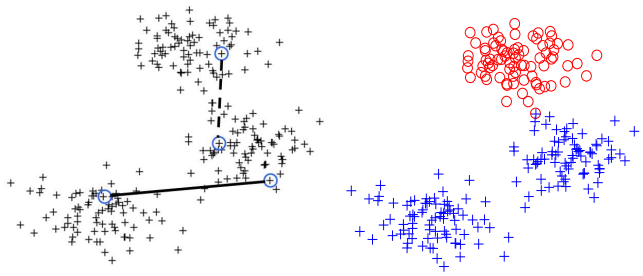
Semi-supervised Clustering with Pairwise Constraints: A Discriminative Approach

Zhengdong Lu **Todd K. Leen**
Department of Computer Science and Engineering
OGI School of Science and Engineering , OHSU
Beaverton, OR 97006
{zhengdon,tleen}@csee.ogi.edu

Abstract

We consider the semi-supervised clustering problem where we know (with varying degree of certainty) that some sample pairs are (or are not) in the same class. Unlike previous efforts in adapting clustering algorithms to incorporate those *pairwise relations*, our work is based on a discriminative model. We generalize the standard Gaussian process classifier (GPC) to express our classification preference. To use the samples not involved in pairwise relations, we employ the graph kernels (covariance matrix) based on the entire data set. Experiments on a variety of data sets show that our algorithm significantly outperforms several state-of-the-art methods.



(a) data and 2 pairwise relations (b) clustering result

Figure 1: Pairwise relations and the clustering on in a 3-component Gaussian mixture. (a): A must-link is specified between samples linked by the solid line, and a cannot-link is denoted as the dashed line. (b): The clustering suggested by the pairwise relations shown in (a).

1 Introduction

There is an emerging interest in semi-supervised clustering algorithms in the machine learning and data mining communities. In addition to the data values, we assume there are a number of instance-level constraints on cluster assignment. More specially, we consider the following two types of *pairwise relations*:

- **Must-link** constraints specify that two samples should be assigned into one cluster.
- **Cannot-link** constraints specify that two samples should be assigned into different clusters.

Figure 1 gives an illustration of pairwise relation constraints and how it affects clustering.

Pairwise relations naturally occur in various domains and applications. In gene classification, our knowledge that two proteins co-occurring in processes can be viewed as a must-link[1]. In information retrieval, the expert critique is often in the form "these two documents shouldn't be in the same cluster", which can be viewed as a cannot-link [2]. Pairwise relations may arise from knowledge of domain experts [3], perceived similarity (or dissimilarity) [4], or even common sense [5]. Unfortunately, those pairwise relations are often determined in a subjective way [2] or with significant uncertainty [4].

Recently several authors have considered using pairwise relations to achieve an intelligent grouping of data [3, 4, 5, 6, 7, 8]. However, prior to this paper, pairwise relations were typically viewed as some kind of side information to a traditional clustering algorithm. Most authors focused on adapting clustering methods, such as K-means [6, 8], or Gaussian mixture models (GMM) [5, 4], to incorporate the pairwise relation constraints. These methods have several inherent drawbacks associated with their basis in clustering (generative) model. Most saliently, they typically need 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 relation, those algorithms will asymptotically degenerate into unsupervised learning (clustering). Another drawback is their limited capability in modeling data distribution within a class. On the other hand, although

discriminative-model based semi-supervised learning algorithms had tremendous success in dealing with partial labeling [9, 10, 11], they are not directly applicable to the pairwise situations.

In this paper, we propose a semi-supervised learning model for pairwise relations loosely based on Gaussian process classifiers (GPC) [12]. We choose the GPC over other discriminative models such as neural networks or the SVM, because it combines two useful properties. First, GPC has an explicit probabilistic interpretation, which facilitates modeling the uncertainty associated with pairwise relations. Second, the covariance matrix (kernel) used in GPC offers a way to use a input-dependent kernel design, and therefore utilize those samples that bear no direct label information.

2 Gaussian Processes for Classification

For simplicity, we consider the binary classification problem. Assume we have data set $\mathbf{X} = \{x_i\}_{i=1}^N$ from two classes with class label $\{+1, -1\}$. The GPC assumes a latent Gaussian process f with zero mean. Let $\mathbf{f} = [f(x_1), f(x_2), ..., f(x_N)]^T$ be the values of f at $\mathbf{X}$, which follows a N-dimensional Gaussian distribution:

$$P(\mathbf{f}) = (2\pi)^{-N/2}|\mathbf{K}|^{-1/2}e^{-\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f}/2} \quad (1)$$

where $\mathbf{K} \in \mathbb{R}^{N\times N}$ is the covariance matrix (kernel). Given the field value at any x_i, the probability that x_i is from class $+1$ is:

$$P(y_i = +1|x_i, f) = \frac{e^{f(x_i)}}{1 + e^{f(x_i)}}, \quad (2)$$

with y_i the class index of x_i. Let $\mathbf{Y} = \{y_i\}_{i=1}^N$ denote the class indices of samples in $\mathbf{X}$. The likelihood of the class labels Y, given the latent variable $\mathbf{f}$, is

$$P(Y|\mathbf{f}) = \prod_{i=1}^N P(y_i|f(x_i)) = \prod_{i=1}^N \frac{e^{f(x_i)\delta(y_i,+1)}}{1 + e^{f(x_i)}}. \quad (3)$$

Our efforts to harness pairwise relations consists of two parts. First, in Section 3, we show that pairwise relations can be treated as observations, and the corresponding likelihood of $\mathbf{f}$ can be given through manipulating Equation (2) and (3). Second, in Section 4, 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 (1). In Section 5, we propose the Semi-supervised Pairwise Gaussian Process Classifier (SPGP) by combining our work on the prior (Section 3) and on the likelihood (Section 4).

3 The Likelihood of Pairwise Relations

3.1 The Formula of Likelihood

In a semi-supervised scenario, we have incomplete knowledge about the class label of samples: it can be a labeled subset [13] (called partial labeling) or some pairwise relations [3] such as discussed in Section 1. Consequently, instead of one, we have a *set* of class assignments $\mathbf{Y}$ consistent with our knowledge. Taking each $\mathbf{Y}$ as an atomic event, our knowledge can be equivalently expressed as a union of all feasible events $\mathbf{Y}$, denoted as Ω. For pairwise relations, we have

$$\Omega = \{\mathbf{Y}|(y_i = y_j, \forall\ (i,j) \in \mathcal{M}) \wedge (y_i \neq y_j, \forall\ (i,j) \in \mathcal{C}\}, \quad (4)$$

where $\mathcal{M}$ and $\mathcal{C}$ are respectively the set of must-links and cannot-links. For any Ω, the likelihood of the latent field $\mathbf{f}$ would be the possibility that Ω happens given $\mathbf{f}$:

$$P(\Omega|\mathbf{f}) = \sum_{\mathbf{Y}\in\Omega} P(\mathbf{Y}|\mathbf{f}) = \sum_{\mathbf{Y}} \{P(\Omega|\mathbf{Y}) \prod_{i=1}^N \frac{e^{f(x_i)\delta(y_i,+1)}}{1 + e^{f(x_i)}}\}, \quad (5)$$

where $P(\Omega|\mathbf{Y}) = 1$ if $\mathbf{Y} \in \Omega$ and $P(\Omega|\mathbf{Y}) = 0$ otherwise.

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

$$P(\mathbf{Y}|\Omega) = \frac{1}{Z_1} e^{\sum_{i<j} w_{ij}\delta(y_i,y_j)} = \frac{1}{Z_1} \prod_{i<j} e^{w_{ij}\delta(y_i,y_j)}, \quad (6)$$

where w_{ij} is the weight for pair (x_i, x_j) and Z_1 is the partition function. We use w_{ij} to express both the type of pairwise relations between (x_i, x_j) and its confidence value γ_{ij} (> 0.5) through

$$\frac{e^{w_{ij}}}{1 + e^{w_{ij}}} = \gamma_{ij}^{L_{ij}}(1 - \gamma_{ij})^{1-L_{ij}}, \quad (7)$$

where $L_{ij} = 1$ if (x_i, x_j) is specified to be must-linked, and $L_{ij} = 0$ for a cannot-link. It follows from Equation (7) that $w_{ij} > 0$ for a must-link between (x_i, x_j), and $w_{ij} < 0$ for a cannot-link. We set $w_{ij} = 0$ if no prior knowledge is available on pair (x_i, x_j). Clearly, $|w_{ij}|$ reflects our confidence since $\frac{e^{|w_{ij}|}}{1+e^{|w_{ij}|}} = \gamma_{ij}$. Using the Bayesian rule, we can get $P(\Omega|\mathbf{Y})$ as follows

$$P(\Omega|\mathbf{Y}) = \frac{P(\mathbf{Y}|\Omega)P(\Omega)}{P(\mathbf{Y})} = \frac{1}{Z_2} \prod_{i<j} e^{w_{ij}\delta(y_i,y_j)}. \quad (8)$$

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 [1]. In Section 3.3 we will show that Z_2 will not affect the final result. From Equation (8), $P(\Omega|\mathbf{Y})$ is larger if $\mathbf{Y}$ satisfies the specified pairwise relations (and vice versa). When $|w_{ij}| \to \infty$, we have $P(\Omega|\mathbf{Y}) = 0$ if (y_i, y_j) violates the specified relation. In this case, we have *hard constraints* between (x_i, x_j); otherwise, the relation is *soft*.

[1] 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 (1) and (3).

When all specified pairwise relations are hard, $P(\Omega|\mathbf{Y})$ degenerates to the extreme case described in Equation (4). Based on Equation (8), the likelihood of $\mathbf{f}$ defined in Equation (5) can be written as:

$$P(\Omega|\mathbf{f}) = \frac{1}{Z_2}\sum_{\mathbf{Y}}\{\prod_{i<j} e^{w_{ij}\ \delta(y_i,y_j)} \cdot \prod_{k=1}^{N} \frac{e^{\delta(y_i,+1)f(x_k)}}{e^{f(x_k)}+1}\}. \tag{9}$$

3.2 Approximation of $P(\Omega|\mathbf{f})$

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

$$P(\Omega|\mathbf{f}) = \frac{1}{Z_2} E_{\mathbf{Y}}\{\prod_{i<j} e^{w_{ij}\delta(y_i,y_j)}\}, \tag{10}$$

where $E_{\mathbf{Y}}\{\cdot\}$ stands for the expectation under distribution $P(\mathbf{Y}|\mathbf{f})$. We get an approximation of $P(\Omega|\mathbf{f})$, denoted $J(\mathbf{f},\Omega)$, by exchanging the order of $\prod$ and $E_{\mathbf{Y}}$ in Equation (10):

$$\begin{aligned} J(\mathbf{f},\Omega) &= \frac{1}{Z_2}\prod_{i<j} E_{\mathbf{Y}}\{e^{w_{ij}\delta(y_i,y_j)}\} \\ &= \frac{1}{Z_2}\prod_{\substack{i<j \\ w_{ij}\neq 0}} \frac{e^{w_{ij}}\{e^{f(x_i)+f(x_j)}+1\} + e^{f(x_i)}+e^{f(x_j)}}{(e^{f(x_i)}+1)(e^{f(x_j)}+1)}. \end{aligned}$$

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. 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

$$\log E_{\mathbf{Y}}\{\prod_{i<j} e^{w_{ij}\delta(y_i,y_j)}\} \geq E_{\mathbf{Y}}\{\log \prod_{i<j} e^{w_{ij}\delta(y_i,y_j)}\}.$$

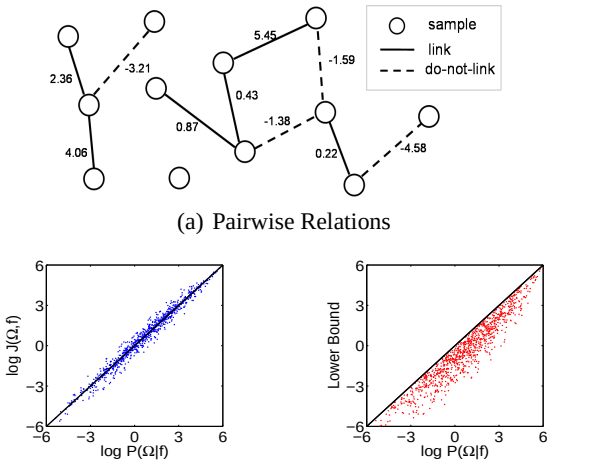
In this case we get a lower bound on $\log P(\Omega|\mathbf{f})$:

$$\log P(\Omega|\mathbf{f}) \geq -\log Z_2 + \sum_{i<j} w_{ij} \frac{e^{f(x_i)+f(x_j)}+1}{(e^{f(x_i)}+1)(e^{f(x_j)}+1)}. \tag{11}$$

Figure 2 compares $\log J(\mathbf{f},\Omega)$ with the lower bound given in Equation (11) on a toy problem. It is clear from Figure 2 (b) and (c) that $\log J(\mathbf{f},\Omega)$ renders a better approximation of $\log P(\Omega|\mathbf{f})$.

3.3 Why Use Maximum a Posteriori (MAP) GPC

The principle Bayesian solution used for standard (supervised) GPC [12] integrates 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:



(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 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 $N(0, 25\mathbf{I}_{12})$. (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 (11) with 1000 random $\mathbf{f}$.

Proposition 1:

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

for $i = 1, 2, \cdots, N$.

The proof of Proposition 1 is simple if one notice that $P(\mathbf{f}|\Omega) = P(-\mathbf{f}|\Omega)$, which can be easily verified using Equation (1) and (9). There are two ways to stay in the standard GPC framework. For a two-class problem, 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 $P(y_i = y_j|\mathbf{X},\Omega)$ for all pair (x_i, x_j), and then use this probability as a 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. In this paper, we instead find the maximum a posteriori (MAP)[2] solution of $\mathbf{f}$ (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:

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}}\{-\log J(\mathbf{f},\Omega) + 1/2\,\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f}\}. \tag{12}$$

We know from the form of $J(\mathbf{f},\Omega)$ that Z_2 only appears in a constant term $\log Z_2$, and therefore will not affect the

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

optimal solution $\hat{\mathbf{f}}$. In Section 4 and 5, we shall show that the optimization in Equation (12) can be simplified. Once $\hat{\mathbf{f}}$ is determined, the classification of $\mathbf{X}$ is carried out with Equation (2).

4 The Prior Probability of Latent Field f

4.1 The Role of the Unconstrained Samples

We divide the data set $\mathbf{X}$ into the constrained set $\mathbf{X}_c = \{x_i| \ \exists j \ \ w_{ij} \neq 0\}$ and unconstrained set $\mathbf{X}_u = \{x_i|\forall j \ \ w_{ij} = 0\}$. We want the unconstrained set to effectively influence the resulting classifier, much the same role played by the unlabeled set in the more familiar partial labeling scenario. Not surprisingly, this intention cannot be realized with a conventional covariance matrix, as we will show presently. Without loss of generality, we assume $\mathbf{X}_c = \{x_1, \cdots, x_{N_c}\}$. Accordingly, we can decompose the field $\mathbf{f}$ as follows:

$$\mathbf{f} = \begin{pmatrix} \mathbf{f}_c \\ \mathbf{f}_u \end{pmatrix}, \tag{13}$$

with $\mathbf{f}_c$ corresponding to the field values at $\mathbf{X}_c$ and $\mathbf{f}_u$ the field values at $\mathbf{X}_u$. 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 that 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 lower-bounded function R can be written as $\hat{\mathbf{f}} = \begin{pmatrix} \hat{\mathbf{f}}_c \\ \hat{\mathbf{f}}_u \end{pmatrix}$, *where*

$$\hat{\mathbf{f}}_c = \arg\min_{\mathbf{f}_c} \{R(\mathbf{f}_c) + \frac{1}{2}\mathbf{f}_c^T\mathbf{K}_c^{-1}\mathbf{f}_c\} \tag{14}$$

$$\hat{\mathbf{f}}_u = \mathbf{K}_{uc}\mathbf{K}_c^{-1}\hat{\mathbf{f}}_c. \tag{15}$$

For a "local" kernel $\mathbf{K}$ [13], e.g. RBF kernel, the entry $\mathbf{K}_{ij}$ only depends on x_i and x_j and *not any other samples*. Proposition 2 tells us that with such a local kernel $\mathbf{K}$, the unconstrained set $\mathbf{X}_u$ is useless for the classification based on Equation (12). Indeed, $\mathbf{X}_u$ does not affect the optimization in Equation (14) (with $R(\mathbf{f}_c)$ set to be $-\log J(\mathbf{f}, \Omega)$), while in Equation (15), $\hat{\mathbf{f}}_u$ is simply interpolated from $\hat{\mathbf{f}}_c$. To overcome this problem, we need a $\mathbf{K}$ with information of $\mathbf{X}_u$ 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.

4.2 Semi-supervised Kernels

Our kernel design strategy largely follows previous work on graph kernel [13, 14]. The key difference is that we fit the kernel to the pairwise relations, instead of some labeled samples as in [10]. Let S be the affinity matrix of $\mathbf{X}$ with $S_{ij} = e^{-||x_i - x_j||^2/s^2}$. The normalized graph Laplacian is defined as $\Delta = \mathbf{I} - D^{-\frac{1}{2}}SD^{-\frac{1}{2}}$, 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 [11] that the eigenvectors $\{\phi_i\}$ provide the harmonic basis with frequency indicated by the eigenvalues $\{\mu_i\}$. Roughly speaking, the higher frequency component has a larger 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 the 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. In this paper, we study the following three types of kernels that have been proposed in literature [9, 10, 15]:

- Step function kernel [3]: $\begin{cases} \lambda & \mu_i \leq \mu_{cut} \\ 0 & \text{otherwise;} \end{cases}$
- Heat Diffusion kernel: $g(\mu_i) = \lambda e^{-t\mu_i}, \ \ t > 0$;
- Lazy-Random-Walk kernel:[4], $g(\mu_i) = \lambda(\mu_i + \sigma^2)^{-1}$.

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 [16], we find $\mathbf{K}$ (or equivalently Θ) that maximizes the alignment score:

$$A(\mathbf{K}, \mathbf{T}) = \frac{<\mathbf{K}, \mathbf{T}>_F}{\sqrt{<\mathbf{K}, \mathbf{K}>_F <\mathbf{T}, \mathbf{T}>_F}}, \tag{16}$$

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,

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

[4] It is also known as Gaussian field kernel [13].

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

$$\bar{A}(\mathbf{K}, \Omega) = \frac{1}{N} < \mathbf{K}, \bar{\mathbf{T}} >_F < \mathbf{K}, \mathbf{K} >_F^{-1/2},$$

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 $P(y_i, y_j|\Omega)$. In this paper, we use a simple approximation for $\bar{\mathbf{T}}_{ij}$ by ignoring the pairwise relations that do not involves x_i or x_j:

$$\bar{\mathbf{T}}_{ij} \approx \sum_{y_i, y_j} y_i y_j \{ \frac{e^{w_{ij}\delta(y_i y_j)}}{1 + e^{w_{ij}}} \prod_{\substack{k:\, w_{ik} \neq 0 \\ w_{jk} \neq 0}} \sum_{y_k} \frac{e^{w_{ik}\delta(y_i, y_k) + w_{jk}\delta(y_j, y_k)}}{(1 + e^{w_{ik}})(1 + e^{w_{jk}})} \}. \quad (17)$$

From Equation (17), the approximation of $\bar{\mathbf{T}}_{ij}$ is non-zero only if $w_{ij} \neq 0$ or both x_i and x_j connected to some sample x_k. Performing the approximation for the 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 in this way since it does not affect the kernel-target alignment score. In our experiment, we use an empirical λ. More systemic methods, like cross validation, are expected to yield better results.

5 Semi-supervised Pairwise Gaussian Process Classifier

We can now combine the likelihood (and its approximation) formulated in Equation (9) and (11), and a Gaussian prior based on the semi-supervised kernel. As mentioned in Section 3.3, the classification is given by the MAP solution of $\mathbf{f}$. According to Proposition 2, the optimization in equation (12) can be divided into the following two steps:

$$\text{step 1: } \hat{\mathbf{f}}_c = \arg\min_{\mathbf{f}_c} \{ \frac{1}{2} \mathbf{f}_c^T \mathbf{K}_c^{-1} \mathbf{f}_c - \sum_{w_{ij} \neq 0} \log \frac{e^{w_{ij}\{e^{f(x_i)+f(x_j)}+1\}+e^{f(x_i)}+e^{f(x_j)}}}{(e^{f(x_i)}+1)(e^{f(x_j)}+1)} \}$$

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

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

$$- \sum_{w_{ij} \neq 0} \log \frac{e^{w_{ij}\{e^{f(x_i)+f(x_j)}+1\}+e^{f(x_i)}+e^{f(x_j)}}}{(e^{f(x_i)}+1)(e^{f(x_j)}+1)},$$

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 $w_{ij} > 0$ (must-link), we tend to have larger $J(\mathbf{f}, \Omega)$ if $f(x_i)$ and $f(x_j)$ are both large (positive) or both small (negative); if $w_{ij} < 0$ (cannot-link), $J(\mathbf{f}, \Omega)$ is larger when one of $f(x_i)$ and $f(x_j)$ is small (negative) and the other is large (positive). The regularizer term enforces a smooth $\mathbf{f}$. Since $\mathbf{K}$ is non-local, $\mathbf{X}_u$ 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).

A visualization of SPGP solution on 2D toy problem can be found in Figure 3. In this toy problem shown in Figure 3 (a), there exist two almost equally good partitions of data into two groups (the upper two components + the lower two components, or the left two components + the right two components). The specified pairwise relations (one must-link + two cannot-links) bias the solution towards the latter one. With a properly designed kernel, SPGP forces the smoothness of $\mathbf{f}$ where the samples are dense, thus the sign of $\mathbf{f}$ can only change in the area where the samples are sparse, as shown in Figure 3 (b). The result given by the MAP solution of $\mathbf{f}$ leads to the second partition, as shown in Figure 3 (c).

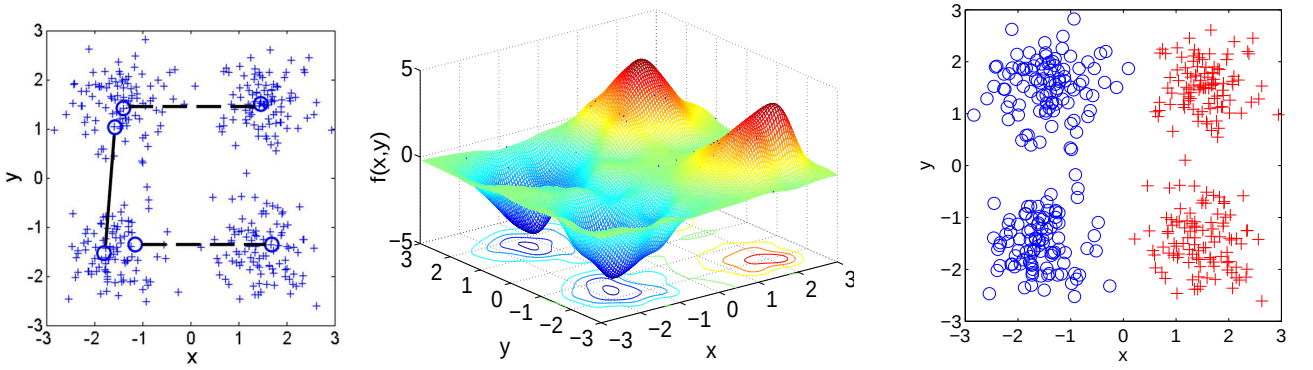
Although in this paper we limited our discussion to two-class cases, SPGP can be readily generalized to M-class ($M > 2$) situations by using M latent processes.

Unlike constrained clustering algorithms [3, 6, 8], SPGP requires at least one cannot-link to work: with *only* must-links, SPGP assigns all samples into one class. This weakness 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 paper.

6 Experiments

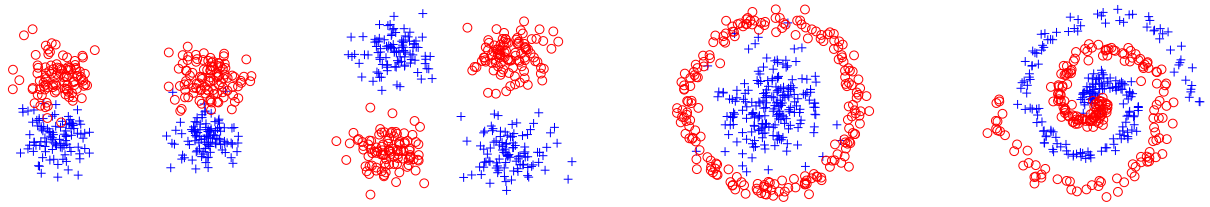
We test SPGP on both artificial data and real-world data, and compare the results with two recently proposed methods: (1) COP-Kmeans [6], a hard-clustering method based on K-means, and (2) Penalized Probabilistic Clustering (PPC) [5][3], a soft-clustering method based on Gaussian

[5]The method in [5] is equivalent to PPC with hard constraints, so the result of it is not included.



(a) data in x-y plane (b) MAP solution of $\mathbf{f}$ (c) clustering result given by the MAP $\mathbf{f}$

Figure 3: SPGP on a two-dimensional toy problem. (a) Two cannot-links (dashed line) and one must-link (solid line) are specified on the data set. (b) A MAP solution of $\mathbf{f}$ as function of coordinates (x, y). Note that discrete values of $\mathbf{f}$ (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 $\mathbf{f}$. (see context in Section 5)



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

Figure 4: Artificial data sets. Classes are denoted by symbols

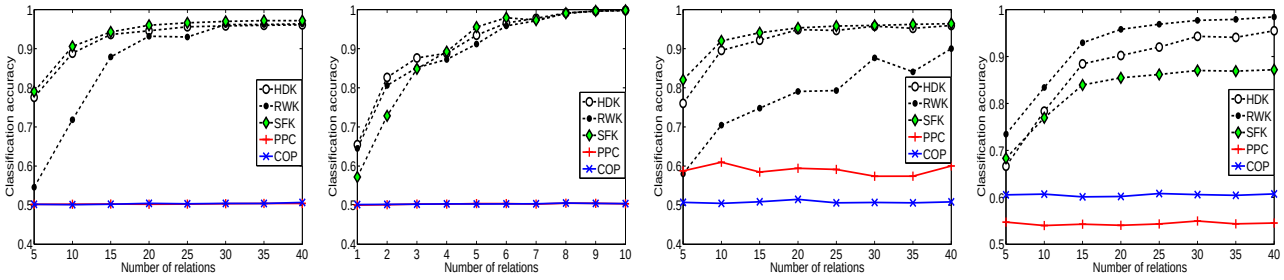
mixture models (GMM). The pairwise relations are randomly generated. The accuracy of each clustering is calculated with a confusion matrix, and we report the classification accuracy averaged over 30 different realizations of pairwise relations.

6.1 Artificial Data (Hard Constraints)

The four 2-dimensional artificial data sets (Figure 4, classes denoted by symbols) are designed to highlight the problems that cannot be effectively solved by centroid-based clustering algorithms. Each data set consists of two classes with 200 samples in each class. We consider the pairwise relations as 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 5 shows classification results for the four data sets with a varying number of pairwise relations. With a small number of pairwise relations, SPGP with all three kernels returns satisfying results, whereas PPC and COP-Kmeans do not respond to them at all.

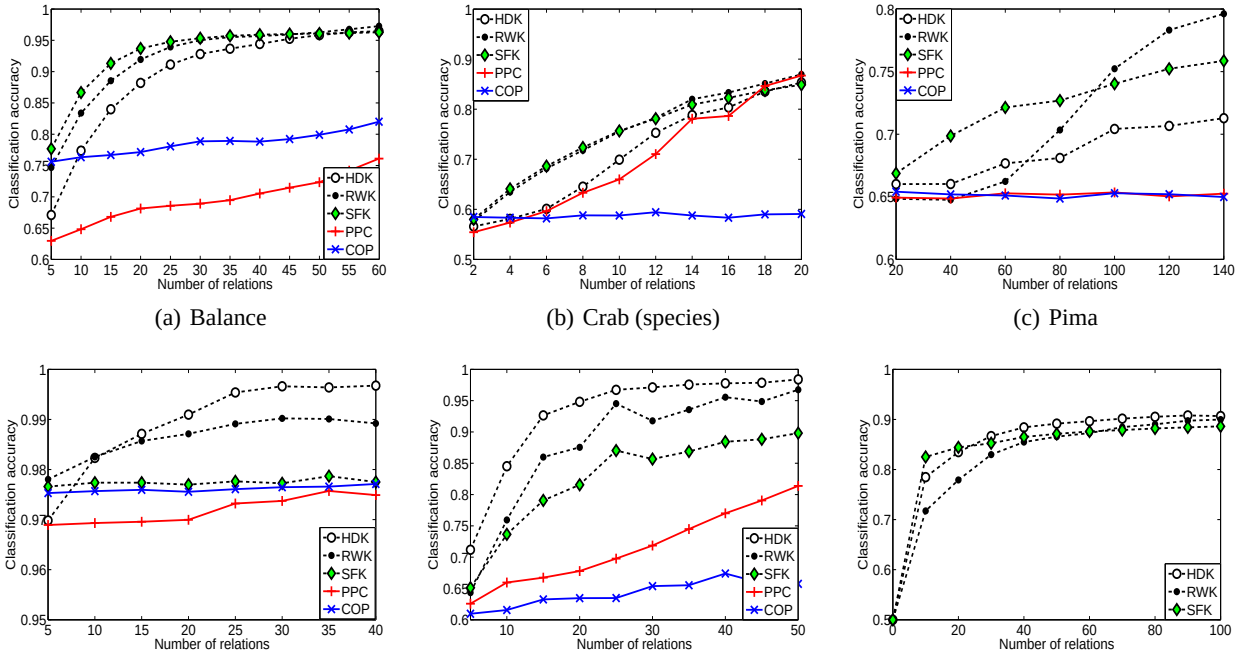
6.2 Real-World Data (Hard Constraints)

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 [12]. `Mac-Windows` has high-dimensional and sparse feature vectors, which makes PPC



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

Figure 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.



(a) Balance (b) Crab (species) (c) Pima

(d) 1-2 (e) Small-Big (f) Mac-Windows

Figure 6: Real-world data. Classification accuracy Vs. number of relations

and COP-Kmeans inapplicable. Therefore on this data we only present SPGP results. Figure 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.

6.3 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), PPC with 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 soft-PPC using Equation (7) 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 6.2. For SPGP, we use the heat-diffusion kernel. In most occasions, soft-SPGP gives the 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 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.

7 Conclusion

We proposed a semi-supervised learning method (SPGP) for pairwise relations based on MAP Gaussian process classifiers. The major contribution of this paper is to give a probabilistic framework, in which classification preference, like pairwise relations, can also be treated as observations. With this probabilistic model, we are able to design discriminative-model based algorithms for semi-supervised clustering with pairwise relation constraints. Also, our model provides a natural way to encode the uncertainty information associated with pairwise relations. Experiments on a variety of data sets show that, compared to traditional constrained clustering methods, our method can achieve decent clustering with significantly fewer pairwise relations.

Acknowledgments

This work was funded by NASA Collaborative Agreement NCC 2-1264 and NSF grant ITR 0121475.

References

[1] H. Wang E. Segal and D. Koller. Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics*, 19:i264–i272, 2003.

[2] D. Cohn, R. Caruana, and A. McCallum. Semi-supervised Clustering with User Feedback. Technical Report TR2003-1892, Cornell University, 2003.

[3] Z. Lu and T. Leen. Semi-supervised learning with penalized probabilistic clustering. In *Advances in NIPS*, volume 17, 2005.

[4] Z. Lu and T. Leen. Penalized probabilistic clustering. *Neural Computation, to appear*.

[5] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing Gaussian mixture models with EM using equivalence constraints. In *Advances in NIPS*, volume 15, 2003.

[6] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained K-means clustering with background knowledge. In *Proceedings of the 18th ICML*, 2001.

[7] D. Klein, S. Kamvar, and C. Manning. From instance level to space-level constraints: making the most of prior knowledge in data clustering. In *Proceedings of the 19th ICML*, 2002.

[8] S. Basu, A. Banerjee, and R. Mooney. Active Semi-Supervision for Pairwise Constrained Clustering. In *Proceedings of the SIAM International Conference on Data Mining*, 2004.

[9] O. Chapelle, J. Weston, and B. Schölkopf. Cluster Kernels for Semi-Supervised Learning. In *Advances in NIPS*, volume 15, 2003.

[10] X. Zhu, J. Kandola, Z. Ghahramani, and J. Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In *Advances in NIPS*, volume 17, 2005.

[11] M. Belkin and P. Niyogi. Semi-supervised learning on manifolds. Technical Report TR-2001-12, University of Chicago, 2002.

[12] C. Williams and David. Barber. Bayesian classification with Gaussian processes. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20:1342–1351, 1998.

[13] X. Zhu, J. Lafferty, and Z. Ghahramani. Semi-supervised learning: from Gaussian field to Gaussian processes. Technical Report CMU-CS-03-175, CMU, 2003.

[14] A. Smola and R. Kondor. Kernels and regularization on graphs. In *Conference on Learning Theory, COLT/KW*, 2003.

[15] D. Zhou and B. Schölkopf. Learning from labeled and unlabeled data using random walks. In *DAGM*, 2004.

[16] N. Cristianini, J. Shawe-Taylorand A. Elisseff, and J. Kandola. On kernel-target alignment. In *Advances in NIPS*, 2001.