Journal of Machine Learning Research 12 (2011) 2649-2680 Submitted 6/09; Revised 5/11; Published 9/11

Bayesian Co-Training

Shipeng Yu SHIPENG.YU@SIEMENS.COM
Balaji Krishnapuram BALAJI.KRISHNAPURAM@SIEMENS.COM
Business Intelligence and Analytics
Siemens Medical Solutions USA, Inc.
51 Valley Stream Parkway
Malvern, PA 19355, USA

Rómer Rosales ROMERR@YAHOO-INC.COM
Yahoo! Labs
4401 Great America Pkwy
Santa Clara, CA 95054, USA

R. Bharat Rao BHARAT.RAO@SIEMENS.COM
Business Intelligence and Analytics
Siemens Medical Solutions USA, Inc.
51 Valley Stream Parkway
Malvern, PA 19355, USA

Editor: Carl Edward Rasmussen

Abstract

Co-training (or more generally, co-regularization) has been a popular algorithm for semi-supervised learning in data with two feature representations (or views), but the fundamental assumptions underlying this type of models are still unclear. In this paper we propose a Bayesian undirected graphical model for co-training, or more generally for semi-supervised multi-view learning. This makes explicit the previously unstated assumptions of a large class of co-training type algorithms, and also clarifies the circumstances under which these assumptions fail. Building upon new insights from this model, we propose an improved method for co-training, which is a novel co-training kernel for Gaussian process classifiers. The resulting approach is convex and avoids local-maxima problems, and it can also automatically estimate how much each view should be trusted to accommodate noisy or unreliable views. The Bayesian co-training approach can also elegantly handle data samples with missing views, that is, some of the views are not available for some data points at learning time. This is further extended to an active sensing framework, in which the missing (sample, view) pairs are actively acquired to improve learning performance. The strength of active sensing model is that one actively sensed (sample, view) pair would improve the joint multi-view classification on all the samples. Experiments on toy data and several real world data sets illustrate the benefits of this approach.

Keywords: co-training, multi-view learning, semi-supervised learning, Gaussian processes, undirected graphical models, active sensing

1. Introduction

In machine learning, data samples may sometimes be characterized in multiple ways. For instance in web page classification, the web pages can be described both in terms of the textual content in each page and the hyperlink structure between them; for cancer diagnosis where the goal is to determine

if the patient has cancer or not, multiple medical imaging techniques (such as CT, Ultrasound and MRI) might be considered to collect complete characteristic of the patient from different perspectives. For learning under such a setting, it has been shown in Dasgupta et al. (2001) that the error rate on unseen test samples can be upper bounded by the disagreement between the classification-decisions obtained from independent characterizations (i.e., *views*) of the data. Thus, in the web page example, *misclassification rate* can be indirectly minimized by reducing the *rate of disagreement* between hyperlink-based and content-based classifiers, provided these characterizations are independent conditional on the class label.

As a completely new learning principle, multi-view consensus learning has been the subject of a large body of research recently. This type of methods were originally developed for semi-supervised learning, where class labels are expensive to obtain but unlabeled data are cheap and abundantly available, such as in web page classification. When the data samples can be characterized in multiple views, the disagreement between the class labels suggested by different views can be computed even when using unlabeled data. Therefore, a natural strategy for using unlabeled data to minimize the misclassification rate is to enforce *consistency* between the classification decisions based on several independent characterizations of the unlabeled samples. For brevity, unless otherwise specified, we shall use the term *co-training* to describe the entire genre of methods that rely upon this intuition, although strictly it should only refer to the original algorithm of Blum and Mitchell (1998).

In this pioneering paper, Blum and Mitchell introduced an iterative, alternating co-training method, which works in a bootstrap mode by repeatedly adding pseudo-labeled unlabeled samples into the pool of labeled samples, retraining the classifiers for each view, and pseudo-labeling additional unlabeled samples where at least one view is confident about its decision. The paper provided PAC-style guarantees that if (a) there exist weakly useful classifiers on each view of the data, and (b) these characterizations of the sample are conditionally independent given the class label, then the co-training algorithm can use the unlabeled data to learn arbitrarily strong classifiers. Later Balcan et al. (2004) tried to reduce the strong theoretical requirements, and they showed that co-training would be useful if (a) there exist low error rate classifiers on each view, (b) these classifiers never make mistakes in classification when they are confident about their decisions, and (c) the two views are not too highly correlated, in the sense that there would be at least some cases where one view makes confident classification decisions while the classifier on the other view does not have much confidence in its own decision. While each of these theoretical guarantees is intriguing and theoretically interesting, they are also rather unrealistic in many application domains. The assumption that classifiers do not make mistakes when they are confident and that of class conditional independence are rarely satisfied in practice. Empirical studies of co-training on many applications show mixed results. See, for instance, Pierce and Cardie (2001) and Kiritchenko and Matwin (2002); Hwa et al. (2003).

A strongly related algorithm is the co-EM algorithm from Nigam and Ghani (2000), which extends the original bootstrap approach of the co-training algorithm to operate simultaneously on all unlabeled samples in an iterative batch mode. Brefeld and Scheffer (2004) used this idea with SVMs as base classifiers, and subsequently in unsupervised learning in Bickel and Scheffer (2005). However, co-EM also suffers from local maxima problems, and while each iteration's optimization step is clear, the co-EM is not really an expectation maximization algorithm (i.e., it lacks a clearly defined overall log-likelihood that monotonically improves across iterations).

In recent years, some co-training algorithms jointly optimize an objective function which includes misclassification penalties (i.e., loss terms) for classifiers from each view, and a regulariza-

tion term that penalizes lack of agreement between the classification decisions of the different views. This *co-regularization* approach has become the dominant strategy for exploiting the intuition behind multi-view consensus learning, rendering obsolete earlier alternating-optimization strategies. Krishnapuram et al. (2004) proposed an approach for two-view consensus learning based on simultaneously learning multiple classifiers by maximizing an objective function which penalized misclassifications by any individual classifier, and included a regularization term that penalized a high level of disagreement between different views. This co-regularization framework improves upon the co-training and co-EM algorithms by maximizing a convex objective function; however the algorithm still depends on an alternating optimization that optimizes one view at a time. This approach was later adapted to two-view spectral clustering in de Sa (2005). The two-view co-regularization approach was subsequently adopted by Sindhwani et al. (2005), Brefeld et al. (2006), Sindhwani and Rosenberg (2008) and Farquhar et al. (2005) for semi-supervised classification and regression based on the reproducing kernel Hilbert space (RKHS). In these approaches a new co-regularization term is added to the objective function which is based on the disagreement of the two views. Representer theorem still holds and solutions can be easily derived by direct optimization. However, it is unclear how to set the regularization parameters (i.e., to control the weight of the co-regularization term). Theoretical analysis of this and other types of algorithms can be found in Balcan and Blum (2006), Sridharan and Kakade (2008), Wang and Zhou (2007) and Wang and Zhou (2010).

Much of these previous work on co-training has been somewhat ad-hoc in nature. Although some algorithms were empirically successful in specific applications, it was not always clear what precise assumptions were made, what was being optimized overall or why they worked well. In this paper we propose a principled undirected graphical model for co-training which we call the *Bayesian co-training*, and show that co-regularization algorithms provide one way for maximum-likelihood (ML) learning under this probabilistic model. By explicitly highlighting previously unstated assumptions, Bayesian co-training provides a deeper understanding of the co-regularization framework, and we are also able to discuss certain fundamental limitations of multi-view consensus learning. Summarizing our algorithmic contributions, we show that co-regularization is exactly equivalent to the use of a novel *co-training kernel* for *support vector machines* (SVMs) and *Gaussian processes* (GP), thus allowing one to leverage the large body of available literature for these algorithms. The kernel is intrinsically *non-stationary*, that is, the level of similarity between any pair of samples depends on *all* the available samples, whether labeled or unlabeled, thus promoting semi-supervised learning. Therefore, this approach is significantly simpler and more efficient than the alternating-optimization that is used in previous co-regularization implementations. Furthermore, we can automatically estimate how much each view should be trusted, and thus accommodate noisy or unreliable views.

The basic idea of Bayesian co-training was published in a short conference paper by Yu et al. (2008). In the current paper we have all the derivation details and more discussions to its related models. More importantly, we extend the Bayesian co-training model to handle data samples with missing views (i.e., some views are missing for certain data samples), and introduce a novel application called the *active sensing*. This makes the current paper significantly different from its conference version.

Active sensing aims to efficiently choose, among all the missing features (grouped in views), what views *and* samples to additionally acquire (or sense) to improve the overall learning performance. This is different from the typical *active learning*, which addresses the problem of efficiently choosing data samples to be labeled in order to improve overall learning performance. From a can-

cer diagnosis perspective, active learning is equivalent to choosing patients to do a biopsy such that the tumor is correctly diagnosed (benign/malignant), whereas active sensing is targeting at collecting (the not-yet-been-collected) medical imaging features (of, e.g., CT, Ultrasound and MRI) from some patients such that all the patients can be better diagnosed. This is important, since a patient does not undergo all possible tests at once (due to various side effects such as radiation and contrast), but these tests are selected based on the evidence collected up to a particular point. This is normally referred to as *differential diagnosis*. Another example is in land mine detection in a sensor network. We may have different types of sensors (as different views) deployed at one location, but some sensors may not be available for all locations due to high cost. So active sensing is to decide which location and which type of sensor we should additionally consider to achieve better detection accuracy. Formulated within the Bayesian co-training framework, two approaches will be discussed for efficiently choosing the (sample, view) pair, based on the mutual information (involving various random variables) and on the predictive uncertainty, respectively.

This active sensing problem is similar to active feature acquisition—see, for example, Melville et al. (2004) and Bilgic and Getoor (2007)—but there is a clear difference. Previous feature acquisition only considers one sample at a time, that is, when one sample is in consideration, the other samples will not be affected. But in active sensing, one actively acquired (sample, view) pair will improve the classification performance of *all* the unlabeled samples via a co-training setting. A related yet different problem was considered in Krause et al. (2008) to identify the optimal spatial locations for placing a single type of sensor to model spatially varying phenomena; however, this work addressed the use of a single type of sensor, and do not consider the scenario of multiple views.

The rest of the paper is organized as follows. We introduce the Bayesian co-training model in Section 2, covering both the undirected graphical model and various marginalizations. Co-training kernel will be discussed in detail to highlight the insight of the approach. The model is extended to handle missing views in Section 4, and this provides the basics for the active sensing solution. The active sensing problem is discussed in Section 5, in which we provide two methods for deciding which incomplete samples should be further characterized, and which sensors should be deployed on them. Experimental results are provided in Section 6, including both some toy problems and real world problems on web page classification and differential diagnosis. We conclude with a brief discussion and future work in Section 7.

2. Bayesian Co-Training

We start from an undirected graphical model for single-view learning with Gaussian processes, and then present Bayesian co-training which is a new undirected graphical model for multi-view learning.

2.1 Single-View Learning with Gaussian Processes

A Gaussian process (GP) defines a nonparametric prior over functions in Bayesian statistics (Rasmussen and Williams, 2006). A random, real-valued function $f : \mathbb{R}^d \to \mathbb{R}$ follows a GP, denoted by $f \sim \mathcal{GP}(h, \kappa)$, if for any finite number of data points $\mathrm{x}_1, \ldots, \mathrm{x}_n \in \mathbb{R}^d$, $\mathrm{f} = \{f(\mathrm{x}_i)\}_{i=1}^n$ follows a multivariate Gaussian distribution $\mathcal{N}(\mathrm{h}, \mathrm{K})$ with mean vector $\mathrm{h} = \{h(\mathrm{x}_i)\}_{i=1}^n$ and covariance matrix defined as $\mathrm{K} = \{\kappa(\mathrm{x}_i, \mathrm{x}_j)\}_{i,j=1}^n$. The functions h and κ are called the mean function and the covariance function, respectively. Conventionally, the mean function is fixed as $h \equiv 0$, and the co-

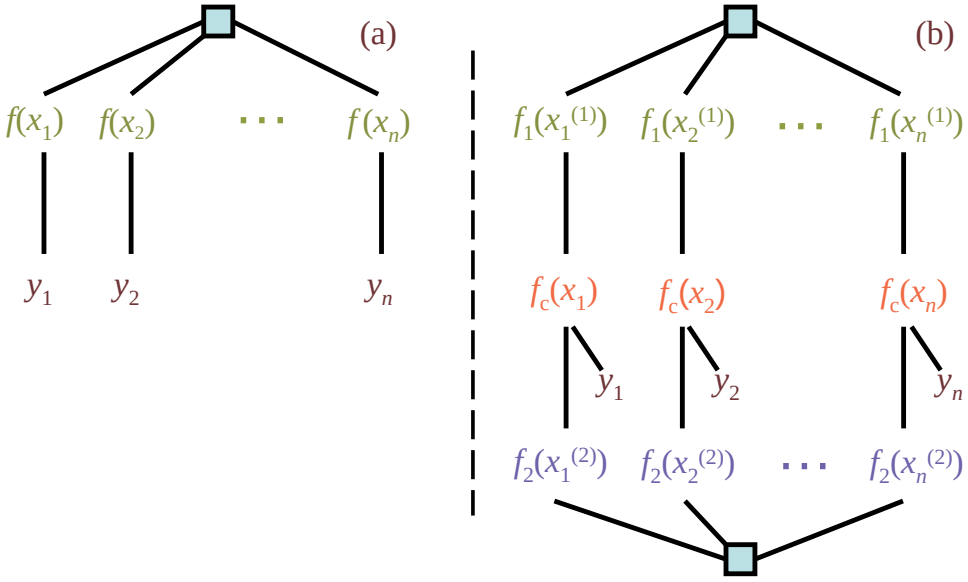


Figure 1: Factor graph for (a) one-view and (b) two-view models.

variance function κ is assumed to take a parametric (and usually stationary) form (e.g., the squared exponential function $\kappa(\mathrm{x}_i,\mathrm{x}_j)=\exp(-\frac{1}{2\rho^2}\|\mathrm{x}_i-\mathrm{x}_j\|^2)$ with $\rho>0$ a *width* parameter).

In a single-view, supervised learning scenario, an output or target y_i is given for each observation x_i (e.g., for regression $y_i\in\mathbb{R}$ and for classification $y_i\in\{-1,+1\}$). In the GP model we assume there is a latent function f underlying the output,

$$p(y_i|\mathrm{x}_i)=\int p(y_i|f,\mathrm{x}_i)\,p(f)\,df=\int p(y_i|f(\mathrm{x}_i))\,p(f)\,df,$$

with the GP prior $p(f)=\mathcal{GP}(h,\kappa)$. Given the latent function f, for regression $p(y_i|f(\mathrm{x}_i))$ takes a Gaussian noise model $\mathcal{N}(y_i|f(\mathrm{x}_i),\sigma^2)$, with $\sigma>0$ a parameter for the noise level; for classification $p(y_i|f(\mathrm{x}_i))$ takes the form of a sigmoid function $\lambda(y_i f(\mathrm{x}_i))$. For instance for GP logistic regression, we have $\lambda(z)=(1+\exp(-z))^{-1}$. See Rasmussen and Williams (2006) for more details on this.

The dependency structure of the single-view GP model can be shown as an undirected graph as in Figure 1(a). The maximal cliques of the graphical model are the fully connected nodes $\{f(\mathrm{x}_1),\ldots,f(\mathrm{x}_n)\}$ and the pairs $\{y_i,f(\mathrm{x}_i)\}$, $i=1,\ldots,n$. Therefore, the joint probability of random variables $\mathrm{f}=\{f(\mathrm{x}_i)\}$ and $\mathrm{y}=\{y_i\}$ is defined as

$$p(\mathrm{f},\mathrm{y})=\frac{1}{Z}\psi(\mathrm{f})\prod_{i=1}^{n}\psi(y_i,f(\mathrm{x}_i)),$$

with potential functions $\psi(\mathrm{f})=\exp(-\frac{1}{2}\mathrm{f}^\top\mathrm{K}^{-1}\mathrm{f})$, and[1]

$$\psi(y_i,f(\mathrm{x}_i))=\begin{cases}\exp(-\frac{1}{2\sigma^2}\|y_i-f(\mathrm{x}_i)\|^2) & \text{for regression,}\\ \lambda(y_i f(\mathrm{x}_i)) & \text{for classification.}\end{cases}\qquad(1)$$

The normalization factor Z hereafter is defined such that the joint probability sums to 1.

1. The definition of ψ in this paper has been overloaded to simplify notation, but its meaning should be clear from the function arguments.

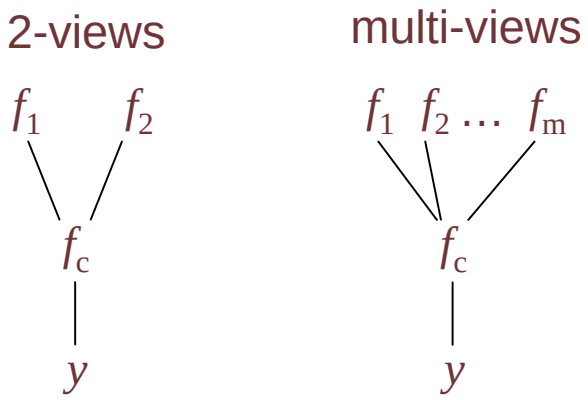


Figure 2: Factor graph in the functional space for 2-view and multi-view learning.

2.2 Undirected Graphical Model for Multi-View Learning

In multi-view learning, suppose we have m different views of a same set of n data samples. Let $\mathrm{x}_i^{(j)} \in \mathbb{R}^{d_j}$ be the features for the ith sample obtained using the jth view, where d_j is the dimensionality of the input space for view j. Note that subscripts index the data sample, and superscripts (with round brackets) index the view. Then the vector $\mathrm{x}_i \triangleq (\mathrm{x}_i^{(1)}, \ldots, \mathrm{x}_i^{(m)})$ is the complete representation of the ith data sample, and $\mathrm{x}^{(j)} \triangleq (\mathrm{x}_1^{(j)}, \ldots, \mathrm{x}_n^{(j)})$ represents all sample observations for the jth view. As in the single-view learning, let $\mathrm{y} = [y_1, \ldots, y_n]^\top$ be the output where y_i is the single output assigned to the ith data point.

One can certainly concatenate the multiple views of the data into a single view, and apply a single-view GP model. But the basic idea of multi-view learning is to introduce *one function per view*, which only uses the features from that specific view to make predictions. Multi-view learning then jointly optimizes these functions such that they come to a consensus. From a GP perspective, let f_j denote the latent function for the jth view (i.e., using features only from view j), and let $f_j \sim \mathcal{GP}(0, \kappa_j)$ be its GP prior in view j with covariance function κ_j. Since one data sample i has only one single label y_i even though it has multiple features from the multiple views (i.e., latent function value $f_j(\mathrm{x}_i^{(j)})$ for view j), the label y_i should depend on *all* of these latent function values for data sample i.

The challenge here is to make this dependency explicit in a graphical model. We tackle this problem by introducing a new latent function, the *consensus function* f_c, to ensure conditional independence between the output y and the m latent functions $\{f_j\}$ for the m views. See Figure 1(b) for the undirected graphical model for multi-view learning. At the functional level, the output y depends *only* on f_c, and latent functions $\{f_j\}$ depend on each other *only via* the consensus function f_c (see Figure 2 for the factor graphs for 2-view and multi-view cases). That is, the joint probability is defined as:

$$p(y, f_c, f_1, \ldots, f_m) = \frac{1}{Z}\psi(y, f_c)\prod_{j=1}^{m}\psi(f_j, f_c), \tag{2}$$

with some potential functions ψ. In the ground network where we have n data samples, let $\mathrm{f}_c = \{f_c(\mathrm{x}_i)\}_{i=1}^n$ and $\mathrm{f}_j = \{f_j(\mathrm{x}_i^{(j)})\}_{i=1}^n$ be the functional values for the consensus view and the jth view,

respectively. The graphical model leads to the following factorization:

$$p(\mathrm{y}, \mathrm{f}_c, \mathrm{f}_1, \ldots, \mathrm{f}_m) = \frac{1}{Z} \prod_{i=1}^{n} \psi(y_i, f_c(\mathrm{x}_i)) \prod_{j=1}^{m} \psi(\mathrm{f}_j) \psi(\mathrm{f}_j, \mathrm{f}_c). \tag{3}$$

Here the *within-view potential* $\psi(\mathrm{f}_j)$ specifies the dependency structure within each view j, and the *consensus potential* $\psi(\mathrm{f}_j, \mathrm{f}_c)$ describes how each latent function f_j is related to the consensus function f_c. With a GP prior for each of the m views, we can define the following potentials:

$$\psi(\mathrm{f}_j) = \exp\left(-\frac{1}{2} \mathrm{f}_j^\top \mathrm{K}_j^{-1} \mathrm{f}_j \right), \quad \psi(\mathrm{f}_j, \mathrm{f}_c) = \exp\left(-\frac{\|\mathrm{f}_j - \mathrm{f}_c\|^2}{2\sigma_j^2} \right), \tag{4}$$

where K_j is the covariance matrix of view j, that is, $\mathrm{K}_j(\mathrm{x}_k, \mathrm{x}_\ell) = \kappa_j(\mathrm{x}_k^{(j)}, \mathrm{x}_\ell^{(j)})$, and $\sigma_j > 0$ is a scalar which quantifies how apart the latent function f_j is from the consensus function f_c. It is seen that the within-view potentials only rely on the *intrinsic structure* of each view, that is, through the covariance matrix in a GP setting. Finally, the *output potential* $\psi(y_i, f_c(\mathrm{x}_i))$ is defined the same as that in (1) for regression or for classification.

The most important potential function in Bayesian co-training is the consensus potential, which simply defines an isotropic multivariate Gaussian for the difference of f_j and f_c, that is, $\mathrm{f}_j - \mathrm{f}_c \sim \mathcal{N}(0, \sigma_j^2 \mathrm{I})$. This can also be interpreted as assuming a conditional isotropic Gaussian for f_j with the consensus f_c being the mean. Alternatively if f_c is of interest, the joint consensus potentials effectively define a conditional Gaussian prior for f_c, $\mathrm{f}_c | \mathrm{f}_1, \ldots, \mathrm{f}_m$, as $\mathcal{N}(\mu_c, \sigma_c^2 \mathrm{I})$ where

$$\mu_c = \sigma_c^2 \sum_j \frac{\mathrm{f}_j}{\sigma_j^2}, \qquad \sigma_c^2 = \left(\sum_j \frac{1}{\sigma_j^2} \right)^{-1}. \tag{5}$$

One can easily verify that this is a product of Gaussian distributions, with each Gaussian being $\mathcal{N}(\mathrm{f}_c | \mathrm{f}_j, \sigma_j^2 \mathrm{I})$.[2] This indicates that, given the latent functions $\{\mathrm{f}_j\}_{j=1}^m$, the posterior mean of the consensus function f_c is a *weighted average* of these latent functions, and the weight is given by the inverse variance (i.e., the precision) of each consensus potential. The higher the variance, the smaller the contribution to the consensus function. In the following we call σ_j^2 the *view variance* for view j. In this paper these view variances are taken as parameters of the Bayesian co-training model, but one can also assign a prior (e.g., a Gamma prior) to them and treat them instead as hidden variables. We will discuss the consensus potential and the view variances in more details in Section 3.

In (3) we assume the output y is available for all the n data samples. More generally we consider *semi-supervised* multi-view learning, in which only a subset of data samples have outputs available. This is actually the setting for which co-training and multi-view learning were originally motivated (Blum and Mitchell, 1998). Formally, let n_l be the number of data samples which have outputs available, and let n_u be the number of data samples which do not. We still keep $n = n_l + n_u$ to be the total number of data samples. Under this setting, we only have outputs available for n_l samples, that is, $\mathrm{y}_l = [y_1, \ldots, y_{n_l}]^\top$.

In the functional space, the undirected graphical model for semi-supervised multi-view learning is the same as in Figure 2. The joint probability is also the same as in (2). In the ground network,

2. Note that this conditional Gaussian for f_c has a normalization factor which depends on $\mathrm{f}_1, \ldots, \mathrm{f}_m$.

since the output vector y_l is only of length n_l, the joint probability is now:

$$p(\mathrm{y}_l, \mathrm{f}_c, \mathrm{f}_1, \ldots, \mathrm{f}_m) = \frac{1}{Z} \prod_{i=1}^{n_l} \psi(y_i, f_c(\mathrm{x}_i)) \prod_{j=1}^{m} \psi(\mathrm{f}_j) \psi(\mathrm{f}_j, \mathrm{f}_c). \tag{6}$$

Note that the product of output potentials contains only that of the n_l labeled data samples, and that $\mathrm{f}_c = \{f_c(\mathrm{x}_i)\}_{i=1}^n$ and $\mathrm{f}_j = \{f_j(\mathrm{x}_i^{(j)})\}_{i=1}^n$ are still of length n. Unlabeled data samples contribute to the joint probability via the within-view potentials $\psi(\mathrm{f}_j)$ and consensus potentials $\psi(\mathrm{f}_j, \mathrm{f}_c)$. All the potentials are defined similarly as in (4). In the following we will mainly discuss this more interesting setting.

3. Inference and Learning in Bayesian Co-Training

In this section we discuss inference and learning in the proposed model, assuming first that there is no missing data in any of the views (the setting with missing data will be discussed in Section 4). Instead of working with the undirected graphical model directly, we show different types of marginalizations under this model. The standard inference task is that of inferring y from the observed data, that is, obtaining $p(\mathrm{y})$; however, in order to gain insight into the proposed model and co-training, we explore different marginalizations. All marginalizations lead to standard Gaussian process inference with different latent function at consideration, but interestingly, these different marginalizations show different insights of the proposed undirected graphical model. One advantage of the marginalizations is that it allows us to see that many existing multi-view learning models are actually special cases of the proposed framework. In addition, this Bayesian interpretation helps us understand both the benefits and the limitations of co-training. For clarity we put the derivations into Appendix A.

3.1 Marginal 1: Co-Regularized Multi-View Learning

Our first marginalization focuses on the joint probability distribution of the m latent functions, when the consensus function f_c is integrated out. This would lead to a GP model in which the latent functions are the view specific functions $\mathrm{f}_1, \ldots, \mathrm{f}_m$. Taking the integral of (3) over f_c (and ignoring the output potential for the moment), we obtain the joint marginal distribution as follows after some mathematics (for derivations see Appendix A.1):

$$p(\mathrm{f}_1, \ldots, \mathrm{f}_m) = \frac{1}{Z} \exp\left\{ -\frac{1}{2} \sum_{j=1}^{m} \mathrm{f}_j^\top \mathrm{K}_j^{-1} \mathrm{f}_j - \frac{1}{2} \sum_{j<k} \left[\frac{\|\mathrm{f}_j - \mathrm{f}_k\|^2}{\sigma_j^2 \sigma_k^2} \Big/ \sum_\ell \frac{1}{\sigma_\ell^2} \right] \right\}. \tag{7}$$

It can be seen that the negation of the logarithm of this marginal recovers the regularization terms in the *co-regularized multi-view learning* (see, e.g., Sindhwani et al., 2005; Brefeld et al., 2006). In particular, we have

$$\begin{aligned} -\log p(\mathrm{f}_1, \ldots, \mathrm{f}_m) &= \frac{1}{2} \sum_{j=1}^{m} \mathrm{f}_j^\top \mathrm{K}_j^{-1} \mathrm{f}_j + \frac{1}{2} \sum_{j<k} \left[\frac{\|\mathrm{f}_j - \mathrm{f}_k\|^2}{\sigma_j^2 \sigma_k^2} \Big/ \sum_\ell \frac{1}{\sigma_\ell^2} \right] + \log Z \\ &= \frac{1}{2} \sum_{j=1}^{m} \Omega_j(\mathrm{f}_j) + \frac{1}{2} \frac{1}{\sum_\ell \frac{1}{\sigma_\ell^2}} \sum_{j<k} L(\mathrm{f}_j, \mathrm{f}_k) + \log Z, \end{aligned}$$

where $\Omega_j(\mathrm{f}_j) \triangleq \mathrm{f}_j^\top \mathrm{K}_j^{-1}\mathrm{f}_j$ regularizes the functional space of each individual view j, and the loss function $L(\mathrm{f}_j,\mathrm{f}_k) \triangleq \|\mathrm{f}_j-\mathrm{f}_k\|^2/\sigma_j^2\sigma_k^2$ measures the disagreement of every pair of the function outputs, inversely weighted by the product of the corresponding variances. The higher the variance σ_j^2 of view j, the less the contribution view j brings to the overall loss. We refer to this as *variance-sensitive co-regularized multi-view learning*. Note that unlike the formulation in Brefeld et al. (2006) where the disagreements are only with respect to the unlabeled data, here we regularize the disagreements of all data samples. From the GP perspective, (7) actually defines a *joint multi-view prior* for the m latent functions, $(\mathrm{f}_1,\ldots,\mathrm{f}_m) \sim \mathcal{N}(0,\Lambda^{-1})$, where Λ is a $mn \times mn$ precision matrix with block-wise definition:

$$\Lambda(j,j) = \mathrm{K}_j^{-1} + \frac{1}{\sum_\ell \frac{1}{\sigma_\ell^2}} \sum_{k\neq j} \frac{1}{\sigma_j^2\sigma_k^2}\mathrm{I}, \quad \Lambda(j,j') = -\frac{1}{\sum_\ell \frac{1}{\sigma_\ell^2}} \frac{1}{\sigma_j^2\sigma_{j'}^2}\mathrm{I}, \quad j' \neq j. \tag{8}$$

It is seen that the block-wise precision matrix for view j has contributions from all the other views.

When we take into account the observed output variable y, we can also easily derive the joint marginal of y with all the latent functions $\mathrm{f}_1,\ldots,\mathrm{f}_m$. For instance for regression, the marginal distribution turns out to be (recall that σ^2 is the variance parameter in the output potential for regression):

$$p(\mathrm{y},\mathrm{f}_1,\ldots,\mathrm{f}_m) = \frac{1}{Z}\exp\left\{ -\frac{1}{2\rho\sigma^2}\sum_j \frac{\sum_{i=1}^n (y_i - f_j(\mathrm{x}_i))^2}{\sigma_j^2} \right.$$
$$\left. -\frac{1}{2}\sum_j \mathrm{f}_j^\top \mathrm{K}_j^{-1}\mathrm{f}_j - \frac{1}{2\rho}\sum_{j<k}\frac{\|\mathrm{f}_j-\mathrm{f}_k\|^2}{\sigma_j^2\sigma_k^2} \right\}. \tag{9}$$

Here $\rho \triangleq \frac{1}{\sigma^2} + \sum_j \frac{1}{\sigma_j^2}$ is the sum of all the inverse variances, including the regression variance. Maximizing this marginal distribution is equivalent to solving a minimization problem in co-regularized multi-view learning with least square loss. It is seen that the least square loss with respect to the jth latent function f_j is inversely weighted by the variance σ_j^2, which indicates again that a higher variance leads to less contribution to the total loss.

3.2 Marginal 2: The Co-Training Kernel

The joint multi-view kernel defined in (8) is interesting, but it has a large dimension and is difficult to work with. A more interesting kernel can be obtained if we instead integrate out all the m latent functions $\mathrm{f}_1,\ldots,\mathrm{f}_m$ in (3). This leads to a standard (transductive) Gaussian process model, with f_c being the latent function realizations, and GP prior being $p(\mathrm{f}_c) = \mathcal{N}(0,\mathrm{K}_c)$ where

$$\mathrm{K}_c = \left[\sum_j (\mathrm{K}_j + \sigma_j^2\mathrm{I})^{-1}\right]^{-1}. \tag{10}$$

See Appendix A.2 for the derivation. This indicates that by marginalization, we can transfer the multi-view problem into a single-view problem with respect to the consensus function f_c, without loss of information. The new kernel matrix K_c is derived via all the m kernels from the m views, and note that each entry (i,j) in K_c depends not only on the features of the corresponding data items x_i and x_j, but also on all the other labeled and unlabeled data points (as seen in (10) through matrix inverse). This is the result of the multi-view dependency in the graphical model in Bayesian

co-training, and it also means that this kernel lacks the marginalization property and can only be used in a transductive setting.

This kernel definition is crucial to Bayesian co-training, and in the following we call K_c the *co-training kernel* for multi-view learning. This marginalization reveals the previously unclear insight of how the kernels from different views are combined together in a multi-view learning framework. This allows us to transform a multi-view learning problem into a single-view problem, and simply use the co-training kernel K_c to solve GP classification or regression. Since this marginalization is equivalent to (7),[3] we end up with solutions that are largely similar to any other co-regularization algorithm, but however a key difference is the Bayesian treatment contrasting previous ML-optimization methods.

Formulation (10) can also be viewed as a *kernel design* for transductive multi-view learning, namely, the inverse of the co-training kernel is the sum of the inverse of all individual kernels, corrected by the view specific variance term. Higher variance leads to less contribution to the overall co-training kernel. In a transductive setting where the data are partially labeled, the co-training kernel between labeled data is also dependent on the unlabeled data. Hence the proposed co-training kernel, by the design in (10), can be used for semi-supervised GP learning (Zhu et al., 2003).

Additional benefits of the co-training kernel include the following:

- With fixed hyperparameters (e.g., σ_j^2), the co-training kernel avoids repeated alternating optimizations with respect to the different views f_j, and directly works with a single consensus view f_c. This reduces both time complexity and space complexity (since we only maintain K_c in memory) of multi-view learning.

- While other alternating optimization algorithms might converge to local minima (because they optimize, not integrate), the single consensus view guarantees the *global optimal inference solution* for multi-view learning since it marginalizes other latent functions and leads to a standard GP inference model.

- Even if all the individual kernels are stationary, K_c is in general *non-stationary*. This is because the inverse-covariances are added and then inverted again.

3.3 Marginal 3: Individual View Learning with Side-Information

In Bayesian co-training model we can also focus on one particular view j by marginalizing all the other views and the consensus view. This is particularly interesting if there is one view that is of the main interest (e.g., it provides the most useful features, or it has the least missing features), and we want to understand how the other views influence this view in the inference process. This can be done by integrating out the other latent functions f_k, $k \neq j$, in (7), and it will lead to another GP formulation with f_j being the latent function. Since (7) represents a jointly Gaussian distribution, we obtain $\mathrm{f}_j \sim \mathcal{N}(0, \mathrm{C}_j)$, where

$$\mathrm{C}_j^{-1} = \mathrm{K}_j^{-1} + \left[\sigma_j^2 \mathrm{I} + \sum_{k \neq j} \left(\mathrm{K}_k + \sigma_k^2 \mathrm{I} \right)^{-1} \right]^{-1}. \tag{11}$$

3. The equivalence is in the sense that both marginalizations are based on the same underlying graphical model, and any optimal solution derived from these marginalizations should be a solution which optimizes the likelihood of the graphical model.

See Appendix A.3 for the derivation. This can be intuitively understood as that the precision matrix of the individual view, C_j^{-1}, is the sum of its original precision matrix and the contributions from other views, weighted by the inverse of the variance. Therefore if σ_k^2 is big for some view k, its contribution to the other views will be compromised. Hence, if one particular view is of interest, we can encode the additional information from the other views into the kernel for the interested view.

Another benefit of this marginalization is the possibility of introducing an inductive inference scheme (rather than transductive as in Section 3.2)—given a new test data x_*, we try to make a prediction of y_* if the jth view $x_*^{(j)}$ is available. Inspired by Yu et al. (2005), let us define $\alpha_j = [\alpha_{j1}, \ldots, \alpha_{jn}]^\top \in \mathbb{R}^n$ such that $f_j(x) = \sum_{i=1}^n \alpha_{ji} \kappa_j(x^{(j)}, x_i^{(j)})$ (this is also motivated by the Representer theorem). On the training data, this yields $f_j = K_j \alpha_j$. From (11) we can see that this re-parameterization leads to a co-training prior for α_j as $\alpha_j \sim \mathcal{N}(0, K_j^{-1} C_j K_j^{-1})$. At testing time when we have the posterior of α_j, y_* can be approximated by $f_j(x_*) = \sum_{i=1}^n \alpha_{ji} \kappa_j(x_*^{(j)}, x_i^{(j)})$. This approach is particularly interesting in the case that one of the views is known to be predictive (i.e., the other views are "side" information to help this primary view), or test data often come with features only in a specific view (since the features from the other views would be disregarded at testing time).

3.4 Optimization of Hyperparameters

One of the advantages of Bayesian co-training is that each view j has a view-specific variance term σ_j^2 to quantify how far the latent function f_j is apart from the consensus view f_c. In particular, a larger value of σ_j^2 implies less confidence on the observation of evidence provided by the jth view. In the perspective of kernel design, this leads to a lesser weight on the kernel K_j. Thus when some views of the data are better at predicting the output than the others, they are weighted more while forming consensus opinions. These variance terms are hyperparameters of the Bayesian co-training model.

To optimize these variance terms together with other hyperparameters involved in each covariance function (e.g., parameter $\rho > 0$ in the Gaussian kernel $\kappa(x_i, x_j) = \exp(-\rho \|x_i - x_j\|^2)$), we can use the *type II maximum likelihood* method (sometimes called evidence approximation), which maximizes the marginal likelihood with respect to each of these hyperparameters. For simplicity we put the derivation and detailed equations in Appendix B. For more details on the type II maximum likelihood in the GP setting, please refer to Rasmussen and Williams (2006).

3.5 Discussions

The proposed undirected graphical model provides better understanding of multi-view learning algorithms. In each of the marginalizations, we end up with a standard GP model for some latent functions (i.e., $\{f_1, \ldots, f_m\}$ in Marginal 1, f_c in Marginal 2, and f_j in Marginal 3). This simplifies learning and inference under the proposed model. Under a transductive setting, the co-training kernel in (10) indicates that *Bayesian co-training is equivalent to single-view learning with a specially designed (non-stationary) kernel*. This is also the preferable way of working with multi-view learning since it avoids alternating optimizations at the inference step.

The proposed graphical model also motivates new methods for unsupervised multi-view learning such as spectral clustering. While the similarity matrix of each view j is encoded in K_j, the

co-training kernel K_c encodes the similarity of two data samples *with multiple views*, and thus can be used directly in spectral clustering.

We would also like to point out the limitations of the proposed consensus-based learning, which are shared by co-training as proposed by Blum and Mitchell (1998) and many other multi-view learning algorithms. As mentioned before, the consensus-based potentials in (4) can be interpreted as defining a Gaussian prior (5) to f_c, where the mean is a *weighted average* of the m individual views. This averaging indicates that the value of f_c is never higher (or lower) than that of any single view. While the consensus-based potentials are intuitive and useful for many applications, they are limited for some real world problems where the evidence from different views should be *additive* (or enhanced) rather than averaging. For instance, when a radiologist is making a diagnostic decision about a lung cancer patient, he or she might look at both the CT image and the MRI image. If either of the two images gives a strong evidence of cancer by that image alone, he or she can make a decision based on a single view (and thus, ignoring the other image completely); if either of the images only gives a moderate evidence (i.e., from a single-view learner which ignores the other image), it would be beneficial to look at both images (i.e., to consider both views), and the final evidence of cancer after observing both images should be higher (or lower, depending on the specific scenario) than either of them if observed individually. It's clear that in this scenario the multiple views are *reinforcing* or *weakening* each other, not averaging. While all the previously proposed co-training and co-regularization algorithms have thus far been based on enforcing consensus between the views explicitly or implicitly, we make this clear from the graphical model perspective, and allow effective tailoring of the view importance from the training data. As part of future work, it would be interesting to explore the possibility of going beyond consensus-based multi-view learning.

4. Bayesian Co-Training with Missing Views

In the previous two sections we assume that the input data are complete, that is, all the views are observed for every data sample. However for many real-world problems, the features could be incomplete or missing for various reasons. For instance, in cancer diagnosis we cannot ask every patient to take all the available imaging tests (e.g., CT, PET, Ultrasound, MRI) for the final diagnosis, so some views (i.e., imaging tests) are missing for certain patients. In this section we extend Bayesian co-training to the case where there are missing (sample, view) pairs in the input data (which can happen both in labeled data and in unlabeled data). The three marginalizations will also be discussed. To the best of our knowledge, this is the first elegant framework to account for the missing views in the multi-view learning setting.

Let each view j be observed for a subset of $n_j \leq n$ samples, and let $\mathbb{I}_j$ denote the indices of these samples in the whole sample set (including labeled and unlabeled data). Note that under this notation, the single-view kernel matrix K_j for view j is of size $n_j \times n_j$, which are defined over the subset of samples denoted by indicator $\mathbb{I}_j$. From the co-training kernel perspective, the difficulty here is to combine the kernels of different sizes together from different views, if at all possible.

We start from the undirected graphical model and make necessary changes to the potentials to account for the missing views. The idea is to treat the missing view information as *hidden* in the graphical model. The undirected graphical model is shown in Figure 3 for Bayesian co-training

with missing views, which is very similar to Figure 1(b). The joint probability can be defined as:

$$p(\mathrm{y}_l, \mathrm{f}_c, \mathrm{f}_1, \ldots, \mathrm{f}_m) = \frac{1}{Z} \prod_{i=1}^{n_l} \psi(y_i, f_c(\mathrm{x}_i)) \prod_{j=1}^{m} \psi(\mathrm{f}_j)\psi(\mathrm{f}_j, \mathrm{f}_c), \tag{12}$$

where $\mathrm{f}_c = \{f_c(\mathrm{x}_i)\}_{i=1}^n \in \mathbb{R}^n$, and $\mathrm{f}_j = \{f_j(\mathrm{x}_i^{(j)})\}_{i \in \mathbb{I}_j} \in \mathbb{R}^{n_j}$. Note that f_j is only realized on a subset of samples and is of length n_j (instead of n). The *within-view potential* $\psi(\mathrm{f}_j)$ is defined via the GP prior, $\psi(\mathrm{f}_j) = \exp(-\frac{1}{2}\mathrm{f}_j^\top \mathrm{K}_j^{-1}\mathrm{f}_j)$, where $\mathrm{K}_j \in \mathbb{R}^{n_j \times n_j}$ is the covariance matrix for view j; the *consensus potential* $\psi(\mathrm{f}_j, \mathrm{f}_c)$ is defined as follows:

$$\psi(\mathrm{f}_j, \mathrm{f}_c) = \exp\left(-\frac{\|\mathrm{f}_j - \mathrm{f}_c(\mathbb{I}_j)\|^2}{2\sigma_j^2}\right), \tag{13}$$

in which $\mathrm{f}_c(\mathbb{I}_j)$ takes the length-n_j subset of vector f_c with indices given in $\mathbb{I}_j$. In other words, the consensus potentials is defined such that

$$\psi(f_j(\mathrm{x}_i), f_c(\mathrm{x}_i)) = \exp\left(-\frac{1}{2\sigma_j^2}\big(f_j(\mathrm{x}_i) - f_c(\mathrm{x}_i)\big)^2\right), \quad i \in \mathbb{I}_j.$$

The idea here is to define the consensus potential for view j using only the data samples observed in view j. The other data samples with missing view information for view j are treated as hidden (or integrated out) in this potential definition. As before, $\sigma_j > 0$ quantifies how far the latent function f_j is apart from f_c. Note that the smaller n_j is, the less the contribution of view j to the overall graphical model.[4] Next we look at the three marginalizations to gain more insight about this graphical model.

4.1 Co-Regularization with Missing Views

It is straightforward to derive all the marginalizations of Bayesian co-training with missing views. For the co-regularization marginal, a simple calculation leads to the following joint distribution for the m latent functions:

$$p(\mathrm{f}_1, \ldots, \mathrm{f}_m) = \frac{1}{Z}\exp\left\{-\frac{1}{2}\sum_{j=1}^{m}\mathrm{f}_j^\top \mathrm{K}_j^{-1}\mathrm{f}_j - \frac{1}{2}\sum_{j<k}\sum_{\mathrm{x}\in\mathbb{I}_j\wedge\mathbb{I}_k}\left[\frac{[f_j(\mathrm{x}) - f_k(\mathrm{x})]^2}{\sigma_j^2\sigma_k^2}\Big/\sum_{\ell:\mathrm{x}\in\mathbb{I}_\ell}\frac{1}{\sigma_\ell^2}\right]\right\}.$$

As in the Bayesian co-training with fully observed views, this provides an equivalent form to co-regularized multi-view learning. The first part regularizes the functional space of each view, and the second part constrains that every pair of views need to agree on the outputs for *co-observed* samples (inversely weighted by view variances and the sum of inverse variances of the views in which the sample is observed). This is very intuitive and naturally extends the joint distribution in (7). If view j and view k do not share any data sample (i.e., no data sample has features from both view j and view k), the view pair (j,k) will not contribute to the joint distribution.[5] A joint probability distribution involving output y_l can also be derived which takes a similar form as in (9).

4. Also note that after hyperparameter learning, σ_j might not fully represent how strongly each view j contributes to the consensus, since the contribution also depends on the number of available data n_j in the view j.
5. Note that view j and view k will still contribute to the overall distribution through other views that they share data samples with.

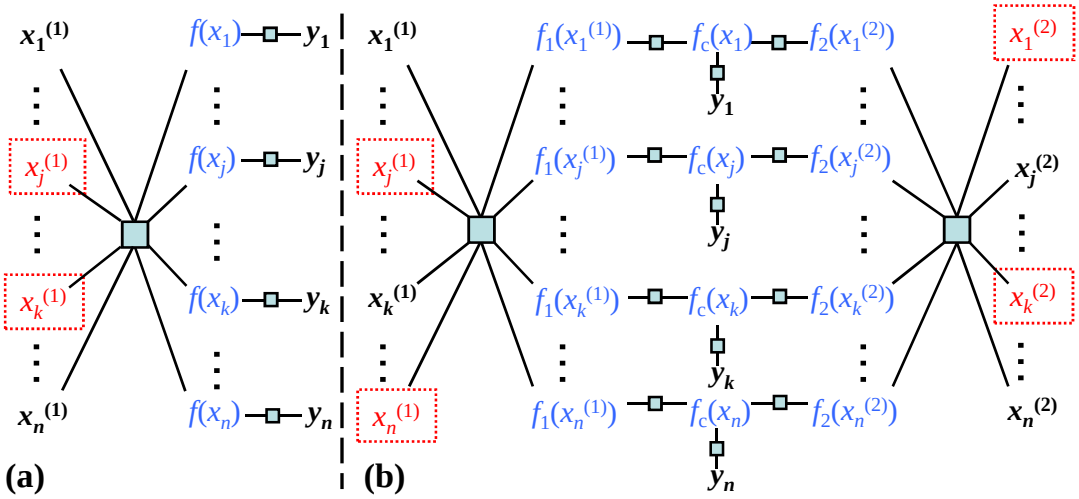


Figure 3: Factor graphs for Bayesian co-training with missing views, for (a) one-view and (b) two-view problems. Observed variables are marked as dark/bold, and unobserved ones are marked as red/non-bold, including functions f_1, f_2, f_c (blue/non-bold). Unobserved variables in a dotted box (such as $\mathrm{x}_j^{(1)}$) are potential observations for active sensing (see Section 5). All labels y are denoted as observed in the graph, but this is not required.

4.2 Co-Training Kernel with Missing Views

We can also derive a co-training kernel K_c by integrating out all the latent functions $\{\mathrm{f}_j\}$ in (12). This leads to a Gaussian prior $p(\mathrm{f}_c) = \mathcal{N}(0, \mathrm{K}_c)$, with

$$\mathrm{K}_c = \Lambda_c^{-1}, \qquad \Lambda_c = \sum_{j=1}^{m} \mathrm{A}_j,$$

where each A_j is a $n \times n$ matrix defined as

$$\mathrm{A}_j(\mathbb{I}_j, \mathbb{I}_j) = (\mathrm{K}_j + \sigma_j^2 \mathrm{I})^{-1}, \text{ and 0 otherwise.} \tag{14}$$

That is, A_j is an expansion of the one-view information matrix $(\mathrm{K}_j + \sigma_j^2 \mathrm{I})^{-1}$ to the full size $n \times n$, with the other (unindexed) entries filled with 0. It is easily seen that such a kernel K_c is indeed positive definite, as long as each one-view kernel K_j is positive definite and at least there are two views sharing one data sample. We also call Λ_c the *co-training precision matrix*. Very importantly, we note that *one additional observation of a (sample, view) pair will affect all the elements of the co-training kernel*. In other words, the kernel value for a pair of samples is potentially changed even when a third (unrelated) object is further characterized by an additional sensor.[6] This property motivates us to do active feature acquisition (or *active sensing*) in the Bayesian co-training framework. Section 5 will discuss this in detail.

6. Note that the marginalizations in Section 4.2 and Section 4.1 are still equivalent (since they come from the same underlying graphical model), despite the fact that additional (sample, view) pair influences the kernels (with dimension $nm \times nm$ in Section 4.1 and $n \times n$ in Section 4.2) differently in these two marginalizations.

4.3 Individual View Learning with Missing Views

If one particular view j is of interest, we can also integrate out the consensus view and all the other views, leading to a GP prior for view j, $\mathrm{f}_j \sim \mathcal{N}(0, \mathrm{C}_j)$, with the precision matrix being

$$\mathrm{C}_j^{-1} = \mathrm{K}_j^{-1} + \left[\sigma_j^2 \mathrm{I} + \Lambda_{c\setminus j}(\mathbb{I}_j, \mathbb{I}_j)^{-1}\right]^{-1}.$$

Here we extract the $(\mathbb{I}_j, \mathbb{I}_j)$ sub-matrix from the *leave-one-view-out* co-training precision matrix $\Lambda_{c\setminus j}$, which is defined as $\Lambda_{c\setminus j} = \sum_{k\neq j} \mathrm{A}_k$. Each A_k is defined as in (14). This marginalization allows us to, for example, measure how much benefit every other view brings to the interested view. An important fact to realize here is that *with an observed (sample, view) pair from another view k, even if this sample is not observed in the primarily interested view j, the kernel of the view j will still be affected so long as* $\mathbb{I}_j \wedge \mathbb{I}_k \neq \emptyset$. One can also introduce the inductive GP inference as in Section 3.3 under this setting.

4.4 Discussion

Bayesian co-training with missing views provides an elegant framework to combine information from multiple views or multiple data sources together, even when different subsets of data samples are measured in different views. For learning and inference, we still prefer using the co-training kernel with the second marginalization due to its simplicity.

We note that the definition of the consensus potentials in (13) implies that the influence of the different pairs of views has been factored into a product. As a consequence, the view-pairs are combined in a linear manner. A way to go beyond this is by using higher-order potentials.

A higher order potential definition $\psi(\mathrm{f}_1, ..., \mathrm{f}_m, \mathrm{f}_c)$, which combines $\mathrm{f}_1, ..., \mathrm{f}_m$ simultaneously, would produce a richer combination of views, but often at the expense of increased inference/computational complexity. It is not clear how to achieve this effect with standard co-training.

Since one observation of a (sample, view) pair will affect the overall co-training kernel, we can derive a framework for *active sensing*, which aims to actively select the best pair for feature acquisition or sensing. This active sensing problem is different from active learning where the goal is to select the best pair for labeling. We discuss this idea in detail in the next section.

5. Active Sensing in Bayesian Co-Training

In active sensing, we are interested in selecting the best unobserved (sample, view) pair for sensing, or for view acquisition, which will improve the overall classification performance. In this section we will focus on logistic regression loss for binary classification. For active sensing we mainly discuss an approach based on the mutual information framework, which measures the expected information gain after observing an additional (sample, view) pair. Another approach based on the predictive uncertainty is also briefly discussed in Section 5.5.

In the following let $\mathcal{D}_O$ and $\mathcal{D}_U$ denote the observed and unobserved (sample, view) pairs, respectively. Recall that under the second marginalization in which only the consensus function f_c is of primary interest, the Bayesian co-training model for binary classification reduces to

$$p(\mathrm{y}_l, \mathrm{f}_c) = \frac{1}{Z}\psi(\mathrm{f}_c)\prod_{i=1}^{n_l}\psi(y_i, f_c(\mathrm{x}_i)),$$

where y_l contains the binary labels for the n_l labeled samples, $\psi(\mathrm{f}_c)$ is defined via the co-training kernel as $\psi(\mathrm{f}_c) = \exp\left\{-\frac{1}{2}\mathrm{f}_c^\top \mathrm{K}_c^{-1}\mathrm{f}_c\right\}$, and $\psi(y_i, f_c(\mathrm{x}_i))$ is the output potential $\lambda(y_i f_c(\mathrm{x}_i))$ with $\lambda(\cdot)$ the logistic function. The log marginal likelihood of the output y_l under this model, conditioned on the input data $\mathrm{X} \triangleq \{\mathrm{x}_i^{(j)}\}$ and model parameters Θ, is:

$$\begin{aligned}\mathcal{L} \triangleq \log p(\mathrm{y}_l|\mathrm{X},\Theta) &= \log \int p(\mathrm{y}_l|\mathrm{f}_c,\Theta)p(\mathrm{f}_c|\mathrm{X},\Theta)\, d\mathrm{f}_c - \log Z \\ &= \log \int \prod_{i=1}^{n_l} \lambda(y_i f_c(\mathrm{x}_i)) \cdot \exp\left\{-\frac{1}{2}\mathrm{f}_c^\top \mathrm{K}_c^{-1}\mathrm{f}_c\right\}\, d\mathrm{f}_c - \log Z.\end{aligned}$$

5.1 Laplace Approximation

To calculate the mutual information we need to calculate the differential entropy of the consensus view function f_c. With co-training kernel and the logistic regression loss, Laplace approximation can be applied to approximate the *a posteriori* distribution of f_c as a Gaussian distribution. The *a posteriori* distribution of f_c, $p(\mathrm{f}_c|\mathcal{D}_O, \mathrm{y}_l, \Theta) \propto p(\mathrm{y}_l|\mathrm{f}_c,\Theta)p(\mathrm{f}_c|\mathcal{D}_O,\Theta)$, is approximately

$$\mathcal{N}(\hat{\mathrm{f}}_c, (\Delta_{\text{post}})^{-1}), \tag{15}$$

where $\hat{\mathrm{f}}_c$ is the maximum *a posteriori* (MAP) estimate of f_c, and the *a posteriori* precision matrix is

$$\Delta_{\text{post}} = \mathrm{K}_c^{-1} + \Phi, \tag{16}$$

with Φ the Hessian of the negative log-likelihood. It turns out that Φ is a diagonal matrix, with $\Phi(i,i) = \eta_i(1-\eta_i)$ where $\eta_i = \lambda(\hat{\mathrm{f}}_c(\mathrm{x}_i))$. The differential entropy of f_c under this Laplace approximation is

$$H(\mathrm{f}_c) = -\frac{n}{2}\log(2\pi e) - \frac{1}{2}\log\det(\Delta_{\text{post}}),$$

where $\det(\cdot)$ denotes the matrix determinant.

5.2 Mutual Information for Active Sensing

Remind that $\mathrm{x}_i^{(j)}$ denote the features in the jth view for the ith sample. In active sensing, the mutual information (MI) between the consensus view function f_c and the unobserved (sample, view) pair $\mathrm{x}_i^{(j)} \in \mathcal{D}_U$ is the *expected decrease in entropy of* f_c when $\mathrm{x}_i^{(j)}$ is observed,

$$I(\mathrm{f}_c, \mathrm{x}_i^{(j)}) = \mathbb{E}[H(\mathrm{f}_c)] - \mathbb{E}[H(\mathrm{f}_c|\mathrm{x}_i^{(j)})] = -\frac{1}{2}\log\det(\Delta_{\text{post}}) + \frac{1}{2}\mathbb{E}\left[\log\det(\Delta_{\text{post}}^{x(i,j)})\right],$$

where the expectation is with respect to $p(\mathrm{x}_i^{(j)}|\mathcal{D}_O, \mathrm{y}_l)$, the distribution of the unobserved (sample, view) pair given all the observed pairs and available outputs. $\Delta_{\text{post}}^{x(i,j)}$ is the *a posteriori* precision matrix, derived from (16), after one pair $\mathrm{x}_i^{(j)}$ is observed.

The maximum MI criterion has been used before to identify the "best" unlabeled sample in active learning (MacKay, 1992). Here we adopt this criterion and choose the unobserved pair which maximizes MI:

$$(i^*, j^*) = \arg\max_{\mathrm{x}_i^{(j)} \in \mathcal{D}_U} I(\mathrm{f}_c, \mathrm{x}_i^{(j)}) = \arg\max_{\mathrm{x}_i^{(j)} \in \mathcal{D}_U} \mathbb{E}\left[\log\det(\Delta_{\text{post}}^{x(i,j)})\right]. \tag{17}$$

5.3 Density Modeling

In order to calculate the expectation in (17), we need a conditional density model for the unobserved pairs, that is, $p(\mathrm{x}_i^{(j)}|\mathcal{D}_O, \mathrm{y}_l)$. This of course depends on the type of the features in each view, and for our applications we use a special Gaussian mixture model (GMM). This model has the nice property that all the marginals are still GMMs, and yet is not too flexible like the full GMM. One can certainly define other density models based on the applications.

For a m-view input data $\mathrm{x} = (\mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(m)})$, let the joint input density be

$$p(\mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(m)}) = p(y=+1)p(\mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(m)}|y=+1) + p(y=-1)p(\mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(m)}|y=-1),$$

and each conditional density takes a *component-wise factorized* GMM form, that is,

$$p(\mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(m)}|y=+1) = \sum_c \pi_c^+ \prod_j \mathcal{N}(\mathrm{x}^{(j)}|\mu_c^{+(j)}, \Sigma_c^{+(j)}),$$
$$p(\mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(m)}|y=-1) = \sum_c \pi_c^- \prod_j \mathcal{N}(\mathrm{x}^{(j)}|\mu_c^{-(j)}, \Sigma_c^{-(j)}).$$

Here, for the positive class, $\mu_c^{+(j)}$ and $\Sigma_c^{+(j)}$ are the mean and covariance matrix for view j in component c, and $\pi_c^+ > 0$, $\sum_c \pi_c^+ = 1$ are the mixture weights. For the negative class we use similar notations. Note that although the conditional density for each mixture component is decoupled for different views, the joint conditional density is not.[7] Under this model, the joint density $p(\mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(m)})$ is also a GMM, and any marginal (conditioned on y or not) density is still a GMM, for example, $p(\mathrm{x}^{(j)}|y=+1) = \sum_c \pi_c^+ \mathcal{N}(\mathrm{x}^{(j)}|\mu_c^{+(j)}, \Sigma_c^{+(j)})$.

Now it is easy to calculate $p(\mathrm{x}_i^{(j)}|\mathcal{D}_O, \mathrm{y}_l)$. Let $\mathrm{x}_i^{(O)}$ be the set of observed views for x_i, we need to distinguish two different settings. When the label y_i is available, for example, $y_i = +1$, we have

$$p(\mathrm{x}_i^{(j)}|\mathcal{D}_O, \mathrm{y}_l) = p(\mathrm{x}_i^{(j)}|\mathrm{x}_i^{(O)}, y_i = +1) = \sum_c \pi_c^{+(j)}(\mathrm{x}_i^{(O)}) \cdot \mathcal{N}(\mathrm{x}_i^{(j)}|\mu_c^{+(j)}, \Sigma_c^{+(j)}), \tag{18}$$

which is again a GMM model, with the mixing weights being

$$\pi_c^{+(j)}(\mathrm{x}_i^{(O)}) = \pi_c^+ \frac{\prod_{k \in O} \mathcal{N}(\mathrm{x}_i^{(k)}|\mu_c^{+(k)}, \Sigma_c^{+(k)})}{p(\mathrm{x}_i^{(O)}|y_i = +1)}.$$

When the label y_i is not available, we need to integrate out the labeling uncertainty and compute

$$\begin{aligned} p(\mathrm{x}_i^{(j)}|\mathcal{D}_O, \mathrm{y}_l) &= p(\mathrm{x}_i^{(j)}|\mathrm{x}_i^{(O)}) \\ &= p(y_i = +1)p(\mathrm{x}_i^{(j)}|\mathrm{x}_i^{(O)}, y_i = +1) + p(y_i = -1)p(\mathrm{x}_i^{(j)}|\mathrm{x}_i^{(O)}, y_i = -1), \end{aligned}$$

which is a GMM model as well, as can be seen from (18).

7. A straightforward EM algorithm can be derived to estimate all these parameters. When labels are only available for a very limited number of samples, one might assume a full generative GMM model neglecting the dependency on labels (instead of a conditional GMM model).

5.4 Expectation Calculation

We are now ready to compute the expectation in (17). The *a posteriori* precision matrix after one (sample, view) pair $\mathrm{x}_i^{(j)}$ is observed, $\Delta_{\text{post}}^{x(i,j)}$, can be calculated as

$$\Delta_{\text{post}}^{x(i,j)} = (\mathrm{K}_c^{x(i,j)})^{-1} + \Phi = \mathrm{A}_j^{x(i,j)} + \sum_{k \neq j} \mathrm{A}_k + \Phi, \tag{19}$$

where $\mathrm{K}_c^{x(i,j)}$ and $\mathrm{A}_j^{x(i,j)}$ are the new K_c and A_j matrices after the new pair is observed. Based on (14), to calculate $\mathrm{A}_j^{x(i,j)}$ we need to recalculate the kernel for the jth view, K_j, after an additional pair $\mathrm{x}_i^{(j)}$ is observed. This is simply done by adding one row and column to the old K_j as:

$$\mathrm{K}_j^{x(i,j)} = \begin{bmatrix} \mathrm{K}_j & \mathrm{b}_j \\ \mathrm{b}_j^\top & a_j \end{bmatrix},$$

where $a_j = \kappa_j(\mathrm{x}_i^{(j)}, \mathrm{x}_i^{(j)}) \in \mathbb{R}$, and $\mathrm{b}_j \in \mathbb{R}^{n_j}$ has the ℓth entry as $\kappa_j(\mathrm{x}_\ell^{(j)}, \mathrm{x}_i^{(j)})$. Then from (14), the non-zero part of $\mathrm{A}_j^{x(i,j)}$ is calculated as

$$\left(\mathrm{K}_j^{x(i,j)} + \sigma_j^2 \mathrm{I}\right)^{-1} = \begin{bmatrix} \mathrm{K}_j + \sigma_j^2 \mathrm{I} & \mathrm{b}_j \\ \mathrm{b}_j^\top & a_j + \sigma_j^2 \end{bmatrix}^{-1} = \begin{bmatrix} \Gamma_j + \lambda_j \Gamma_j \mathrm{b}_j \mathrm{b}_j^\top \Gamma_j & -\lambda_j \Gamma_j \mathrm{b}_j \\ -\lambda_j \mathrm{b}_j^\top \Gamma_j & \lambda_j \end{bmatrix}, \tag{20}$$

using the block-matrix inverse formula, where $\Gamma_j = (\mathrm{K}_j + \sigma_j^2 \mathrm{I})^{-1}$ and $\lambda_j = \frac{1}{a_j + \sigma_j^2 - \mathrm{b}_j^\top \Gamma_j \mathrm{b}_j}$.

As seen from (19) and (20), it is difficult to directly calculate the expectation in (17). Since for any matrix Q, $\mathbb{E}[\log\det(\mathrm{Q})] \leq \log\det(\mathbb{E}[\mathrm{Q}])$ due to the concavity of $\log\det(\cdot)$, we alternatively take the upper bound $\log\det(\mathbb{E}[\Delta_{\text{post}}^{x(i,j)}])$ as the selection criteria and also take the risk that the best pair (i, j) that optimizes $\log\det(\mathbb{E}[\Delta_{\text{post}}^{x(i,j)}])$ doesn't necessarily optimize $\mathbb{E}[\log\det(\Delta_{\text{post}}^{x(i,j)})]$. From (19) and (20), this reduces to computing $\mathbb{E}[\lambda_j], \mathbb{E}[\lambda_j \mathrm{b}_j]$ and $\mathbb{E}[\lambda_j \mathrm{b}_j \mathrm{b}_j^\top]$, where the expectations are with respect to $p(\mathrm{x}_i^{(j)} | \mathcal{D}_O, \mathrm{y})$, a GMM model (cf. Section 5.3). In general one needs to calculate these expectations numerically, as different kernel functions lead to different integrals. As another approximation one might assume each of the GMM component is a point-mass such that the mean is used for the calculation.

5.5 Discussion

The mutual information based approach directly measures the expected information gain for every (sample, view) pair. A different (and simpler) approach is based on the predictive uncertainty, in which the most *uncertain* sample (after the current classifier is trained) is selected for view acquisition. This approach was taken for a different problem in Melville et al. (2004). This uncertainty (i.e., predictive variance) is estimated as the diagonal entries of the *a posteriori* covariance matrix $(\Delta_{\text{post}})^{-1}$, as seen from (15). However it is not clear what view to acquire for this sample (if more than one view is missing for the sample). The advantage of this approach is that no density modeling is necessary for unobserved views.

6. Experiments

For the first part of the experiments we empirically evaluate some single-view and multi-view learning algorithms on several toy data and two real world data sets. We compare the proposed Bayesian

co-training models with the original co-training method proposed by Blum and Mitchell (1998), and several single-view learning algorithms. Since this co-training algorithm—sometimes we call it the *canonical co-training* algorithm—was proposed for classification problems, we focus on classification in this section and compare all the methods with the logistic regression loss. We show both problems where co-training works and does not work (i.e., is not better compared to the single-view learning counterpart).

In the second part we evaluate the active sensing algorithms in the Bayesian co-training setting. We are given a classification task with missing views, and at each iteration we are allowed to select an unobserved (sample, view) pair for sensing (i.e., feature acquisition). The proposed methods are compared with random sensing in which a random unobserved (sample, view) pair is selected for sensing.

6.1 Toy Examples for Bayesian Co-Training

First of all, we show some 2D toy classification problems to visualize the co-training result in Figure 4. We assume each of these 2D problems is a two-view problem, in which one view only contains one single feature. Canonical co-training is applied by iteratively training one classifier based on one view, adding the most confident unlabeled data from one view to the training pool of the other classifier, and retraining each classifier till convergence (i.e., no confident unlabeled data can be added further). In Bayesian co-training we use the squared exponential covariance function as mentioned in Section 2, and the width ρ is set to $1/\sqrt{2}$ which yields the optimal performance.

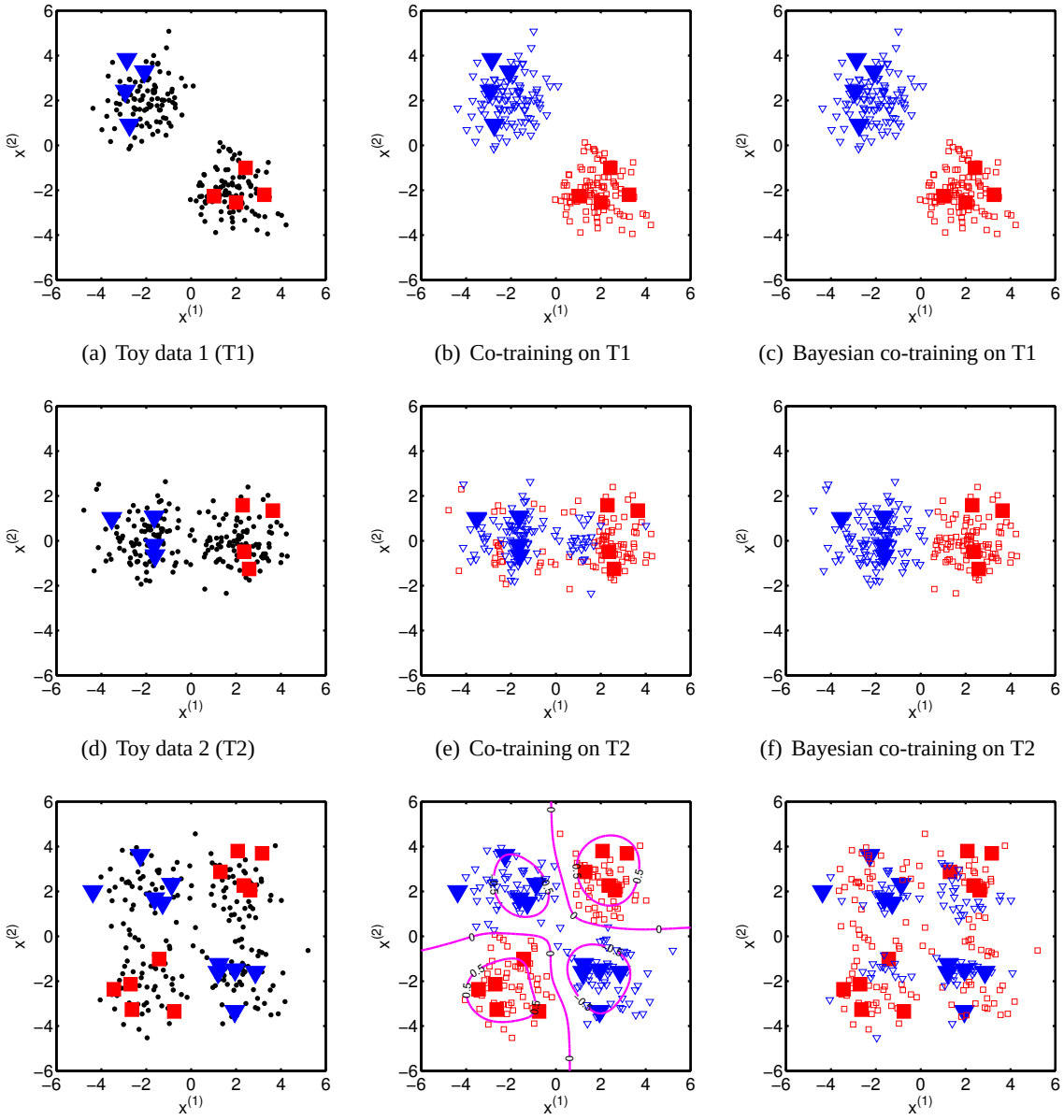
Our first example is a two-Gaussian case with mean $(2,-2)$ and $(-2,2)$, where either feature $x^{(1)}$ or $x^{(2)}$ can be used alone to fully solve the problem (Figure 4(a)). This is an ideal case for co-training, since: 1) each single view is sufficient to train a classifier, and 2) both views are conditionally independent given the class labels. Therefore we see that both canonical co-training and Bayesian co-training yield the same perfect result (Figure 4(b),(c)).

For the second toy data (Figure 4(d)) we assume the two Gaussians are aligned to the $x^{(1)}$-axis (with mean $(2,0)$ and $(-2,0)$). In this case the feature $x^{(2)}$ is totally irrelevant to the classification problem. The canonical co-training fails here (Figure 4(e)) since when we add labels using the $x^{(2)}$ feature , noisy labels will be introduced and expanded to future training. The Bayesian co-training model can handle this situation since we can adapt the weight of each view and penalize the feature $x^{(2)}$ (Figure 4(f)).

The third toy data follows an XOR shape where the data from four Gaussians (with mean $(2,2)$, $(-2,2)$, $(2,-2)$, $(-2,-2)$) lead to a binary classification problem that is not linearly separable (Figure 4(g)). In this case both the two assumptions mentioned above are violated, and neither canonical nor Bayesian co-training will work (Figure 4(i)).[8] On the other hand, a supervised GP classification model with squared exponential covariance function can easily recover the non-linear underlying structure (see Figure 4(h)). This indicates that the learning a multi-view classifier for this problem with the current co-training type algorithms will not succeed. From a kernel design perspective, the consensus based co-training kernel K_c is not suitable for this type of problem.

In summary, these toy problems indicate that when co-training works, Bayesian co-training performs better than or at least as well as canonical co-training models. But since Bayesian co-training is fundamentally a kernel design for a single-view supervised learning, it will not work when the problem calls for more flexible kernel form (e.g., in Figure 4(g)).

8. We also tried other types of covariance functions but they yield similar results.



(a) Toy data 1 (T1) (b) Co-training on T1 (c) Bayesian co-training on T1

(d) Toy data 2 (T2) (e) Co-training on T2 (f) Bayesian co-training on T2

(g) Toy data 3 (T3) (h) GP classification on T3 (i) (Bayesian) co-training on T3

Figure 4: Toy problems for co-training. (b)∼(c) show canonical and Bayesian co-training results on two-Gaussian data (a); (e)∼(f) show the results on two-Gaussian data (d); (h) shows GP classification result on four-Gaussian XOR data (g); (i) shows (Bayesian) co-training result on data (g). Square exponential covariance function was used with width 1 for GP classification and $1/\sqrt{2}$ for each feature in two-view learning. In the toy data big red-square/blue-triangle markers denote the $+1/-1$ labeled points, and black dots denote the unlabeled points.

	# TRAIN +2/-10		# TRAIN +4/-20	
MODEL	AUC	F1	AUC	F1
TEXT	0.5725 ± 0.0180	0.1359 ± 0.0565	0.5770 ± 0.0209	0.1443 ± 0.0705
INBOUND LINK	0.5451 ± 0.0025	0.3510 ± 0.0011	0.5479 ± 0.0035	0.3521 ± 0.0017
OUTBOUND LINK	0.5550 ± 0.0119	0.3552 ± 0.0053	0.5662 ± 0.0124	0.3600 ± 0.0059
TEXT+LINK	0.5730 ± 0.0177	0.1386 ± 0.0561	0.5782 ± 0.0218	0.1474 ± 0.0721
CO-TRAINED GPLR	0.6459 ± 0.1034	0.4001 ± 0.2186	0.6519 ± 0.1091	0.4042 ± 0.2321
BAYESIAN CO-TRAINING	$\mathbf{0.6536 \pm 0.0419}$	$\mathbf{0.4210 \pm 0.0401}$	$\mathbf{0.6880 \pm 0.0300}$	$\mathbf{0.4530 \pm 0.0293}$

Table 1: Results for Citeseer with different numbers of labeled training data (positive/negative). The first three lines are supervised learning results using only the single-view features. The fourth line shows the supervised learning results by combining features from all the three views. The fifth and sixth lines are the co-training results. Bold face indicates the best performance.

MODEL	# TRAIN +2/-2		# TRAIN +4/-4	
	AUC	F1	AUC	F1
TEXT	0.5767 ± 0.0430	0.4449 ± 0.1614	$\mathbf{0.6150 \pm 0.0594}$	0.5338 ± 0.1267
INBOUND LINK	0.5211 ± 0.0017	$\mathbf{0.5761 \pm 0.0013}$	0.5210 ± 0.0019	$\mathbf{0.5758 \pm 0.0015}$
TEXT+LINK	0.5766 ± 0.0429	0.4443 ± 0.1610	$\mathbf{0.6150 \pm 0.0594}$	0.5336 ± 0.1267
CO-TRAINED GPLR	0.5624 ± 0.1058	0.5437 ± 0.1225	0.5959 ± 0.0927	0.5737 ± 0.1203
BAYESIAN CO-TRAINING	$\mathbf{0.5794 \pm 0.0491}$	0.5562 ± 0.1598	0.6140 ± 0.0675	0.5742 ± 0.1298

Table 2: Results for WebKB with different numbers of labeled training data (positive/negative). The first two lines are supervised learning results using only the single-view features. The third line shows the supervised learning results by combining features from both views. The fourth and fifth lines are the co-training results. Bold face indicates the best performance.

6.2 Bayesian Co-Training for Web Page Classification

We use two sets of linked documents for our experiment. The main purpose of these empirical studies is to show the benefit of the proposed Bayesian co-training method compared to single-view learning and the canonical co-training algorithms, and also highlight the limitations of co-training type algorithms. As will be seen later, we show one case that co-training works, in which case Bayesian co-training yields the best performance; we also show one case that co-training does not improve over the single-view counterpart, in which case Bayesian co-training is slightly better than canonical co-training. As the co-training kernel based approach is equivalent to the adaptive co-regularized multi-view learning (since they are based on the same underlying graphical model), we do not include a separate line of results for the co-regularization methods.

The *Citeseer* data set contains 3,312 documents that belong to six classes. There are three natural views for each document: the text view consists of title and abstract of the paper; the two link views are inbound and outbound references. The bag-of-words features are extracted from each view, which amount to 3,703 for the text view, 1,107 for the inbound view and 903 for the outbound view. We pick up the largest class which contains 701 documents and test the one-vs-rest classification performance. The *WebKB* data set is a collection of 4,501 academic web pages

manually grouped into six classes (student, faculty, staff, department, course, project). There are two views containing the text on the page (24,480 features) and the anchor text (901 features) of all inbound links, respectively. We consider the binary classification problem "student" against "faculty", for which there are 1,641 and 1,119 documents, respectively. The preprocessed data sets are kindly shared by Steffen Bickel at `http://www.mpi-inf.mpg.de/~bickel/mvdata/`.

We compare the single-view learning methods based on logistic regression with Gaussian processes (using features in the single view such as TEXT, INBOUND LINK, and OUTBOUND LINK), concatenated-view method based on logistic regression with Gaussian processes (TEXT+LINK), and co-training methods CO-TRAINED GPLR (which stands for Co-Trained Gaussian Process Logistic Regression using canonical co-training) and BAYESIAN CO-TRAINING (using co-training kernel with logistic regression loss function). Linear kernels are used for all the competing methods since it is very robust from our experience in these experiments. For CO-TRAINED GPLR method, we repeat the procedure 50 times, and in each iteration we add the most predictable 1 positive sample and r negative samples into the training set where r depends on the number of negative/positive ratio of each training data set. The classifier we use is the Gaussian process classifier with logistic regression loss (or GPLR for short). For BAYESIAN CO-TRAINING, we use the co-training kernel approach with the same GPLR classifier. Performance is evaluated using AUC score and F1 measure. We vary the number of labeled training documents as seen in Table 1 and 2 (with ratio proportional to the true positive/negative ratio). Single-view learning methods use only the labeled data, and co-training algorithms are allowed to use all the unlabeled data in the training process. The experiments are repeated 20 times and the prediction means and standard deviations are shown in Table 1 and 2.

It can be seen that for the binary classification problem in Citeseer data set, the co-training methods are better than the single-view methods. In this case BAYESIAN CO-TRAINING is better than CO-TRAINED GPLR and achieves the best performance. For WebDB, however, CO-TRAINED GPLR is not as good as the single-view counterparts, and thus BAYESIAN CO-TRAINING is also worse than the purely supervised methods though it is slightly better than CO-TRAINED GPLR. This is maybe because the TEXT and LINK features are not independent given the class labels (especially when two classes "faculty" and "staff" might share features). CO-TRAINED GPLR has higher standard deviations than other methods due to the possibility of adding noisy labels. We have also tried other number of iterations but 50 seems to give an overall best performance.

Note that the single-view learning with TEXT almost achieves the same performance as concatenated-view method. This might be because the number of text features are much more than the link features (e.g., for WebKB there are 24,480 text features and only 901 link features). So these multiple views are very unbalanced and should be taken into account in co-training with different weights. Bayesian co-training provides a natural way of doing it.

6.3 Active Sensing on Toy Data

We show some empirical results on active sensing in this and the following subsections. Suppose we are given a classification task with missing views, and at each iteration we are allowed to select an unobserved (sample, view) pair for sensing (i.e., feature acquisition). We compare the classification performance on unlabeled data using the following three sensing approaches:

- **Active Sensing MI**: The pair is selected based on the mutual information criteria (17).

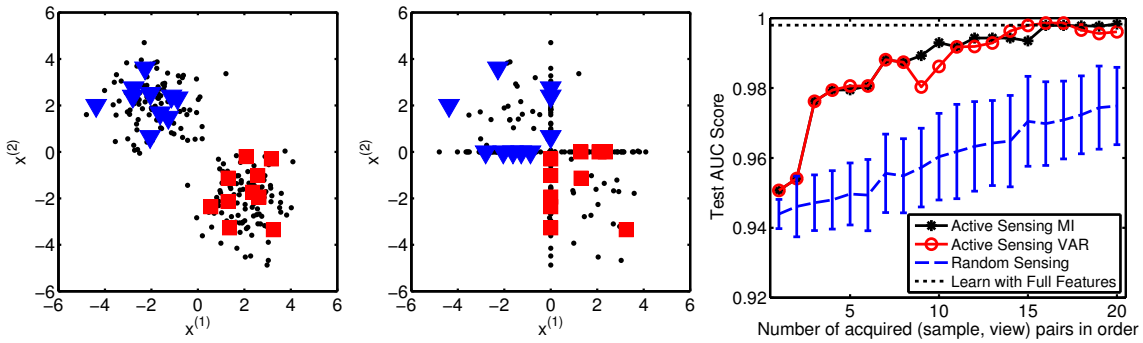


Figure 5: Toy data for active sensing (left). Big red-square/blue-triangle markers denote $+1/-1$ labeled points, and black dots denote unlabeled points. Data are sampled from two Gaussians with mean $(2,-2)$, $(-2,2)$ and unit variance. After "hiding" one feature for some of the data points, the data look like (middle) with removed features replaced with 0. Comparison of active sensing with random sensing is shown on the right. The x-axis labels each acquired pair in order.

- **Active Sensing VAR**: A sample is selected first which has the maximal predictive variance and has missing views, and then one of the missing views is randomly selected for sensing.

- **Random Sensing**: A random unobserved (sample, view) pair is selected for sensing.

After the pair is acquired in each iteration, learning is done using the Bayesian co-training model (with missing views), as discussed in Section 4. Note that for all the three approaches, the acquired (sample, view) pair will affect all the samples in the next iteration (via the co-training kernel). In active sensing with MI, we use EM algorithm to learn the GMM structure with missing entries, and the GMM model is re-estimated after each pair is selected and filled in (this is fast thanks to the incremental updates in the EM algorithm).

We first illustrate active sensing with a toy example. Figure 5 (left) shows a well separated two-class problem which is similar to the one shown in Figure 4(a). To simulate our active sensing experiment, we randomly "hide" one of the two features of each sample with 40% probability each, and with 20% probability observe both features. The final incomplete training data are shown in Figure 5 (middle) with the incomplete samples shown along the first or second axis. It can be seen that only 2 fully observed positive and negative samples are available. For active sensing MI we use the Gaussian kernel with width 0.5, and let the GMM choose the number of clusters automatically (see, e.g., Corduneanu and Bishop, 2001). Standard transductive setting is applied where all the unlabeled data are available for co-training kernel calculation. In Figure 5 (right) we compare active sensing with random sensing, using AUC for the unlabeled data. This indicates that active sensing is much better than random sensing in improving the classification performance. The Bayes optimal accuracy (reachable when there is no missing data) is reached by the 16th query by active sensing whereas random sensing improves much slower with the number of acquired pairs. The two active sensing algorithms show similar results.

Features for NSCLC 2-years Survival Prediction

Feature	Description	View
GENDER	1-Male, 2-Female	1st
WHO	WHO performance status	1st
FEV1	Forced expiratory volume in 1 second	1st
GTV	Gross tumor volume	2nd
NPLN	Number of positive lymph node stations	2nd

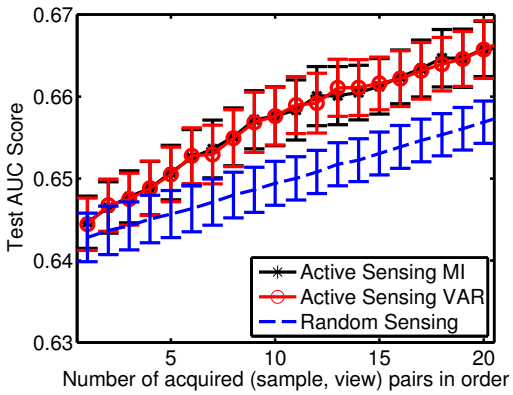


Figure 6: Experiments on NSCLC survival prediction. The features for the 2 views are listed in the left table, and the performance comparison of active sensing and random sensing is shown in the right figure. As baselines, training with full features (i.e., no sensing needed) yields 0.73; training with mean imputation (i.e., using the mean of each feature to fill in the missing entries) yields 0.62.

6.4 Active Sensing in Survival Prediction for Lung Cancer

We consider 2-year survival prediction for advanced non-small cell lung cancer (NSCLC) patients treated with (chemo-)radiotherapy. This is currently a very challenging problem in clinical research, since the prognosis of this group of patients is very poor (less than 40% survive two years). Currently most models in the literature rely on various clinical factors of the patient such as gender and the WHO performance status. Very recently, imaging-related factors such as the size of the tumor and the number of positive lymph node stations are shown to be better predictors (Dehing-Oberije et al., 2009). However, it is expensive to obtain the images and to manually measure these factors. Therefore we study how to select the best set of patients to go through imaging to get additional features. All the relevant factors are listed in Figure 6 (left) with short descriptions. These factors are all known to be predictive based on Dehing-Oberije et al. (2009). From Bayesian co-training point of view we have 2 views, with 3 features in the first (clinical feature) view and 2 features in the second (imaging-based feature) view.

Our study contains 233 advanced NSCLC patients treated at the MAASTRO Clinic in the Netherlands from 2002 to 2006, among which 77 survived 2 years (labeled +1). All the features are available for these patients, and are normalized to have zero mean and unit variance before training. We randomly choose 30% of the patients as training samples (with labels known), and the rest 70% as unlabeled samples. We use linear kernel for each view, and let the GMM algorithm automatically choose the number of clusters. As the active sensing setup, the first view is available for all the patients, and the second view is available only for randomly chosen 50% patients. So our goal is to sequentially select patients to acquire features in view 2, such that the overall classifier performance is maximized. Figure 6 (right) shows the test AUC scores (with error-bars) of active sensing and random sensing, with different number of acquired pairs. Performance is averaged over 20 runs with randomly chosen 50% patients at the start. Active sensing in general yields better performance, and is significantly better after 5 first pairs. Active sensing based on MI and VAR again yield very

similar results. We have also tested other experimental settings, and the comparison is not sensitive to this setup.

6.5 Active Sensing in pCR Prediction for Rectal Cancer

Our second example is to predict tumor response after chemo-radiotherapy for locally advanced rectal cancer. This is important in individualizing treatment strategies, since patients with a pathologic complete response (pCR) after therapy, that is, with no evidence of viable tumor on pathologic analysis, would need less invasive surgery or another radiotherapy strategy instead of resection. Most available models combine clinical factors such as gender and age, and pre-treatment imaging-based factors such as tumor length and SUV_{max} (from CT/PET imaging), but it is expected that adding imaging data collected *after* therapy would lead to a better predictive model (though with a higher cost). In this study we show how to effectively select patients to go through pre-treatment and post-treatment imaging to better predict pCR.

We use the data from Capirci et al. (2007) which contains 78 prospectively collected rectal cancer patients. All patients underwent a CT/PET scan before treatment and 42 days after treatment, and 21 of them had pCR (labeled +1). We split all the features into 3 views (clinical, pre-treatment imaging, post-treatment imaging), and the features are listed in Figure 7 (left). For active sensing, we assume that all the (labeled or unlabeled) patients have view 1 features available, 70% of the patients have view 2 features available, and 40% of the patients have view 3 features available. This is to account for the fact that view 3 features are most expensive to get. All the other settings are the same as the NSCLC survival prediction study. Figure 7 (right) shows the performance comparison of active sensing with random sensing, and it is seen that after about 18 pair acquisitions, active sensing is significantly better than random sensing. Active sensing MI and VAR share a similar trend, and the MI based active sensing is overall better than VAR based active sensing. The difference is however not statistically significant. The optimal AUC (when there are no missing features) is shown as a dotted line, and we see that with around 34 actively acquired pairs, active sensing can almost achieve the optimum. It takes however much longer for random sensing to reach this performance.

7. Conclusion

This paper has two principal contributions. We have proposed a graphical model for combining multi-view data, and shown that previously derived co-regularization based training algorithms maximize the likelihood of this model. In the process, we showed that these algorithms have been making an intrinsic assumption of the form $p(f_c, f_1, f_2, \ldots, f_m) \propto \psi(f_c, f_1)\psi(f_c, f_2)\ldots\psi(f_c, f_m)$, even though it was not explicitly realized earlier. We also studied circumstances when this assumption proves unreasonable. Thus, our first contribution was to clarify the implicit assumptions and limitations in multi-view consensus learning in general, and co-regularization in particular.

Motivated by the insights from the graphical model, our second contribution was the development of alternative algorithms for co-regularization; in particular the development of a non-stationary co-training kernel. Unlike previously published co-regularization algorithms, our approach handles all the following in an elegant framework: (a) handles naturally more than 2 views; (b) automatically learns which views of the data should be trusted more while predicting class labels; (c) shows how to leverage previously developed methods for efficiently training GP/SVM; (d) clearly explains our assumptions, for example, what is being optimized *overall*; (e) does not suffer

Features for pCR Prediction in Rectal Cancer

Feature	Description	View
GENDER	1-Male, 2-Female	1st
AGE	Age in years	1st
STAGE	Staging of cancer	1st
LENGTH	Max diameter of the tumor	2nd
SUVPre	SUV_{max} before treatment	2nd
ΔSUV	Absolute difference of SUV_{max} before and after treatment	3rd
RI	Response Index, ΔSUV in %	3rd

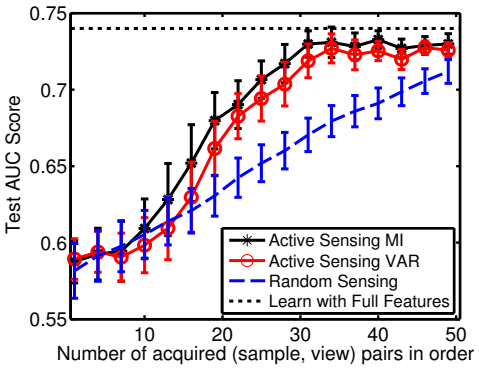


Figure 7: Experiments on pCR prediction for rectal cancer. The features for the 3 views are listed in the left table, and the performance comparison of active sensing and random sensing is shown in the right figure. As baselines, training with full features (i.e., no sensing needed) yields 0.74 (shown as a dotted line); training with mean imputation (i.e., using the mean of each feature to fill in the missing entries) yields 0.55 (not shown).

from local maxima problems; (f) is less computationally demanding in terms of both speed and memory requirements.

We also extend this framework to handle multi-view data with missing features, and introduce an active sensing framework which allows us to actively acquiring missing (sample, view) pairs to maximize performance. In the future we plan to study alternative potentials based on the proposed graphical model, and explore inductive multi-view learning in a more principled manner.

Appendix A. Derivations of the Marginalizations

In this appendix we provide the derivations of the various marginalizations of the Bayesian co-training model, described in Section 3. The joint probability of all the variables is defined as in (6) and is repeated here:

$$p(\mathrm{y}_l, \mathrm{f}_c, \mathrm{f}_1, \ldots, \mathrm{f}_m) = \frac{1}{Z} \prod_{i=1}^{n_l} \psi(y_i, f_c(\mathrm{x}_i)) \prod_{j=1}^{m} \psi(\mathrm{f}_j) \psi(\mathrm{f}_j, \mathrm{f}_c). \tag{21}$$

Recall that the following integration result is true for any $\mathrm{x} \in \mathbb{R}^p$, $\mathrm{b} \in \mathbb{R}^p$, and symmetric matrix $\mathrm{A} \in \mathbb{R}^{p \times p}$.

$$\int \exp\left\{-\frac{1}{2}\mathrm{x}^\top \mathrm{A}\mathrm{x} + \mathrm{b}^\top \mathrm{x}\right\} d\mathrm{x} = \sqrt{\det(2\pi \mathrm{A}^{-1})} \exp\left\{\frac{1}{2}\mathrm{b}^\top \mathrm{A}^{-1}\mathrm{b}\right\}. \tag{22}$$

A.1 Marginal 1: Co-Regularized Multi-View Learning

The first marginalization integrates out the latent consensus function f_c in (21). Ignoring the output consensus function $\psi(y_i, f_c(\mathrm{x}_i))$ for the moment, we derive the joint likelihood

$$\begin{aligned} p(\mathrm{f}_1,\ldots,\mathrm{f}_m) &= \frac{1}{Z}\int \prod_{j=1}^{m} \psi(\mathrm{f}_j)\psi(\mathrm{f}_j,\mathrm{f}_c)\, d\mathrm{f}_c \\ &= \frac{1}{Z}\int \prod_{j=1}^{m} \exp\left\{-\frac{1}{2}\mathrm{f}_j^\top \mathrm{K}_j^{-1}\mathrm{f}_j^\top - \frac{\|\mathrm{f}_j-\mathrm{f}_c\|^2}{2\sigma_j^2}\right\} d\mathrm{f}_c \\ &= \frac{1}{Z}\int \exp\left\{-\frac{1}{2}\sum_{j=1}^{m}\left[\mathrm{f}_j^\top \mathrm{K}_j^{-1}\mathrm{f}_j^\top + \frac{\|\mathrm{f}_j-\mathrm{f}_c\|^2}{\sigma_j^2}\right]\right\} d\mathrm{f}_c \\ &= \frac{1}{Z}\int \exp\left\{-\frac{1}{2}\mathrm{f}_c^\top \mathrm{A}\mathrm{f}_c + \mathrm{b}^\top \mathrm{f}_c + C\right\} d\mathrm{f}_c, \end{aligned}$$

in which we define

$$\mathrm{A} = \sum_j \frac{1}{\sigma_j^2}\mathrm{I}, \quad \mathrm{b} = \sum_j \frac{\mathrm{f}_j}{\sigma_j^2}, \quad C = -\frac{1}{2}\sum_j \left[\mathrm{f}_j^\top \mathrm{K}_j^{-1}\mathrm{f}_j + \frac{\|\mathrm{f}_j\|^2}{\sigma_j^2}\right]. \tag{23}$$

Note that C does not depend on f_c. Applying (22) and absorbing the constants into the normalization factor Z, we have

$$\begin{aligned} p(\mathrm{f}_1,\ldots,\mathrm{f}_m) &= \frac{1}{Z}\exp\left\{-\frac{1}{2}\sum_j \mathrm{f}_j^\top \mathrm{K}_j^{-1}\mathrm{f}_j - \frac{1}{2}\sum_j \frac{\|\mathrm{f}_j\|^2}{\sigma_j^2} + \frac{1}{2}\frac{1}{\sum_j \frac{1}{\sigma_j^2}}\left\|\sum_j \frac{\mathrm{f}_j}{\sigma_j^2}\right\|^2\right\} \\ &= \frac{1}{Z}\exp\left\{-\frac{1}{2}\sum_j \mathrm{f}_j^\top \mathrm{K}_j^{-1}\mathrm{f}_j - \frac{1}{2}\frac{1}{\sum_j \frac{1}{\sigma_j^2}}\left[\sum_j \frac{1}{\sigma_j^2}\cdot\sum_j \frac{\|\mathrm{f}_j\|^2}{\sigma_j^2} - \left\|\sum_j \frac{\mathrm{f}_j}{\sigma_j^2}\right\|^2\right]\right\} \\ &= \frac{1}{Z}\exp\left\{-\frac{1}{2}\sum_j \mathrm{f}_j^\top \mathrm{K}_j^{-1}\mathrm{f}_j - \frac{1}{2}\frac{1}{\sum_j \frac{1}{\sigma_j^2}}\sum_{j<k}\frac{\|\mathrm{f}_j-\mathrm{f}_k\|^2}{\sigma_j^2\sigma_k^2}\right\}. \end{aligned}$$

This recovers the marginal 1 as in (7). To see the GP view of this marginal as in (8), we just need to notice that (7) is a quadratic form of the joint latent functions $(\mathrm{f}_1,\ldots,\mathrm{f}_m)$, and relocate the terms in (7) in the GP format.

When the output potentials $\psi(y_i, f_c(\mathrm{x}_i))$ are taken into account, the whole derivation follows with the only difference that there is an additional term with respect to y in each summation in (23). So we obtain (9) as the joint marginal likelihood.

A.2 Marginal 2: The Co-Training Kernel

To get the co-training kernel we integrate out all the m latent functions in (21), leaving only f_c and y_l. We calculate the marginal distribution of y_l and f_c as follows:

$$\begin{aligned} p(\mathrm{y}_l,\mathrm{f}_c) &= \int p(\mathrm{y}_l,\mathrm{f}_c,\mathrm{f}_1,\ldots,\mathrm{f}_m)\, d\mathrm{f}_1\ldots d\mathrm{f}_m \\ &= \frac{1}{Z}\prod_{i=1}^{n_l}\psi(y_i, f_c(\mathrm{x}_i))\prod_{j=1}^{m}\int \psi(\mathrm{f}_j)\psi(\mathrm{f}_j,\mathrm{f}_c)\, d\mathrm{f}_j, \end{aligned} \tag{24}$$

and

$$\int \psi(\mathrm{f}_j)\psi(\mathrm{f}_j,\mathrm{f}_c)\,d\mathrm{f}_j = \int \exp\left\{-\frac{1}{2}\mathrm{f}_j^\top \mathrm{K}_j^{-1}\mathrm{f}_j - \frac{\|\mathrm{f}_j-\mathrm{f}_c\|^2}{2\sigma_j^2}\right\} d\mathrm{f}_j$$

$$= \int \exp\left\{-\frac{1}{2}\mathrm{f}_j^\top\left(\mathrm{K}_j^{-1}+\frac{1}{\sigma_j^2}\mathrm{I}\right)\mathrm{f}_j + \frac{\mathrm{f}_c^\top}{\sigma_j^2}\mathrm{f}_j - \frac{\|\mathrm{f}_c\|^2}{2\sigma_j^2}\right\} d\mathrm{f}_j \tag{25}$$

$$= \exp\left\{\frac{1}{2}\frac{\mathrm{f}_c^\top}{\sigma_j^2}\left(\mathrm{K}_j^{-1}+\frac{1}{\sigma_j^2}\mathrm{I}\right)^{-1}\frac{\mathrm{f}_c}{\sigma_j^2} - \frac{\|\mathrm{f}_c\|^2}{2\sigma_j^2}\right\} \tag{26}$$

$$= \exp\left\{-\frac{1}{2}\mathrm{f}_c^\top \mathrm{A}_j \mathrm{f}_c\right\}, \tag{27}$$

where

$$\mathrm{A}_j \triangleq \frac{1}{\sigma_j^2}\mathrm{I} - \frac{1}{\sigma_j^2}\left(\mathrm{K}_j^{-1}+\frac{1}{\sigma_j^2}\mathrm{I}\right)^{-1}\frac{1}{\sigma_j^2} = \left(\mathrm{K}_j+\sigma_j^2\mathrm{I}\right)^{-1}.$$

Note that from (25) to (26) we applied the integration result (22). Therefore, from (24) and (27) we have

$$p(\mathrm{y}_l,\mathrm{f}_c) = \frac{1}{Z}\prod_{i=1}^{n_l}\psi(y_i, f_c(\mathrm{x}_i))\exp\left\{-\frac{1}{2}\mathrm{f}_c^\top\left(\sum_j \mathrm{A}_j\right)\mathrm{f}_c\right\},$$

in which the output potentials are equivalent to the conditional density $p(\mathrm{y}_l|\mathrm{f}_c)$, and the big exponential term can be seen as a *prior term* for the consensus function f_c. This leads to the co-training Gaussian prior $p(\mathrm{f}_c) = \mathcal{N}(0,\mathrm{K}_c)$, with $\mathrm{K}_c = (\sum_j \mathrm{A}_j)^{-1}$ being the co-training kernel (10).

A.3 Marginal 3: Individual View Learning with Side-Information

The third marginalization leaves out only the latent function f_j and integrates out the consensus function f_c and all the other latent functions $\{\mathrm{f}_k\}_{k\neq j}$. Ignoring the output potentials for the moment, based on (27) and (22) we have

$$p(\mathrm{f}_j) = \int p(\mathrm{f}_c,\mathrm{f}_1,\ldots,\mathrm{f}_m)\,d\mathrm{f}_c\,d\mathrm{f}_1\ldots d\mathrm{f}_{j-1}\,d\mathrm{f}_{j+1}\ldots d\mathrm{f}_m$$

$$= \frac{1}{Z}\psi(\mathrm{f}_j)\int\left(\psi(\mathrm{f}_j,\mathrm{f}_c)\prod_{k\neq j}\int\psi(\mathrm{f}_k)\psi(\mathrm{f}_k,\mathrm{f}_c)\,d\mathrm{f}_k\right)d\mathrm{f}_c$$

$$= \frac{1}{Z}\psi(\mathrm{f}_j)\int\exp\left\{-\frac{\|\mathrm{f}_j-\mathrm{f}_c\|^2}{2\sigma_j^2} - \frac{1}{2}\mathrm{f}_c^\top\left(\sum_{k\neq j}\mathrm{A}_k\right)\mathrm{f}_c\right\}d\mathrm{f}_c$$

$$= \frac{1}{Z}\psi(\mathrm{f}_j)\int\exp\left\{-\frac{1}{2}\mathrm{f}_c^\top\left(\sum_{k\neq j}\mathrm{A}_k+\frac{1}{\sigma_j^2}\mathrm{I}\right)\mathrm{f}_c + \frac{\mathrm{f}_j^\top}{\sigma_j^2}\mathrm{f}_c - \frac{\|\mathrm{f}_j\|^2}{2\sigma_j^2}\right\}d\mathrm{f}_c$$

$$= \frac{1}{Z}\exp\left\{-\frac{1}{2}\mathrm{f}_j^\top\mathrm{K}_j^{-1}\mathrm{f}_j\right\}\exp\left\{\frac{1}{2}\frac{\mathrm{f}_j^\top}{\sigma_j^2}\left(\sum_{k\neq j}\mathrm{A}_k+\frac{1}{\sigma_j^2}\mathrm{I}\right)^{-1}\frac{\mathrm{f}_j}{\sigma_j^2} - \frac{\|\mathrm{f}_j\|^2}{2\sigma_j^2}\right\}$$

$$= \frac{1}{Z}\exp\left\{-\frac{1}{2}\mathrm{f}_j^\top\mathrm{C}_j^{-1}\mathrm{f}_j\right\},$$

where in the last line we define

$$\begin{aligned} \mathrm{C}_j^{-1} &= \mathrm{K}_j^{-1} + \frac{1}{\sigma_j^2}\mathrm{I} - \frac{1}{\sigma_j^2}\left(\sum_{k\neq j}\mathrm{A}_k + \frac{1}{\sigma_j^2}\mathrm{I}\right)^{-1}\frac{1}{\sigma_j^2} \\ &= \mathrm{K}_j^{-1} + \left(\sigma_j^2\mathrm{I} + \sum_{k\neq j}\mathrm{A}_k\right)^{-1}. \end{aligned}$$

This yields the Equation (11). If we consider the output potentials, a similar GP prior for f_j holds but takes a more sophisticated form.

Appendix B. Optimization of the View Variance Parameters

In this appendix we derive the equations to optimize the view variance σ_j^2 for each view j using the type II maximum likelihood. Under the second marginalization in which only the consensus function f_c is of primary interest, the Bayesian co-training model reduces to

$$p(\mathrm{y}_l, \mathrm{f}_c) = \frac{1}{Z}\psi(\mathrm{f}_c)\prod_{i=1}^{n_l}\psi(y_i, f_c(\mathrm{x}_i)),$$

where $\psi(y_i, f_c(\mathrm{x}_i))$ is the output potential as defined in (1), and $\psi(\mathrm{f}_c)$ is defined via the co-training kernel as

$$\psi(\mathrm{f}_c) = \frac{1}{Z}\exp\left\{-\frac{1}{2}\mathrm{f}_c^\top \mathrm{K}_c^{-1}\mathrm{f}_c\right\}. \tag{28}$$

Note that f_c is of length $n \geq n_l$. This defines a single-view learning problem, and we are effectively assigning a GP prior to f_c with the co-training kernel K_c. The log marginal likelihood of the output y_l under this model, conditioned on the input data $\mathrm{X} \triangleq \{\mathrm{x}_i^{(j)}\}$ and model parameters Θ, is:

$$\mathcal{L} \triangleq \log p(\mathrm{y}_l|\mathrm{X},\Theta) = \log\int p(\mathrm{y}_l|\mathrm{f}_c,\Theta)p(\mathrm{f}_c|\mathrm{X},\Theta)\,d\mathrm{f}_c. \tag{29}$$

In (29) all the probabilities are conditional probabilities, in which $p(\mathrm{y}_l|\mathrm{f}_c,\Theta)$ is defined via (1) and $p(\mathrm{f}_c|\mathrm{X},\Theta)$ is a Gaussian distribution defined via the co-training kernel (28). Here the model parameters Θ contain all the view variance parameters $\{\sigma_j^2\}$, all kernel parameters and other parameters involved in the output potentials. In type II maximum likelihood we maximize (29) with respect to these model parameters. In the following we derive the equations in the regression case, that is, the output potential is a Gaussian noise model. Similar but more complicated equations can be derived for classification case and readers please refer to Rasmussen and Williams (2006) for details.

When the outputs y_l are regression outputs, the integral in (29) can be computed analytically as

$$\mathcal{L} = -\frac{1}{2}\mathrm{y}_l^\top \mathrm{G}^{-1}\mathrm{y}_l - \frac{1}{2}\log\det\mathrm{G} - \frac{n}{2}\log 2\pi,$$

in which for simplicity we rename $\mathrm{G} \triangleq \mathrm{K}_c(1:n_l, 1:n_l) + \sigma^2\mathrm{I}$. Note that since y_l is only of length $n_l \leq n$, matrix G only involves the $n_l \times n_l$ sub-matrix of K_c. For each $\theta \in \Theta$, the partial derivative

of $\mathcal{L}$ with respect to θ is calculated as:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \theta} &= \frac{1}{2} y_l^\top G^{-1} \frac{\partial G}{\partial \theta} G^{-1} y_l - \frac{1}{2} \mathrm{tr}\left[G^{-1} \frac{\partial G}{\partial \theta} \right] \\ &= \frac{1}{2} \mathrm{tr}\left[\left(\alpha\alpha^\top - G^{-1} \right) \frac{\partial G}{\partial \theta} \right], \end{aligned} \tag{30}$$

where $\alpha = G^{-1} y_l$, and $\mathrm{tr}(\cdot)$ denote the matrix trace. We are now ready to calculate the partial derivative of $\mathcal{L}$ with respect to each view variance σ_j^2. We first compute the partial derivative of K_c with respect to σ_j^2 as:

$$\begin{aligned}\frac{\partial K_c}{\partial \sigma_j^2} &= \frac{\partial}{\partial \sigma_j^2} \left[\sum_j \left(K_j + \sigma_j^2 I \right)^{-1} \right]^{-1} \\ &= -K_c \cdot \frac{\partial}{\partial \sigma_j^2} \left(K_j + \sigma_j^2 I \right)^{-1} \cdot K_c \\ &= K_c \left(K_j + \sigma_j^2 I \right)^{-1} \cdot \frac{\partial}{\partial \sigma_j^2} \left(K_j + \sigma_j^2 I \right) \cdot \left(K_j + \sigma_j^2 I \right)^{-1} K_c \\ &= K_c \left(K_j + \sigma_j^2 I \right)^{-1} \left(K_j + \sigma_j^2 I \right)^{-1} K_c. \end{aligned}$$

Then if we name matrix $B_j \triangleq K_c (K_j + \sigma_j^2 I)^{-1} (K_j + \sigma_j^2 I)^{-1} K_c$, we have

$$\frac{\partial G}{\partial \sigma_j^2} = \frac{\partial}{\partial \sigma_j^2} K_c(1:n_l, 1:n_l) = B_j(1:n_l, 1:n_l). \tag{31}$$

This equation follows since we have

$$\begin{aligned}\frac{\partial}{\partial \sigma_j^2} K_c(1:n_l, 1:n_l) &= \frac{\partial}{\partial \sigma_j^2} \begin{pmatrix} I_{n_l} & 0 \end{pmatrix} \cdot K_c \cdot \begin{pmatrix} I_{n_l} \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} I_{n_l} & 0 \end{pmatrix} \cdot \frac{\partial}{\partial \sigma_j^2} K_c \cdot \begin{pmatrix} I_{n_l} \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} I_{n_l} & 0 \end{pmatrix} \cdot B_j \cdot \begin{pmatrix} I_{n_l} \\ 0 \end{pmatrix} \\ &= B_j(1:n_l, 1:n_l). \end{aligned}$$

Note that even though we only need to consider the top left corner of matrix B_j in the derivative calculation, each entry in this sub-matrix depends both on labeled data and on unlabeled data. This provides some additional insight since even with f_c integrated out, the marginal likelihood still depends on unlabeled data, so as the optimization of the hyperparameters σ_j^2.

With (30) and (31) we can calculate $\partial \mathcal{L} / \partial \sigma_j^2$ and then use conjugate gradients to find the optimal σ_j^2. Since the derivatives for the different σ_j^2 are coupled, one needs to iteratively optimize each σ_j^2 until convergence. The partial derivative for σ^2 can be easily computed as $\frac{\partial G}{\partial \sigma^2} = I_{n_l}$. Similarly one can derive the partial derivatives for other kernel parameters inside each kernel K_j and we omit the details.

References

M. Balcan and A. Blum. A PAC-style model for learning from labeled and unlabeled data. In *Semi-Supervised Learning*, pages 111–126. MIT Press, 2006.

M. Balcan, A. Blum, and K. Yang. Co-training and expansion: Towards bridging theory and practice. In *NIPS*, 2004.

S. Bickel and T. Scheffer. Estimation of mixture models using Co-EM. In *ECML*, 2005.

M. Bilgic and L. Getoor. VOILA: Efficient feature-value acquisition for classification. In *AAAI*, 2007.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

U. Brefeld and T. Scheffer. Co-EM support vector learning. In *ICML*, 2004.

U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel. Efficient co-regularised least squares regression. In *ICML*, pages 137–144, 2006.

C. Capirci, L. Rampin, P. Erba, F. Galeotti, G. Crepaldi, E. Banti, M. Gava, S. Fanti, G. Mariani, P. Muzzio, and D. Rubello. Sequential FDG-PET/CT reliably predicts response of locally advanced rectal cancer to neo-adjuvant chemo-radiation therapy. *Eur J Nucl Med Mol Imaging*, 34, 2007.

A. Corduneanu and C. M. Bishop. Variational Bayesian model selection for mixture distributions. In *Workshop AI and Statistics*, pages 27—34, 2001.

S. Dasgupta, M. Littman, and D. McAllester. PAC generalization bounds for co-training. In *NIPS*, 2001.

V. de Sa. Spectral clustering with two views. In *ICML Workshop on Learning With Multiple Views*, 2005.

C. Dehing-Oberije, S. Yu, D. De Ruysscher, S. Meerschout, K. van Beek, Y. Lievens, J. van Meerbeeck, W. de Neve, G. Fung, B. Rao, S. Krishnan, H. van der Weide, and P. Lambin. Development and external validation of prognostic model for 2-year survival of non-small-cell lung cancer patients treated with chemoradiotherapy. *Int J Radiat Oncol Biol Phys*, 2009.

J. Farquhar, D. Hardoon, H. Meng, J-S. Taylor, and S. Szedmak. Two view learning: SVM-2K, Theory and Practice. In *NIPS*, 2005.

R. Hwa, M. Osborne, A.Sarkar, and M. Steedman. Corrected co-training for statistical parsers. In *ICML Workshop The Continuum from Labeled to Unlabeled Data*, 2003.

S. Kiritchenko and S. Matwin. Email classification with co-training. Technical report, University of Ottawa, 2002.

A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *JMLR*, 9:235–284, 2008.

B. Krishnapuram, D. Williams, Y. Xue, A. Hartemink, L. Carin, and M. Figueiredo. On semi-supervised classification. In *NIPS*, 2004.

D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4:590–604, 1992.

P. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney. Active feature-value acquisition for classifier induction. In *IEEE International Conference on Data Mining*, 2004.

K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Workshop on information and knowledge management*, 2000.

D. Pierce and C. Cardie. Limitations of co-training for natural language learning from large datasets. In *EMNLP-2001*, 2001.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

V. Sindhwani and D. S. Rosenberg. An RKHS for multi-view learning and manifold co-regularization. In *ICML*, 2008.

V. Sindhwani, P. Niyogi, and M. Belkin. A co-regularization approach to semi-supervised learning with multiple views. *ICML Workshop on Learning With Multiple Views*, 2005.

K. Sridharan and S. M. Kakade. An information theoretic framework for multi-view learning. In *COLT*, 2008.

W. Wang and Z.-H. Zhou. Analyzing co-training style algorithms. In *European Conference on Machine Learning*, 2007.

W. Wang and Z.-H. Zhou. A new analysis of co-training. In *International Conference on Machine Learning*, 2010.

K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian processes from multiple tasks. In *International Conference on Machine Learning*, 2005.

S. Yu, B. Krishnapuram, R. Rosales, H. Steck, and B. Rao. Bayesian co-training. In *NIPS*, 2008.

X. Zhu, J. Lafferty, and Z. Ghahramani. Semi-supervised learning: from Gaussian fields to Gaussian processes. Technical report, CMU-CS-03-175, 2003.