W}$

$$\bar{\mathbf{T}}_{ij} = \sum_{y_i,y_j} y_i y_j \{\frac{e^{\mathbf{W}_{ij}\delta(y_i y_j)}}{1+e^{\mathbf{W}_{ij}}} \prod_{\substack{k\,:\,\mathbf{W}_{ik}\neq 0\\ \mathbf{W}_{jk}\neq 0}} \sum_{y_k} \frac{e^{\mathbf{W}_{ik}\delta(y_i,y_k)+\mathbf{W}_{jk}\delta(y_j,y_k)}}{(1+e^{\mathbf{W}_{ik}})(1+e^{\mathbf{W}_{jk}})}\}.$$

step 3 Initialize $\mathbf{f}_c$.

step 4 Perform the optimization in

$$\hat{\mathbf{f}}_c = \arg\min_{\mathbf{f}_c}\{\frac{1}{2}\mathbf{f}_c^T\mathbf{K}_c^{-1}\mathbf{f}_c - \sum_{\mathbf{W}_{ij}\neq 0}\log\frac{e^{\mathbf{W}_{ij}}\{e^{f(\mathbf{x}_i)+f(\mathbf{x}_j)}+1\}+e^{f(\mathbf{x}_i)}+e^{f(\mathbf{x}_j)}}{(e^{f(\mathbf{x}_i)}+1)(e^{f(\mathbf{x}_j)}+1)}\}$$

with a conjugated gradient optimization, and find a local optima of $\mathbf{f}_c$.

step 5 Go back to step 3 until satisfied and denote the obtained local optimum of $\mathbf{f}_c$ as $\hat{\mathbf{f}}_c$.

step 6 Propagate the value of $\mathbf{f}_c$ to $\mathbf{f}_u$ with

$$\hat{\mathbf{f}}_u = \mathbf{K}_{uc}\mathbf{K}_c^{-1}\hat{\mathbf{f}}_c,$$

and obtain $\hat{\mathbf{f}} = \begin{pmatrix}\hat{\mathbf{f}}_c \\ \hat{\mathbf{f}}_u\end{pmatrix}$.

step 7 For $i = 1, 2, \cdots, N$, estimate the probability of label

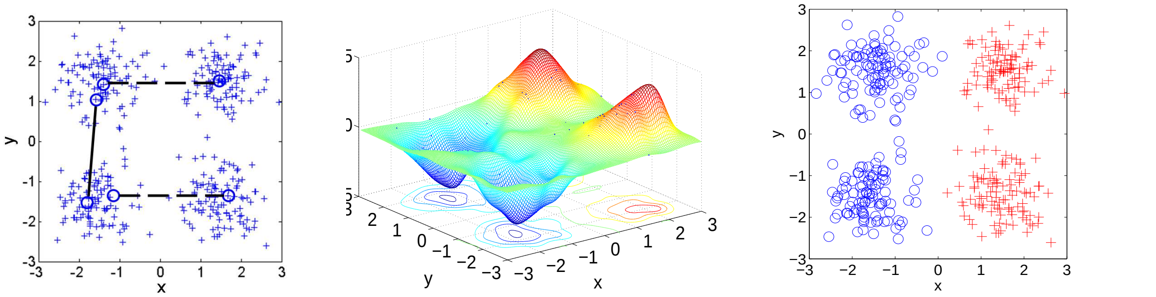
$$P(y|\hat{f}(\mathbf{x}_i)) = \begin{cases} \frac{e^{\hat{f}(\mathbf{x}_i)}}{1+e^{\hat{f}(\mathbf{x}_i)}} & y = +1, \\ \frac{1}{1+e^{\hat{f}(\mathbf{x}_i)}} & y = -1 \end{cases}$$

and assign the sample to the most probable cluster.

A visualization of PGSP solution on 2D toy problem can be found in Figure 4.3. In this toy problem shown in panel (a), there exist two almost equally good partitions of data into two groups(upper two components + lower two components Vs. left two components + right two components). The specified pairwise relations (one link + two do-not-links) biases towards the latter solution. With properly designed kernel, PGSP forces the smoothness of **f** where samples are dense, thus the sign of **f** can only change in the area where samples are sparse, as shown in panel (b). The result given by the MAP solution of **f** leads to the second partition, as shown in panel (c).

Although in this research we limited our discussion to two-class cases, SPGP can be readily generalized to M-class ($M > 2$) situations by using M latent processes.

Unlike constrained clustering algorithms [82, 53, 4], SPGP requires at least one do-not-link to work: with *only* links, SPGP assigns all samples into one class. This weakness



(a) data in x-y plane

(b) MAP solution of **f**

(c) clustering result given by the MAP **f**

Figure 4.3: SPGP on a two-dimensional toy problem. (a) Two do-not-links (dashed line) and one link (solid line) are specified on the data set. (b) A MAP solution of **f** as function of coordinates (x, y). Note that discrete values of **f** (the black dots on the surface) have been interpolated to the x-y plane for visualization purpose. We used heat diffusion kernel for this example. (c) The clustering results given by the MAP **f**. (see context in Section 4.6)

can be alleviated by adding into the objective function an extra term that penalizes the unbalanced distribution of samples among different classes. This extension will not be discussed in this thesis.

4.7 Experiments

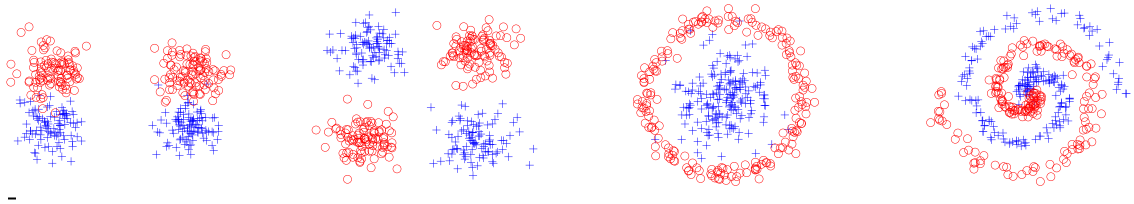
We test SPGP on both artificial data and real-world data, and compare the results with two recently proposed methods: (1) COP-Kmeans [82], a hard-clustering method based on K-means, and (2) Penalized Probabilistic Clustering (PPC) [8](Chapter 2). The pairwise relations are randomly generated. We use a confusion matrix between each clustering result and the label to calculate the classification accuracy. We report the classification accuracy averaged over 30 different realizations of relations.

4.7.1 Artificial Data (Hard Constraints)

The four 2-dimensional artificial data sets (Figure 4.4, classes denoted by symbols) are designed to highlight the problems that cannot be effectively solved by centroid-based

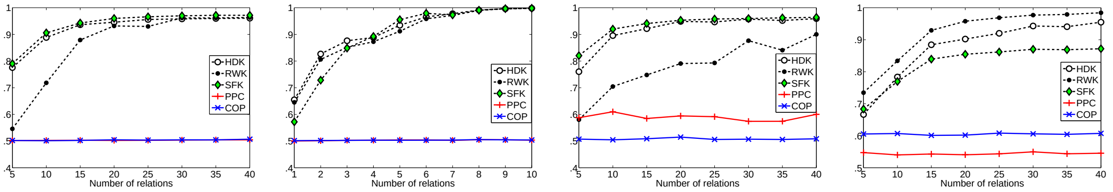
[8] As shown in Chapter 3, the method in [69] is equivalent to PPC with hard constraints, so the result of it is not included.

clustering algorithms. Each data set consists of two classes with 200 data in each class. We consider the pairwise relations highly reliable knowledge and set them to be hard constraints. Intuitively, the classification problems presented in the first two data sets (`Four-Gaussians` and `Noisy-Xor`) can be solved with a constrained clustering, like COP-Kmeans or PPC. However, it requires many pairwise relations to fight with the unconstrained data, which clearly suggest a poor maximum-likelihood solution. The other two data sets (`Doughnut` and `Two-Spirals`) are tasks that are not achievable with a centroid-based clustering algorithm. Figure 4.5 shows classification results for the four data sets with a varying number of pairwise relations. SPGP with all three kernels returns satisfying results with a small number of pairwise relations, whereas PPC and COP-Kmeans do not respond to them at all.



(a) Four-Gaussians (b) Xor (c) Doughnut (d) Two-Spirals

Figure 4.4: Artificial data sets. Classes are denoted by symbols.



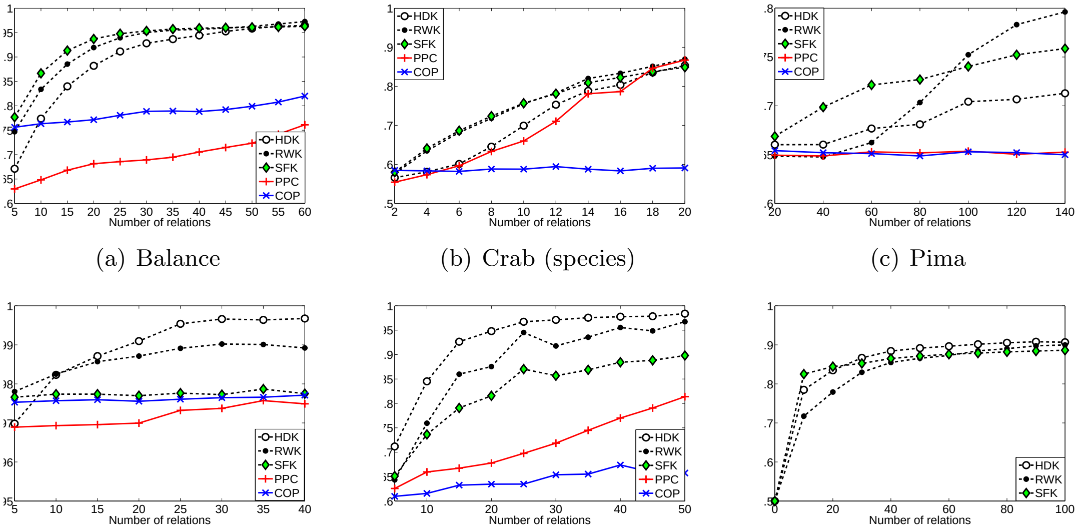
(a) Four-Gaussians (b) Noisy-Xor (c) Doughnut (d) Two-Spirals

Figure 4.5: Classification accuracy Vs. number of relations on artificial data set result. In the legend, HDK: heat diffusion kernel, RWK: lazy-random-walk kernel, SFK: step function kernel, PPC: Penalized Probabilistic Clustering, COP: COP-Kmeans.

4.7.2 Real-World Data (Hard Constraints)

We also present results on six well-known real-world data sets with different characteristics. `Balance-Scale`: we use only class L and R, 576 samples, 5 dimensions; `Crab`(species):

200 samples, 5 dimensions; `Pima`: 768 samples, 8 dimensions; `1-2` and `Small-Big` are handwritten digits recognition tasks with 64 dimension and around 370 samples for each digit. The `1-2` contains digits “1” and “2” in 739 samples. The `Small-Big` is an artificial task with two classes (digits ‘1, 2, 3’ Vs. ‘7, 8, 9’) and 2307 samples. For these two tasks, we use the first 20 principal components as the feature vector for PPC. `Mac-Windows` is a text classification task from the 20-newsgroup data set consisting of 7511-dim TFIDF vectors and 1956 samples. Among these data sets, `Crab` and `1-2` are relatively easy for centroid-based clustering algorithms. `Balance` and `Small-Big` are examples of highly non-Gaussian distribution of samples within each class. `Pima` is difficult even for sophisticated supervised learning methods [83]. `Mac-Windows` has very high-dimensional and sparse feature vectors, which makes PPC and COP-Kmeans inapplicable. Therefore on this data we only present SPGP results. Figure 4.6 summaries the classification results of the three methods. On all data except `Crab`, SPGP outperforms PPC and COP-Kmeans. On `Crab`, SPGP is still the best when pairwise relations are scarce (< 20) whereas PPC gives the highest classification accuracy after 20 relations. We also notice that no single kernel is consistently better than the others.



(a) Balance

(b) Crab (species)

(c) Pima

(d) 1-2

(e) Small-Big

(f) Mac-Windows

Figure 4.6: Classification accuracy Vs. number of relations for real-world data.

4.7.3 Real-World Data (Soft Relations)

We also consider the situation where our pairwise relations come with significant uncertainty. Here we simulate this uncertainty by randomly flipping the specified relations with probability q. We assess the performance of SPGP with soft relations (soft-SPGP) on those noisy relations, and compare it with SPGP with hard constraints (hard-SPGP), PPC with soft relations (soft-PPC) and hard constraints (hard-PPC), and COP-Kmeans. We try two different noise levels: $q = 0.1$ and $q = 0.2$, and set the weight of specified relations for soft-SPGP and hard-PPC using Equation (4.17) with $\gamma_{ij} = q$. Table 1 summarizes the performance of the five algorithms with noisy relations on the six real-world data sets used in Section 4.7.2. For SPGP, we use heat-diffusion kernel. In most occasions, soft-SPGP gives best results among all five methods. In other occasions, hard-SPGP gives slightly better or comparable results.

dataset		Soft-SPGP	Hard-SPGP	Soft-PPC	Hard-PPC	COP-Kmeans
Balance	$q = 0.1$	**0.9568**	0.9477	0.6406	0.6997	0.7870
(60)	$q = 0.2$	**0.9417**	0.6427	0.6406	0.6846	0.7766
Crab	$q = 0.1$	**0.9030**	0.9010	0.8045	0.8933	0.5933
(40)	$q = 0.2$	0.8448	**0.8520**	0.6577	0.7800	0.5902
Pima	$q = 0.1$	**0.7317**	0.6936	0.6510	0.6510	0.6510
(200)	$q = 0.2$	**0.7197**	0.6863	0.6510	0.6510	0.6510
1-2	$q = 0.1$	**0.9955**	0.9922	0.9698	0.9684	0.9753
(30)	$q = 0.2$	**0.9902**	**0.9902**	0.9697	0.9662	0.9740
Small-Big	$q = 0.1$	**0.9357**	0.9332	0.6092	0.7201	0.6542
(40)	$q = 0.2$	**0.9176**	0.6433	0.5928	0.7076	0.6487
Mac-Win	$q = 0.1$	**0.8150**	0.7533	N/A	N/A	N/A
(40)	$q = 0.2$	**0.7799**	0.5580	N/A	N/A	N/A

Table 4.1: Classification accuracy with noisy pairwise relations. Each row contains results for one data set with two different q, and the number of relations is in the parenthesis. The bold face number is the best result among all five methods.

4.8 Discussion and Conclusion

In this Chapter, we discussed a semi-supervised clustering model, called Semi-supervised Pairwise Gaussian process classifiers (SPGP). Experiments on a variety of data sets show that SPGP is much more efficient on using the pairwise constraints, than constraint-based algorithms such as COP-Kmeans and PPC. Moreover, SPGP inherits the modeling flexility of the Gaussian process classifiers, and thus is more suitable for clustering problems in which data from each class is far from a Gaussian.

Despite its efficiency and flexibility, SPGP suffers from the following two limitations:

- The central optimization is not convex and thus prone to local optima. One way to alleviate the problem is to use deterministic annealing. Initializing $\mathbf{f}_c$ with other clustering algorithm, such as PPC, may also help.

- When we have $M(> 2)$ classes, we have to use M Gaussian processes instead of one, which, intuitively, will further increase the complexity of the cost landscape for $\mathbf{f}$, and therefore will increase the chance of getting trapped in poor local optima.

Chapter 5

Constrained Clustering: Contribution and Future Research

5.1 Contribution

In Chapter 2 and 3, we introduced Penalized Probabilistic Clustering (PPC), a constrained clustering model adapted from the Gaussian mixture model we first proposed in NIPS 2004 [53] and later gave a more detailed exposition [54].

In PPC, our belief on the pairwise relations are expressed as a prior probability on the assignments of data points to clusters. Unlike previous effort in this direction, this clustering model naturally accommodates both hard constraints and soft preferences in a framework. We lately extend PPC's modeling flexibility by showing in [54] that PPC provides a rich framework that can encompass many constraint-based models as special cases. The PPC model has been discussed in Chapter 2 and its connection to other constraint-based clustering models has been given in Chapter 3.

Although PPC and its follow-up models are successful in many applications, they also suffer from their limited modeling capability and inefficiency on using the pairwise constraints, as will be discussed in Chapter 2. Our second approach [55], named Semi-supervised Pairwise Gaussian Process classifier (SPGP), is specifically designed to address these two limitations. Instead of adapting a traditional clustering model, we start from the Gaussian process classifiers (GPCs), a type of discriminative models carefully chosen for our specific constrained clustering requirement, and treat the pairwise relations as a special form of observation. The prior probability of the latent process is controlled with a kernel designed using the graph Laplacian of all the available data, thus making use of the samples that are not involved in pairwise relations. Experiments on a variety of data sets show that SPGP is much more efficient on using the pairwise constraints, than constraint-based algorithms such as COP-Kmeans and PPC. Moreover, SPGP inherits the modeling flexility of the Gaussian process classifiers, and thus is more suitable for clustering problems in which data from each class is far from a Gaussian. This approach has been discussed in Chapter 4.

Despite its efficiency and flexibility, SPGP suffers from the following two limitations. First the central optimization is not convex and thus prone to local optima. One way to alleviate the problem is to use deterministic annealing. Initializing $\mathbf{f}_c$ with other clustering algorithm, such as PPC, may also help.Second, when we have $M(>2)$ classes, we have to use M Gaussian processes instead of one, which, intuitively, will further increase the complexity of the cost landscape for $\mathbf{f}$, and therefore will increase the chance of getting trapped in poor local optima.

5.2 Future Research

It is certainly useful to have a constrained version of graph based method, such as normalized cut, or its more heuristical variant, spectral clustering. One obvious choice is to cast it into a constrained optimization problem while using the pairwise relations as the constraints on graph partition, as in [28]. This method and all its variants to facilitate the optimization, share the same weakness with generative model based clustering methods. That is, they are inefficient on using the constraints. Indeed, in a constrained graph partition proposed in [28], the effect of constraints is almost limited to the constrained samples, while those unconstrained samples are largely following a unsupervised graph partition. Simple heuristic has been proposed to remedy this obvious limitation, such as the constrained image segmentation algorithm , called “grouping with bias”, by Yu et al. [88]. In their algorithm, a constrained pixel is enforced to be in the same cluster as its neighbors in a vicinity specified by user. Therefore the pairwise constraints between two pixels are “propagated” to their adjacent pixels in an ad hoc way. One interesting research direction for the future is a more principled way of this constraints propagation so that (1) it can work on non-image data, and (2)no further human interference is needed.

Part II
Cognitive Decline Detection

Chapter 6

Introduction to Cognitive Decline Detection

6.1 Cognitive Decline Detection: Background

With increased life-span, there are more aged people suffering from impairment, for which the loss of cognitive ability has become the most threatening cause. People with cognitive impairment (particularly those at risk for dementia) require substantial assistance even though they are healthy otherwise. It is thus critical to detect cognitive decline quickly and reliably. Early detection of cognitive decline provides the opportunity for more effective medical intervention, planning for compensation strategies, and assistance [12, 33, 66].

Research by our group and others show that motor changes, such as in walking and finger tapping rates, can effectively predict cognitive decline several years before impairment is manifest [15, 35, 79, 84]. The Bronx Aging Study [79] found that subjects with syndromically rated neurologic gait abnormalities had increased risk of non-Alzheimer's dementia but not of Alzheimer's disease (AD). Their Cox proportional regression shows that individuals with any of the considered neurological gait abnormalities are 3.51 times more likely to develop non-Alzheimer's dementia. Other groups have investigated the connection between motor performance and cognitive impairment. Wilson et al. [84] use proportional hazards models to show that progression of parkinsonian signs, revealed by a (modified) Unified Parkinson's Disease Rating Scale (UPDRS), predicts increased risk of developing AD. In a later study from the same group, Aggarwal et al. [1] report that among people with mild cognitive impairment (MCI)[1],those with lower extremity motor performance, Parkinsonian gait, and bradykinesia have a larger risk of developing AD than individuals with good lower limb function.

A large body of literature indicates that there are clinical observations strongly predictive of future cognitive decline years before the symptom becomes manifest. These studies require a longitudinal cohort design such that baseline or early measures of function are then used to prospectively predict those that will develop mild cognitive impairment (MCI) or dementia. This body of work shows that baseline cognitive function assessed up to decades prior to developing dementia is highly predictive of dementia. In general, tests assessing delayed recall [2] of information, as well as more generalized cognitive function, "executive function", or control processes [37, 18, 27] are all predictive. Similar to cognitive function, motor function has also been identified as a complementary, but independent predictor [58, 15, 79, 84] of future cognitive decline and dementia. This has been shown both generally as overall slowing (bradykinesia) or loss of trunk and lower extremity automaticity measured by clinical signs on motor rating scales [84, 57] and more specifically related to gait speed or timed walking [58, 15, 30, 1].

There is a consistent pattern of change in cognitive and motor domains that occurs presymptomatically leading to MCI and dementia. Unfortunately, although these test domains (cognitive and motor) predict group outcomes years later, they are difficult to be applied to individual subjects to determine if they will decline with a degree of certainty that is clinically helpful. This limitation comes from the analysis methodologies as well as the data types employed in analysis to date.

The long-term goal of the research is a detection system that reliably predicts the cognitive decline in the individual, using the motor and cognitive test data from both clinical and in-home assessment technology. Ultimately this detection system should give an estimate of the time horizon to a clinical diagnosis, or the time horizon to decline to more severe impairment for a mildly impaired individual. In addition, each estimation should come with a certainty value to facilitate the integration in a clinical decision system.

[1] See Section 6.3 for the clinical definition.

[2] See Section 6.3 for description.

Our initial aim is to predict if an individual will become impaired, or if already mildly impaired whether or not they will continue to decline.

6.2 Thesis Contribution and Roadmap

In this part of the thesis we will present our initial effort towards a sophisticated and reliable cognitive decline detection system. My contributions include an improved modeling tool for longitudinal clinical data on both population and individual levels, and developing powerful discriminant machinery based on modern statistical learning theory and population-informed models of individual subject time series.

Our work can be roughly divided into two related parts. In the first part, we studied modeling a population of longitudinal time series with the mixed-effect models and propose to apply the fit mixed-effect model (or a mixture of mixed-effect models) as a prior for a new individual. This part of work is summarized in a manuscript in review [56], and will be discussed in full detail in Chapter 7 and 8. The second part of our project is a cross-sectional study partially based on the mixed-effect models. In this part of the study, we attempted to build models predicting whether a cognitively healthy subject will later develop into cognitive impairment based on his or her clinical records. This static classification is useful as an aid for diagnosis, and it can also be viewed as the first step towards a more sophisticated cognitive decline detection system that dynamically estimates a subject's risk of dementia at different age. In Chapter 9, we studied likelihood ratio classifiers based on mixed-effect models. As a justification for modeling the individual effects and the measurement noise seperately, we show that as a classifier the mixed-effect model outperforms a baseline generative model assuming only observation noise. To further improve the classification performance, we also examined the discriminative models based on the state-of-the-art technology, as will be discussed in Chapter 10. We extended the Fisher kernel, a generative model-based kernel design method, to deal with time series with unequal lengths by exploiting the latent structure of mixed-effect models. In addition we developed a new distance measure between time series based on Gaussian processes, functional Bregman divergence, and the theory of reproducing kernel Hilbert space (RKHS).

6.3 Data Description

Our research focuses on using clinical motor behavior and psychometric data from the Oregon Brain Aging Study (OBAS) [31]. The cohort consists of 216 subjects, with 91 males and 125 females. All subjects are healthy at entry, and when the data were drawn 78 of them had developed into mild cognitive impairment (MCI) or worse, while 138 remained cognitively healthy. In this study, we considered a subject as MCI or worse if he or she had two consecutive Clinical Dementia Rating (CDR)[3] score of 0.5 or greater. If the CDR is over 0.5 at least once but never in two consecutive clinical visits, the subjects are tagged *questionable dementia*. We divide all the subjects into the impaired group and the normal group according to their state when the data were drawn from the database. In our current study, we pool the questionable dementia subjects with the normal group.

Since we are mostly interested in the pre-diagnosis prediction, we retain only data taken before a clinical diagnosis of MCI or dementia is made. Thus, when we speak of 'impaired' subjects, we are referring to subjects who become impaired at a later date. Our classification is predictive of future impairment. For a reliable prediction for individual subjects, we only consider subjects with at least four motor measurements before the cut-off date, which reduces the number of qualified subjects to less than 150, with 46 in impaired group and less than 100 (varying with the types of measurements) in normal group.

The measurements used include four motor behaviors (seconds, steps, tappingD, tappingN) and two neuro-psychological tests (delayed-recall, logical memory II)[31]:

- seconds: The time in seconds the subject takes to walk 9 meters ($\sim$ 30 feet). Usually, the value of seconds increases as the subjects age or become impaired. We have 46 qualified subjects in impaired group and 97 in normal group.

- steps: The number of steps the subject takes to walk 9 meters ($\sim$ 30 feet). Usually,

[3] The CDR value can only take values 0, 0.5,1,and 2, while 0 stands for the normal, and the other values stand for increasing level of impairment.

the value of steps increases increases as the subjects age or become impaired. We have 46 qualified subjects in impaired group and 97 in normal group.

- tappingD: The number of the tappings the subject does in 10 seconds (averaged over 3 trials) with his or her dominant hand. Usually, the value of tappingD decreases as the subjects age or become impaired. We have 46 qualified subjects in impaired group and 97 in normal group.

- tappingN: The number of the tappings the subject does in 10 seconds (averaged over 3 trials) with his or her non-dominant hand. Usually, the value of tappingN decreases as the subjects age or become impaired. We have 46 qualified subjects in impaired group and 97 in normal group.

- delayed-recall: The number of words (out of 10) a subject can recall one minute after the words are read to him or her. Usually, the value of delayed-recall decreases as the subjects age or become impaired. We have 46 qualified subjects in impaired group and 86 in normal group.

- logical memory II: The subject is asked to repeat a story that was told 15 $\sim$ 20 minutes ago, and is graded according to the level of matching between the repeated story and the original one. Usually, the value of logical memory II decreases as the subjects age or become impaired, but it is observed that cognitively healthy people can often learn to improve their score in the first 2 $\sim$ 3 years, and levels out afterwards. We have 46 qualified subjects in impaired group and 82 in normal group.

Figure 6.1 shows a sample of the seconds data. In the left panel, we give eight measurements of seconds test (across seven years) of one subject who later developed into dementia, with each measurement plotted as a circle and consecutive measurement pairs connected by a line. In the right panel we plot all the seconds measurements from all 46 subjects in the impaired group. Plot like this are referred to as spaghetti plots.

One of our initial aims is to predict whether an individual will become impaired. The evaluation of the prediction model for this requires the ground-truth about individuals' "final" cognitive state. In our study, we group the subjects based on whether they are

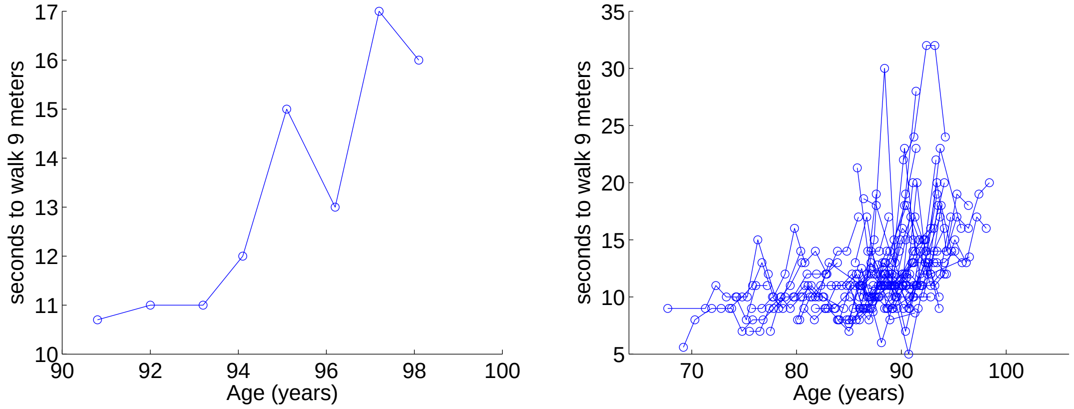


Figure 6.1: Sample spaghetti plots of seconds from the impaired group. Left panel: one example subject with eight measurements of seconds test (across seven years) who later developed into impairment. Each measurement is plotted as a circle and any two consecutive measurements are connected by a line. Right panel: seconds measurements from all 46 subjects in the impaired group.

diagnosed as MCI or dementia when the data were drawn. This grouping is potentially inaccurate due to the possibility that those mentally healthy subjects can later develop into dementia given enough time, which is known as right censoring in the survival analysis [47]. In the light of this, the classification approaches (and the way they are evaluated) we will discuss in Chapter 9 and 10 should be considered as an approximation. We expect that the future extension of our method will enable us to predict for each subject the probability that he or she becomes impaired at any future age, which is then consistent with the survivor function in the classic survival analysis.

Notation

Following are some notation conventions we will adopt through this part of thesis. (Note that they might be different from the notation we used in Chapter 1- Chapter 4.)

- We use $P(A)$ to denote the probability of discrete event A, and $p(x)$ as the continuous probability density at x.

- We use $P(A;\Theta)$ (or $p(x;\Theta)$) to denote the probability of A (density of x) with model parameter Θ. We use $P(A|B;\Theta)$ (or $p(x|B;\Theta)$) to denote the probability of A (density of x) conditioned on event B with model parameter Θ.

- We use $\mathbb{I}$ to denote the identity matrix, and save the notation $\mathbf{I}$ for Fisher information matrix (Chapter 10).

- We use a superscript for index of subjects and subscript for the index of time, for example, we use y^i_n for the n^{th} observation of the i^{th} subject taken at t^i_n. Also, we use $\mathbf{y}^i$ to denote the *set* of observations of the i^{th} subject, that is

  $$\mathbf{y}^i = [y^i_1 \; y^i_2 \; \cdots \; y^i_{N_i}]^T$$

  and $\mathbf{t}^i$ is the corresponding age when the observations are made

  $$\mathbf{t}^i = \{t^i_1, t^i_2, \cdots, t^i_{N_i}\}$$

  where N_i is the total number of observations available for the i^{th} subject. We use the same symbol without a superscript to denote the a sample in test set, often for the out-of-sample testing of prediction or classification models. For example, we use $\mathbf{t}$ and $\mathbf{y}$ to denote the time and clinic observations associated with a subject not used for model training.

- We use $\tilde{\Theta}$ to denote the parameters for the mixture models with the parameters and prior (mixing proportion) of the m^{th} component denoted as Θ_m and π_m. That is

  $$\tilde{\Theta} = \{\pi_1, \Theta_1, \pi_2, \Theta_2, \cdots, \pi_M, \Theta_M\}.$$

  where M is the number of components in the mixture model.

- We use boldface letter for vectors and matrices. However when the matrix or vector is a function evaluated on an ordered set of inputs, we still use regular font to denote that function. For example, we use $K(x, y)$ to denote a kernel function but $\mathbf{K}$ to denote a kernel matrix.

- Suppose $f(\cdot)$ is a regular function with scalar output, we use $f(\mathbf{x})$ with $\mathbf{x} = [x_1, x_2, \cdots, x_n]$ to denote the vector $[f(x_1), f(x_2), \cdots, f(x_n)]$. Similarly, when $f(x, y)$ is a scalar function of both x and y, we use $f(\mathbf{x}, \mathbf{y})$ to denote the following matrix

$$\begin{pmatrix} f(x_1, y_1) & f(x_1, y_2) & \cdots & f(x_1, y_m) \\ f(x_2, y_1) & f(x_2, y_2) & \cdots & f(x_2, y_m) \\ \vdots & \vdots & \vdots & \vdots \\ f(x_n, y_1) & f(x_n, y_2) & \cdots & f(x_2, y_m) \end{pmatrix}$$

with $\mathbf{x} = [x_1, x_2, \cdots, x_n]$ and $\mathbf{y} = [y_1, y_2, \cdots, y_m]$.

Chapter 7

Mixed-effect Model for Longitudinal Data

Mixed-effect models provide a flexible and powerful tool for the analysis of a set of similar data, which arises in diversified areas. Among their appealing properties, mixed-effect models offer the flexibility to model the within-group correlation often present in a population. Mixed-effect models have long been used for analyzing longitudinal data [48, 23], and are a suitable modeling tool for the clinical data in our research. Most interestingly, it provides a principled way to summarize a population and thus a means to compare the characteristic of different populations. This property is of fundamental importance to us since we need to discriminate the group of cognitively impaired people from the group of healthy people. In this chapter, we will give a brief introduction to the mixed-effect model, with a great emphasis on linear mixed-effect models. The mixed-effect models on the longitudinal data in OBAS research will be given.

7.1 Regression Models

In this thesis, we confine attention to parametric regression[1]. Suppose there are k individuals (indexed by $i = 1, \ldots, k$) contributing data to the sample, and we have observations $\{t_n^i, y_n^i\}$, $n = 1, \ldots, N^i$ as a function of time t for individual i. Suppose that the data are modeled as

$$y_n^i \;=\; f(t_n^i; \gamma^i) \;+\; \epsilon_n^i \tag{7.1}$$

where γ^i are the regression parameters and ϵ_n^i is zero-mean white Gaussian noise with (unknown) variance σ^2. The superscript on the model parameters γ^i indicates that the

[1] Nonparametric mixed-effect regression is discussed by Guo [32].

generative model is different for each individual contributing to the population. Since the model parameters vary across individuals, it is natural to consider them generated by the sum of a fixed and a random piece

$$\gamma^i \;=\; \alpha \;+\; \beta^i \tag{7.2}$$

where β^i is assumed distributed $\mathcal{N}(0, \mathbf{D})$ with unknown covariance $\mathbf{D}$. The expected parameter vector α, called *fixed effect* or *population model*, determines the model for the population as a whole. This intuition is most precise for the linear case

$$f(t;\gamma) \;=\; \gamma^T\,\Phi(t) \;=\; \alpha^T\,\Phi(t) \;+\; \beta^T\,\Phi(t) \tag{7.3}$$

where $\Phi(t) = [\phi_1(t), \phi_2(t), ..., \phi_d(t)]^T$ denotes a vector of basis functions[2]. Using $\mathcal{M} = \{\alpha, \mathbf{D}, \sigma\}$ to denote the mixed-effect model, the generative process of observations $\mathbf{y}^i(= [y^i_1, y^i_2, \cdots, y^i_{N^i}]^T)$ at times $\mathbf{t}^i(= [t^i_1, t^i_2, \cdots, t^i_{N^i}]^T)$, can be summarized with the graphical model in Figure 7.1 with plate notation [3]. Model fitting will use the entire collection of data $\{\mathbf{t}^i, \mathbf{y}^i\}$, $i = 1, \ldots, k$ to determine the parameters $\mathcal{M}$ by maximum likelihood considering the random effects $\{\beta^i\}$ as latent variables.

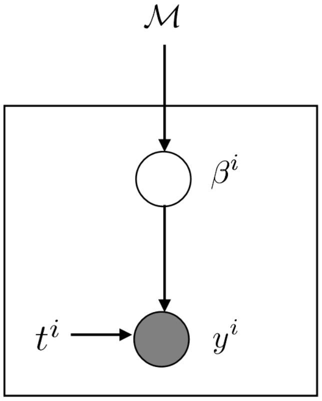


Figure 7.1: The graphical model of the mixed-effect model.

[2] More generally, the fixed and random effects can be associated with different basis functions.

[3] The plate notation simply groups together nodes that are duplicated–that is, have the same structure inside the plate and are linked to nodes outside the plate in the same way. In this thesis we use plate notation since all the individuals are generated the same way.

7.2 Maximum Likelihood Fitting

The likelihood of the data $\{\mathbf{t}^i, \mathbf{y}^i\}$ given the mixed-effect model $\mathcal{M} = \{\alpha, \mathbf{D}, \sigma\}$ is

$$\begin{aligned} p(\mathbf{y}^i;\, \mathbf{t}^i, \mathcal{M}) &= \int p(\mathbf{y}^i|\, \beta^i; \mathbf{t}^i, \sigma) p(\beta^i|\, \mathcal{M}) d\beta^i \\ &= (2\pi)^{-N^i/2} |\Sigma^i|^{-1/2} \exp((\mathbf{y}^i - \alpha^T \Phi(\mathbf{t}^i))^T (\Sigma^i)^{-1} (\mathbf{y}^i - \alpha^T \Phi^i(\mathbf{t}^i))) \end{aligned}$$

where

$$\begin{aligned} \Sigma^i &= \sum_{n=1}^{N^i} \Phi(t_n^i) \mathbf{D} \Phi(t_n^i)^T + \sigma^2 \mathbf{I} \\ \Phi(\mathbf{t}^i) &= [\Phi(t_1^i), \Phi(t_2^i), \cdots, \Phi(t_n^i)]^T \end{aligned}$$

The data likelihood for $\mathbf{Y} = \{\mathbf{y}^1, \mathbf{y}^2, \cdots, \mathbf{y}^k\}$ with $\mathbf{T} = \{\mathbf{t}^1, \mathbf{t}^2, \cdots, \mathbf{t}^k\}$ is then

$$p(\mathbf{Y};\, \mathbf{T}, \mathcal{M}) = \prod_{i=1}^{k} p(\mathbf{y}^i|\, \mathbf{t}^i; \mathcal{M}).$$

The maximum likelihood solution of $\{\alpha, \mathbf{D}, \sigma\}$ can be found with the Expectation-Maximization algorithm [24] with $\{\beta^1, \beta^2, \cdots, \beta^k\}$ considered as the latent variable:

$$\text{E-step:} \qquad Q(\mathcal{M}, \mathcal{M}^g) = E_{\{\beta^i\}}[\log p(\mathbf{Y}, \{\beta^i\}; \mathbf{T}, \mathcal{M}) | \mathbf{Y}; \mathbf{T}, \mathcal{M}^g] \tag{7.4}$$

$$\text{M-step:} \qquad \mathcal{M} = \arg\max_{\mathcal{M}} Q(\mathcal{M}, \mathcal{M}^g), \tag{7.5}$$

where $\mathcal{M}^g$ stands for the estimation of the mixed-effect model obtained in previous step and the expectation in the E-step is with respect to the posterior distribution of on $\{\beta^i\}$ when $\mathbf{Y}$ is known and the model parameter is $\mathcal{M}^g$. For the linear mixed-effect model in Equation (7.3), the M-step has a closed form:

$$\alpha = (\sum_{i=1}^{k} \Phi(\mathbf{t}^i)^T \Phi(\mathbf{t}^i))^{-1} \sum_{i=1}^{k} \sum_{n=1}^{N^i} (\mathbf{y}_n^i - E(\beta^i | \mathbf{y}^i, ; \mathbf{t}^i, \mathcal{M}^g)^T \Phi(\mathbf{t}^i)) \tag{7.6}$$

$$\mathbf{D} = \frac{1}{k} \sum_{i=1}^{k} E(\beta^i (\beta^i)^T | \mathbf{y}^i; \mathbf{t}^i, \mathcal{M}^g) \tag{7.7}$$

$$\sigma^2 = \frac{1}{\sum_{i=1}^{k} N^i} \sum_{i=1}^{k} E(||\epsilon^i||^2 \,|\, \mathbf{y}^i; \mathbf{t}^i, \mathcal{M}^g) \tag{7.8}$$

The calculation of the expectations in Equation (7.6)-(7.8), performed in the E-step, are straightforward since they are all expectation of linear or quadratic function of Gaussian variables. We first notice that

$$\hat{\beta}^i \equiv E(\beta^i|\mathbf{y}^i, ; \mathbf{t}^i, \mathcal{M}^g) = \mathbf{D}^g \Phi(\mathbf{t}^i)^T \mathbf{W}^i (\mathbf{y}^i - (\alpha^g)^T \Phi(\mathbf{t}^i)) \tag{7.9}$$

$$\hat{\epsilon}^i \equiv E(\epsilon^i|\mathbf{y}^i, ; \mathbf{t}^i, \mathcal{M}^g) = \mathbf{y}^i - (\alpha^g + E(\beta^i|\mathbf{y}^i, ; \mathbf{t}^i, \mathcal{M}^g))^T \Phi(\mathbf{t}^i), \tag{7.10}$$

where $\mathbf{W}^i \equiv ((\sigma^g)^2 \mathbb{I} + \Phi(\mathbf{t}^i) \mathbf{D}^g \Phi(\mathbf{t}^i)^T)^{-1}$ with $\mathbb{I}$ standing for the identity matrix. It is easy to see that Equation (7.6) and (7.9) give the closed-form update for α in the M-step. After some algebra, we can rewrite Equation (7.7)-(7.8) as

$$\mathbf{D} = \mathbf{D}^g + \frac{1}{k}(\sum_{i=1}^{k} \hat{\beta}^i (\hat{\beta}^i)^T - \mathbf{D}^g \sum_{i=1}^{k} \Phi(\mathbf{t}^i)^T \mathbf{W}^i \Phi(\mathbf{t}^i)) \tag{7.11}$$

$$\sigma^2 = (\sigma^g)^2 + \frac{1}{\sum_{i=1}^{k} N^i}(\sum_{i=1}^{k} ||\hat{\epsilon}^i||^2 - (\sigma^g)^4 tr(\sum_{i=1}^{k} \mathbf{W}^i)). \tag{7.12}$$

The maximum likelihood fitting of mixed-effect model can be summarized as follows:

Fitting Mixed-effect Model

Input: Observations $\{\mathbf{t}^i, \mathbf{y}^i\}_{i=1}^{k}$ $(\mathbf{t}^i, \mathbf{y}^i \in \mathbb{R}^{N_i}, i = 1, 2, \cdots, k)$from a population of subjects.

step 0 Select the basis functions $\Phi(t) = [\phi_1(t), \phi_2(t), ..., \phi_d(t)]^T$.

step 1 Initialize the mixed-effect model parameters $\mathcal{M} = \{\alpha, \mathbf{D}, \sigma\}$, where $\mathbf{D}$ has to be positive definite.

step 2 Let $\mathcal{M}^g = \mathcal{M}$ and calculate the $Q(\mathcal{M}, \mathcal{M}^g) = E_{\{\beta^i\}_{i=1}^{k}|\mathbf{Y}}[\log p(\mathbf{Y}; \mathbf{T}, \mathcal{M})|\mathbf{Y}, \mathcal{M}^g]$.

step 3 Find $\mathcal{M} = \arg\max_{\mathcal{M}} Q(\mathcal{M}, \mathcal{M}^g)$ using

$$\alpha = (\sum_{i=1}^{k} \Phi(\mathbf{t}^i)^T \Phi(\mathbf{t}^i))^{-1} \sum_{i=1}^{k} \sum_{n=1}^{N^i} (\mathbf{y}_n^i - \mathbf{D}^g \Phi(\mathbf{t}^i)^T \mathbf{W}^i (\mathbf{y}^i - (\alpha^g)^T \Phi(\mathbf{t}^i)))$$

$$\mathbf{D} = \mathbf{D}^g + \frac{1}{k}(\sum_{i=1}^{k} \hat{\beta}^i (\hat{\beta}^i)^T - \mathbf{D}^g \sum_{i=1}^{k} \Phi(\mathbf{t}^i)^T \mathbf{W}^i \Phi(\mathbf{t}^i))$$

$$\sigma^2 = (\sigma^g)^2 + \frac{1}{\sum_{i=1}^{k} N^i}(\sum_{i=1}^{k} ||\hat{\epsilon}^i||^2 - (\sigma^g)^4 tr(\sum_{i=1}^{k} \mathbf{W}^i)),$$

where $\mathbf{W}^i \equiv ((\sigma^g)^2 \mathbb{I} + \Phi(\mathbf{t}^i) \mathbf{D}^g \Phi(\mathbf{t}^i)^T)^{-1}$.

step 4 Go back to step 2 until convergence. Return $\mathcal{M}$.

7.2.1 Restricted Maximum Likelihood Fitting

We may also consider the fixed effect α as a Gaussian variable $\sim \mathcal{N}(\bar{\alpha}, \mathbf{A})$, which leads to a mixed-effect model with parameters $\mathcal{M} = \{\bar{\alpha}, \mathbf{A}, \mathbf{D}, \sigma\}$. The data likelihood is then

$$p(\mathbf{Y}; \mathcal{M}) \;=\; \prod_{i=1}^{k} \int p(\alpha | \mathcal{M}) \int p(\mathbf{y}^i | \alpha, \beta^i; \mathbf{t}^i, \sigma) p(\beta^i | \mathcal{M}) d\beta^i d\alpha,$$

which is called restricted maximum likelihood (REML) for mixed-effect models. Like the maximum likelihood case, the REML fitting can also be done with EM algorithm (See Appendix C for the details).

It is argued by Laird and Ware [48] that REML is more robust when the observations are sparse. In our experiments, the ML fitting and REML fitting return very similar models. That is, from REML fitting we get $\bar{\alpha}$ close to α, a very small $||\mathbf{A}||$, and the estimated random effect covariance $\mathbf{D}$ close to that from ML fitting.

7.3 Mixed-effect Models on OBAS Data

In this section we present the mixed-effect models trained to maximize the data likelihood. We use the linear mixed-effect model with the polynomial basis function as $\Phi(t) = [1, t]^T$ (order 1) and $\Phi(t) = [1, t, t^2]^T$ (order 2). From Equation (7.3) the order-1 regression is a linear function of t, and the order-2 regression model is a quadratic function[4] of t. We considered four motor observations (steps, seconds,tappingD, tappingN) and two representative neuro-psychological test scores (delayed-recall, logical memory II). For the four motor behavior measurements, we use the logarithm of measurement as the output to achieve a better symmetry of the residual.

We show in Figure 7.3 the fit mixed-effect models for the four motor behaviors, in Figure 7.4 the models for the two neuro-psychological test. The model parameter of the fit mixed-effect models are given in Appendix B. The mixed-effect models are visualized by plotting the fixed effect regression $\alpha^T \Phi(t)$ (red curve), and the expected standard deviation of individual model from random effect (green curves) and measurement noise

[4] Do not confuse the linearity to the input variable t with the linearity with respect to the model parameter γ.

(dashed black curve, see caption). Clearly, for steps and seconds, the fixed effect for the impaired group shows a faster pace of increase than the one for the normal group. For tappingD and tappingN, the fixed effect model for the impaired group decreases faster than the one for the normal group, while for tappingN the difference is less salient. Also for tappingN and tappingD, the variance from the random effect in impaired group is significantly larger than the one in normal group, while this effect is less obvious for the other four measurements. For the two neuro-psychological test scores, the difference between models of impaired group and normal group are significant with both order 1 and order 2 regression models. Interestingly, for the logical memory II, we observe an increasing performance in the fixed effect for the normal group, which is explained by the fact that normal people can learn to do the test and improve their performance in the first $2 \sim 3$ years and their performance usually level out afterwards.

From Figure 7.3 and 7.4, it appears that the mixed-effect models based on quadratic regression model fit the data better than the one based on linear model. However, our later experiments show that the mixed-effect model based on quadratic regression model does not bring a significant improvement for individual time series modeling, a result we will discuss in Chapter 8. Moreover, for the discrimination between the two groups, as we will discuss in Chapter 9, the quadratic model actually performs worse than the linear models. There are two reasons, we speculate, for this: First, there are more parameters in $\mathbf{D}$ to estimate for the quadratic model and we may not have enough samples for a reliable estimation; Second, the distribution of the random effect parameter β may not be well captured by a higher dimensional Gaussian. In the next three chapters, we will limit our discussion to mixed-effect linear regression models.

7.4 Extensions of the Mixed-effect Model

7.4.1 Mixture of Mixed-effect Models

A population may consist of several sub-populations with different characteristics. Indeed, as shown in Section 7.3, the motor ability of individuals destined to become cognitively impaired declines more dramatically than in individuals that remain cognitively healthy

[15, 58]. It is sensible to describe the population with people from the two groups with a mixture of two mixed-effect models[5]: one fit on the normal group (denoted $\mathcal{M}_0$) and one fit on impaired group (denoted $\mathcal{M}_1$), with

$$\mathcal{M}_m = \{\alpha_m, \mathbf{D}_m, \sigma_m\}, \;\; m = 0, 1.$$

Here, we use $\widetilde{\mathcal{M}} = \{\pi_0, \mathcal{M}_0, \pi_1, \mathcal{M}_1\}$ to denote the parameters of this mixture, where π_0 and π_1 are the mixing proportions (prior) estimated from the individuals in the training data. The overall generative process for any individual time series $(\mathbf{t}^i, \mathbf{y}^i)$ can be summarized as a graphical model shown in Figure 7.2. Here $z^i \in \{0, 1\}$ is the latent variable indicating which mixed-effect model component is used in generating y^i. The generative process consists of the following three steps:

1. We randomly set the value of z^i as in $\{0, 1\}$ with probability π_0 and π_1. (That is equivalent to choosing the generative component from the mixture.)

2. We randomly draw γ^i from the Gaussian distribution $\mathcal{N}(\alpha_{z^i}, \mathbf{D}_{z^i})$, where α_{z^i} and $\mathbf{D}_{z^i}$ are respectively the fixed effect and the covariance of the random effect in model $\mathcal{M}_{z^i}$.

3. Let $y_n^i = (\gamma^i)^T \Phi(t_n^i) + \epsilon_n^i$, where ϵ_n^i is drawn from $\mathcal{N}(0, \sigma_{z^i}^2)$.

7.4.2 Mixed-effect Model with Multiple Output Variables

Different types of clinical measurements provide different but related information about cognitive decline, and it is thus desirable to fuse these information sources. Not surprisingly, this can be done with the mixed-effect model with multiple output variables, which assumes for each output variable a different regression model, coupled through the correlation between the regression model parameters. With a simple transformation, this multiple output case can be described with the model same as the single output case. Assuming that for each individual i, there are L types of measurements we want to model jointly

$$\{t_n^{(l),i}, y_n^{(l),i}\}, \, l = 1, \cdots, L, \;\; n = 1, \ldots, N^{(l),i},$$

[5] It is straightforward to construct such a mixture with more than two components.

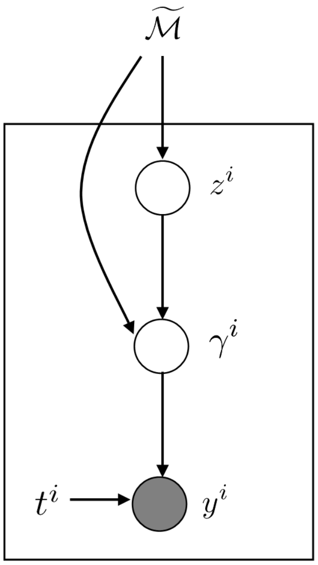


Figure 7.2: The graphical model of the mixture of mixed-effect models.

which are individually modeled as

$$\begin{aligned} y_n^{(1),i} &= \alpha^{(1)}\Phi^{(1)}(t_n^{(1),i}) + \beta^{(1),i}\Phi^{(1)}(t_n^{(1),i}) + \epsilon_n^{(1),i},\ n = 1, \cdots, N^{(1),i} \\ &\cdots \qquad\qquad\qquad\qquad\qquad\qquad \cdots \\ y_n^{(L),i} &= \alpha^{(L)}\Phi^{(L)}(t_n^{(L),i}) + \beta^{(L),i}\Phi^{(L)}(t_n^{(L),i}) + \epsilon_n^{(L),i},\ n = 1, \cdots, N^{(L),i} \end{aligned}$$

where $\alpha^{(1)}$ and $\beta^{(l),i}$ are respectively the fixed and random effect (for individual i) for measurement of type l, and $\Phi^{(l)}(\cdot)$ are the corresponding basis functions. Note it is not required that different measurements are made the same time.

We first concatenate the output variables into a longer one

$$\bar{\mathbf{y}}^i = \{y_1^{(1),i}, \cdots, y_{N^{(1),i}}^{(1),i},\ \cdots,\ y_1^{(L),i},\ \cdots,\ y_{N^{(L),i}}^{(L),i}\}$$

with observation times

$$\bar{\mathbf{t}}^i = \{t_1^{(1),i}, \cdots, t_{N^{(1),i}}^{(1),i},\ \cdots,\ t_1^{(L),i}, \cdots, t_{N^{(L),i}}^{(L),i}\}.$$

For the mixed-effect model, we let

$$\begin{aligned}
\bar{\alpha} &= [\alpha^{(1)}; \cdots; \alpha^{(L)}] \\
\bar{\beta}^i &= [\beta^{(1),i}; \cdots; \beta^{(L),i}], \\
\bar{\Phi}^i &= \begin{bmatrix} \Phi^{(1)}(\mathbf{t}^{(1),i}) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \Phi^{(2)}(\mathbf{t}^{(2),i}) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \Phi^{(L)}(\mathbf{t}^{(L),i}) \end{bmatrix} \\
\bar{\epsilon}^i &= [\epsilon_1^{(1),i}, \epsilon_2^{(1),i}, \cdots, \epsilon_{N^{(1),i}}^{(1),i}, \ \cdots, \ \epsilon_1^{(L),i}, \epsilon_2^{(L),i}, \cdots, \epsilon_{N^{(L),i}}^{(L),i}]^T
\end{aligned}$$

The concatenated vector $\bar{\mathbf{y}}^i$ is modeled as follows, with $\bar{\mathbf{y}}^i = [\bar{y}_1^i, \bar{y}_2^i, \cdots, \bar{y}^i_{\sum_{l=1}^L N^{(l),i}}]^T$,

$$\bar{y}_n^i = \begin{cases} \alpha^{(1)}\Phi^1(t_n^{(1),i}) + \beta^{(1),i}\Phi^1(t_n^{(1),i}) + \epsilon_n^{(1),i} & 1 \leq n \leq N^{(1),i} \\ \cdots & \cdots \\ \alpha^{(L)}\Phi^{(L)}(t^{(L),i}_{n-\sum_{l=1}^{L-1} N^{(l),i}}) + \beta^{(L),i}\Phi^{(L)}(t^{(L),i}_{n-N^{(L),i}})\epsilon^{(L),i}_{n-\sum_{l=1}^{L-1} N^{(l),i}} & \sum_{l=1}^{L-1} N^{(l),i} + 1 \leq n \leq \sum_{l=1}^{L} N^{(l),i}. \end{cases}$$

which is equivalent to

$$\bar{\mathbf{y}}^i = (\bar{\alpha})^T \bar{\Phi}^i + (\bar{\beta}^i)^T \bar{\Phi}^i + \bar{\epsilon}^i. \tag{7.13}$$

Clearly, Equation (7.13) is a single-variable regression model, and the set of measurements $\{\bar{\mathbf{y}}^i, \bar{\mathbf{t}}^i\}$ can thus be modeled with the single-variable mixed-effect model described above. If we further assume $\bar{\beta}^i$ and $\epsilon^{(l),i}$ $(l = 1, 2, \cdots, L)$ are independent Gaussian variables with variance $\mathbf{D}$ and $(\sigma^{(l)})^2$, we can optimize over the parameter $\overline{\mathcal{M}} = \{\bar{\alpha}, \mathbf{D}, \sigma^{(1)}, \cdots, \sigma^{(L)}\}$ with the EM algorithm described above. Note that the different measurements are coupled by the joint covariance matrix $\mathbf{D}$ [6]. This joint mixed-model for multiple measurements will be referred to as the *multivariate mixed-effect model* as opposed to the *univariate*

[6] When we ignore the correlation between random effects corresponding to different types of measurements, we get a block-diagonal $\mathbf{D}$. That is equivalent to assume all different type of measurements from the same individual are independent

$$p(\bar{\mathbf{y}}^i; \bar{\mathbf{t}}^i, \overline{\mathcal{M}}) = \prod_{l=1}^{L} p(\mathbf{y}^{(l),i}; \mathbf{t}^{(l),i}, \mathcal{M}^{(l)}), \tag{7.14}$$

where $\mathcal{M}^{(l)}$ is the mixed-effect model for the l^{th} type of measurement. The generative model in Equation (7.14) is called a naive Bayesian model [59].

mixed-effect model, as we described in Section 7.1 and 7.2, although the modeling and training of which are not vastly different from that in a single type case.

In our experiments, we tried to combine two or more measurements out of the six measurements, for example seconds and tappingD. We expect to gain some extra information about one measurement from observing the other and thus achieve a better prediction, with the help of the correlation between them modeled by the $\mathbf{D}$ in the joint mixed-effect model. Unfortunately, our experiments show that this information gain is insignificant. The reason, we speculate, is that the distribution of the joint random effect can not be well captured by the higher dimensional Gaussian distribution $N(\bar{\alpha},, \mathbf{D})$ in the multivariate mixed-effect model. Note that this situation is not caused by over-fitting (although it might exist) since the information gain is measured on training data, and therefore cannot be solved by putting prior on $\mathbf{D}$. The classification results in Chapter 9 and Chapter 10 suggest the similar futility of combining multiple measurements with the current configuration of mixed-effect model. In the rest of this thesis, we will limit ourselves to modeling each type measurement separately.

7.5 Applications of Mixed-effect Model

In Chapter 8, 9 and 10, we will discuss the two applications of the mixed-effect models. We know from this chapter that mixed-effect models can be used as the generative models for the longitudinal time series. We can get an informative prior for the regression model of the new time series by assuming it is generated from the same mixed-effect model. We will explore this idea in Chapter 8 and show that the learnt prior significantly improve the quality of regression. We have demonstrated the mixed-effect models as a means to manifest the difference between the normal group and the impaired group on a population level. We further show that the mixed-effect model can also be used in differing the two group on an individual level. Indeed, in Chapter 9, we build a probabilistic classifier of time series by directly exploiting the generative property of mixed-effect models. In Chapter 10, we will discuss the potential advantages and drawbacks of classification based on mixed-effect models, and in the same chapter, we show how to avoid those drawbacks by combining them with a discriminative model.

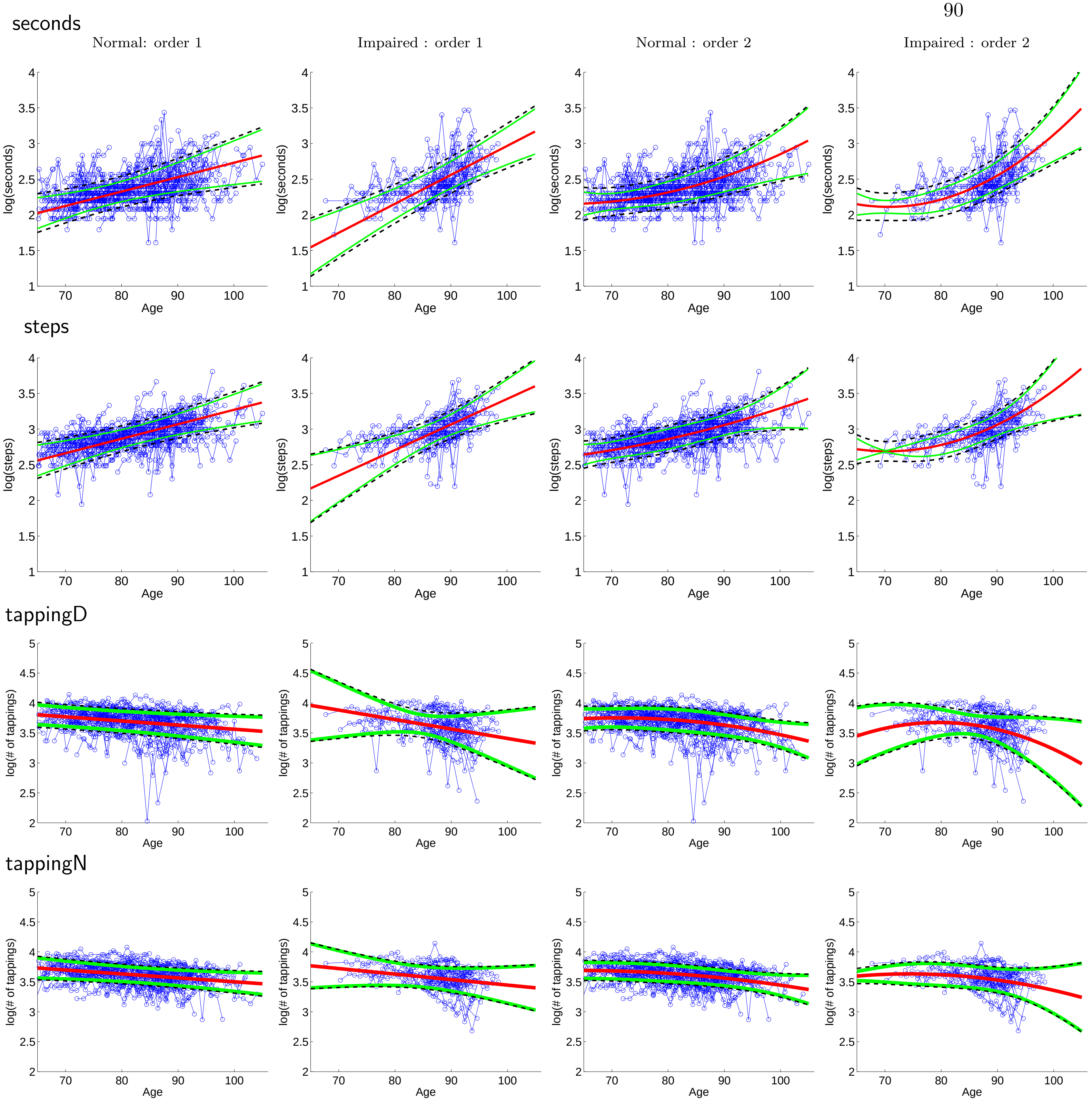


Figure 7.3: The fit mixed-effect models for the four motor behaviors. The linear mixed model with basis function $\Phi(t)$ as order 1 (linear) and order 2 (quadratic) polynomials of t. In each panel, the red line stands for the model given by the fixed effect $\alpha^T\Phi(t)$. The two green lines stand for $\alpha^T\Phi(t) \pm \sqrt{\Phi^T(t)\mathbf{D}\Phi(t)}$, i.e., the population model ± the s.t.d. of the deviation from the uncertainty of the β. The black dash line is the s.t.d of the deviation when we consider the observation noise ϵ, that is $\alpha^T\Phi(t) \pm \sqrt{\Phi^T(t)\mathbf{D}\Phi(t) + \sigma^2}$

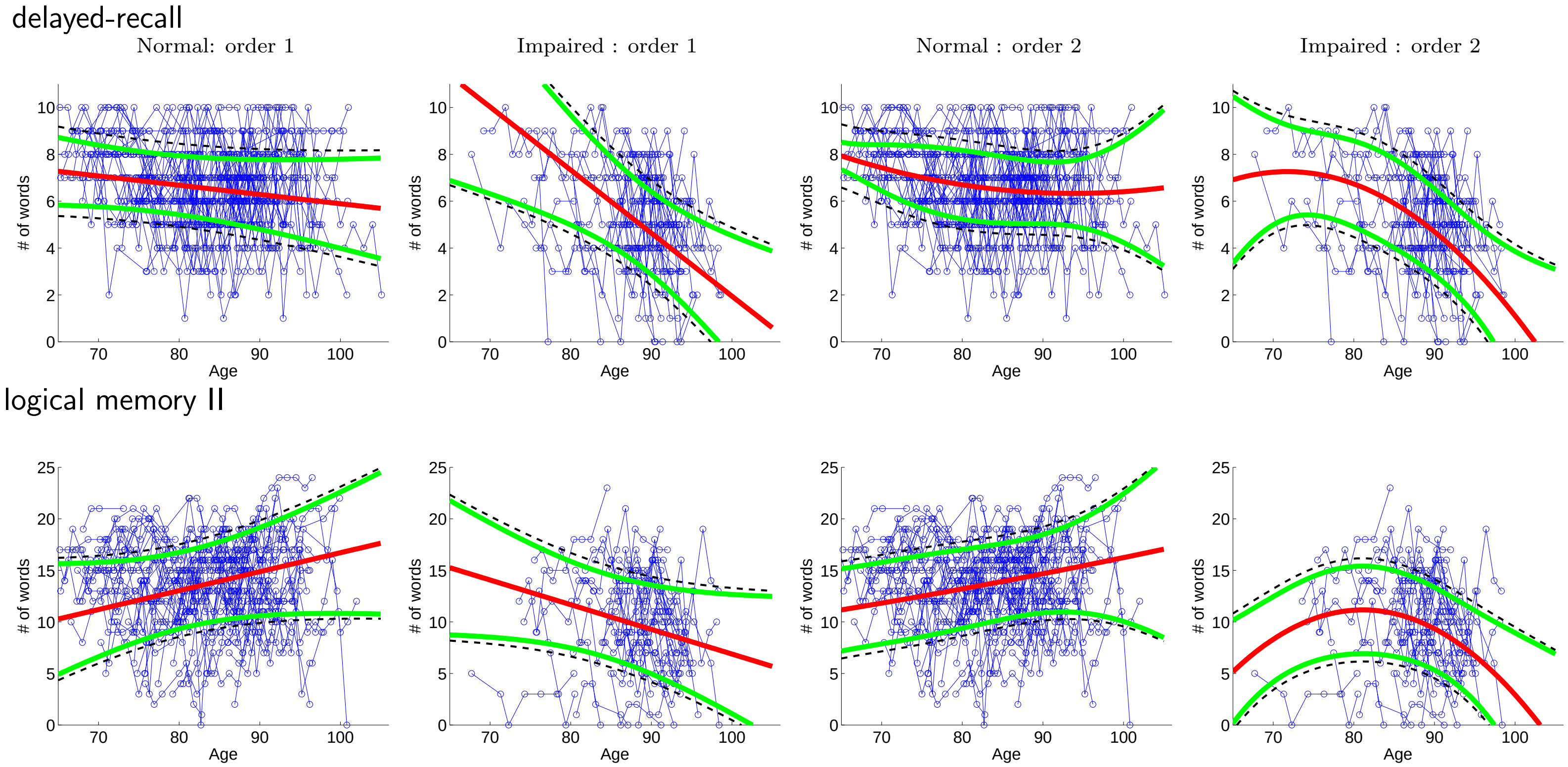


Figure 7.4: The fit mixed-effect models for the two neuro-psychological tests. The linear mixed model with basis function $\Phi(t)$ as order 1 (linear) and order 2 (quadratic) polynomials of t. In each panel, the red line stands for the model given by the fixed effect $\alpha^T\Phi(t)$. The two green lines stand for $\alpha^T\Phi(t) \pm \sqrt{\Phi^T(t)\mathbf{D}\Phi(t)}$, i.e., the population model ± the s.t.d. of the deviation from the uncertainty of the β. The black dash line is the s.t.d of the deviation when we consider the observation noise ϵ, that is $\alpha^T\Phi(t) \pm \sqrt{\Phi^T(t)\mathbf{D}\Phi(t) + \sigma^2}$

Chapter 8

Good Priors from Population Maximum Likelihood

It is clinically helpful to predict the future cognitive decline of a subject based on his or her historical clinical observations. One way to do that is to build a parametric regression model fit to the existing observations. However, regression models, even the simplest ones, often suffer from overfitting, especially when the observations are sparse and noisy. As a conventional way to circumvent overfitting, we often take the problem into the Bayesian framework and put a prior on the model parameter, through which we convey our belief that certain models are more likely than others. However the priors are usually not specific to the problem considered, but rather chosen based on either convention (for example a penalty on the average square of the second derivative to insure smoothness), or mathematical and computational convenience (for example we may choose a conjugate prior for analytical tractability, or a ridge regression because it is simple and works adequately). In this chapter, we describe our work on learning an informative prior from a population of similar tasks based on mixed-effect model.

8.1 Informative Priors

It is appealing to find a framework where more problem specific prior can be extracted. We are interested in the situation in which a data sample is generated by a number of different individuals. More specifically, in this chapter, the individuals and the data considered are measurements over time of subjects in a clinical longitudinal study. The same ideas should apply to other situations, for example, time series from a collection of stocks representing a market sector, or utterances from a set of speakers recruited to build a speech recognition system. In each of these examples, there are models at two levels: the underlying model for each individual, and the collective model that describes the entire population.

Within this framework, we are interested in building models for new individuals that make efficient use of their data by incorporating priors derived from a population of similar tasks. A simple heuristic approach might fit separate models to each individual, examine the mean and variance of the individual models to extract a prior to use in generating models for new individuals. A more rigorous approach would correct the asymmetry in the heuristic by regularizing each of the individual models using priors derived from the entire data sample. Although not generally viewed as such, this is precisely the maximum likelihood fitting of mixed-effect models, as discussed in Chapter 7. In this Chapter, we will show that this framework is a very effective way to derive informative priors for fitting out-of-sample data.

8.2 Regression Models

We start by reviewing the notation for regression models that we used in Chapter 7. Suppose there are k individuals (indexed by $i = 1, \ldots, k$) contributing data to the sample, and we have observations $\{t^i_n, y^i_n\}$, $n = 1, \ldots, N^i$ as a function of time t for individual i. Suppose that the data are modeled as

$$y^i_n \;=\; f(t^i_n;\, \gamma^i) \;+\; \epsilon_n \tag{8.1}$$

where ϵ is zero-mean Gaussian noise with (unknown) variance σ^2. We use $\mathbf{y}^i = \{y^i_1, y^i_2, \cdots, y^i_{N^i}\}$ as the set of the observations for individual i collected at $\mathbf{t}^i = \{t^i_1, t^i_2, \cdots, t^i_{N^i}\}$. As discussed in Chapter 7, we consider the model parameters generated by the sum of a fixed and a random piece

$$\gamma^i \;=\; \alpha \;+\; \beta^i \tag{8.2}$$

where β is assumed distributed $\mathcal{N}(0, \mathbf{D})$ with unknown covariance $\mathbf{D}$. In this thesis, we are mainly concerned with the linear case

$$f(t;\gamma) \;=\; \gamma^T\,\Phi(t) \;=\; \alpha^T\,\Phi(t) \;+\; b^T\,\Phi(t) \tag{8.3}$$

where $\Phi(t) = [\phi_1(t), \phi_2(t), ..., \phi_d(t)]^T$ denotes a vector of basis functions.

8.3 Estimation of Parameters

8.3.1 Mixed-effect Model as Prior

We use $\mathcal{M} = \{\alpha, \mathbf{D}, \sigma\}$ to denote the fit mixed-effect model. It is easy to see that $\mathcal{M}$ defines a distribution of the model parameter γ:

$$p(\gamma|\mathcal{M}) \propto \exp(\frac{1}{2}(\gamma - \alpha)^T \mathbf{D}^{-1}(\gamma - \alpha)) \tag{8.4}$$

Using this distribution as a prior, we can write the posterior probability of model parameter γ for any novel individual after its observation $(\mathbf{y}, \mathbf{t})$ is known:

$$\begin{aligned} p(\gamma|\mathbf{y}; \mathbf{t}, \mathcal{M}) &\propto p(\gamma|\mathcal{M})p(\mathbf{y}\,|\gamma,\, \mathbf{t}; \mathcal{M}) & (8.5)\\ &\propto \exp(\frac{1}{2}(\gamma - \alpha)^T \mathbf{D}^{-1}(\gamma - \alpha)) \exp(\frac{1}{2\sigma^2}\sum_{n=1}^{N} ||y_n - f(t_n; \gamma)||^2) & (8.6)\\ &= \exp(\frac{1}{2\sigma^2}\sum_{n=1}^{N} ||y_n - f(t_n; \gamma)||^2 + \frac{1}{2}(\gamma - \alpha)^T \mathbf{D}^{-1}(\gamma - \alpha)). & (8.7) \end{aligned}$$

When $f(t, \gamma)$ is linear in γ, the posterior probability of γ is a Gaussian. Therefore the maximum a posteriori (MAP) solution and the posterior mean of γ are the same [67]:

$$\hat{\gamma}^B = \alpha + (\frac{1}{\sigma^2}\sum_{i=1}^{n} \Phi(t_n)\Phi(t_n)^T + \mathbf{D}^{-1})^{-1} \sum_{i=1}^{n} \Phi^T(t_n)(y_n - \alpha^T \Phi(t_n)). \tag{8.8}$$

Note that Equation (8.8) is essentially the same as finding $E(\beta^i|\mathbf{y}^i; \mathbf{t}^i, \mathcal{M}^g)$ in the E-step (Equation (7.4)) .

8.3.2 Relation to Ridge Regression

The common ridge regression assumes a simple spherical prior $p(\gamma) \propto \exp(-\frac{\lambda}{2}||\gamma||^2)$. Finding the MAP solution is then equivalent to the following optimization

$$\hat{\gamma}^R = \arg\max_{\gamma}[\exp(\frac{1}{2\sigma^2}\sum_{n=1}^{N} ||y_n - f(t_n; \gamma)||^2 + \frac{\lambda}{2}||\gamma||^2)]. \tag{8.9}$$

which can be given in closed form

$$\hat{\gamma}^R = (\sum_{n=1}^{N} \Phi(t_n)\Phi(t_n)^T + \sigma^2\lambda\mathbb{I})^{-1} \sum_{n=1}^{N} \Phi^T(t_n)y_n. \tag{8.10}$$

Equation (8.10) can be viewed as a special case of Equation (8.8) with $\alpha = 0$ and $\mathbf{D} = \lambda^{-1}\mathbb{I}$. Not surprisingly, if Equation (8.4) defines the true prior of parameters γ , the parameters given by Equation (8.8) are generally better than those given by Equation (8.10).

This property can be more rigorously expressed through the generalization error [60]. The generalization error for any particular parameter estimator $\hat{\gamma}$ is defined as the expectation of test error

$$\begin{aligned} \langle \varepsilon_{test} \rangle_{\epsilon,\epsilon',\gamma} &= \left\langle \sum_{i=1}^{n'} ||y'_n - f(t'_n;\hat{\gamma})||^2 \right\rangle_{\epsilon,\epsilon',\gamma} & (8.11) \\ &= \int\int\int p(\gamma|\mathcal{M})p(\epsilon)p(\epsilon') \sum_{n=1}^{N'} ||y'_n - f(t'_n;\hat{\gamma})||^2 d\epsilon\, d\epsilon'\, d\gamma & (8.12) \end{aligned}$$

where $y'_n = f(t'_n;\gamma) + \epsilon'_n$ is the observation made at *new* times $\{t'_1, t'_2, \cdots, t'_{n'}\}$ with observation noise ϵ' independent of the noise ϵ in training data. Actually, we can further show that the regression given by

$$f(t;\hat{\gamma}^B) = (\hat{\gamma}^B)^T \Phi(t)$$

gives the least generalization error among all regression models of the form of Equation (8.3), as proven as follows. Letting $\mathbf{R} = \sum_{n=1}^{N} \Phi(t_n)\Phi(t_n)^T$ and $\mathbf{R}' = \sum_{n=1}^{N'} \Phi(t'_n)\Phi(t'_n)^T$, the expected generalization error is

$$\begin{aligned} \langle \varepsilon_{test} \rangle^{BLUP}_{\epsilon,\epsilon'} &= \left\langle \sum_{n=1}^{N'} (\gamma^T\Phi(t'_n) + \epsilon'_n - \hat{\gamma}^B\Phi(t'_n))^2 \right\rangle_{\epsilon,\epsilon'} \qquad (\hat{\gamma}^B \text{ is a function of } \epsilon) \\ &= tr\{\mathbf{R}'(\mathbf{R} + (\frac{\mathbf{D}}{\sigma^2})^{-1})^{-2}(\frac{\mathbf{D}}{\sigma^2})^{-2}(\gamma-\alpha)(\gamma-\alpha)^T) + \sigma^2\mathbf{R}'\mathbf{R}(\mathbf{R} + (\frac{\mathbf{D}}{\sigma^2})^{-1})^{-2}\} + n'\sigma^2 \end{aligned}$$

If we integrate out γ based on its distribution $\mathcal{N}(\alpha, \mathbf{D})$, we get

$$\langle \varepsilon_{test} \rangle^{BLUP}_{\epsilon,\epsilon',\gamma} = \int \langle \varepsilon_{test} \rangle^{BLUP}_{\epsilon,\epsilon'} p(\gamma) d\gamma = tr\{\mathbf{R}'(\frac{\mathbf{R}}{\sigma^2} + \mathbf{D}^{-1})^{-1}\} + n'\sigma^2.$$

$\langle \varepsilon_{test} \rangle^{BLUP}_{\epsilon,\epsilon',\gamma}$ is the smallest among these achieved by all $\hat{\gamma}$ in the form of $\mathbf{a} + \mathbf{b}^T\mathbf{y}$. One can see that by calculating the derivatives of $\langle \varepsilon_{test} \rangle_{\epsilon,\epsilon',\gamma}$ to $\mathbf{a}$ and $\mathbf{b}$ and setting them to zero. That is why Equation (8.8) is also known as the best linear unbiased prediction (BLUP) [67].

8.3.3 Mixture of Mixed-effect Models as Prior

When a sample time series is from a population consisting of several sub-populations with different characteristics, it is suitable to use a mixture of mixed-effect model as the generative model, as discussed in Section 7.4.1. We use $\widetilde{\mathcal{M}} = \{\pi_0, \mathcal{M}_0, \pi_1, \mathcal{M}_1\}$ to denote the parameters of this mixture, with π_m being the mixing proportion of the m^{th} component mixed-effect model and $\mathcal{M}_m = \{\alpha_m, \mathbf{D}_m, \sigma_m\}$ being its parameters. For any novel subject with measurement $\mathbf{y}$, the joint probability of γ and $\mathbf{y}$ is

$$p(\mathbf{y}, \gamma; \mathbf{t}, \widetilde{\mathcal{M}}) \;=\; \sum_{m=0}^{1} \pi_m p(\gamma; \mathcal{M}_m) p(\mathbf{y}|\, \mathbf{t}; \gamma)\ . \tag{8.13}$$

Using Bayes rule, the posterior of parameter γ is given by

$$p(\gamma|\, \mathbf{y}, \mathbf{t}, \widetilde{\mathcal{M}}) \;=\; \frac{p(\gamma, \mathbf{y}; \mathbf{t}, \widetilde{\mathcal{M}})}{p(\mathbf{y}; \mathbf{t}, \widetilde{\mathcal{M}})} = \frac{\sum_{z=0}^{1} p(z, \mathbf{y}; \mathbf{t}, \widetilde{\mathcal{M}})}{p(\mathbf{y}; \mathbf{t}, \widetilde{\mathcal{M}})} \tag{8.14}$$

$$\;=\; \sum_{m=0}^{1} P(z = m|\, \mathbf{y}; \mathbf{t}, \widetilde{\mathcal{M}}) p(\gamma|\, \mathbf{y}; \mathbf{t}, \mathcal{M}_m). \tag{8.15}$$

From Section 8.3.1, when $f(t;\gamma)$ is linear in γ, $p(\gamma|\mathbf{y}, \mathbf{t}, \mathcal{M}_m)$ is Gaussian, and thus $p(\gamma|\, \mathbf{y}; \mathbf{t}, \widetilde{\mathcal{M}})$ is a mixture of Gaussians[1]. Due to the linearity, the optimal estimator of γ that minimizes the generalization error is the posterior mean of γ

$$\hat{\gamma}^M \;=\; \arg\min_{\gamma} \left\langle \sum_{n=1}^{N'} ||y'_n - f(t'_n; \hat{\gamma})||^2 \right\rangle_{\epsilon, \epsilon', \gamma} = E(\gamma|\, \mathbf{y}, \mathbf{t}, \widetilde{\mathcal{M}}) \tag{8.16}$$

$$= \sum_{m=1}^{M} P(z = m|\, \mathbf{y}; \mathbf{t}, \widetilde{\mathcal{M}}) \{\alpha_m + (\frac{1}{\sigma_m^2} \sum_{n=1}^{N} \Phi(t_n)\Phi(t_n)^T + \mathbf{D}_m^{-1})^{-1} \sum_{n=1}^{N} \Phi^T(t_n)(y_n - \alpha_m^T \Phi(t_n))\} \tag{8.17}$$

In the remainder of the paper, we will refer to the regression model in Equation (8.17) as MixPrior [2].

[1] Note the parameters of this Gaussian mixture is functions of $\mathbf{y}$ and $\mathbf{t}$, and not constant.

[2] If $f(t;\gamma)$ is not linear in γ, finding the solution that minimizes the generalization error is generally difficult. Instead, we can find the MAP solution by maximizing the posterior probability of γ formulated in Equation (8.15). When $\sigma_m \approx \sigma$ for any $m \in \{1, 2, \cdots, M\}$, the right hand side of Equation (8.15) can be approximated as

$$p(\gamma|\mathbf{y}; \mathbf{t}, \widetilde{\mathcal{M}}) \;\approx\; p(\mathbf{y}|\gamma; \mathbf{t}, \sigma) \sum_{m=1}^{M} P(z = m|\, \mathbf{y}; \mathbf{t}, \widetilde{\mathcal{M}})(\gamma|\{\alpha_m, \mathbf{D}_m\}). \tag{8.18}$$

Here the first factor on right hand side of Equation (8.18) tells us how well the model γ describes the data $\mathbf{y}$, and the second factor is about the prior distribution of γ.

8.4 Experimental Results

In this section we give the results of fitting individual curves using the population model as the prior. We consider two motor behaviors steps and seconds, for which the reading generally increases with age and cognitive decline. The population (with 143 individuals) consists of the normal group (97 individuals) and impaired group (46 individuals), with people in impaired group deteriorate faster than the normal group. As in Chapter 7, we use the mixed-effect model with basis functions $\Phi(t) = [1, t]^T$ and $\Phi(t) = [1, t, t^2]^T$. The corresponding mixed-effect models fit on the two data can be found in Figure 7.3.

As we discussed in Section 8.3, the ridge regression, BLUP, and MixPrior can all be viewed as MAP estimation of parameters γ with different priors. In Figure 8.1, we give an example on how the probability of γ changes after some observations are incorporated. The particular individual in this example is from impaired group. It consists of 15 observations and we used the first 5 for training and the remaining 10 for testing, as shown on the bottom panel in Figure 8.1 (see caption). We show with the upper row (three panels) in Figure 8.1 the prior probability density of γ given by the ridge regression, BLUP, and MixPrior, and with the middle row (three panels) the corresponding posterior probability densities of γ after the first five observations are incorporated. For BLUP, we used the mixed-effect model fit on impaired group as the prior. For MixPrior the mixing proportion are estimated based on the number of individual from each group. The bottom panel shows the curve fitting result with different regression models. In this example, with only 5 observations, the BLUP and MixPrior reached a solution close to the one that minimize total test error for all the 15 points for this individual.

The following two experiments are designed to show the modeling ability of our algorithm under two different assumptions. In the first experiment (Figure 8.2 and 8.3), we suppose the diagnosis for each individual is known. We can thus use BLUP with the correct population prior as the modeling method. In the second experiment (Figure 8.4), we suppose the diagnosis for each individual is unknown, and correspondingly, we use MixPrior for individual modeling.

The experiments are carried out in a“leave-one-out” fashion to prevent the model are trained and evaluated on different sets of individuals. For example, when we try to model individual i, the population model will be trained on individual $\{1, 2, \cdots, i-1, i+1, \cdots, k\}$. For each individual, we used the first several observations for curve fitting and rest observations for testing. The thing of interest to us is the behavior of different algorithms with a varying number of training points.

In Figure 8.2 and Figure 8.3, we compare the test error from BLUP with that from LSQ and ridge regression. For ridge regression, the penalty coefficient λ is manually chosen to minimize the total test error of all the test points. The reported test error for each method is averaged over all the test points. It is clear from Figure 8.2 and 8.3 that BLUP achieves a decent fitting with as few as 4 points for training, while with the same number of training points, LSQ and ridge regression give considerably higher test error. This observation indicates that when the information from the observations is limited, the BLUP prior plays an important role in finding proper parameters. When the number of training points increases, the difference between BLUP and ridge regression decreases, although BLUP is still consistently better than ridge regression. Not surprisingly, both regularized models achieves less test error than LSQ.

In Figure 8.4, we mix all the individuals from two groups and for each method, the test error is averaged over all the test points in the two groups. This case corresponds to the scenario that we do not know the correct membership of the individual we are studying, and therefore we used the mixture of mixed-effect model as prior. For comparison, we also list the BLUP result with the correct group prior. The result shows that MixPrior gives a slightly higher test error than BLUP, but is still considerably better than ridge regression and LSQ fitting. The meaning of this result is twofold. First, when the group membership information is available, the prior used in BLUP is more accurate than MixPrior. Second, when the correct group is unknown, the mixture of the mixed-effect models is still a highly informative prior in helping find the proper model parameters, as compared to ridge regression and LSQ.

It is also revealed from Figure 8.2 $\sim$ 8.4 that the BLUP regression with quadratic functions does not yield significant improvement over that with linear function. Furthermore, even though the quadratic and linear *population* models in Figure 7.3 are substantially

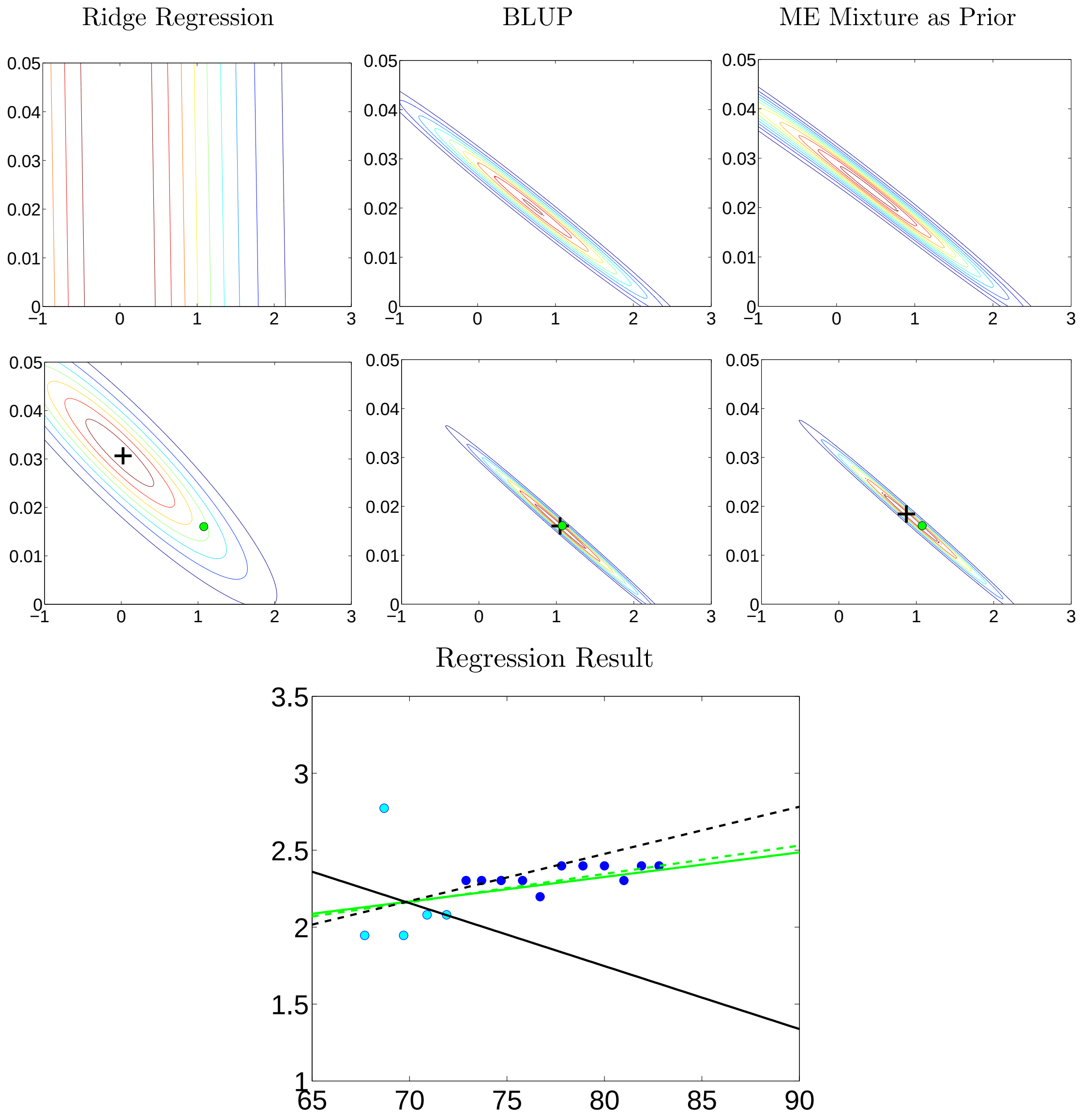


Figure 8.1: The probability density of γ before and after the observations are incorporated. The first row, the prior probability of γ given by ridge regression, BlUP, and mixture of ME. The second row, the posterior probability of γ after $\{y_1, y_2, \cdots, y_5\}$ is known. Note the red dot is the location of parameters minimizing the $\sum_{i=1}^{15}(f(t_n;\gamma) - y_n)^2$ and the black cross is the location of the MAP estimation. The bottom panel is the curve fitting result from different models. **Cyan dots:** points used for training; **Blue dots:** dots for testing; **Green line:** the BLUP solution; **Green dashed line:** MixPrior. **Black line:** LSQ; **Black dashed line:** ridge regression.

different, for most *particular subjects* the BLUP regression with linear and quadratic functions are very similar. It appears that for most *individual subjects* the observations do not span a long enough time range to distinguish between the linear and quadratic models.

8.5 Summary

In this chapter, we discussed improving the prediction of motor ability decline on individual level by learning a prior from similar subjects. We examined regression in a hierarchical model framework, where the hierarchy consists of models for each of several individuals that contribute to the population sample, and an overall population model. In our work, this hierarchical model can either be a mixed-effect model or a mixture of mixed-effect models (when the population consists of several sub-populations). The distribution of model parameters across the population then serves as a model prior for regression on out-of-sample data. With the learnt prior, our regression model can produce accurate predictions with small amounts of fitting data. We have demonstrated both theoretically and empirically that this framework provides priors substantially superior to that used in ridge regression.

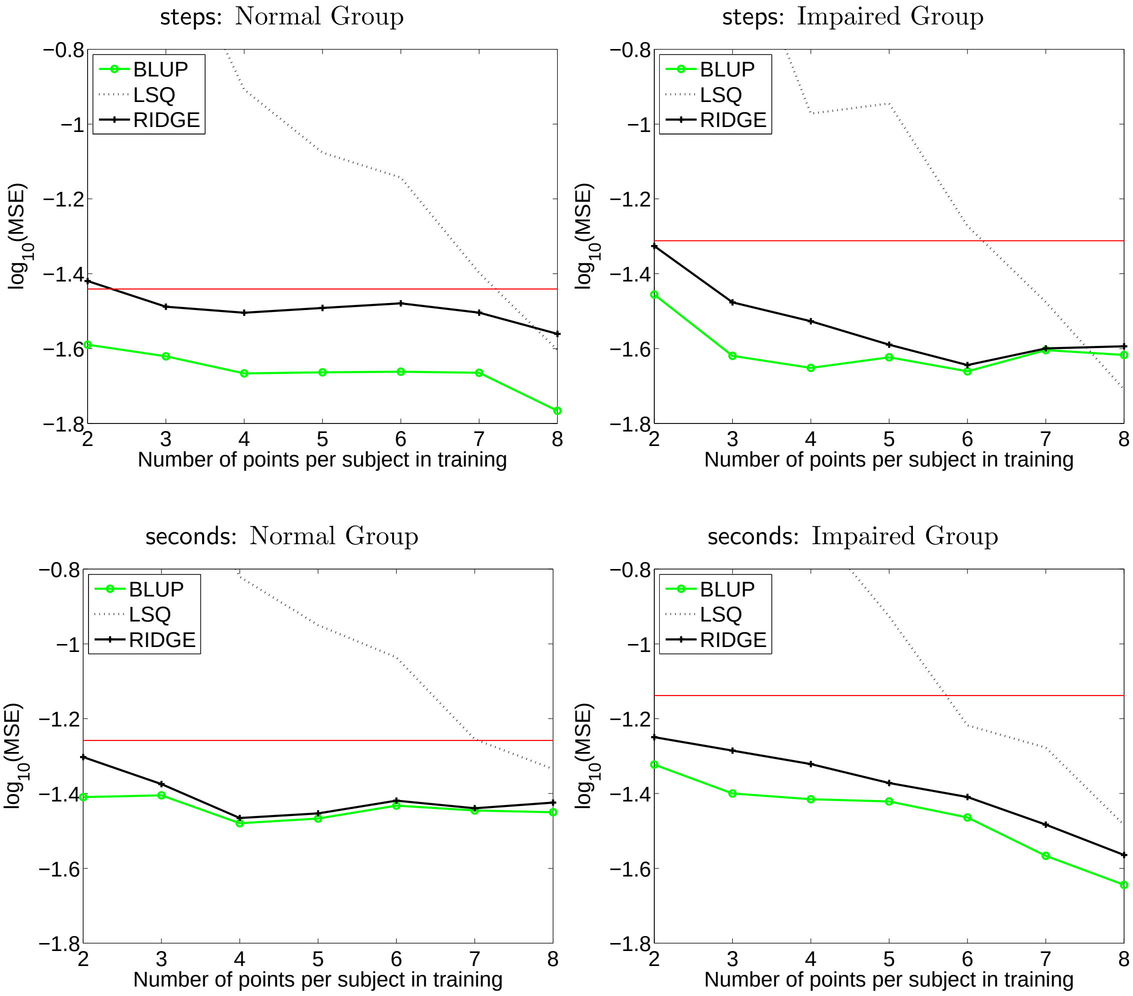


Figure 8.2: The test error with order-1 polynomial (linear) regression model. The test error with order-1 model. The logarithm of mean square error from different models with varying number of the training points. The red line the error by using the population mean α as the model parameters.

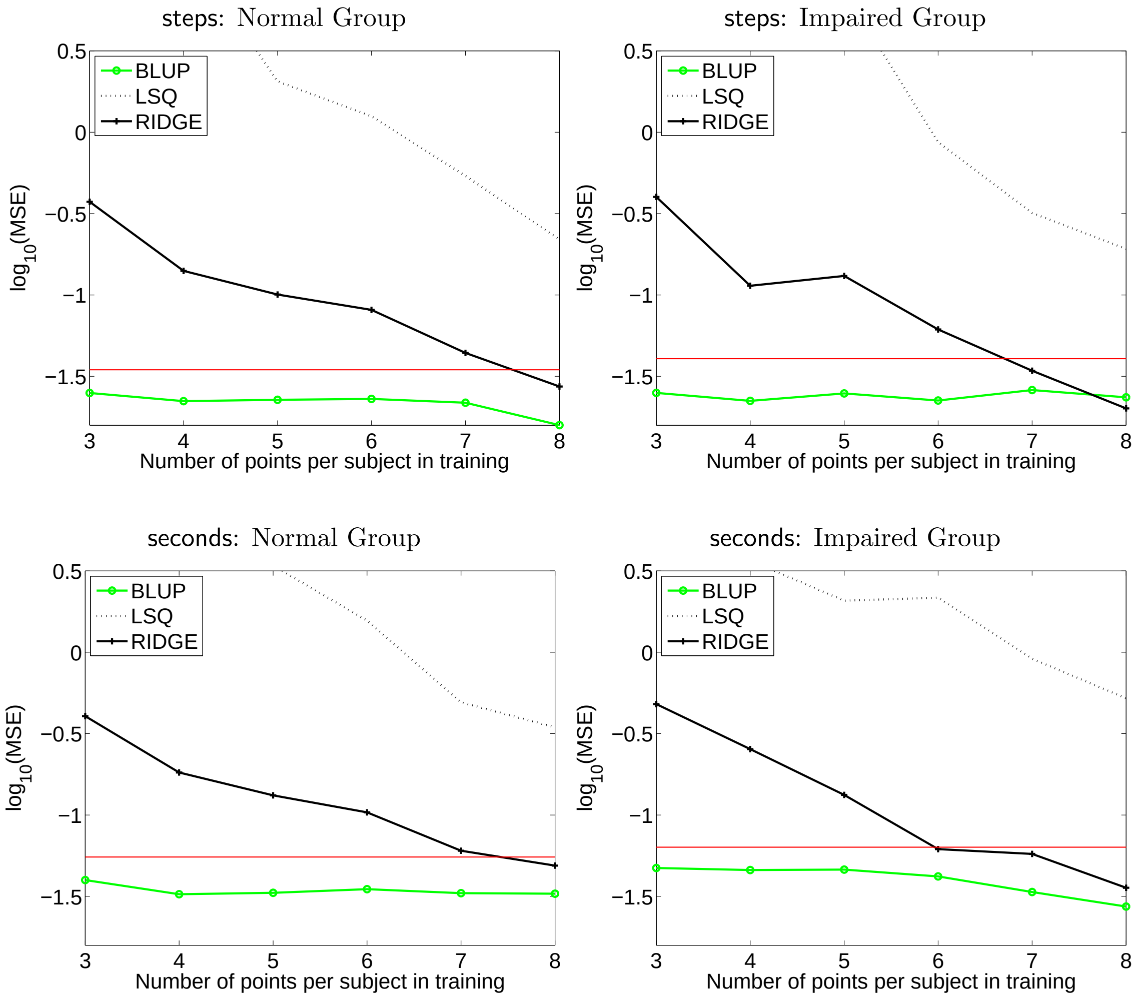
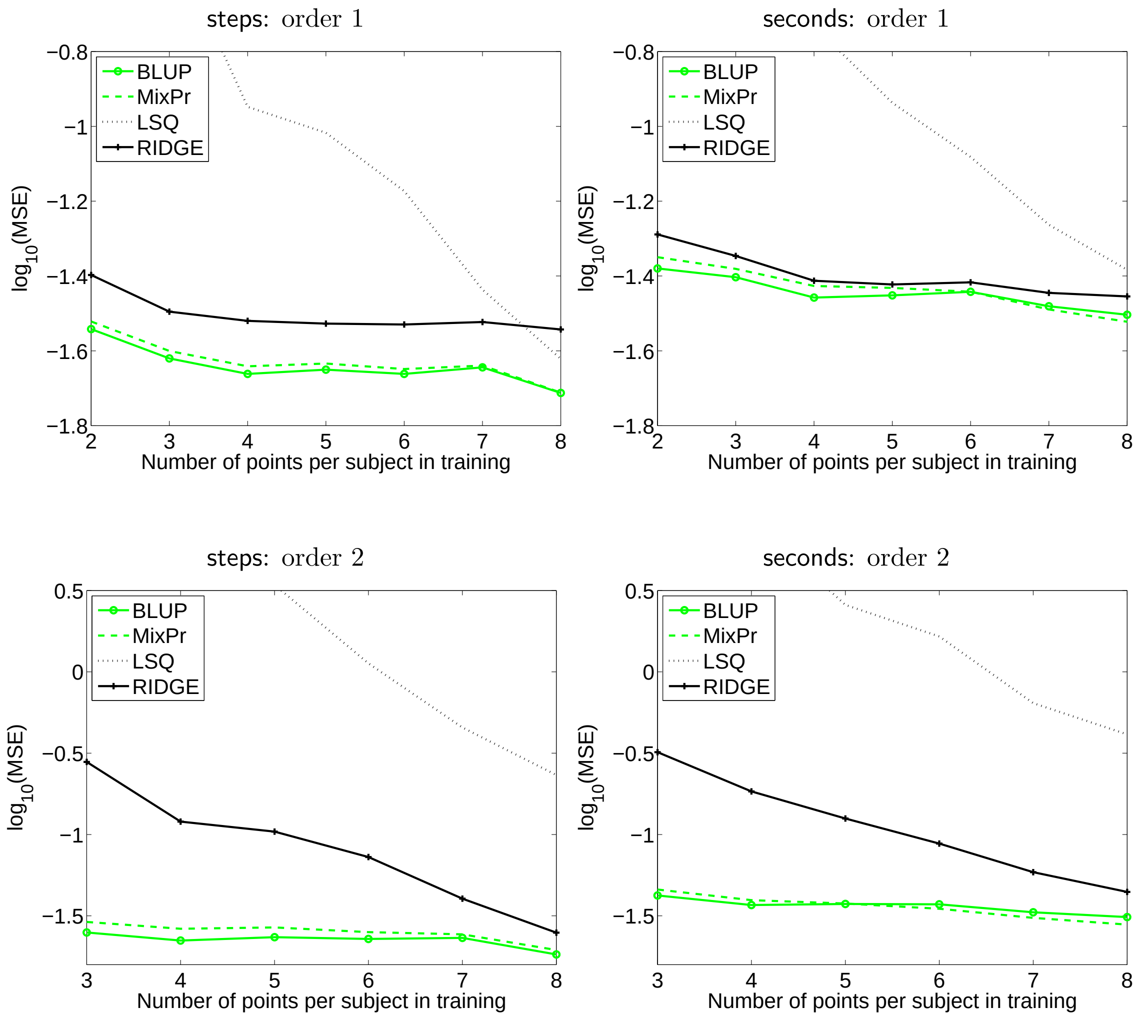


Figure 8.3: The test error with order-2 polynomial (quadratic) model. The logarithm of mean square error from different models with varying number of the training points. The red line the error by using the population mean α as the model parameters.



steps: order 1

seconds: order 1

steps: order 2

seconds: order 2

Figure 8.4: The test error with mixture of mixed-effect models as prior. The logarithm mean square error from different models with varying number of the training points. The data is the union of normal and impaired group.

Chapter 9

Detection of Cognitive Decline Based on Mixed-effect Models

In Chapter 8, we discussed the prediction of individual subject's future motor ability based on historical observations. Left unanswered, is how to identify those people with high risk of developing into impairment based on their sparse historic clinical observations. We considered several solutions to this classification problem. In this chapter, we will discuss building a classifier based on the mixed-effect models from a generative model perspective. Later in Chapter 10, we will discuss the classifiers based on discriminative models.

9.1 Likelihood Ratio Based on Mixed-effect Models

Again, let us consider the mixed effect model mixture $\widetilde{\mathcal{M}}$ consisting of two component models, $\mathcal{M}_0$ or impaired group and $\mathcal{M}_1$ for normal group, with mixing proportion (prior) π_0 and π_1. For any given observation $(\mathbf{t}, \mathbf{y})$, the posterior probability that this observation is generated from $\mathcal{M}_0$ is given by

$$P(z=0|\mathbf{y};\mathbf{t},\widetilde{\mathcal{M}}) = \frac{\pi_0 p(\mathbf{y};\mathbf{t},\mathcal{M}_0)}{p(\mathbf{y};\mathbf{t},\widetilde{\mathcal{M}})} = \frac{\pi_0 p(\mathbf{y};\mathbf{t},\mathcal{M}_0)}{\pi_0 p(\mathbf{y};\mathbf{t},\mathcal{M}_0) + \pi_1 p(\mathbf{y};\mathbf{t},\mathcal{M}_1)} \tag{9.1}$$

where to get $p(\mathbf{y};\mathbf{t},\mathcal{M}_m)$ we need to integrate out the random effect parameter β. That is

$$\begin{aligned} p(\mathbf{y};\mathbf{t},\mathcal{M}_m) &= \int_{\mathbb{R}^d} p(\mathbf{y};\mathbf{t},\alpha_m+\beta)p(\beta;\mathcal{M}_m)d\beta \\ &= (2\pi)^{-n/2}|\Sigma_m|^{-1/2}\exp((\mathbf{y}-\alpha_m^T\Phi)^T(\Sigma_m)^{-1}(\mathbf{y}-\alpha_m^T\Phi)) \end{aligned}$$

where

$$\begin{aligned}\Sigma_m &= \sum_{n=1}^{N} \Phi(t_n)\mathbf{D}_m\Phi(t_n)^T + \sigma_m^2\mathbf{I}_{n\times n} \\ \mathbf{\Phi} &= [\Phi(t_1), \Phi(t_2), \cdots, \Phi(t_n)]^T.\end{aligned}$$

The classification decision can be made based on the posterior probability

$$c = \begin{cases} 0 & P(z=0|\mathbf{t},\mathbf{y};\widetilde{\mathcal{M}}) \geq 0.5 \\ 1 & \text{otherwise} \end{cases} \tag{9.2}$$

Equation (9.2) is the optimal Bayesian classifier that minimizes the expected the 0-1 loss

$$P(z=0|\mathbf{t},\mathbf{y};\widetilde{\mathcal{M}})I(c\neq 0) + P(z=1|\mathbf{t},\mathbf{y};\widetilde{\mathcal{M}})I(c\neq 1) \tag{9.3}$$

where $I(\cdot)$ is a function with boolean input and binary output:

$$I(\omega) = \begin{cases} 1 & \omega \text{ is true} \\ 0 & \text{otherwise} \end{cases}.$$

If we considered unequal cost for misclassification cost to different classes, we instead minimize the following cost function:

$$C_H P(z=0|\mathbf{t},\mathbf{y};\widetilde{\mathcal{M}})I(c\neq 0) + C_I\, P(z=1|\mathbf{t},\mathbf{y};\widetilde{\mathcal{M}})I(c\neq 1). \tag{9.4}$$

The optimal classifier is given by

$$c = \begin{cases} 0 & \frac{p(\mathbf{y}^i;\mathbf{t}^i,\mathcal{M}_0)}{p(\mathbf{y}^i;\mathbf{t}^i,\mathcal{M}_1)} \geq \frac{\pi_1}{\pi_0}\frac{C_I}{C_H} \\ 1 & \text{otherwise} \end{cases} \tag{9.5}$$

where $\frac{p(\mathbf{y}^i;\mathbf{t}^i,\mathcal{M}_0)}{p(\mathbf{y}^i;\mathbf{t}^i,\mathcal{M}_1)}$is referred to as likelihood ratio.

We define the *detection rate* probability that subjects who later become impaired are correctly identified, and the false alarm rate as the probability that subjects who remain healthy are incorrectly predicted to become impaired. We are interested in the performance of the detection rate given a false alarm rate, which is given by the receiver operating characteristic (ROC) curve [64].

Classification Based on Multivariate Mixed-effect Model It is straightforward to extend the generative model-based classifier to the multivariate (Section 7.4.2) as a way to fuse several clinical measurements in classification decision. As we noted in Chapter 7, the multiple variate mixed-effect, the full covariance case or the naive Bayesian case, is not a satisfying way to describe the joint probability of the measurements of different types. This statement is further verified by our pilot research on using the multivariate mixed-effect model on classification. Generally, the classification result given by the joint model is not significantly better than the one given by the best univariate model, most times worse.

9.2 Evaluation of Classifiers

We assess the performance of a classifier as the area under the ROC curve (AUC), which can be empirically estimated

$$\text{AUC} = \frac{1}{k_I k_H} \sum_{i=1}^{k_I} \sum_{j=1}^{k_H} \{ I(Y_H^i > Y_I^j) + \frac{1}{2} I(Y_H^i = Y_I^j) \} \tag{9.6}$$

where Y_H^i is the classifier output for subject i in impaired group and the Y_I^j is classifier output for subject j in the normal group. To compare two classifiers A and B we calculate the difference between the two corresponding AUCs

$$\Delta\text{AUC} = \text{AUC}_A - \text{AUC}_B.$$

The null-hypothesis $\text{AUC}_A = \text{AUC}_B$ is tested by comparing the test statistic $\Delta\text{AUC}/\sqrt{\text{var}\,\{\Delta\text{AUC}\}}$ with a standard Gaussian distribution (Z-test), where var $\{\Delta\text{AUC}\}$ is the sample variability. In the rest of this thesis, we will refer to the significance level of rejecting null-hypothesis test value as the significance of the statement "Classifier A is different from Classifier B".

In our case, all classifiers are evaluated on the same subjects, and var $\{\Delta\text{AUC}\}$ between classifier A and B is estimated as

$$\frac{\sum_{i=1}^{k_I} v\hat{a}r(\hat{S}_{H,A}(Y_{H,A}^i) - \hat{S}_{H,B}(Y_{H,B}^i))}{k_I} + \frac{\sum_{j=1}^{k_H} v\hat{a}r(\hat{S}_{I,A}(Y_{I,A}^j) - \hat{S}_{I,B}(Y_{I,B}^j))}{k_H}$$

where (as defined in [64] page 69 and 108)

- $Y^i_{H,A}$ is the output from classifier A for the i^{th} subject in the normal group
- $Y^i_{I,A}$ is the output from classifier A for the i^{th} subject in the impaired group
- $Y^j_{H,B}$ is the output from classifier B for the j^{th} subject in the normal group
- $Y^j_{I,B}$ is the output from classifier B for the j^{th} subject in the impaired group
- $\hat{S}_{H,A}$ is the empirical non-disease placement value for classifier A estimated as

$$\hat{S}_{H,A}(Y) = \frac{1}{k_H} \sum_{i=1}^{k_H} I(Y^i_{H,A} \leq Y)$$

- $\hat{S}_{I,A}$ is the empirical disease placement value for classifier A estimated as

$$\hat{S}_{H,A}(Y) = \frac{1}{k_I} \sum_{j=1}^{k_I} I(Y^j_{I,A} \leq Y)$$

- $\hat{S}_{H,B}$ is the empirical non-disease placement value for classifier B estimated as

$$\hat{S}_{H,B}(Y) = \frac{1}{k_H} \sum_{i=1}^{k_H} I(Y^i_{H,B} \leq Y)$$

- $\hat{S}_{I,B}$ is the empirical non-disease placement value for classifier B estimated as

$$\hat{S}_{I,B}(Y) = \frac{1}{k_I} \sum_{i=1}^{k_I} I(Y^j_{I,B} \leq Y)$$

The sampling $v\hat{a}r$ is estimated using the jackknife sampling [26] as follows:

$$v\hat{a}r(\hat{S}_{H,A}(Y^i_{H,A}) - \hat{S}_{H,B}(Y^i_{H,B})) = \frac{k_H}{k_H - 1} \sum_{j=1}^{k_H} (\hat{S}^{(j)}_{H,A}(Y^i_{H,A}) - \hat{S}^{(j)}_{H,B}(Y^i_{H,B}))^2.$$

9.3 Comparing to A Baseline Generative Model

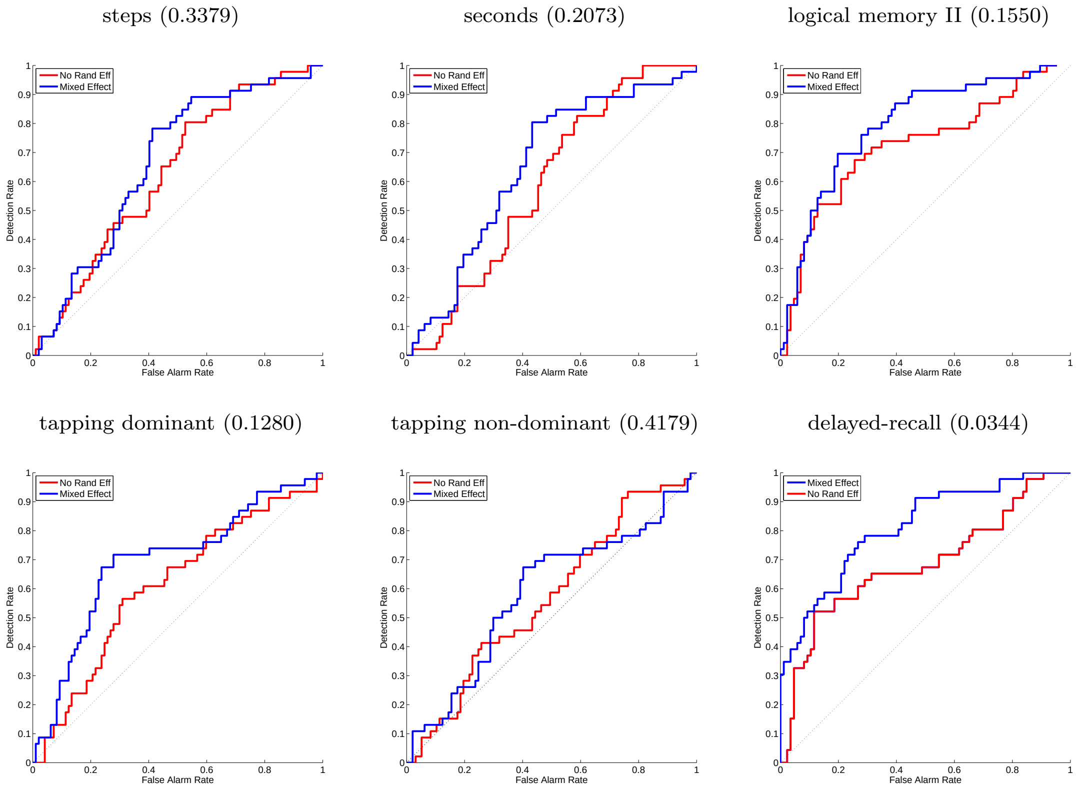
To demonstrate the discriminative power we gain from distinguishing random effect from observation noise in the mixed-effect models, we compare it to a simpler generative model that assumes no random effect and explains all the deviation from the population model as (white) observation noise. The maximum-likelihood fitting of this model is actually a least square fitting to all the observations without distinguishing the individuals. Equivalently this simplified model can be viewed as a mixed-effect model with constraint $\mathbf{D} = \mathbf{0}$. Using $\mathcal{S}_0$ and $\mathcal{S}_1$ to denote the simplified model fit on the normal group and impaired group, with $\mathcal{S}_m = \{\mu_m, s_m\}$, where μ_m is the fit regression model and s_m is the observation noise.[1] Once the models are fit, we can calculate the likelihood of any novel sample $\mathbf{y}$ is generated by each $\mathcal{S}_m$

$$p(\mathbf{y}; t, \mathcal{S}_m) \quad = \quad (2\pi)^{-n/2}\sigma^{-n} \exp(\frac{||\mathbf{y} - \mu_m^T \Phi||^2}{2s_m^2}),$$

with $\mathbf{\Phi} = [\Phi(t_1), \Phi(t_2), \cdots, \Phi(t_n)]^T$, based on which we can build the likelihood ratio classifier. We expect the classifiers based on mixed-effect models can outperform this baseline classifier since we believe the random effects better describe the individual data than merely the observation noise.

We compare the classification result of mixed-effect model and LSQ model in Figure 9.1. It is clear from Figure 9.1 that mixed-effect model is generally better than the simplified model, but this superiority is not statistically significant except on the delayed-recall.

[1] Generally $\mu_m \neq \alpha_m$, although in our experiments they are fairly close.



steps (0.3379)

seconds (0.2073)

logical memory II (0.1550)

tapping dominant (0.1280)

tapping non-dominant (0.4179)

delayed-recall (0.0344)

Figure 9.1: The ROC curves of the likelihood ratio classifiers. The number in the parenthesis is the *p*-value (Z-test) for the null-hypothesis "AUC of mixed-effect model is same as the AUC of least square fitting".

Chapter 10

Detection of Cognitive Decline: Discriminative Approaches

In Chapter 9, we discussed cognitive detection with mixed-effect models as the generative models. It is long known in machine learning community that generative model-based classifiers are often inferior to discriminative methods in terms of classification accuracy. One important reason, as discussed by many authors [25, 42], is the following. In a generative model-based classifier, the model for each class is usually trained separately based on maximum-likelihood (ML) or maximum a posteriori (MAP) criterion. Following this training fashion comes the inefficiency as a classifier in using the model parameters. That is, the generative models often spend most their parameters on modeling either the "typical" data far away from class boundary (for example Gaussian mixture model), or the generative properties that are common to all the classes (for example hidden Markov Model in speech recognition). Discriminative models, on the other hand, concentrate their modeling effort into defining the decision boundary and do not spend resource modeling the data far from the decision boundary. Nevertheless generative models have their advantages over discriminative models [63, 36]. For example, generative models are usually more compatible with human prior knowledge, better at dealing with missing data, and less prone to over-fitting. Moreover, generative models may contain useful information about the class distribution hard to capture with simple discriminative models. As we showed in Chapter 9, mixed-effect models trained on the normal group and impaired group manifest the difference between the two populations, whereas there is no simple way to do so with a discriminative model, mainly because the longitudinal observations can not be directly used as feature for discriminative models.

It is now a common practice to take advantage of the two schools of models by combining generative and discriminative approaches. In Chapter 8 and Chapter 9, we have demonstrated the mixed-effect model as a good modeling tool for the longitudinal data on a population and individual level. We will show in this chapter that the mixed-effect model can also be used in feature extraction for discriminative models, more specifically support vector machine [78]. Also we will present the performance of discriminative models with other feature extraction routines we developed for the longitudinal data.

10.1 Background: Support Vector Machine

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. A special property of SVMs is that they simultaneously minimize the empirical classification error and maximize the geometric margin; hence they are also known as maximum margin classifiers.

Viewing the input data as two sets of vectors in an n-dimensional space, an SVM will construct a separating hyperplane in that space, one which maximizes the "margin" between the two data sets. To calculate the margin, we construct two parallel hyperplanes, one on each side of the separating one, which are "pushed up against" the two data sets. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the neighboring data points of both classes. The hope is that, the larger the margin or distance between these parallel hyperplanes, the better the generalization error of the classifier will be. A good tutorial of SVM is given by Burges [14].

10.1.1 Formalization

We are given some training data, a set of points of the form

$$\{(\mathbf{x}_1, c_1), (\mathbf{x}_2, c_2), \ldots, (\mathbf{x}_n, c_n)\}$$

where the ci is either 1 or -1, indicating the class to which the point $\mathbf{x}_i$ belongs. Each $\mathbf{x}_i$ is a p-dimensional real vector. We want to give the maximal-margin hyperplane which divides the points having $c_i = 1$ from those having $c_i = -1$. Any hyperplane can be written as the set of points $\mathbf{x}$ satisfying Maximum-margin hyperplane and margins for a SVM trained with samples from two classes. Samples on the margin are called the support vectors.

$$\mathbf{w} \cdot \mathbf{x} - b = 0.$$

The vector $\mathbf{w}$ is a normal vector: it is perpendicular to the hyperplane. The parameter b determines the offset of the hyperplane from the origin along the normal vector $\mathbf{w}$.

We want to choose the $\mathbf{w}$ and b to maximize the margin, or distance between the parallel hyper-planes that are as far apart as possible while still separating the data. These hyper-planes can be described by the equations

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x} - b &= 1 \qquad (10.1) \\ \mathbf{w} \cdot \mathbf{x} - b &= -1. \qquad (10.2) \end{aligned}$$

Note that if the training data are linearly separable, we can select the two hyper-planes of the margin in a way that there are no points between them and then try to maximize their distance. By using geometry, we find the distance between these two hyper-planes is $2/|w|$, so we want to minimize $|w|$. As we also have to prevent data points falling into the margin, we add the following constraint: for each i either

$$\mathbf{w} \cdot \mathbf{x}_i - b \geq 1 \qquad \text{for } \mathbf{x}_i \text{ for the first class}$$

or

$$\mathbf{w} \cdot \mathbf{x}_i - b \leq -1 \qquad \text{for } \mathbf{x}_i \text{ of the second.}$$

This can be rewritten as:

$$c_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1, \quad \text{for all } 1 \leq i \leq n. \qquad (1)$$

We can put this together to get the optimization problem:

$$\begin{aligned} &\text{choose } \mathbf{w}, b \text{ to minimize } |\mathbf{w}| \\ &\text{subject to } c_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1, \quad \text{for all } 1 \leq i \leq n \end{aligned}$$

Primal Form The optimization problem presented in the preceding section is hard to optimize because it depends on the absolute value $|\mathbf{w}|$. The reason is that, in mathematical terms, it is a non-convex optimization problem which are known to be much more difficult to solve. Fortunately it is possible to alter the equation by substituting $|w|$ with $\frac{1}{2}||\mathbf{w}||^2$ without changing the solution (the minimum of the original and the modified equation have the same $\mathbf{w}$ and b). This is a quadratic programming (QP) optimization problem. More clearly,

$$\text{minimize } \frac{1}{2}||\mathbf{w}||^2, \text{ subject to } c_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1, \quad 1 \leq i \leq n..$$

This problem can now be solved by standard quadratic programming techniques and programs.

Dual Form Writing the classification rule in its unconstrained dual form reveals that the maximum margin hyperplane and therefore the classification task is only a function of the support vectors, the training data that lie on the margin. The dual of the SVM can be shown to be:

$$\max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j c_i c_j \mathbf{x}_i^T \mathbf{x}_j \text{ subject to } \alpha_i \geq 0, \text{ and } \sum_{i=1}^{n} \alpha_i c_i = 0 \tag{10.3}$$

where the α terms constitute a dual representation for the weight vector in terms of the training set:

$$\mathbf{w} = \sum_i \alpha_i c_i \mathbf{x}_i$$

Soft Margin Vapnik [78] suggested that a modified maximum margin idea that allows for mislabeled examples. If there exists no hyperplane that can split the "positive" and "negative" examples, the Soft Margin method will choose a hyperplane that splits the examples as cleanly as possible, while still maximizing the distance to the nearest cleanly split examples. This work popularized the expression Support Vector Machine or SVM. The method introduces slack variables, ξ_i, which measure the degree of misclassification of the datum $\mathbf{x}_i$

$$c_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 - \xi_i \quad 1 \leq i \leq n \qquad (2). \tag{10.4}$$

The objective function is then increased by a function which penalizes non-zero ξ_i, and the optimization becomes a trade off between a large margin, and a small error penalty. If the penalty function is linear, the Equation (10.4) now transforms to

$$\min \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i \quad \text{such that} \quad c_i(\mathbf{w}\cdot\mathbf{x}_i - b) \geq 1 - \xi_i \quad 1 \leq i \leq n. \tag{10.5}$$

This constraint in Equation (10.5) along with the objective of minimizing $|w|$ can be solved using Lagrange multipliers. The key advantage of a linear penalty function is that the slack variables vanish from the dual problem, with the constant C appearing only as an additional constraint on the Lagrange multipliers. Non-linear penalty functions have been used, particularly to reduce the effect of outliers on the classifier, but unless care is taken, the problem becomes non-convex, and thus it is considerably more difficult to find a global solution.

Non-linear classification It follows Equation (10.3) that we only need the dot products of samples instead of the explicit vector values of x_i. This property enables us to replace every dot product with a non-linear kernel function, for example the popular Gaussian RBF kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2s^2}).$$

This allows the algorithm to fit the maximum-margin hyperplane in the transformed feature space. The transformation may be non-linear and the transformed space high dimensional; thus though the classifier is a hyperplane in the high-dimensional feature space it may be non-linear in the original input space. In the rest of this section, we construct a kernel K from either mixed-effect model or other feature extraction routines.

10.2 The Optimization of ROC Curves

In Chapter 9 we can get an optimal ROC by thresholding the likelihood ratio with varying values. We get this modeling simplicity from the probabilistic interpretation of the classifier. That is, if the classifier assumes the class distribution $P(z = 0|t, y; \widetilde{\mathcal{M}})$, the classification decision will be automatically given by the following likelihood ratio

$$c = \begin{cases} 0 & \frac{p(\mathbf{y}^i;\mathbf{t}^i,\mathcal{M}_0)}{p(\mathbf{y}^i;\mathbf{t}^i,\mathcal{M}_1)} \geq \frac{\pi_1}{\pi_0}\frac{C_I}{C_H} \\ 1 & \text{otherwise} \end{cases}$$

no matter how the mis-classification cost $\{C_H, C_I\}$ are specified.

Unfortunately, things are more involved for discriminative methods such as SVM, with which we model the decision boundary instead of the class distribution. The modeled decision boundary (or more accurately, the parameters in the classifiers) is tuned for a certain setting of mis-classification cost, and its shape may change as the mis-classification cost varies, as illustrated in Figure 10.1. Suppose we are classifying through a parametric-function $f(\mathbf{x};\theta,h)$ with parameters θ and threshold h. With a certain mis-classification cost (C_H, C_I), we try to find the optimal parameters θ^* and threshold h through the following optimization[1]

$$(\theta^*, h^*) = \arg\min_{\theta,h} \{C_I \sum_{i=1}^{k_I} e(f(x_I^i, \theta, h), L_I) + C_H \sum_{j=1}^{k_H} e(f(x_H^j, \theta, h), L_H) + \text{regulizer}(\theta)\} \tag{10.6}$$

where $e(f(x_H^i, \theta, h), L_H)$ and $e(f(x_I^j, \theta, h), L_I)$ are respectively the empirical error associated with the i^{th} sample from the normal (labeled as L_H) and the j^{th} sample from impaired group (labeled as L_I). Suppose with parameter θ^* and threshold h^* we get a false alarm rate R_F^* and detection rate R_D^* on held-out test set. In the extreme case when the number of samples goes to infinity, the regularizer becomes immaterial and therefore only the ratio $r_c = \frac{C_I}{C_H}$ matters. We then have the following two nice properties:

Property 1: The training error is the same as the test error

Property 2: If empirical error is the 0-1 loss, R_D^* is the highest detection rate one can get with false alarm rate R_F^*.

This extreme can serve as a simplified scenario for our analysis, in which we only have to worry about the ratio $r_c = \frac{C_I}{C_H}$ and the training error. Suppose we get (θ^*, h^*) from

[1] For simplicity we only give the general form here, and assume we can find the global optimal solution

the optimization in Equation (10.6), with which we achieve a false alarm rate R_F^* and a detection rate R_D^*. According to Property 2, R_D^* is the highest detection rate we can achieve with a false alarm rate R_F^*. However this nice property no longer holds if we change *only* the threshold $h' \neq h^*$ to get a different classifier, which gives the false alarm rate R'_F and the detection rate R'_D. That is, R'_D is generally not the optimal detection rate for false alarm rate R'_F. When we move along the false alarm rate axis, we need only to change r_c in $(0, +\infty)$, retrain the classifier for every new r_c with the optimization in Equation (10.6), and collect all the (R_F^*, R_D^*) pairs. The optimal ROC can then be obtained by putting all the points (R_F^*, R_D^*) together[2].

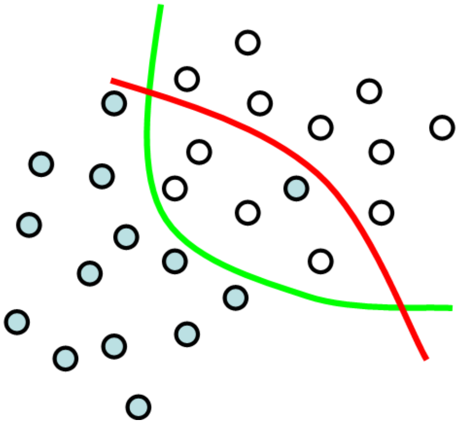


Figure 10.1: An illustrative example of decision boundaries for different mis-classification cost. We use gray dots for positive examples and white dots for negative examples. The green curve stands for the decision boundary when $C_I = C_H$, and the red curve stands for the boundary when $C_I \gg C_H$.

In our case, we use the support vector machine with soft margin as the classifier, which adopts a empirical error as the sum of the distances of the support vectors to the bounder of margins [14] instead of 0-1 loss. Moreover the training samples (< 200) are not abundant enough to ignore the regularizer. Consequently, the aforementioned two properties do not hold. However, we found that in all our experiments, the exact values of C_I and C_H (in a reasonably big range) have only a small influence on the performance of the classification

[2] Bach et al. [3] suggests we have to run through a finite set of values for $\frac{C_I}{C_H}$ by taking advantage of the theory of the optimization path [34]

as long as the ratio $\frac{C_I}{C_H}$ is kept unchanged. Figure 10.2 gives an example of classification on seconds with SVM (with details revealed later), in which we have ROC curves computed two classifiers trained with different setting of mis-classification cost. The first classifier is trained with $C_I = C_H = 10$, while for the second $C_I = 20$ and $C_H = 10$. For each setting of C_I and C_H, we can obtain an optimal (θ^*, h^*), and thus a false alarm rate R_F^* and detection rate R_F^* estimated on a held-out test set. Based on each θ^*, we can get an ROC curve by letting the threshold h vary in a big enough range. On the resulted ROC, the point (R_F^*, R_D^*) is called operating point (shown as crosses in Figure 10.2). We then have two ROC curves corresponding to the two settings of (C_H, C_I), each with its own operating point. For comparison, we also show for each (θ^*, h^*) the false alarm rate $\tilde{R}_F^*$ and detection rate $\tilde{R}_D^*$ estimated on the training set (shown as circles). In left panel of Figure 10.2, we show the ROC curves and the operating points for the two classifiers (blue for Classifier 1, and red for Classifier 2). Apparently, for each classifier the operating point (R_F^*, R_D^*) is different from the $(\tilde{R}_F^*, \tilde{R}_D^*)$ estimated on training set (circle), thus the Property 1 does no hold. Our another observation is that the operating point does not always give the optimal detection rate for the false alarm rate R_F^*, as the operating point (red cross) on the ROC for Classifier 2 (red curve) has been outperformed by Classifier 1 (blue curve) with the same false alarm rate. Nevertheless, it is still true that the ROC curve obtained with a larger $\frac{C_I}{C_H}$ tends to do better with large false alarm rate. Although we cannot build the optimal ROC simply by putting together all the operating points for various value of (C_H, C_I), we can still get a reasonable approximation by concatenating segments of ROCs from different classifiers, with ones trained with small $\frac{C_I}{C_H}$ in charge of the region of small false alarm and ones trained with large $\frac{C_I}{C_H}$ in charge of the region of large false alarm. This procedure consists of the following three steps:

1. Train classifiers with a number of pairs (C_H, C_I), and for each classifier obtain its ROC curve by varying the threshold h. In the experiments, we only consider two different settings: $C_H = 10, C_I = 10$, and $C_H = 10, C_I = 20$.

2. For each classifier, identify the regime of false alarm rate in which it performs the best, and keep the segment of ROC curve above that regime. In this thesis, since

there are only two different cost settings, we divide the false alarm as the higher half and lower half with a proper bounder.

3. Concatenate the different segments of ROC curves gotten from step 2 into a complete piece.

A simple example is given in right panel of Figure 10.2, in which we concatenate the ROCs for two classifiers shown in the left panel.

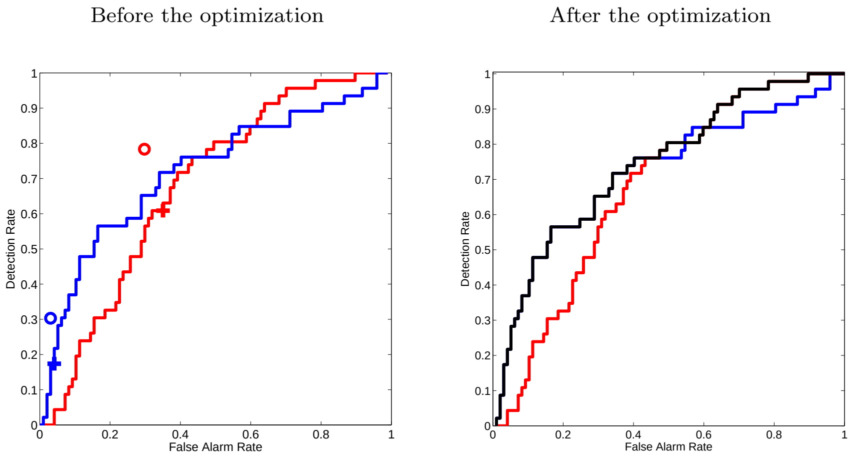


Figure 10.2: Optimization of the ROC curves based on the asymmetry of cost. In the left panel, the blue curves stands for the ROC of the Classifier 1 with $C_I = C_H = 10$, and the red curve stands for the ROC of Classifier 2 with $C_I = 20, C_H = 10$. For each ROC, the cross stands for the operating point estimated on the test set while the circle stands for the (false alarm rate, detection rate) estimated on training set.

10.3 Fisher Kernel and Its Extensions

10.3.1 Fisher Kernel

The Fisher kernel [40] provides a way to extract features from a generative model, and those features will be used in discriminative classifiers. For any θ-parameterized model $p(\cdot;\theta)$, the Fisher kernel between x^i and x^j is defined as

$$K(x^i, x^j) = (\nabla_\theta \log p(x^i;\theta))^T \mathbf{I}^{-1} \nabla_\theta \log p(x^j;\theta), \tag{10.7}$$

where $\mathbf{I}$ is the Fisher information matrix with the (n, m) entry defined as

$$\mathbf{I}_{n,m} = \int_x \frac{\partial \log p(x;\theta)}{\partial \theta_n} \frac{\partial \log p(x;\theta)}{\partial \theta_m} p(x|\theta) dx. \tag{10.8}$$

The Fisher kernel entry $K(x^i, x^j)$ can be viewed as the inner product of the natural gradient $\mathbf{I}^{-1}\nabla_\theta \log p(x;\theta)$ at x^i and x^j with metric $\mathbf{I}$, and is invariant to re-parametrization. Furthermore [40] have shown that a linear classifier based on the Fisher kernel returns a classification result at least as good as the generative model.

It is common in practice that $p(x^i;\theta)$ is a different distribution for different i due to some individual specific property, such as the length of the sequence in HMM modeling. When this is the case, the Fisher information matrix in Equation (10.8) cannot be uniformly defined for all individuals. This difficulty is usually circumvented by simply using a identity matrix in place of Fisher information matrix [39, 61, 25], that is,

$$K(x^i, x^j) = (\nabla_\theta \log p(x^i|\theta))^T \nabla_\theta \log p(x^j|\theta). \tag{10.9}$$

which, unfortunately, renders the kernel dependent on the particular parametrization choice in θ. This is a significant issue: the particular coordinate system (parametrization) used to describe the distribution is immaterial. Under the setting of probabilistic kernel regression [41] this problem becomes asymptotically immaterial when the number of training samples goes to infinity. However, for other applications such as the widely used support vector machine [78], this difference cannot be neglected.

In our project, for any time series $\{\mathbf{t}^i, \mathbf{y}^i\}$, the Fisher score is defined as $\phi_{\mathbf{y}^i} = \nabla_{\widetilde{\mathcal{M}}} \log p(\mathbf{y}^i; \mathbf{t}^i, \widetilde{\mathcal{M}})$. Due to the individual specific observation time $\mathbf{t}^i$, $p(\mathbf{y}^i; \mathbf{t}^i, \widetilde{\mathcal{M}})$ are actually different distribution, and therefore the Fisher information matrix cannot be defined. Simply using identity matrix is particularly unsuitable here since in mixed-effect models the parameters can have vastly different influence on the distribution, and thus the gradient with respect to them. For the mixed-effect model with polynomials as the basis function, the Fisher score entries associated with higher order term (such as slope) is far larger than the entries associated with lower order term (such as intercept). Without proper normalization provided by the Fisher information matrix, the defined kernel will be dominated by higher order entries. Instead of re-weighting the Fisher score entries based on some other heuristic, we propose a principled extension to Fisher kernel which allows the strict calculation of Fisher information.

Our design of kernel is based on the generative hierarchy of mixture of mixed-effect models, as shown in Figure 10.3. We notice that the individual-specific information $\mathbf{t}^i$ enter into this generative process at the last step, and before that the "latent" variables γ^i and z^i are drawn from the same distribution for all individuals. It is thus possible to build a standard Fisher kernel for the latent variables, and based on which, we can further build the kernel between individuals by associating each individual with the corresponding latent variable. We use v^i to denote the latent variables for individual i and $K(v^i, v^j)$ for the Fisher kernel between v^i and v^j. The kernel for $\mathbf{y}^i$ and $\mathbf{y}^j$ is defined as the expectation of $K(v^i, v^j)$ given the observation $\mathbf{y}^i$ and $\mathbf{y}^j$

$$\begin{aligned} K(\mathbf{y}^i, \mathbf{y}^j) &= E[K(v^i, v^j) |\, \mathbf{y}^i, \mathbf{y}^j; \mathbf{t}^i, \mathbf{t}^j, \widetilde{\mathcal{M}}] && (10.10) \\ &= \int\int K(v^i, v^j) p(v^i |\, \mathbf{y}^i; \mathbf{t}^i, \widetilde{\mathcal{M}}) p(v^j |\, \mathbf{y}^j; \mathbf{t}^j, \widetilde{\mathcal{M}}) dv^i dv^j && (10.11) \end{aligned}$$

Based on the choice of latent variable v and the kernel form for $K(v^i, v^j)$, we have the kernel design strategies in the next three subsections. This extension to the Fisher kernel enables us to deal with time series with unequal length, and to our knowledge it has not been reported in literature.

10.3.2 Design A

This kernel design takes the $\{\gamma^i\}$ as the latent variables and chooses to marginalize out the higher level latent variable $\{z^i\}$. That is, we consider each individual regression model parameters γ is drawn from the following mixture of Gaussian distributions

$$p(\gamma | \widetilde{\mathcal{M}}) = \pi_0 p(\gamma;\, \alpha_0, \mathbf{D}_0) + \pi_1 p(\gamma;\, \alpha_1, \mathbf{D}_1) \equiv p(\gamma; \tilde{\Theta})$$

where $\tilde{\Theta} = \{\pi_0, \alpha_0, \mathbf{D}_0, \pi_1, \alpha_1, \mathbf{D}_1\}$ are the parameters of the corresponding Gaussian mixture model, and $p(\gamma; \alpha_m, \mathbf{D}_m)$ $(m = 0, 1)$ is simply a Gaussian distribution with mean α_m and covariance $\mathbf{D}_m$. This generative process is illustrated in Figure 10.4, which is the same graphical model in Figure 10.3 with latent variable z^i integrated out[3]. The Fisher

[3] Strictly speaking, we cannot integrate out z^i at this step since the group membership is used later in generating the observation noise. However this is a reasonable approximation here since the observation

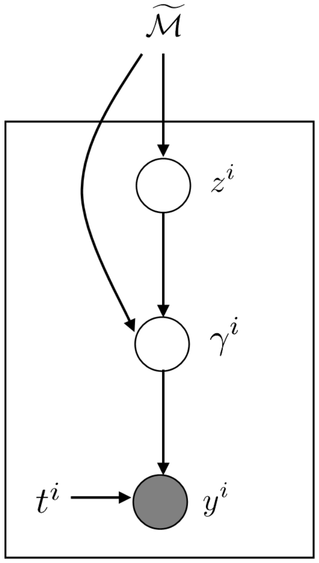


Figure 10.3: The graphical model of the mixture of mixed-effect models, diagram for the Fisher kernel extensions. Same as Figure 7.2.

kernel for γ is defined as

$$K(\gamma^i, \gamma^j) = (\nabla_{\tilde{\Theta}} \log p(\gamma^i|\tilde{\Theta}))^T (\mathbf{I}^\gamma)^{-1} \nabla_{\tilde{\Theta}} \log p(\gamma^i|\tilde{\Theta}), \tag{10.12}$$

where the Fisher score is defined

$$\nabla_{\tilde{\Theta}} \log p(\gamma^i; \tilde{\Theta}) = [\frac{\partial \log p}{\partial \pi_0}; \frac{\partial \log p}{\partial \alpha_0}; \frac{\partial \log p}{\partial \mathbf{D}_0}; \frac{\partial \log p}{\partial \pi_1}; \frac{\partial \log p}{\partial \alpha_1}; \frac{\partial \log p}{\partial \mathbf{D}_1}]^T.$$

and the Fisher information matrix $\mathbf{I}^\gamma$ is defined as

$$\mathbf{I}^\gamma_{n,m} = \int_x \frac{\partial \log p(\gamma; \tilde{\Theta})}{\partial \tilde{\Theta}_n} \frac{\partial \log p(\gamma; \tilde{\Theta})}{\partial \tilde{\Theta}_m} p(\gamma|\tilde{\Theta}) d\gamma. \tag{10.13}$$

Once $K(\gamma^i, \gamma^j)$ is defined, we can further define the kernel between $\mathbf{y}^i$ and $\mathbf{y}^j$ as the expectation of $K(\gamma^i, \gamma^j)$ given $\mathbf{y}^i$ and $\mathbf{y}^j$:

$$\begin{aligned}
K(\mathbf{y}^i, \mathbf{y}^j) &= E[K(\gamma^i, \gamma^j)|\, \mathbf{y}^i, \mathbf{y}^j; \mathbf{t}^i, \mathbf{t}^j, \widetilde{\mathcal{M}}] && (10.14)\\
&= \int\int K(\gamma^i, \gamma^j) p(\gamma^i|\mathbf{y}^i; \mathbf{t}^i, \widetilde{\mathcal{M}}) p(\gamma^j|\mathbf{y}^j; \mathbf{t}^j, \widetilde{\mathcal{M}}) d\gamma^i d\gamma^j && (10.15)\\
&= (\int \nabla_{\tilde{\Theta}} \log p(\gamma^i|\tilde{\Theta}) p(\gamma^i|\, \mathbf{y}^i; \mathbf{t}^i, \widetilde{\mathcal{M}}) d\gamma^i)^T (\mathbf{I}^\gamma)^{-1} \int \nabla_{\tilde{\Theta}} \log p(\gamma^j|\tilde{\Theta}) p(\gamma^j|\, \mathbf{y}^j; \mathbf{t}^j \widetilde{\mathcal{M}}) d\gamma^j. && (10.16)
\end{aligned}$$

noise specified by $\mathcal{M}_0$ and $\mathcal{M}_1$ have almost the same variance, see Appendix B.

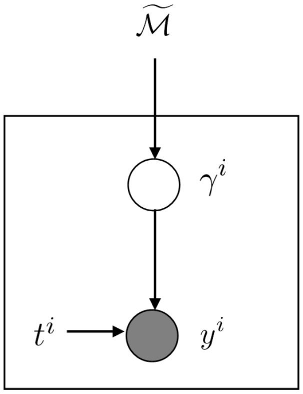


Figure 10.4: The graphical model of the mixture of mixed-effect models for Design A, where the latent variable z^i is marginalized out.

The computational drawback of this design strategy is that the integral in evaluating $\mathbf{I}^r$ and $\int \nabla_{\tilde{\Theta}} \log p(\gamma^j|\tilde{\Theta})p(\gamma^j|\,\mathbf{y}^j;\mathbf{t}^j\widetilde{\mathcal{M}})d\gamma^j$ generally does not have an analytical solution. In our experiments, we estimated the integral by Monte-Carlo sampling [17].

10.3.3 Design B

This design strategy takes both γ^i and z^i as joint latent variables and builds a Fisher kernel for them. The probability model, as summarized in Figure 10.5 for the latent variable would be

$$p(z^i, \gamma^i; \tilde{\Theta}) = \pi_{z^i} p(\gamma_i; \alpha_{z^i}, \mathbf{D}_{z^i})$$

where $\tilde{\Theta}$ is defined the same as in Design A.

The Fisher score would be

$$\nabla_{\tilde{\Theta}} \log p(z^i, \gamma^i; \tilde{\Theta}) = [\frac{\partial \log p}{\partial \pi_0}; \frac{\partial \log p}{\partial \alpha_0}; \frac{\partial \log p}{\partial \mathbf{D}_0}; \frac{\partial \log}{\partial \pi_1}; \frac{\partial \log p}{\partial \alpha_1}; \frac{\partial \log p}{\partial \mathbf{D}_1}]^T.$$

and the Fisher kernel for the joint variable (γ^i, z^i) is defined as

$$K((z^i, \gamma^i), (z^j, \gamma^j)) = (\nabla_{\tilde{\Theta}} \log p(\gamma^i|\tilde{\Theta}))^T (\mathbf{I}^{z,\gamma})^{-1} \nabla_{\tilde{\Theta}} \log p(\gamma^i|\tilde{\Theta}), \tag{10.17}$$

where $\mathbf{I}^{z,\gamma}$ is the Fisher information matrix. The kernel for $\mathbf{y}^i$ and $\mathbf{y}^j$ is defined in a similar

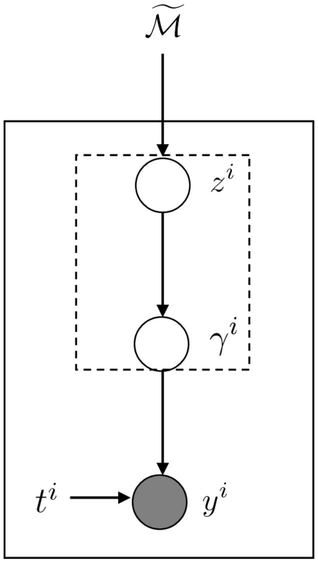


Figure 10.5: The graphical model of the mixture of mixed-effect models for Design B. The variables in the dashed box are considered the joint latent variable.

way as in Design A:

$$
\begin{aligned}
K(\mathbf{y}^i, \mathbf{y}^j) &= E_{z^i,\gamma^i,z^j,\gamma^j}[K((z^i, \gamma^i), (z^j, \gamma^j))|\,\mathbf{y}^i, \mathbf{y}^j; \mathbf{t}^i, \mathbf{t}^j \widetilde{\mathcal{M}}] && (10.18)\\
&= \int\int \sum_{z^i} \sum_{z^j} K((z^i, \gamma^i), (z^j, \gamma^j)) p(z^i, \gamma^i|\,\mathbf{y}^i; \mathbf{t}^i, \widetilde{\mathcal{M}}) p(z^j, \gamma^j|\,\mathbf{y}^j; \mathbf{t}^j, \widetilde{\mathcal{M}}) d\gamma^i d\gamma^j && (10.19)
\end{aligned}
$$

The integration in Equation (10.19)and the calculation of Fisher information matrix can be carried out analytically.

This kernel design is interestingly connected to the so-called marginalized kernel proposed by Tsuda et al. [77]. The marginalized kernel also considers a distribution with discrete latent variable h (indicating the generating component) and observable x, which form a *complete* variable $\mathbf{x} = (h, x)$. The kernel for observable x^i and x^j is defined as

$$K(x^i, x^j) = \sum_{h^i} \sum_{h^j} P(h^i|x^i) P(h^j|x^j) K(\mathbf{x}^i, \mathbf{x}^j)$$

where $K(\mathbf{x}^i, \mathbf{x}^j)$ is the joint kernel for complete variables, which usually takes the form

$$K(\mathbf{x}^i, \mathbf{x}^j) = \delta(h^i, h^j) K_{h^i}(x^i, x^j). \tag{10.20}$$

with $K_{h^i}(x^i, x^j)$ is the kernel pre-defined for the h^i component generative model. It is clear from Equation (10.20) that $K(\mathbf{x}^i, \mathbf{x}^j)$ is zero if x^i and x^j are generated from different component models ($h^i \neq h^j$), otherwise it takes the value of kernel defined for the m^{th} component model assuming $h^i = h^j = m$.

As an alternative to the definition in Equation (10.17), we can define a joint kernel for (z^i, γ^i) in a way similar to the marginalized kernel as in

$$\widetilde{K}((z^i, \gamma^i), (z^j, \gamma^j)) = K_{z^i}(\gamma^i, \gamma^j)\delta(z^i, z^j), \tag{10.21}$$

where $K_m(\gamma^i, \gamma^j)$ is the Fisher kernel between γ^i and γ^j with the m^{th} component in mixture $\tilde{\Theta}$ as the generative model

$$K_m(\gamma^i, \gamma^j) = (\nabla_{\Theta_m} \log p(\gamma^i; \alpha_m, \mathbf{D}_m))^T \mathbf{I}_m^{-1} \nabla_{\Theta_m} \log p(\gamma^i; \alpha_m, \mathbf{D}_m). \tag{10.22}$$

Clearly the kernel between (z^i, γ^i) and (z^j, γ^j) is non-zero only if they are drawn from the same component mixed-effect model. Again we define the kernel between $\mathbf{y}^i$ and $\mathbf{y}^j$ as

$$\widetilde{K}(\mathbf{y}^i, \mathbf{y}^j) \quad = \quad E_{z^i,\gamma^i,z^j,\gamma^j}[\widetilde{K}((z^i, \gamma^i), (z^j, \gamma^j))|\, \mathbf{y}^i, \mathbf{y}^j; \mathbf{t}^i, \mathbf{t}^j, \widetilde{\mathcal{M}}] \tag{10.23}$$

Interestingly, the kernel $K((z^i, \gamma^i), (z^j, \gamma^j))$ from Equation (10.17) and $\widetilde{K}((z^i, \gamma^i), (z^j, \gamma^j))$ in Equation (10.21) are related through the following equation

$$K((z^i, \gamma^i), (z^j, \gamma^j)) = \frac{1}{\pi_{z^i}}\widetilde{K}((z^i, \gamma^i), (z^j, \gamma^j)) + \frac{1}{\pi_{z^i}}\delta(z^i, z^j).$$

with full derivation in Appendix B.

10.3.4 Design C

We can also base the kernel design on one mixed-effect model component instead on the mixture. Equivalently, we assume that the mixture model contains only one component in Design A or Design B, as illustrated in Figure 10.6.

For the mixed-effect model for the group indexed m, the Fisher score for the i^{th} individual

$$\nabla_{\Theta_m} \log p(\gamma^i; \Theta_m)$$

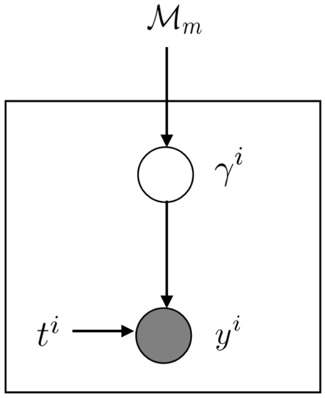


Figure 10.6: The graphical model for the mixed-effect models for Design C.

describes how the log probability $p(\gamma^i; \Theta_m)$ responds to the change of mixed-effect model parameters Θ_m. This is a valid feature for classification since the likelihood of γ_i for individuals from different groups are likely to have different responses to the change of parameters Θ_m). The kernel between γ^i and γ^j is same as the one defined in Equation (10.22) in Design B

$$K_m(\gamma^i, \gamma^j) = (\nabla_{\Theta_m} \log p(\gamma^i; \alpha_m, \mathbf{D}_m))^T \mathbf{I}_m^{-1} \nabla_{\Theta_m} \log p(\gamma^i; \alpha_m, \mathbf{D}_m), m = 0, 1$$

And then the kernel for $\mathbf{y}^i$ and $\mathbf{y}^j$:

$$K(\mathbf{y}^i, \mathbf{y}^j) = E[K(\gamma^i, \gamma^j)|\,\mathbf{y}^i, \mathbf{y}^j; \mathbf{t}^i, \mathbf{t}^j, \mathcal{M}_m] \tag{10.24}$$
$$= \int\int K(\gamma^i, \gamma^j) p(\gamma^i|\mathbf{y}^i; \mathbf{t}^i, \mathcal{M}_m) p(\gamma^j|\mathbf{y}^j; \mathbf{t}^j, \mathcal{M}_m) d\gamma^i d\gamma^j \tag{10.25}$$
$$= (\int \nabla_{\tilde{\Theta}} \log p(\gamma^i|\tilde{\Theta}) p(\gamma^i|\,\mathbf{y}^i; \mathbf{t}^i, \mathcal{M}_m) d\gamma^i)^T \mathbf{I}_m^{-1} \int \nabla_{\tilde{\Theta}} \log p(\gamma^j|\tilde{\Theta}) p(\gamma^j|\,\mathbf{y}^j; \mathbf{t}^j, \mathcal{M}_m) d\gamma^j . \tag{10.26}$$

The integration in Equation (10.26)and the calculation of Fisher information matrix can be carried out analytically.

We can choose to use the mixed-effect model trained on the impaired group or the normal group. Not surprisingly, mixed-effect model fit on different groups describe the data in quite different ways, and therefore lead to different kernels. Indeed, our experiments show that the kernel based on the impaired group is significantly better than others (details omitted). Therefore this kernel is chosen as the representative of Design C. In this case, the designed kernel is essentially a special case of Design A or Design B when $\pi_0 = 1$ and $\pi_1 = 0$.

10.3.5 Empirical Comparison: Three Designs

We tested the three Fisher kernel extensions on the four motor behaviors: steps, seconds, tappingD, and tappingN, and the two neuro-psychological tests: delayed-recall and logical memory II, with the mixed-effect models for each measurement trained separately. We use the order-1 polynomials (linear)[4] as the basis functions for the mixed-effect models. For each measurement, the constructed kernels are used in support vector machines for classification.

We compare the performance of two kernels by comparing the ROC curves generated based on them (see Section 9.2). The reported ROC curves are estimated through a leave-one-out cross-validation and the optimization procedure described in Section 10.2. In each validation round, we use $k-1$ subjects training a support vector machine, and test the trained classifier on the left out subject. We report the average test classification accuracy over all k validation rounds. The same training-test strategy will be used with other kernel designs in the rest of this section.

We first compare Design A and Design B, two designs based on mixture of mixed-effect models, in Figure 10.7. Clearly the two kernel has very much comparable performance except on except on tappingD, on which ROC Design A is better than that from Design B with $p = 0.136$. We then compare Design A and Design C in Figure 10.8, from which we find that Design C yields slightly better ROC curves than Design A ((which in turn is slightly better than Design B) on all four motor behaviors, and is comparable to Design A (then Design B) on the two neuro-psychological tests.

[4] We did not report the result of order-2 polynomials (quadratic) since they yield significantly worse result than order-1 polynomials.

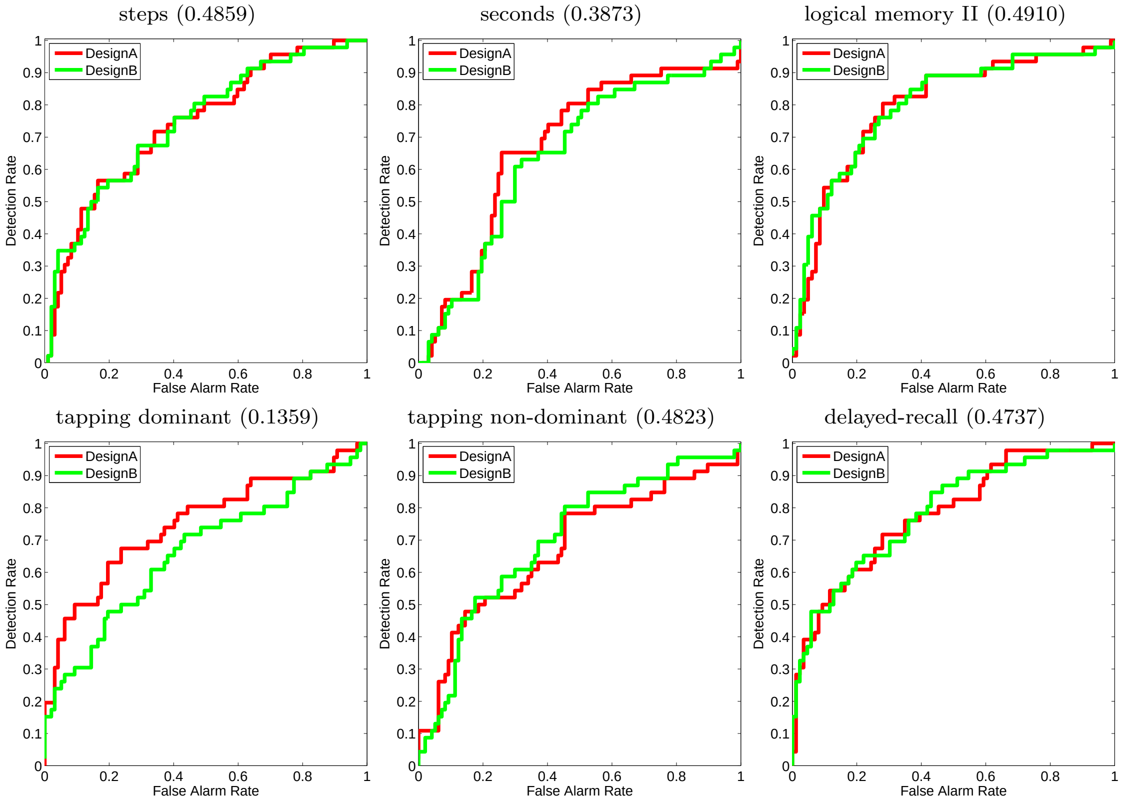
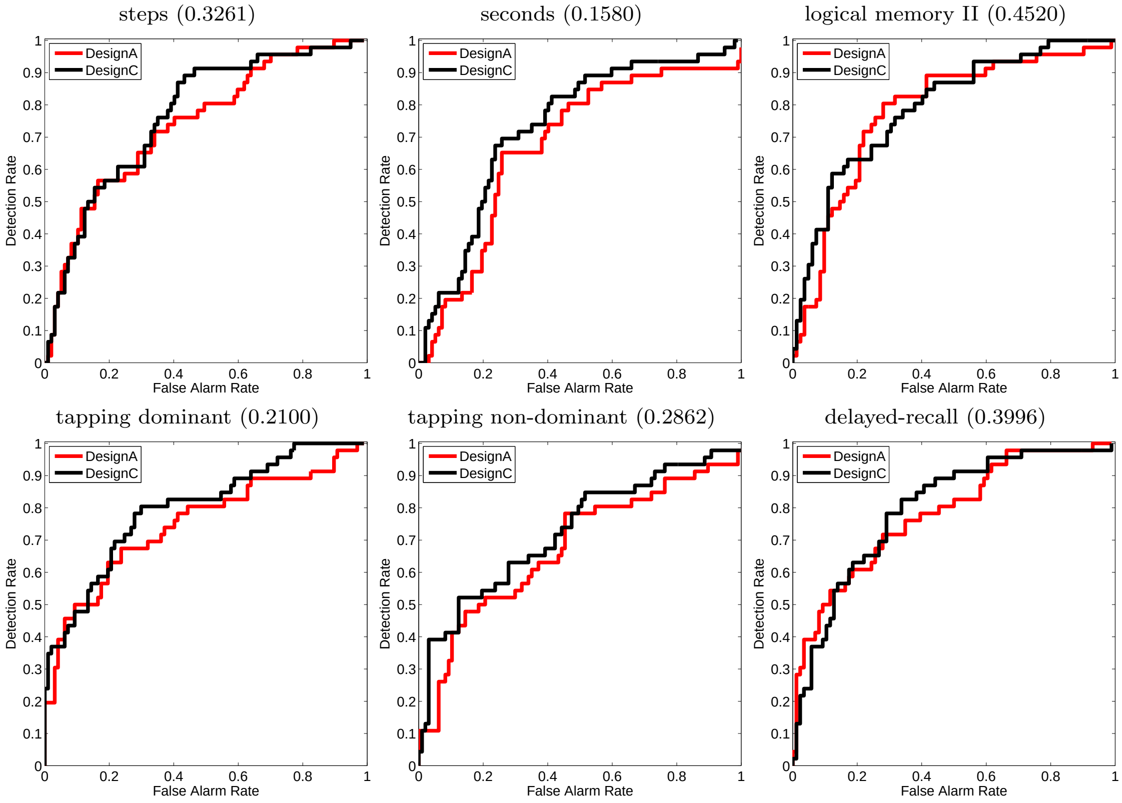


Figure 10.7: Design A Vs. Design B. The number in the parenthesis is the p-value (Z-test) for the null-hypothesis "the AUC of Design A is the same as the AUC of Design B".

10.4 Feature Extraction Routines Independent of Generative Models

We will discuss in this section two other feature extraction routines that are not based on mixed-effect models or any other generative models. The first one models each subject with an independent polynomial curve and uses the least-square fitting coefficients as the feature vector; The second one takes a non-parametric approach fitting the observations of each subject with a smooth curve, and uses it as the summarizing feature for the classification afterwards.



steps (0.3261)

seconds (0.1580)

logical memory II (0.4520)

tapping dominant (0.2100)

tapping non-dominant (0.2862)

delayed-recall (0.3996)

Figure 10.8: Design A Vs. Design C. The number in the parenthesis is the p-value (Z-test) for the null-hypothesis "the AUC of Design A is the same as the AUC of Design C".

10.4.1 Feature Extraction 1: Parametric

We can summarize each individual time series with the least-square fit coefficients for a d-degree polynomial regression model. That is, for subject i, the $d+1$-dimensional feature $\mathbf{p}^i$ is

$$\mathbf{p}^i = \arg\min_{\mathbf{p}} \sum_{j=1}^{N^i} (\sum_{l=0}^{d} p_l (t_j^i)^l - y_j^i)^2. \tag{10.27}$$

where $\mathbf{p}^i$ is the vector of coefficients for individual i. We consider only $d = 1$ since a substantial proportion of subjects have no more than five observations, not enough for a reliable fitting of polynomial with $d \geq 2$. We normalize the entries in $\mathbf{p}^i$ to the estimated standard deviation before we use them as input to a support vector machine

$$\hat{p}_l^i = \frac{p_l^i}{\sqrt{\frac{1}{k-1}\sum_j (p_l^j - \overline{p}_l)^2}}, \ l = 0, \cdots, d, \tag{10.28}$$

with $\overline{p}_l = \frac{1}{k}\sum_{i=1}^k p_l^i$. We use a RBF kernel based on the square Euclidean distance

$$\mathbf{K}_{ij} = \exp(-\frac{||\hat{\mathbf{p}}^i - \hat{\mathbf{p}}^j||_2^2}{2s^2}), \tag{10.29}$$

where $\hat{\mathbf{p}}^i = [\hat{p}_0^i, \cdots, \hat{p}_d^i]^T$. In the remainder of the thesis, we will refer to the matrix $\mathbf{K}$ defined in Equation (10.29) as the LSQ kernel, although the feature defined in Equation (10.27) and (10.28) can be readily used in other classifiers such as mulit-layer perceptron (MLP) [11].

10.4.2 Feature Extraction 2: Non-parametric

We can extend the feature extraction described in Section 10.4.1 to non-parametric form as follows. Our model is based on Gaussian processes [65] and reproducing kernel Hilbert space (RKHS). Basically, we assume the observation for each individual is generated from a independent Gaussian process indexed by age. So for each individual i, its n^{th} observation is

$$y_n^i = f^i(t_n^i) + \epsilon_n^i, \ \ n = 1, 2, \cdots, N_i$$

where ϵ_n^i is a white observation noise with standard deviation σ. Assuming the f^i with different i have the same covariance function, denoted as C, we can then summarize each individual with the following smooth function of time:

$$\hat{f}^i(t) = E[f(t)|\mathbf{y}^i; \mathbf{t}^i, C, \sigma] = C(t, \mathbf{t}^i)(C(\mathbf{t}^i, \mathbf{t}^i) + \sigma^2\mathbb{I})^{-1}\mathbf{y}^i$$

where $\mathbf{y}^i$ is the vector of observations for individual i and $C(\mathbf{t}^i, \mathbf{t}^i)$ is the matrix with its entries (n, m) being $C(t_n^i, t_m^i)$. The difference between two individuals can be measured as the distance between two summarizing smooth curves in a Hilbert space

$$d_{ij} = ||\hat{f}^i - \hat{f}^j||_{\mathcal{H}}^2, \tag{10.30}$$

where $||\cdot||_{\mathcal{H}}$ is a norm in Hilbert space $\mathcal{H}$ [5].

When $\mathcal{H}$ is the reproducing kernel Hilbert space (RKHS) induced by the covariance function C, this distance measure is well-defined, and can be calculated as

$$\begin{aligned} d_{ij} &= ||\hat{f}^i - \hat{f}^j||_{\mathcal{H}} \\ &= ||C(t,\mathbf{t}^i)(C(\mathbf{t}^i,\mathbf{t}^i)+\sigma^2\mathbb{I})^{-1}\mathbf{y}^i - C(t,\mathbf{t}^j)(C(\mathbf{t}^j,\mathbf{t}^j)+\sigma^2\mathbb{I})^{-1}\mathbf{y}^j||_{\mathcal{H}} \\ &= < C(t,\mathbf{t}^i)\mathbf{v}^i - C(t,\mathbf{t}^j)\mathbf{v}^j, C(t,\mathbf{t}^i)\mathbf{v}^i - C(t,\mathbf{t}^j)\mathbf{v}^j >_{\mathcal{H}} \end{aligned}$$

where $\mathbf{v}^i = (C(\mathbf{t}^i,\mathbf{t}^i)+\sigma^2\mathbb{I})^{-1}\mathbf{y}^i$. Considering that

$$< C(t_n,t), C(t_m,t) >_{\mathcal{H}} = C(t_n,t_m),$$

the distance measurement can be simplified as

$$d_{ij} = (\mathbf{v}^i)^T C(\mathbf{t}^i,\mathbf{t}^i)\mathbf{v}^i + (\mathbf{v}^j)^T C(\mathbf{t}^i,\mathbf{t}^i)\mathbf{v}^j - 2(\mathbf{v}^i)^T C(\mathbf{t}^i,\mathbf{t}^j)\mathbf{v}^j$$

The distance d_{ij} can also be interpreted as the Bregman divergence on f with $||f||^2_{\mathcal{H}}$ as the seed functional [29].

Based on the proposed distance, we defined the following kernel

$$\mathbf{K}_{ij} = \exp(-\frac{d_{ij}}{2s^2}). \tag{10.31}$$

The matrix $\mathbf{K}$ defined in Equation (10.31) is Mercer kernel, simply because it can be rewritten as RBF kernel after we embed the distance d_{ij} into a N-dimensional Euclidean space with $N = \sum_{i=1}^{N_i}$. In the remainder of the thesis, we will refer to this kernel as Gaussian process kernel. To get such a kernel, we need specify the covariance function C and s used in Equation (10.31). In this thesis, we use another RBF kernel for C.

10.4.3 Comparison of the Parametric and Non-parametric Feature Extraction

We compare classifiers built on the parametric and non-parametric kernels in Figure 10.9. For each design, we set the radius s to maximize the performance at the operating point.

[5] One might want to use $E[||f^i - f^j||^2_{\mathcal{H}}|\mathbf{y}^i,\mathbf{y}^j;\mathbf{t}^i,\mathbf{t}^j]$ as the measure of distance. Unfortunately, this expectation goes to infinity. One can understand this by imagining Gaussian process as a multivariate Gaussian distribution with infinite (usually uncountable) dimensions.

As Figure 10.9 shows, Gaussian process kernel generally yields larger AUC than the LSQ kernel, while this superiority is significant ($p < 0.05$) only on tappingN, while on delayed-recall and logical memory II, the performance of the two kernels are very much comparable.

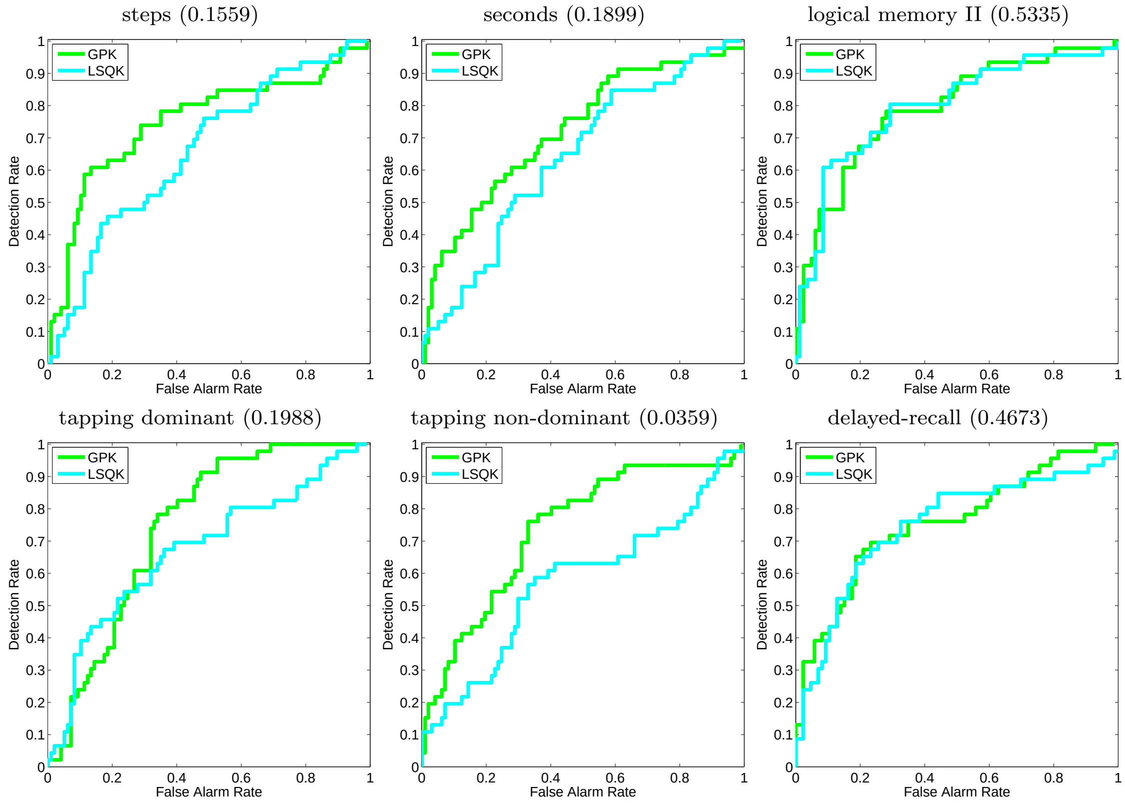


Figure 10.9: Gaussian process kernel (GPK) vs. least squares kernel (LSQK). The number in the parenthesis is the p-value of the Z-test for the null-hypothesis "the AUC of Gaussian process kernel is the same as the AUC of LSQ kernel".

10.5 Comparing Generative and Discriminative Models

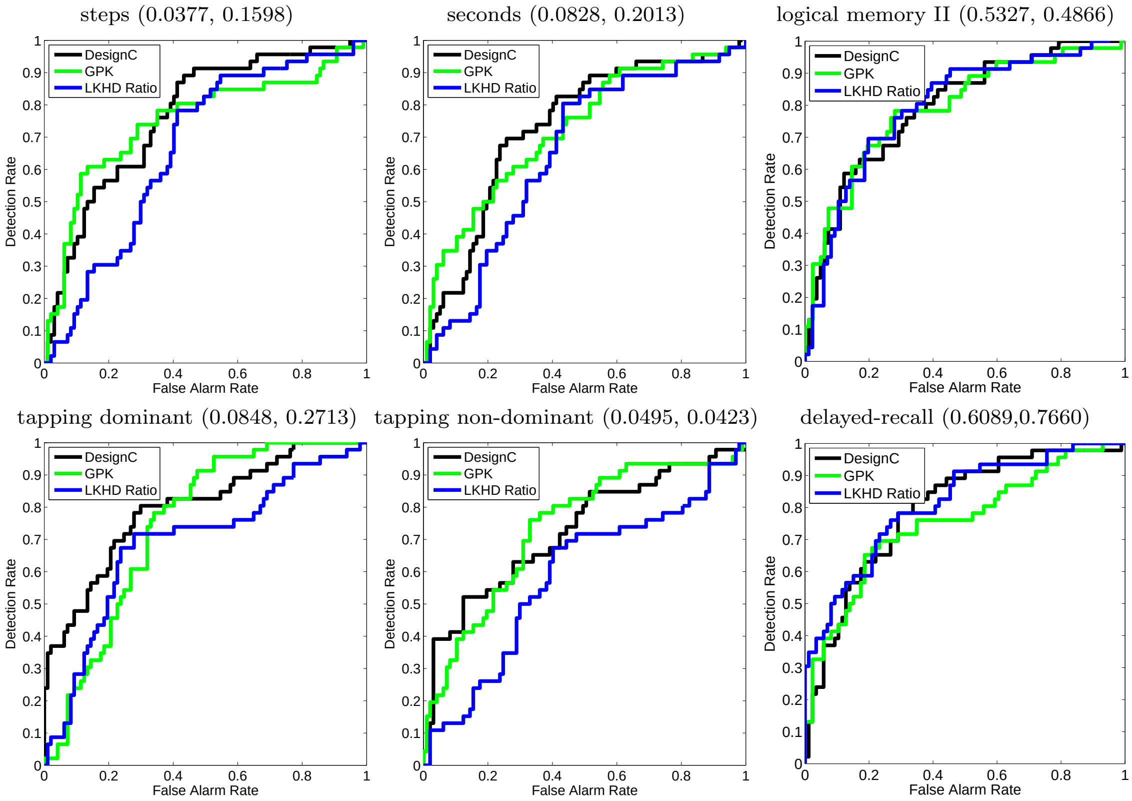
Finally, we compare the best performer in each of the three categories of methods

- For generative models (Chapter 9), we choose the one based on mixed-effect model, since it is slightly better than the one based on the simplified model that assumes no random effect.

- For the discriminative models based on the extensions to the Fisher kernel (Section 10.3), we pick Design C. It is consistently better than Design A and B on all six measurements, although the difference is not statistically significant.
- For the feature extraction models independent of the mixed-effect model (Section 10.4), we pick Gaussian process kernels since it is better than the LSQ kernel on all six measurements, although the difference is not statistically significant except on tappingN.

Figure 10.10 compares the three best performers in their own category. As it shows, both discriminative models are better than the likelihood ratio classifier except on delayed-recall and logical memory II, on which the two schools of models have similar performance. On the four motor measurements the two discriminative models work better than likelihood ratio classifiers. Among the discriminative models, the one based on Fisher kernel extension (Design C) outperforms the likelihood ratio classifiers with $p < 0.1$, while the superiority of the Gaussian process kernel over likelihood ratio classifiers is generally not statistically significant.

It is also meaningful to group the classifiers we have discussed so far into parametric ones (including mixed-effect model, the Fisher kernel extensions, and the LSQ kernel), and non-parametric ones (Gaussian process kernel). All the three parametric classifiers adopt polynomial regression models (in our experiments we use a linear model) for individual subjects, while the non-parametric one uses Gaussian process as the individual regression model. Intuitively, the performance of the three parametric classifiers depends to a great extent on the suitability of polynomial as the regression model. As we can see from the mixed-effect models fit on the medical data (Figure 7.3 and 7.4), for the seconds, steps, tappingD, delayed recall, and logical memory II, there is a significance difference between the linear regression models from the impaired group and the normal group, while on tappingN, the difference is not as significant. This observation is further verified in Chapter 9 where we showed that the performance of the mixed-effect model-based classifier on tappingN is the worst among all six measurements. Not surprisingly, a direct comparison of results on tappingD and tappingN shows, if either of other two parametric models is used, tappingD yields better result than tappingN. On the other hand, the non-parametric one (Gaussian process kernel) seems to immune to this, since the results on tappingD and tappingN are much comparable.


steps (0.0377, 0.1598)

seconds (0.0828, 0.2013)

logical memory II (0.5327, 0.4866)

tapping dominant (0.0848, 0.2713)

tapping non-dominant (0.0495, 0.0423)

delayed-recall (0.6089,0.7660)

Figure 10.10: The comparison between Fisher kernel extension (Design C), Gaussian process kernel and likelihood ratio classifiers (mixed-effect model). For each measurement, the first number in the parenthesis is the *p*-value (Z-test) for the null hypothesis " the AUC of Design C is same as the AUC of likelihood ratio classifier", and the second number is the *p*-value (Z-test) for the null hypothesis "the AUC of Gaussian process kernel is the same as the AUC of likelihood ratio classifier".

Chapter 11

Cognitive Decline Detection: Contribution and Future Research

11.1 Contribution

In Chapter 9 and Chapter 10, we presented a number of models for predicting whether a cognitively healthy subject will later develop into cognitive impairment based on his or her clinical records. These binary classification algorithms should be viewed as an initial step towards a more sophisticated cognitive decline detection system that can estimates for each subject the risk of dementia at any given age.

We developed and compared two schools of methods: the generative models and discriminative models. For generative models, we adopt the mixed-effect model, for its modeling ability on both population and individual level. In Chapter 7 we have discussed the mixed-effect models in detail and shown that the mixed-effect models can capture the difference on a population level between normal group and impaired group. In Chapter 8, we have further demonstrated the mixed-effect model as a proper generative model, by showing that the model fit on a proper population can serve as an informative prior for the regression modeling of a unseen subject. In Chapter 9, we explored the discriminative capability of a mixed-effect model on individual level, and built a likelihood ratio classifier with the mixed-effect models for two groups as the generative model. This classifier yields reasonable classification results on four motor behaviors, and has excellent performance two neuro-psychological tests.

The discriminative models are proposed to improve the classification performance, for which we proposed two categories of methods. The first category is a hybrid of discriminative and generative models, which employs the widely used support vector machine as the classifiers but rely on the mixed-effect models for feature extraction. We extended the Fisher kernel, an conventional way of extracting feature from generative models, to deal with our situation where different time series have unequal observation time, by exploiting the latent structure of the mixed-effect model. The second category of methods consists of two other feature extraction routines, parametric and non-parametric, that are independent of the mixed-effect models. Among them, the non-parametric one actually gives a new distance measure for a wide range of time series, which is potentially useful in a variety of other applications. Our experiments show that the discriminative methods yield significant better classification performance than generative models on motor behaviors, and are comparable to mixed-effect models on neuro-psychological tests.

11.2 Future Research

11.2.1 Evaluating the Risk at Different Ages

Ultimately we want the detection system to give an estimate of the time horizon to a clinical diagnosis, or the time horizon to decline to more severe impairment for a mildly impaired individual. In addition, each estimation should come with a certainty value to facilitate the integration in a clinical decision system. In Chapter 9 and Chapter 10, we described approaches to predict if an individual will become impaired. This static classification approach we adopted in this thesis does not provide the risk information useful for the clinical decision making, and is potentially inaccurate due to the possibility that people in the normal group could develop into impairment later. Both drawbacks can be avoided in the standard survival analysis with the so called *survival function.* It is thus desired to combine the kernel methods we developed in Chapter 10 with survival analysis to achieve a kernelized survival analysis, which is, to our knowledge, a novel synthesis..

11.2.2 Combining Kernels from Different Measurements

The kernel design method described in Section 10.3 can also be applied to the mixed-effect model with multiple output variable, thus give a joint kernel for measurements of different types. As a special case, when we use the naive Bayesian as the joint mixed-effect model, the kernel (for Design A, B or C) for the joint variable $\bar{\mathbf{y}}^i$ (see Section 7.4.2) is

$$K(\bar{\mathbf{y}}^i, \bar{\mathbf{y}}^j) = K^{(1)}(\mathbf{y}^{(1),i}, \mathbf{y}^{(1),j}) + K^{(2)}(\mathbf{y}^{(2),i}, \mathbf{y}^{(2),j}) + \cdots + K^{(L)}(\mathbf{y}^{(L),i}, \mathbf{y}^{(L),j}),$$

where $K^{(l)}$ is the kernel for measurement of type l. As we have discussed in Chapter 7, the multi-variate mixed-effect model does not provides a fruitful fusion of the measurements of different types. Not surprisingly, the kernel based on this does not show an improved discriminative power over the kernel based on mixed-effect model for individual measurement types.

Besides fusing the different measurement on the level of generative model, we can also do it in the later feature extraction phase for discriminative models. It is suggested by Lanckriet et al. [49] that we may construct kernels from different sources separately, and try to combine them through a proper convex combination of them in the following form,

$$K(\bar{\mathbf{y}}^i, \bar{\mathbf{y}}^j) = \lambda^{(1)} K^{(1)}(\mathbf{y}^{(1),i}, \mathbf{y}^{(1),j}) + \lambda^{(2)} K^{(2)}(\mathbf{y}^{(2),i}, \mathbf{y}^{(2),j}) + \cdots + \lambda^{(L)} K^{(L)}(\mathbf{y}^{(L),i}, \mathbf{y}^{(L),j}), \quad (11.1)$$

with $\sum_{l=1}^{L} \lambda^{(l)} = 1$ to be decided. Intuitively we desire a kernel K that fits the class label well. One criteria for such a fitness, as suggested by [49], is the kernel-target alignment, which is also known to render the optimization over $\{\lambda^{(l)}\}$ convex. Another way to fuse the kernel, known as kernel extrapolation [80], can be obtained by viewing the kernels as the reproducing kernel for a RKHS, and working directly with the norm of vector in the RKHS. The resulted kernel is of the following form:

$$K(\bar{\mathbf{y}}^i, \bar{\mathbf{y}}^j) = (\lambda^{(1)} K^{(1)}(\mathbf{y}^{(1),i}, \mathbf{y}^{(1),j})^{-1} + \lambda^{(2)} K^{(2)}(\mathbf{y}^{(2),i}, \mathbf{y}^{(2),j})^{-1} + \cdots + \lambda^{(L)} K^{(L)}(\mathbf{y}^{(L),i}, \mathbf{y}^{(L),j})^{-1})^{-1},$$

where again the weight $\{\lambda^{(1)}, \lambda^{(2)}, \cdots, \lambda^{(L)}\}$ are to be decided.

11.2.3 Extension to Unobtrusive Home Assessment

It is useful to extend our method to unobtrusive home assessments, such as walking speed estimated by home-installed motion sensors and finger tapping measured based on everyday computer use. These home assessments are cheap to obtain in a relatively high frequency (e.g., on daily basis), but subject to various form of interfering factors. A sensible use of home assessments needs to consider all these factors that can affect the measurement, such as absence from home or having guests, which themselves could be hard to identify based on the relatively sparse information collected by the installed sensors. In addition, we need to tune the mixed-effect models and the discriminative approaches to the characteristics of home assessment. All these problems remain to be solved.

Bibliography

[1] Aggarwal, N. T., Wilson, R. S., Beck, T. L., Bienias, J. L., and Bennett, D. A. Motor dysfunction in mild cognitive impairment and the risk of incident alzheimer diseas. *Arch. Neurol. 64* (2006), 1763–1769.

[2] Ambroise, C., Dang, M., and Govaert, G. Clustering of spatial data by the EM algorithm. In *Geostatistics for Environmnental Applications* (1997), vol. 3, pp. 493–504.

[3] Bach, F., Heckerman, D., and Horvitz, E. On the path to an ideal roc curve: Considering cost asymmetry in learning classifiers. In *AISTATS* (2005).

[4] Basu, S., Banerjee, A., and Mooney, R. Active Semi-Supervision for Pairwise Constrained Clustering. In *Proceedings of the SIAM International Conference on Data Mining* (2004), pp. 333–344.

[5] Basu, S., Bannerjee, A., and Mooney, R. Semi-supervised clustering by seeding. In *Proceedings of the Nineteenth International Conference on Machine Learning* (2002), pp. 19–26.

[6] Basu, S., Bilenko, M., and Mooney, R. J. A probabilistic framework for semi-supervised clustering. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2004), pp. 59–68.

[7] Belkin, M., and Niyogi, P. Semi-supervised learning on manifolds. Tech. Rep. TR-2001-12, University of Chicago, 2002.

[8] Bie, T., Momma, M., and Cristianini, N. Efficiently learning the metric using side information. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)* (2003), pp. 175–189.

[9] Bilenko, M., Basu, S., and Mooney, R. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the Twenty-first International Conference on Machine Learning* (2004), C. Brodley, Ed., ACM, pp. 11–18.

[10] Bilenko, M., and Mooney, R. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)* (2003), pp. 39–48.

[11] Bishop, C. M. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[12] Boise, L., Morgan, D., Kaye, J., and Camicioli, R. Delays in the diagnosis of dementia: Perspectives of family caregivers. *American Journal of Alzheimer's Disease and Other Dementias 14* (1999), 20–26.

[13] Bouman, C., and Shapiro, M. A multiscale random field model for Bayesian image segmentation. *IEEE Trans. Image Processing 3* (Mar. 1994), 162–177.

[14] Burges, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery 2*, 2 (1998), 121–167.

[15] Camicioli, R., Howieson, D., Oken, B., Sexton, G., and Kaye, J. Motor slowing precedes cognitive impairment in the oldest old. *Neurology 50* (1998), 1496–1498.

[16] Chapelle, O., Weston, J., and Schölkopf, B. Cluster Kernels for Semi-Supervised Learning. In *Advances in NIPS* (2003), vol. 15, pp. 585–592.

[17] Chen, M.-H., Shao, Q.-M., and Ibrahim, J. G. *Monte Carlo Methods in Bayesian Computation.* Springer, New York, 2000.

[18] Chen, P., Ratcliff, G., Phil, D., Belle, S., Cauley, J., Kosky, S. D., and Ganguli, M. Cognitive tests that best discriminate between presymptomatic and those who remain nondemented. *Neurology 55* (2000), 1847–1853.

[19] Cohn, D., Caruana, R., and McCallum, A. Semi-supervised Clustering with User Feedback. Tech. Rep. TR2003-1892, Cornell University, 2003.

[20] Cristianini, N., Elisseff, J. S.-T. A., and Kandola, J. On kernel-target alignment. In *Advances in NIPS* (2001), pp. 367–373.

[21] Davidson, I., and Ravi, S. Hierarchical clustering with constraints: Theory and practice. In *In Proceedings of the Nineth European Principles and Practice of KDD (PKDD)* (2005), pp. 59–70.

[22] Davis, J., Kulis, B., Jain, P., Sra, S., and Dhillon, I. Information-theoretic metric learning. In *The 24th Annual International Conference on Machine Learning (ICML)* (2006), pp. 209–216.

[23] Demidenko, E. *Mixed Models, Theory and Applications*. John Wiley and Sons, Inc., Hoboken, NJ, 2004.

[24] Dempster, A., Laird, N., and Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. B 39* (1977), 1–39.

[25] Druck, G., Pal, C., Zhu, X., and McCallum, A. Semi-supervised classification with hybrid generative/discriminative methods. In *Conference on Knowledge Discovery and Data Mining (KDD)* (2007), pp. 280–289.

[26] Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification*. John Wiley and Sons, Inc., Hoboken, NJ, 2001.

[27] Elias, M., Beiser, A., Wolf, P., Au, R., White, R., and DAgostino, R. The preclinical phase of alzheimer disease. *Archives of Neurology 57* (2000), 1847–1853.

[28] Eriksson, A., Olsson, C., and Kahl, F. Normalized cuts revisited: A reformulation for segmentation with linear grouping constraints. In *ICCV* (2007), pp. 1–8.

[29] Frigyik, B., Srivastava, S., and M, G. Functional bregman divergence and bayesian estimation of distributions, 2006. arXiv:cs/0611123.

[30] Goldman, W. P., Baty, J. D., Buckles, V. D., Sahrmann, S., and Morris, J. C. Motor dysfunction in mildly demented ad individuals without extrapyramidal signs. *Neurology 53* (1999), 956–962.

[31] Green, M., Kaye, J., and Ball, M. The oregon brain aging study: Neuropathology accompanying healthy aging in the oldest old. neurology. *Neurology 54*, 1 (2000), 105–113.

[32] Guo, W. Functional mixed effects models. *Biometrics 58* (2002), 121–128.

[33] Gwyther, L. Family issues in dementia: Finding a new normal. *Neurologic Clinics 18* (2000), 993–1010.

[34] Hastie, T., Rosset, S., Tibshirani, R., and Zhu, J. The entire regularization path for the support vector machine. *Journal of Machine Learning Research 5* (2005), 1391–1415.

[35] Holtzer, R., Verghese, J., Xue, X., and Lipton, R. B. Cogntive processes related to gait velocity: Results from the einstein aging study. *Neuropsychology 20* (2006), 215–223.

[36] Holub, A., Welling, M., and Perona, P. Combining generative models and fisher kernels for object recognition. In *International Conference on Computer Vison* (2005), pp. 136–143.

[37] Howieson, D., Dame, A., Camicioli, R., Sexton, G., Payami, H., and Kaye, J. Cognitive markers preceding palzheimers dementia in the healthy oldest old. *J. Am. Geriatr. Soc. 45* (1997), 584–589.

[38] Jaakkola, T. Tutorial on variational approximation methods. In *Proceedings of the Twenty-first International Conference on Machine Learning* (2004), C. Brodley, Ed., ACM, pp. 11–18.

[39] Jaakkola, T., Diekhaus, M., and Haussler, D. Using the fisher kernel method to detect remote protein homologies. *7th Intell. Sys. Mol. Biol.* (1999), 149–158.

[40] Jaakkola, T., and Haussler, D. Exploiting generative models in discriminative classifiers. Tech. rep., Dept. of Computer Science, Univ. of California, 1998.

[41] Jaakkola, T., and Haussler, D. Probabilistic kernel regression models. In *AISTATS* (1999).

[42] Jaakkola, T., Meila, M., and Jebara, T. Maximum entropy discrimination. Tech. Rep. AITR-1668, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1999.

[43] Jain, A., and Dubes, R. *Algorithms for Clustering Data.* Prentice Hall, NJ, 1988.

[44] Jain, K., and Vazirani, V. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of ACM 48* (2001), 274–296.

[45] Kamvar, S., Klein, D., and Manning, C. Spectral Learning. In *IJCAI* (2003), pp. 561–566.

[46] Klein, D., Kamvar, S., and Manning, C. From instance Level to space-level constraints: making the most of prior knowledge in data clustering. In *Proceedings of the Nineteenth International Conference on Machine Learning* (2002), pp. 307–313.

[47] Klein, J., and Moeschberger, M. *Survival Analysis.* Springer, New York, 2003.

[48] Laird, N. M., and Ware, J. H. Random-effects models for longitudinal data. *Biometrics 38*, 4 (1982), 963–974.

[49] Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L., and Jordan, M. Learning the kernel with semidefinite Programming. *Journal of Machine Learning Research 5* (2004), 27–72.

[50] Lange, T., Law, M., Jain, A., and Buhmann, J. Learning with constrained and unlabelled data. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005), pp. 730–737.

[51] Law, M., Topchy, A., and Jain, A. Clustering with soft and group constraints. In *Joint IAPR International Workshop on Syntactical and Structural Pattern Recognition and Statistical Pattern Recognition* (2004), pp. 662–670.

[52] Law, M., Topchy, A., and Jain, A. Model-based clustering with probabilistic constraints. In *Proceedings of SIAM Data Mining* (2005), pp. 641–645.

[53] Lu, Z., and Leen, T. Semi-supervised learning with penalized probabilistic clustering. In *Advances in NIPS* (2005), vol. 17, pp. 849–856.

[54] Lu, Z., and Leen, T. Penalized probabilistic clustering. *Neural Computation 19*, 6 (2007), 1528–1567.

[55] Lu, Z., and Leen, T. Semi-supervised clustering with pairwise constraints: A discriminative approach. In *Eleventh International Conference on Artificial Intelligence and Statistics*. 2007.

[56] Lu, Z., and Leen, T. K. Good prior from population maximum likelihood. unpublished, 2007.

[57] M. Richards, Y. S., and Mayeux, R. Subtle extapyramidal signs can predict the development of dementia in elderly individuals. *Neurology 43* (1993), 2184–2188.

[58] Marquis, S., Moore, M., Howieson, D. B., Sexton, G., Payami, H., Kaye, J. A., and Camicioli, R. Independent predictors of cognitive decline in healthy elderly persons. *Arch. Neurol. 59* (2002), 601–606.

[59] Mitchell, T. *Machine Learning.* McGraw Hill, Columbus, OH, 1997.

[60] Moody, J. Prediction risk and architecture selection for neural networks. In *From Statistics to Neural Networks: Theory and Pattern Recognition Applications.* Springer, NATO ASI Series F, 1994, pp. 1335–1350.

[61] Moreno, P., and Rifkin, R. Using the fisher kernel method for web audio classification. In *ICASSP* (2000), pp. 2417–2420.

[62] Neal, R. Probabilistic inference using Markov Chain Monte Carlo methods. Tech. Rep. CRG-TR-93-1, Computer Science Department, Toronto University, 1993.

[63] Ng, A., and Jordan, M. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing System* (2002), pp. 287–296.

[64] Pepe, M. S. *The Statistical Evaluation of Medical Tests for Classification and Prediction*. Oxford University Press, Oxford, 2003.

[65] Rasmussen, C., and Williams, C. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[66] Riefler, V., and Larson, E. Excess disability in demented elderly outpatients: The rule of halves. *Journal of the American Geriatrics Society 47* (1988), 1065–1072.

[67] Robinson, G. That BLUP is a good thing: The estimation of random effects. *Statistical Science 6* (1991), 15–51.

[68] Seeger, M. Covariance Kernels from Bayesian Generative Models. In *Neural Information Processing Systems 14* (2002), pp. 905–912.

[69] Shental, N., Bar-Hillel, A., Hertz, T., and Weinshall, D. Computing Gaussian Mixture Models with EM using Side-Information. In *ICML workshop on the Continuum from labeled to unlabeled data in machine learning and data mining* (2003).

[70] Shental, N., Bar-Hillel, A., Hertz, T., and Weinshall, D. Computing Gaussian mixture models with EM using equivalence constraints. In *Advances in NIPS*, L. Saul, Y. Weiss, and L. Bottou, Eds., vol. 16. 2004, pp. 505–512.

[71] Sill, J., and Abu-Mostafa, Y. Monotonicity Hints. In *Advances in NIPS*, D. Touretzky, M. Mozer, and M. Hasselmo, Eds., vol. 8. Cambridge, MA: MIT Press, 1996, pp. 634–640.

[72] Sindhwani, V., Niyogi, P., and Belkin, M. Beyond Cloud Point Cloud from Transductive to Semi-supervised Learning. In *ICML* (2005), pp. 824–831.

[73] Smola, A., and Kondor, R. Kernels and regularization on graphs. In *Conference on Learning Theory, COLT/KW* (2003), pp. 144–158.

[74] Srivastava, A., and Stroeve, J. Onboard Detection of Snow, Ice and Other Geophysical Processes Using Kernel Methods. In *ICML 2003 Workshop on Machine Learning Technologies for Autonomous Space Sciences* (2003).

[75] Szummer, M., and Jaakkola, T. Partially labeled classification with Markov random walks. In *Advances in NIPS*, vol. 14. pp. 145–152.

[76] Theiler, J., and Gisler, G. A contiguity-enhanced K-means clustering algorithm for unsupervised multispectral image segementation. In *Proceedings of SPIE* (1997), vol. 3159, pp. 108–118.

[77] Tsuda, K., Kin, T., and Asai, K. Marginalized kernels for biological sequences. *Bioinformatics 1*, 1 (2002), 1–8.

[78] Vapnik, V. *Statistical Learning Theory.* Wiley, New York, 1998.

[79] Verghese, J., Lipton, R., Hall, C., Kuslansky, G., Katz, M., and Buschke, H. Abnormality of gait as a predictor of non-alzheimers dementia. *N. Engl. J. Med. 347*, 22 (2002), 1761–1768.

[80] Vishwanathana, S., Borgwardtc, K. M., Guttmana, O., and Smolaa, A. Kernel extrapolation. *Neurocomputing 69* (2006), 721–729.

[81] Wagstaff, K. *Intellgient clustering with instance-level constraints.* PhD thesis, Cornell University, 2002.

[82] Wagstaff, K., Cardie, C., Rogers, S., and Schroedl, S. Constrained K-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning* (2001), pp. 577–584.

[83] Williams, C., and Barber, D. Bayesian classification with Gaussian processes. *IEEE Transaction on Pattern Analysis and Machine Intelligence 20* (1998), 1342–1351.

[84] Wilson, R. S., Schneider, J. A., Bienias, J. L., Evans, D. A., and Bennett, D. A. Parkinsonianlike signs and risk of incident alzheimer disease in older persons. *Arch. Neurol. 60* (2003), 539–544.

[85] Xing, E., Ng, A., Jordan, M., and Russe, S. Distance metric learning with applications to clustering with side information. In *Advances in NIPS* (2003), vol. 15, pp. 505–512.

[86] Yan, R., Zhang, J., Yang, J., and Hauptmann, A. G. A discriminative learning framework with pairwise constraints for video object classification. *IEEE Trans. Pattern Anal. Mach. Intell. 28*, 4 (2006), 578–593.

[87] Yarowsky, D. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics* (1995), pp. 189–196.

[88] Yu, X., and Shi, J. Grouping with bias. In *Advances in NIPS* (2001), vol. 13, pp. 283–290.

[89] Zhao, Q., and Miller, D. Mixture modeling with pairwise, instance-level class constraints. *Neural Computation 17* (2005), 2482–2507.

[90] Zhou, D., and Schölkopf, B. Learning from labeled and unlabeled data using random walks. In *26th annual pattern recognition symposium DAGM* (2004), pp. 237–244.

[91] Zhu, X., Kandola, J., Ghahramani, Z., and Lafferty, J. Nonparametric transforms of graph kernels for semi-supervised learning. In *Advances in NIPS* (2005), vol. 17, pp. 255–262.

[92] Zhu, X., Lafferty, J., and Zoubin. Semi-supervised learning: from Gaussian field to Gaussian processes. Tech. Rep. CMU-CS-03-175, Computer Science Department, CMU, 2003.

Appendix A

Deriving W from Prior Knowledge

In this appendix, we show how to derive weight $\mathbf{W}$ from the certainty value γ_{ij} for each pair $(\mathbf{x}_i, \mathbf{x}_j)$. Let E denote those original (noise-free) labeled pairwise relations and $\tilde{E}$ the noisy version delivered to us. If we know the original pairwise relations E, we only have to consider the cluster assignments that are consistent with E and neglect the others, that is, the prior probability of Z is

$$P(Z|\Theta, E) \quad = \quad \begin{cases} \frac{1}{\Omega_E} P(Z|\Theta) & Z \text{ is consistent with } E \\ 0 & \text{otherwise,} \end{cases}$$

where Ω_E is the normalization constant for E: $\Omega_E = \sum_{Z:\text{ consistent with } E} P(Z|\Theta)$. Since we know $\tilde{E}$ and the associated certainty values $\Gamma = \{\gamma_{ij}\}$, we know

$$\begin{aligned} P(Z|\Theta, \tilde{E}, \Gamma) &= \sum_{E} P(Z|\Theta, E, \tilde{E}, \Gamma) P(E|\tilde{E}, \Gamma) && \text{(A.1)} \\ &= \sum_{E} P(Z|\Theta, E) P(E|\tilde{E}, \Gamma). && \text{(A.2)} \end{aligned}$$

Let $E(Z) \equiv$ the unique E that is consistent with Z, from equation (A.2) we know

$$\begin{aligned} P(Z|\Theta, \tilde{E}, \Gamma) &= P(Z|\Theta, E(Z)) P(E(Z)|\tilde{E}, \Gamma) \\ &= \frac{1}{\Omega_E} P(Z|\Theta) P(E(Z)|\tilde{E}, \Gamma) = \frac{1}{\Omega_E} P(E(Z)|\tilde{E}, \Gamma) P(Z|\Theta). \end{aligned}$$

If we ignore the variation of Ω_E over E, we can get an approximation of $P(Z|\Theta, \tilde{E}, \Gamma)$, denoted as $P_a(Z|\Theta, \tilde{E}, \Gamma)$:

$$\begin{aligned} P_a(Z|\Theta, \tilde{E}, \Gamma) &= \frac{1}{\Omega_a} P(Z|\Theta) P(E(Z)|\tilde{E}, \Gamma) \\ &= \frac{1}{\Omega_a} P(Z|\Theta) \prod_{i<j} \gamma_{ij}^{H_{ij}(\tilde{E}, z_i, z_j)} (1 - \gamma_{ij})^{1 - H_{ij}(\tilde{E}, z_i, z_j)} \end{aligned}$$

where Ω_a is the new normalization constant: $\Omega_a = \sum_Z P(Z|\Theta)P(E(Z)|\tilde{E},\Gamma)$ and

$$H_{ij}(\tilde{E}, z_i, z_j) = \begin{cases} 1 & (z_i, z_j) \text{ is consistent with } \tilde{E} \\ 0 & \text{otherwise} \end{cases}.$$

We argue that $P_a(Z|\Theta,\tilde{E},\Gamma)$ is equal to a PPC prior probability $P(Z|\Theta,\mathbf{W})$ with

$$\mathbf{W}_{ij} = \begin{cases} \frac{1}{2}\log(\frac{\gamma_{ij}}{1-\gamma_{ij}}) & (z_i, z_j) \text{ is specified as must-linked in } \tilde{E} \\ -\frac{1}{2}\log(\frac{\gamma_{ij}}{1-\gamma_{ij}}) & (z_i, z_j) \text{ is specified as cannot-linked in } \tilde{E} \\ 0 & \text{otherwise.} \end{cases} \tag{A.3}$$

This can be easily proven by verifying

$$\frac{P(Z|\Theta,\mathbf{W})}{P_a(Z|\Theta,\tilde{E},\Gamma)} = \frac{\Omega_a}{\Omega_w} \prod_{i<j,\mathbf{W}_{ij}\neq 0} \gamma_{ij}^{\text{sign}(\mathbf{W}_{ij})-1}(1-\gamma_{ij})^{-\text{sign}(\mathbf{W}_{ij})} = \text{constant}.$$

Since both $P_a(Z|\Theta,\tilde{E},\Gamma)$ and $P(Z|\Theta,\mathbf{W})$ are normalized, we know

$$P_a(Z|\Theta,\tilde{E},\Gamma) = P(Z|\Theta,\mathbf{W}).$$

Appendix B

Mixed-effect Models on OBAS Data

Seconds

Linear Regression Model (order = 1)

Normal Group

$$\alpha = \begin{bmatrix} 0.7111 \\ 0.0202 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 0.9714 & -0.0122 \\ -0.0122 & 0.0002 \end{bmatrix} \qquad \sigma = 0.1635$$

Impaired Group

$$\alpha = \begin{bmatrix} -1.0952 \\ 0.0406 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 1.6234 & -0.0182 \\ -0.0182 & 0.0002 \end{bmatrix} \qquad \sigma = 0.1709$$

Quadratic Regression Model (order = 2)

Normal Group

$$\alpha = \begin{bmatrix} 3.8942 \\ -0.0570 \\ 0.0005 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 17.8055 & -0.4408 & 0.0026 \\ -0.4408 & 0.0111 & -6.7 \times 10^{-5} \\ 0.0026 & -6.7 \times 10^{-5} & 4.1 \times 10^{-7} \end{bmatrix} \qquad \sigma = 0.1603$$

Impaired Group

$$\alpha = \begin{bmatrix} 7.4632 \\ -0.1543 \\ 0.00115 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 26.7031 & -0.6809 & 0.0042 \\ -0.6809 & 0.0175 & -1.0 \times 10^{-4} \\ 0.0042 & -1.0 \times 10^{-4} & 6.8 \times 10^{-7} \end{bmatrix} \qquad \sigma = 0.1689$$

Steps

Linear Regression Model (order = 1)

Normal Group

$$\alpha = \begin{bmatrix} 1.2408 \\ 0.0203 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 0.7019 & -0.0083 \\ -0.0083 & 0.0001 \end{bmatrix} \qquad \sigma = 0.1304$$

Impaired Group

$$\alpha = \begin{bmatrix} -0.1618 \\ 0.0358 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 2.8816 & -0.0326 \\ -0.0326 & 3.7 \times 10^{-4} \end{bmatrix} \qquad \sigma = 0.1359$$

Quadratic Regression Model (order = 2)

Normal Group

$$\alpha = \begin{bmatrix} 2.7642 \\ -0.0152 \\ 0.0002 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 18.7414 & -0.4767 & 0.0029 \\ -0.4767 & 0.0122 & -7.6 \times 10^{-5} \\ 0.0029 & -7.6 \times 10^{-5} & 4.7 \times 10^{-7} \end{bmatrix} \qquad \sigma = 0.1274$$

Impaired Group

$$\alpha = \begin{bmatrix} 7.5445 \\ -0.1377 \\ 0.0010 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 50.9682 & -1.2890 & 0.0080 \\ -1.2890 & 0.0326 & -0.0002 \\ 0.0080 & -0.0002 & 1.3 \times 10^{-6} \end{bmatrix} \qquad \sigma = 0.1360$$

TappingD

Linear Regression Model (order = 1)

Normal Group

$$\alpha = \begin{bmatrix} 4.2518 \\ -0.0069 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 0.1980 & -0.0024 \\ -0.0024 & 3.2 \times 10^{-5} \end{bmatrix} \qquad \sigma = 0.1284$$

Impaired Group

$$\alpha = \begin{bmatrix} 4.9858 \\ -0.0158 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 5.7261 & -0.0672 \\ -0.0672 & 0.0008 \end{bmatrix} \qquad \sigma = 0.1626$$

Quadratic Regression Model (order = 2)

Normal Group

$$\alpha = \begin{bmatrix} 1.9844 \\ 0.0496 \\ -0.0003 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 10.4208 & -0.2607 & 0.0016 \\ -0.2607 & 0.0065 & -4.0 \times 10^{-5} \\ 0.0016 & -4.0 \times 10^{-5} & -2.5 \times 10^{-7} \end{bmatrix} \qquad \sigma = 0.1254$$

Impaired Group

$$\alpha = \begin{bmatrix} -1.8806 \\ 0.1446 \\ -0.0009 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 306.3862 & -7.5113 & 0.0458 \\ -7.5113 & 0.1842 & -0.0011 \\ 0.0458 & -0.0011 & 0.0000 \end{bmatrix} \qquad \sigma = 0.1558$$

TappingN

Linear Regression Model (order = 1)

Normal Group

$$\alpha = \begin{bmatrix} 4.1580 \\ -0.0066 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 0.2672 & -0.0030 \\ -0.0030 & 3.5 \times 10^{-5} \end{bmatrix} \qquad \sigma = 0.1012$$

Impaired Group

$$\alpha = \begin{bmatrix} 4.3362 \\ -0.0092 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 1.9736 & -0.02291 \\ -0.02291 & 2.7 \times 10^{-4} \end{bmatrix} \qquad \sigma = 0.1041$$

Quadratic Regression Model (order = 2)

Normal Group

$$\alpha = \begin{bmatrix} 2.9223 \\ 0.0241 \\ -0.0002 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 3.8314 & -0.1005 & 0.0006 \\ -0.1005 & 0.0027 & -1.7 \times 10^{-5} \\ 0.0006 & -1.7 \times 10^{-5} & 1.1 \times 10^{-7} \end{bmatrix} \qquad \sigma = 0.0997$$

Impaired Group

$$\alpha = \begin{bmatrix} 7.5445 \\ -0.1377 \\ 0.0010 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 26.2063 & -0.6838 & 0.0043 \\ -0.6838 & 0.0179 & -0.0001 \\ 0.0043 & -0.0001 & 7.0 \times 10^{-7} \end{bmatrix} \qquad \sigma = 0.1013$$

Delayed Recall

Linear Regression Model (order = 1)

Normal Group

$$\alpha = \begin{bmatrix} 9.8435 \\ -0.0395 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 24.3887 & -0.2972 \\ -0.2972 & 0.0039 \end{bmatrix} \qquad \sigma = 1.2240$$

Impaired Group

$$\alpha = \begin{bmatrix} 28.8741 \\ -0.2693 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 236.8424 & -2.6264 \\ -2.6264 & 0.0295 \end{bmatrix} \qquad \sigma = 1.3725$$

Quadratic Regression Model (order = 2)

Normal Group

$$\alpha = \begin{bmatrix} 23.2888 \\ -0.3616 \\ 0.0019 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 1189.7 & -31.1731 & 0.1965 \\ -31.1731 & 0.8196 & -0.0052 \\ 0.1965 & -0.0052 & 3.3 \times 10^{-5} \end{bmatrix} \qquad \sigma = 1.2151$$

Impaired Group

$$\alpha = \begin{bmatrix} -43.0368 \\ 1.3364 \\ -0.0089 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 12119 & -289.3461 & 1.6981 \\ -289.3461 & 6.9383 & -0.0409 \\ 1.6981 & -0.0409 & 0.0002 \end{bmatrix} \qquad \sigma = 1.3200$$

Logical Memory II

Linear Regression Model (order = 1)

Normal Group

$$\alpha = \begin{bmatrix} -1.6665 \\ 0.1838 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 399.3892 & -4.7566 \\ -4.7566 & 0.0587 \end{bmatrix} \qquad \sigma = 2.5274$$

Impaired Group

$$\alpha = \begin{bmatrix} 30.8248 \\ -0.2394 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 514.0913 & -5.9009 \\ -5.9009 & 0.0700 \end{bmatrix} \qquad \sigma = 2.7569$$

Quadratic Regression Model (order = 2)

Normal Group

$$\alpha = \begin{bmatrix} 6.2120 \\ 0.0362 \\ 0.0006 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 2637.8 & -72.8670 & 0.4700 \\ -72.8670 & 2.0666 & -0.0137 \\ 0.4700 & -0.0137 & 0.0001 \end{bmatrix} \qquad \sigma = 2.5093$$

Impaired Group

$$\alpha = \begin{bmatrix} -130.2617 \\ 3.5094 \\ -0.0218 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 5081.6 & -137.6516 & 0.8867 \\ -137.6516 & 3.7491 & -0.0243 \\ 0.8867 & -0.0243 & 0.0002 \end{bmatrix} \qquad \sigma = 2.6582$$

Appendix C

Restricted Maximum Likelihood (REML) Fitting of Mixed-effect Models

The data likelihood is then

$$p(Y;\mathcal{M}) \quad = \quad \prod_{i=1}^{k}\int p(\alpha|\,\mathcal{M})\int p(\mathbf{y}^i|\,\alpha,\beta^i;\mathbf{t}^i,\sigma)p(\beta^i|\,\mathcal{M})d\beta^i d\alpha,$$

The maximum likelihood solution of $\{\alpha, \mathbf{A}, \mathbf{D}, \sigma\}$ can be found with the Expectation-Maximization algorithm [24] with $\{\beta^1, \beta^2, \cdots, \beta^k\}$ considered as the latent variable:

$$\text{E-step:} \qquad Q(\mathcal{M},\mathcal{M}^g) = E_{\{\beta^i\}}(\log p(\mathbf{Y},\{\beta^i\};\mathbf{T},\mathcal{M})|\mathbf{Y};\mathbf{T},\mathcal{M}^g) \tag{C.1}$$

$$\text{M-step:} \qquad \mathcal{M} = \arg\max_{\mathcal{M}} Q(\mathcal{M},\mathcal{M}^g), \tag{C.2}$$

where $\mathcal{M}^g$ stands for the estimation of the mixed-effect model obtained in previous step and the expectation in the E-step is with respect to the posterior distribution of on $\{\beta^i\}$ when $\mathbf{Y}$ is known and the model parameter is $\mathcal{M}^g$. For the linear mixed-effect model in Equation (7.3), the M-step has a closed form:

$$\bar{\alpha} \quad = \quad (\sum_{i=1}^{k}\Phi(\mathbf{t}^i)^T\Phi(\mathbf{t}^i))^{-1}\sum_{i=1}^{k}\sum_{n=1}^{N^i}(\mathbf{y}_n^i - E(\beta^i|\mathbf{y}^i,;\mathbf{t}^i,\mathcal{M}^g)^T\Phi(\mathbf{t}^i)) \tag{C.3}$$

$$\mathbf{A} \quad = \quad (\sum_{i=1}^{k}\Phi(\mathbf{t}^i)^T\mathbf{W}^i\Phi(\mathbf{t}^i))^{-1} \tag{C.4}$$

$$\mathbf{D} \quad = \quad \mathbf{D}^g + \frac{1}{k}(\sum_{i=1}^{k}\hat{\beta}^i(\hat{\beta}^i)^T - \mathbf{D}^g\sum_{i=1}^{k}\Phi(\mathbf{t}^i)^T\mathbf{P}^i\Phi(\mathbf{t}^i)) \tag{C.5}$$

$$\sigma^2 \quad = \quad (\sigma^g)^2 + \frac{1}{\sum_{i=1}^{k}N^i}(\sum_{i=1}^{k}||\hat{\epsilon}^i||^2 - (\sigma^g)^4 tr(\sum_{i=1}^{k}\mathbf{P}^i)) \tag{C.6}$$

where $\mathbf{W}^i = ((\sigma^g)^2\mathbb{I} + \Phi(\mathbf{t}^i)\mathbf{D}^g\Phi(\mathbf{t}^i)^T)^{-1}$ (as defined in Chapter 6) and

$$\mathbf{P}^i \quad \equiv \quad \mathbf{W}^i(\mathbb{I} - \Phi(\mathbf{t}^i)(\sum_{i=1}^{k}\Phi(\mathbf{t}^i)^T\mathbf{W}^i\Phi(\mathbf{t}^i))^{-1}\Phi(\mathbf{t}^i)^T\mathbf{W}^i).$$

Appendix D

Two Kernels in Design B

The Fisher score is the gradient of the log likelihood

$$\phi_{\widetilde{\Theta}}(z^i, \gamma^i) \equiv \nabla_{\widetilde{\Theta}} \log p(z^i, \gamma^i; \widetilde{\Theta}) = [\frac{\partial \log p}{\partial \pi_0}; \frac{\partial \log p}{\partial \alpha_0}; \frac{\partial \log p}{\partial \mathbf{D}_0}; \frac{\partial \log}{\partial \pi_1}; \frac{\partial \log p}{\partial \alpha_1}; \frac{\partial \log p}{\partial \mathbf{D}_1}]^T.$$

and the Fisher kernel for the joint variable (γ^i, z^i) is defined as

$$K((z^i, \gamma^i), (z^j, \gamma^j)) = (\nabla_{\widetilde{\Theta}} \log p(\gamma^i|\widetilde{\Theta}))^T (\mathbf{I}^{z,\gamma})^{-1} \nabla_{\widetilde{\Theta}} \log p(\gamma^i|\widetilde{\Theta}), \tag{D.1}$$

where $\mathbf{I}^{z,\gamma}$ is the Fisher information matrix. In the equation (D.1), we have for $m = 0, 1$

$$\begin{aligned}
\frac{\partial \log p(z^i, \gamma^i; \widetilde{\Theta})}{\partial \pi_m} &= \delta(z^i, m)\frac{1}{\pi_m} \\
\frac{\partial \log p(z^i, \gamma^i; \widetilde{\Theta})}{\partial \alpha_m} &= \delta(z^i, m)\mathbf{D}_m^{-1}(\alpha_m - \gamma^i) \\
\frac{\partial \log p(z^i, \gamma^i; \widetilde{\Theta})}{\partial \mathbf{D}_m} &= \delta(z^i, m)\{-\frac{1}{2}\mathbf{D}_m^{-1} + \frac{1}{2}\mathbf{D}_m^{-1}(\alpha_m - \gamma^i)(\alpha_m - \gamma^i)^T\mathbf{D}_m^{-1}\}
\end{aligned}$$

Note that

$$\frac{\partial \log p(z^i, \gamma^i; \widetilde{\Theta})}{\partial \alpha_m} = \delta(z^i, m)\frac{\partial \log p(\gamma^i; \Theta_m)}{\partial \alpha_m} \tag{D.2}$$

$$\frac{\partial \log p(z^i, \gamma^i; \widetilde{\Theta})}{\partial \mathbf{D}_m} = \delta(z^i, m)\frac{\partial \log p(\gamma^i; \Theta_m)}{\partial \mathbf{D}_m}. \tag{D.3}$$

Denoting $\Theta_m = \{\alpha_m, \mathbf{D}_m\} \quad m = 0, 1$, the equation (D.2) and equation (D.3) can be summarized as

$$\frac{\partial \log p(z^i, \gamma^i; \widetilde{\Theta})}{\partial \Theta_m} = \delta(z^i, m)\frac{\partial \log p(\gamma^i; \Theta_m)}{\partial \Theta_m}. \tag{D.4}$$

The Fisher information matrix $\mathbf{I}^{z,\gamma}$ is defined as

$$\begin{aligned}
\mathbf{I}^{z,\gamma} &= E_{z^i,\gamma^i}(\phi^T_{\widetilde{\Theta}}(z^i,\gamma^i)\phi_{\widetilde{\Theta}}(z^i,\gamma^i)|\widetilde{\Theta}) & \text{(D.5)}\\
&= \begin{bmatrix}
E(\frac{\partial \log p}{\partial \pi_0}(\frac{\partial \log p}{\partial \pi_0})^T|\widetilde{\Theta}) & E(\frac{\partial \log p}{\partial \pi_0}(\frac{\partial \log p}{\partial \Theta_0})^T|\widetilde{\Theta}) & E(\frac{\partial \log p}{\partial \pi_0}(\frac{\partial \log p}{\partial \pi_1})^T|\widetilde{\Theta}) & E(\frac{\partial \log p}{\partial \pi_0}(\frac{\partial \log p}{\partial \Theta_1})^T|\widetilde{\Theta})\\
E(\frac{\partial \log p}{\partial \Theta_0}(\frac{\partial \log p}{\partial \pi_0})^T|\widetilde{\Theta}) & E(\frac{\partial \log p}{\partial \Theta_0}(\frac{\partial \log p}{\partial \Theta_0})^T|\widetilde{\Theta}) & E(\frac{\partial \log p}{\partial \Theta_0}(\frac{\partial \log p}{\partial \pi_1})^T|\widetilde{\Theta}) & E(\frac{\partial \log p}{\partial \Theta_0}(\frac{\partial \log p}{\partial \Theta_1})^T|\widetilde{\Theta})\\
E(\frac{\partial \log p}{\partial \pi_1}(\frac{\partial \log p}{\partial \pi_0})^T|\widetilde{\Theta}) & E(\frac{\partial \log p}{\partial \pi_1}(\frac{\partial \log p}{\partial \Theta_0})^T|\widetilde{\Theta}) & E(\frac{\partial \log p}{\partial \pi_1}(\frac{\partial \log p}{\partial \pi_1})^T|\widetilde{\Theta}) & E(\frac{\partial \log p}{\partial \pi_1}(\frac{\partial \log p}{\partial \Theta_1})^T|\widetilde{\Theta})\\
E(\frac{\partial \log p}{\partial \Theta_1}(\frac{\partial \log p}{\partial \pi_0})^T|\widetilde{\Theta}) & E(\frac{\partial \log p}{\partial \Theta_1}(\frac{\partial \log p}{\partial \Theta_0})^T|\widetilde{\Theta}) & E(\frac{\partial \log p}{\partial \Theta_1}(\frac{\partial \log p}{\partial \pi_1})^T|\widetilde{\Theta}) & E(\frac{\partial \log p}{\partial \Theta_1}(\frac{\partial \log p}{\partial \Theta_1})^T|\widetilde{\Theta}).
\end{bmatrix} & \text{(D.6)}
\end{aligned}$$

It is straightforward to verify that for $m = 0, 1$

$$\begin{aligned}
E_{z^i,\gamma^i}(\frac{\partial \log p(z^i,\gamma^i;\widetilde{\Theta})}{\partial \pi_m}(\frac{\partial \log p(z^i,\gamma^i;\widetilde{\Theta})}{\partial \pi_m})^T|\widetilde{\Theta}) &= \frac{1}{\pi_m}\\
E_{z^i,\gamma^i}(\frac{\partial \log p(z^i,\gamma^i;\widetilde{\Theta})}{\partial \pi_m}(\frac{\partial \log p(z^i,\gamma^i;\widetilde{\Theta})}{\partial \Theta_m})^T|\widetilde{\Theta}) &= 0\\
E_{z^i,\gamma^i}(\frac{\partial \log p(z^i,\gamma^i;\widetilde{\Theta})}{\partial \pi_m}(\frac{\partial \log p(z^i,\gamma^i;\widetilde{\Theta})}{\partial \pi_{1-m}})^T|\widetilde{\Theta}) &= 0\\
E_{z^i,\gamma^i}(\frac{\partial \log p(z^i,\gamma^i;\widetilde{\Theta})}{\partial \pi_m}(\frac{\partial \log p(z^i,\gamma^i;\widetilde{\Theta})}{\partial \Theta_{1-m}})^T|\widetilde{\Theta}) &= 0\\
E_{z^i,\gamma^i}(\frac{\partial \log p(z^i,\gamma^i;\widetilde{\Theta})}{\partial \Theta_m}(\frac{\partial \log p(z^i,\gamma^i;\widetilde{\Theta})}{\partial \Theta_m})^T|\widetilde{\Theta}) &= \pi_m E_{z^i,\gamma^i}(\frac{\partial \log p(\gamma^i;\Theta_m)}{\partial \Theta_m}(\frac{\partial \log p(\gamma^i;\Theta_m)}{\partial \Theta_m})^T|\Theta_m)
\end{aligned}$$

from which equation (D.6) can be simplified as

$$\mathbf{I}^{z,\gamma} = \begin{bmatrix}
\frac{1}{\pi_0} & 0 & 0 & 0\\
0 & \pi_0 E(\frac{\partial \log p}{\partial \Theta_0}(\frac{\partial \log p}{\partial \Theta_0})^T|\Theta_0) & 0 & 0\\
0 & 0 & \frac{1}{\pi_1} & 0\\
0 & 0 & 0 & \pi_1 E(\frac{\partial \log p}{\partial \Theta_1}(\frac{\partial \log p}{\partial \Theta_1})^T|\Theta_1)
\end{bmatrix}. \qquad \text{(D.7)}$$

It is not hard to see from here that

$$\begin{aligned}
K((z^i,\gamma^i),(z^j,\gamma^j)) &= (\nabla_{\widetilde{\Theta}} \log p(\gamma^i|\widetilde{\Theta}))^T(\mathbf{I}^{z,\gamma})^{-1}\nabla_{\widetilde{\Theta}} \log p(\gamma^i|\widetilde{\Theta})\\
&= \frac{1}{\pi_{z^i}}\delta(z^i,z^j)(1+K_{z^i}(\gamma^i,\gamma^j))\\
&= \frac{1}{\pi_{z^i}}\widetilde{K}((z^i,\gamma^i),(z^j,\gamma^j)) + \frac{1}{\pi_{z^i}}\delta(z^i,z^j).
\end{aligned}$$

Biographical Note

Zhengdong Lu was born in Minggguang, China on February 5, 1977. In 1998, he earned Bachelors of Engineering degree in Computer Science from Xi'an Institute of Post & Telecommunication, Xi'an. He continued on to graduate school, receiving a Masters of Engineering degree in Electronic Engineering in 2001, from Tsinghua University. He joined the Ph.D. program at the OGI School of Science & Engineering in 2002 where he pursued a doctoral degree in Computer Science until he graduated in 2008. During this time of higher academic pursuits, he not only made new friends, explored the Pacific Northwest, he also met his future wife, Haiming Zheng, whom he married in January 2004.

His broad range of interests include (but are not limited to) many aspects of machine learning. Zhengdong Lu is author of the following publications:

- Lu Z., Leen, T. K., Huang, Y., and Erdogmus, D. A Reproducing Kernel Hilbert Space Framework for Pairwise Time Series Distance. In *The 25th International Conference on Machine Learning (ICML)*, 2008

- Lu, Z., and Carreira-Perpiñán, M. Á. Constrained Spectral Clustering through Affinity Propagation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008

- Carreira-Perpiñán, M. Á., and Lu, Z. Dimension Reduction with Unsupervised Regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008

- Lu, Z., Carreira-Perpiñán, M. Á., and Sminchisescu, C. People Tracking with the Laplacian Eigenmaps Latent Variable Model. To appear in *Advances in Neural Information Processing System (NIPS)* 20, 2008.

- Lu, Z., and Leen, T. K. Constrained Clustering: Advances in Algorithms, Theory, and Applications, Chapman & Hall/CRC Press 2008 -Chapter 4: Pairwise Constraints as Prior in Probabilistic Clustering.

- Huang, Y., Erdogmus, D., Lu, Z., Leen, T. K. Detection Mild Cognitive Loss with Continuous Monitoring of Medication Adherence. In *The 33rd International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008.

- Lu, Z. The Laplace Approximation of a Gaussian Process Mixture *Snowbird Learning Workshop,* 2007.

- Lu, Z., and Leen, T. K. Semi-supervised Clustering with Pairwise Constraints: A Discriminative Approach. *The 11th International Conference on Artificial Intelligence and Statistics (AISTATS'2007),* Puerto Rico, 2007.

- Carreira-Perpiñán, M. Á., and Lu, Z. The Laplacian Eigenmaps Latent Variable Model. *The 11th International Conference on Artificial Intelligence and Statistics (AISTATS'2007),* Puerto Rico, 2007.

- Leen, T. K., Lu, Z., Hayes, T., and Kaye, J. Detection of Early Cognitive Loss from Medication Adherence Behavior. *The 2nd International Conference on Technology and Aging*, Toronto, 2007.

- Lu, Z., and Leen, T. K. Penalized Probabilistic Clustering. *Neural Computation*, 19(6), 1528-1567, 2007.

- van der Merwe, R., Leen, T. K., Lu, Z., Frolov, S., and Baptista, A. M. Fast Neural Network Surrogates for Very High Dimensional Physics-based Models in Computational Oceanography, *Neural Networks*, 20, 462-478, 2007.

- Frolov, S., Baptista, A. M., Lu, Z., van der Merwe, R., and Leen T. K. Fast Data Assimilation with Model Surrogates: Application to A Highly Stratified Estuary, *Ocean Modeling*, 2007.

- Lu, Z., van der Merwe, R., Leen T. K., Frolov, S., and Baptista, A. M. Sequential Data Assimilation with Sigma-point Kalman Filter on Low-dimensional Manifold. CMOP Technical report TR-07-001

- Frolov, S., Lu, Z., van der Merwe, R., Leen, T. K., and Baptista, A. M. Fast and Model-independent Data Assimilation of Estuarine Circulation, Using Neural Networks. in Eos Trans. AGU, 87(36), *Ocean Sci. Meet. Suppl.*, Abstract OS26O-06, Honolulu, HI, 2006.

- Lu, Z., and Leen, T. K. Semi-supervised Learning with Penalized Probabilistic Clustering. *Advances in Neural Information Processing System (NIPS)* 17, 849-856, 2005.

- Lu, Z., and Yan, P. Utilizing Oscillator Neural Networks to Realize Multi-resolution Pattern Recognition. *International Conference on Neural Information Processing, 2002*

- Ouyang, K., Yang, K., Jia, W., Gao, J., Yan, P., Lu, Z., and Ouyang, H. A Study of Dynamics of the Rabbit's Olfactory System: A New Approach for Pattern Recognition. *Neural Networks*, 2, 1077-1082, 2001