CoNet: Feature Generation for Multi-View Semi-Supervised Learning with Partially Observed Views

Brian Quanz and Jun Huan
Information and Telecom. Tech. Center
Dept. of Electrical Eng. and Computer Science
University of Kansas, Lawrence, KS 66045
{bquanz,jhuan}@ittc.ku.edu

ABSTRACT

Multi-view semi-supervised learning methods try to exploit the combination of multiple views along with large amounts of unlabeled data in order to learn better predictive functions when limited labeled data is available. However, lack of complete view data limits the applicability of multi-view semi-supervised learning to real world data. Commonly, one data view is readily and cheaply available, but additionally views may be costly or only available in some cases. This work aims to make multi-view semi-supervised learning approaches more applicable to real world data specifically by addressing the issue of missing views. We introduce CoNet, a feature generation method that learns a mapping from one view to another that is specifically designed to produce features that are useful for multi-view semi-supervised learning algorithms. The mapping is then used to fill in views as pre-processing. Our comprehensive experimental study demonstrates the utility of our method as compared to the state-of-the-art multi-view semi-supervised learning methods for this scenario of partially observed views.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining; I.5.2 [**Design Methodology**]: Feature evaluation and selection

Keywords

semi-supervised learning, multi-view learning, missing data

1. INTRODUCTION

With the fast development of cost-effective data collection methods in imaging, the health care industry, the web, social networks, and sensor networks, data from multi-sensory devices, i.e., multi-view data, become ubiquitous. In the multi-view data setting, information collected from each sensory device is a "view". Often individual views are sufficient for prediction tasks given enough labeled data. Multi-view semi-supervised learning methods aim to take advantage of large amounts of unlabeled data by enforcing view-specific predictor consensus on the unlabeled data. Multi-view semi-supervised learning (MVSSL) has been shown to be effective in a variety of applications including text mining [5, 45, 46], image annotation [14, 39], and chemical classification [11, 12].

A key limitation that restricts the wide application of existing MVSSL approaches to a wide range of real-world data sets is that those approaches require the completeness of the data set. Complete multi-view data, however, are rare and a much more common scenario is *incomplete* multi-view data where views may only be available for a subset of samples. For example, for prediction tasks involving chemicals, molecular structure features based on chemical graphs (view 1) can be readily obtained, but obtaining the chemical bioactivity data (e.g., chemical-protein interaction profiles) for a set of proteins (view 2) can be costly and time-consuming. As another example in medical diagnostics [45] where additional views correspond to expensive tests like MRI imaging, information from such views are subject to opportunity. Yet another example of incomplete views comes from webpage classification where incoming link text features provide a convenient second view [5]. Such information may not be always available for new webpages since it requires time and resources to collect.

This case of MVSSL with various amounts of incomplete view data, which we call *multi-view semi-supervised learning with partially observed views*, is commonly encountered in many real-world applications but has barely been addressed in the data mining and machine learning literature. The first method to claim credit for considering missing views in the MVSSL setting is the Gaussian process co-regularization (GPCR) approach [45]. Under this approach missing views are handled in a Bayesian framework by integrating out the missing view function values. Though it has achieved promising preliminary results, GPCR has several limitations. First, GPCR is built on a particular MVSSL framework, co-regularization, which is not always the best or most appropriate for a given application. Second, GPCR essentially ignores those unlabeled data points without a second view, limiting its applicability to cases with little-to-no second view data. A closely related direction to handling partially observed views is the study of MVSSL methods when there is no second view data [6, 7, 16, 27, 31, 41, 48, 49]. The most recent, state-of-the-art method in this category is pseudo multi-view co-training (PMC) [7], which is also the first in this category to explicitly consider conditions for the success of MVSSL algorithms. This method works by choosing a feature partition at each iteration in order to artificially derive two views. However all of the methods in this category completely ignore additional view data and hence cannot take advantage of such data when available. Furthermore, whereas appropriate real data inherently satisfies the desired conditions, with artificially constructed view data the satisfaction of such conditions can only be approximately estimated. In addition feature-splitting approaches like PMC will fail when all or most of the features in a view are needed for a predictor to achieve

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

high-performance. Furthermore the transformation needed to result in two sufficient views may be more complex than a simple partition. Additionally these methods are also often tied to a particular MVSSL algorithm, e.g., PMC is closely integrated with the co-training algorithm and it is not clear if it could even be applied to a co-regularization algorithm, for example.

We aim to extend MVSSL to handle cases with partially observed views. In our study, we assume there is one view that is present in all data. The rest of the views may only be partially observed. Although this assumption may seem restrictive at first glance, it is quite generic in real-world examples. For example, in the chemical activity prediction example that we cited previously, features computed from chemical structures are always available (since those features are computed). As another example, in the webpage classification example, for every webpage, features computed from the content of the page itself (e.g., the bag-of-word representation of the page) are always available but the incoming link information may be missing.

To solve the problem, we have designed a unified approach, CoNet, which uses a feature-generation network for learning a mapping to fill in missing views. A motivating observation is that feature generation approaches are widely used to improve performance for standard supervised learning tasks, therefore we might expect a feature generation approach to also be helpful in the MVSSL setting. However, a key difference is that the goal for the generated data is different - in this case the generated view data should have properties making it useful for MVSSL, that is in conjunction with the original data. We start with the idea of using random nonlinear feature generation functions to generate new view data. Random nonlinear features allow variability in the generated view: the data points are "scattered" to some extent so that labeled data points may be closest to different unlabeled data points in the generated view. This helps ensure that conditions sufficient for the success of MVSSL algorithms are met, in particular the "expansion" condition [3] requiring that there is some chance that some unlabeled data instances can be labeled with "confidence" in one view but not the other. By incorporating these features together in a network structure, we can then fine-tune the collective set of feature generation functions to further ensure that the conditions for MVSSL algorithms are met, namely label consistency and view variability, and additionally that the generated features are consistent with any partial view data available. This results in a very natural approach to generating features for MVSSL. Our approach has the key advantages of operating as a pre-processing step which allows the subsequent application of the most application-appropriate MVSSL algorithm to the completed data, efficient out-of-sample extension, and the ability to make use of additional view data when available. Our comprehensive experimental study demonstrates the utility of the CoNet method as compared to the state-of-the-art MVSSL methods GPCR and PMC.

2. RELATED WORK

Multi-view semi-supervised learning has attracted significant research interest in recent years [9, 42, 43]. Methods for multi-view semi-supervised learning generally exploit in some way the idea of predictive agreement on unlabeled data for ideal functions from each view, whether explicitly or implicitly. MVSSL approaches can be roughly divided into three major categories: pseudo-labeling approaches, which iteratively label unlabeled instances [5]; co-regularization approaches, which incorporate the agreement idea into an optimization problem via constraints or regularization terms [14, 37, 47]; and active learning approaches, which use the agreement idea to select unlabeled instances for labeling by a human [25].

View Generating Functions. Theoretical results were established and verified in experiments showing that improved generalization error could be achieved by using pre-defined view-generating functions mapping one view to another to fill in missing views and effectively increasing the training set size for each view [1]. The limitation of this work is that the existence of "natural" view mapping functions (e.g., translators for cross language text categorization) is assumed. Such natural view mapping functions do not exist for many applications.

View Splitting for MVSSL. One extreme case of partially observed views is the case of having only a single view. There are several approaches that aim to extend the ideas of multi-view semi-supervised learning to single view learning, following a general idea of splitting the features of one view into multiple sets [6, 27]. Recently, one such approach was proposed in which features are split into two views according to criteria that included satisfying the expansion condition for co-training [3], by finding a split such that some unlabeled instances are labeled with confidence in one view but not the other given the current view models [7]. However feature splitting approaches rely on the assumption that the split sets of features will be sufficient for learning. This means they cannot be applied to data where most of the features are needed for learning a good predictor, for example, see Figure 3; splitting the features in this case would result in overlapping classes in each new view. Secondly, even if useful redundancy is present in a single view, this redundancy may be in the form of arbitrary linear combinations of the features or more complex functions of the features, as opposed to the more restricted mapping of feature partitioning.

Additionally for the single view case, several approaches based on using diverse predictors have been proposed [16, 41, 48, 49]. However, in addition to restricting the choice of algorithms, these approaches do not have a clear way for choosing which predictors to use. For instance in one approach co-training was performed using k-nearest-neighbor regressors with different distance metrics and/or values of k in place of different views, but mixed results were obtained depending on the arbitrary choices [49], and further this limits what methods can be used and diversity may come at the cost of worse performance for the individual predictors used.

It is also worth mentioning that many latent model, multi-modal fusion methods [8, 22, 26] might also be used to estimate missing views, but these approaches have the goal of combining different views into one as opposed to exploiting the variability in distinct views, and as such they do not consider the subsequent application of MVSSL algorithms.

When we say that one view is "missing" in MVSSL for a data instance, we mean that all the feature values in that view are not recorded. In this sense we are discussing structured missing values, which is dramatically different from handling random missing values [24].

3. BACKGROUND

3.1 Notation and Setting

We use the following notations throughout the rest of the paper. We use lowercase letters to represent scalar values, lower-case letters with an arrow to represent vectors (e.g., $\vec{x}$), uppercase letters to represent matrices, and uppercase calligraphic letters to represent sets. We use $||\vec{a}||_p = (\sum_{i=1}^{k} |a_i|^p)^{1/p}$ to denote the L_p norm of a k-dimensional vector $\vec{a}$. Unless stated otherwise, all vectors are column vectors.

In MVSSL with partially observed views, we have two sets of data. One set is a set of n labeled samples, e.g., $\{(\vec{x}_1^1, \vec{x}_1^2, \ldots, \vec{x}_1^V, y_1), \ldots, (\vec{x}_n^1, \vec{x}_n^2, \ldots, \vec{x}_n^V, y_n)\} \in \mathcal{X}^1 \times \mathcal{X}^2 \times \mathcal{Y}$. Additionally we have

a set of m unlabeled data points from the same spaces, $\{(\vec{x}^1_{n+1}, \vec{x}^2_{n+1}, \ldots, \vec{x}^V_{n+1}), \ldots, (\vec{x}^1_{n+m}, \vec{x}^2_{n+m}), \ldots, \vec{x}^V_{n+m})\} \in \mathcal{X}^1 \times \mathcal{X}^2$. V is the number of views.

For simplicity we will restrict further discussion to the case of $V = 2$ views, though all the proposed methods can be extended to more than two views. We take $\mathcal{X}^1$ to be $\mathbb{R}^{p_1}$ and $\mathcal{X}^2$ to be $\mathbb{R}^{p_2}$ for some positive integers p_1 and p_2, i.e., view 1 has p_1 features and view 2 p_2 features. We also restrict the label space to $\mathcal{Y} = \{-1, 1\}$ since all of the applications discussed and tested in the experiments deal with binary classification. Additionally we assume that one view is always present but the other is potentially missing in some samples, for two reasons. First, this is the scenario encountered in all data sets used in the proposed experiments, and is the most commonly encountered one. Second, solving this case immediately provides a solution to the case of additional views that may also have missing view cases, simply by computing pair-wise feature generation functions for filling in each view.

3.2 View Expansion in Multi-view Learning

There has been much research on the conditions for which MVSSL may lead to improved predictive performance. There are at least four directions. First originally the condition of conditional independence of views given the class label was proposed as the required condition for the success of co-training [5]. Second for the co-regularization method [46] showed how the co-regularization approach was equivalent to using a special data-dependent kernel for the support vector machine. [38] simplified the theoretical analysis and established similar bounds as [34] and further proposed a co-regularized alternative to manifold regularization [4] that offered significant empirical improvement in their experiments. Following this direction [45] designed a Bayesian MVSSL algorithm that handles missing views.

We follow a different direction of view expansion. It has been shown that an "expansion" condition, weaker than conditional independence, is sufficient for MVSSL to improve over single view learning [3]. This condition requires that there exist some instances whose labels are not confidently[1] known in one view but are confidently known in the other view, so that labels could be propagated iteratively between views. One illustrative way of thinking about this is with the following example with two data views. Suppose an unlabeled instance $\vec{x}^1$ in view 1 is in a region in which a given predictive model is confident corresponds to label y, e.g., due to being close to many y-labeled instances in that view. It may be reasonable to assume with confidence that the label of $\vec{x}^1$ is also y. Then the expansion condition would require that the same unlabeled instance, $(\vec{x}^1, \vec{x}^2)$ not be in such a confident region when restricted to the second view, $\vec{x}^2$ in view 2, at least for some such $(\vec{x}^1, \vec{x}^2)$ in the unlabeled data. For example, $\vec{x}^2$ may only be near other unlabeled instances in view 2. If this condition always holds as confident labels are propagated between views, than all of the instances can be labeled. This example is illustrated in Figure 1, where the solid rectangle corresponds to the positive class and the dotted box shows a possible "expanded" region for the location of the corresponding view 2 point. This potential shuffling means that labeled points can end up near different unlabeled points in the second view and therefore label confidence (based on proximity) can be transferred to the unlabeled points.

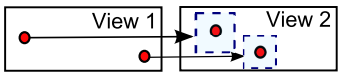


Figure 1: An Example Illustrating View Expansion.

This condition motivates the idea proposed here of using the distances between the *profiles* of the data in each view for determining if pairs of views provide sufficiently complementary information, when evaluating candidate values for filling in missing views. Here "profile" refers to a vector capturing the relationship between a data instance $\vec{x}^j$ in view j and all of the unlabeled data in that view, $\vec{x}^j_{n+1}, \ldots, \vec{x}^j_{n+m}$. Specifically here the profile vector $\vec{v}^j$ in view j of distances between $\vec{x}^j$ and each $\vec{x}^j_i$ is given by $\vec{v}^j_i = d(\vec{x}^j, \vec{x}^j_{n+i})$ for $i = 1, \ldots, m$ for a distance function d. An additional motivation for this idea comes from theoretical analysis for co-regularization [38]. In providing a generalization error bound, the authors also found that the key factor that reduced the bound was a sum of distances between the profiles of the labeled data in each view, with the profiles calculated using a kernel function [38]. The greater these differences in profiles between the views are, the greater the bound on generalization error is reduced.

This motivating difference in profiles idea is incorporated into the proposed approach through a term in the objective function for a feature generation mapping that encourages the sum of squared profile differences $\sum_i \hat{d}(\vec{v}^1_i, \vec{v}^2_i)^2$ to be large, where $\vec{v}^2$ is the profile in the second view which may be generated and $\hat{d}$ is a distance function, potentially different from d. We call this "contrasting view regularization" and this term is described in Section 4.4. The intuition is that, by finding a feature generation function that causes labeled data to be nearby different unlabeled data in the generated view than in the original view, this creates the potential for MVSSL methods to effectively utilize the multiple views, for example, through confident label propagation as previously described.

4. METHODOLOGY

4.1 CoNet Overview

The main idea behind our approach is to use random nonlinear feature functions to introduce variability in generated views, and to fine-tune these functions to match sufficient conditions for the success of multi-view semi-supervised learning methods and to be consistent with available view 2 data. Matching the available view 2 data also helps to ensure the generated second view is useful for classifying the data. To generate random nonlinear feature functions, we generate random projection directions by iteratively sampling a vector $\vec{w}$ from a p_1-dimensional spherical Gaussian and then normalizing $\vec{w}$ to have length 1. We than choose an initial offset uniformly at random in the range of the values taken by the projected data (both labeled and unlabeled). A sigmoid transfer function, $f(x) = 1/(1 + \exp(-x))$ is then applied to introduce nonlinearity.

In order to allow easy fine-tuning of the feature functions, we group functions together into a multi-layered network, i.e., our approach fits naturally into a neural network framework. The final layer is the feature output layer of the network, and each feature function shares all lower layers to allow easier fine-tuning. Each layer is initially generated using the random projection procedure as described above. In our experiments we take the approach of

[1]In the theoretical results of the cited paper "confident" means "with probability one" i.e., absolute certainty. The authors consider particular scenarios where certain regions of the input space can be labeled with absolute certainty. In practice this is relaxed to mean "relative confidence" for the specific model being used, for example, if a linear model is used the unlabeled instances whose labels are considered to be the most confidently known are usually taken as those farthest from the hyperplane defined by the linear model.

using a single hidden layer followed by the feature output layer, as using a large enough number of hidden nodes can allow sufficient expressivity [10].

In addition we consider the recent advancement from the side of neural networks and explore the initialization strategy of deep belief networks - pre-training the network as a generative model using contrastive divergence [20]. This alternative for initializing the feature generation network potentially provides better performance and stability as it may capture the data manifold and prevent overfitting - identifying an accurate lower-dimensional feature representation for the data could facilitate the feature generation network learning.

Subsequently the first condition to ensure through fine-tuning is consistency with available labeled data, which we achieve by adding an additional output node to the network and using a typical loss function for this output node in an overall objective function for the network. Another term is added to the objective function penalizing the distance between generated view 2 instances and actual view 2 instances when available. Finally, although using random nonlinear features can already help to shuffle the distances between labeled and unlabeled points, we add a "contrasting view regularization" (Section 3.2) term to the objective to help ensure this characteristic. Details are given in the following sub-sections.

4.2 Proposed Feature Generation Method

A neural-network model is proposed for the feature generation network, mapping one view to another. The general model is depicted in Figure 2, which shows a particular network with three input features in view 1, three output features in view 2, and one hidden layer of three units.

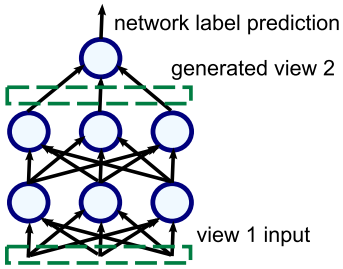


Figure 2: Example feature generation network model, where inputs are entered at the bottom and computations propagate through to the top.

An input $\vec{x}^1$ from view 1 is presented to the network, each set of values is transformed by a linear function at each node and passed through a nonlinear transformation $f()$ to get the output of the node, here we use the sigmoid transformation $f(a) = 1/(1 + \exp(-a))$. Thus the vector of outputs for a layer j is given by $\vec{f}_j \triangleq \vec{f}(W_j \vec{f}_{j-1} + \vec{b}_j)$ where W_j and $\vec{b}_j$ corresponds to the weight matrix and bias vector for the j^{th} layer of the network, respectively, $\vec{f}_0 \triangleq \vec{x}^1$ for $j = 1, \ldots, K$ where K is the number of layers in the network. The generated feature view, which corresponds to the second view and also must have the same number of features as the second view if available, here corresponds to the output of the second-to-last set of nodes in the network, counting from the bottom. In order to also incorporate good performance on the labeled training data, the network's final output is the predicted label.

The weights and biases are then learned from the available data by attempting to find a local minimum of an objective function. In its most basic form, corresponding to a basic feature generation, or neural, network, the objective function is just the sum of a loss term approximating misclassification error. The basic objective function is given by Equation 1, where $\vec{f}_{j,i}$ is the output of the j^{th} layer on an input to the network of $\vec{x}_i^1$.

$$\text{argmin.}_{W_j, \vec{b}_j, \forall j} \quad \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i f_{K,i})) \tag{1}$$

Since the objective function and all transfer functions are differentiable, gradients are straight-forward to compute using the chain rule which results in backpropagation with the network structure. A gradient descent approach is then used to find a local solution.

Once the weights and biases are learned from the data, the model can be applied to each instance missing another view, to generate the missing view for that instance. To ensure generated view data is on the same scale as the available view 2 data, we first generate all view 2 data instances, normalize the data, and then (optionally) fill in the available real view 2 data. Afterwards, any desired multi-view semi-supervised learning algorithm can be applied to the completed data.

4.3 Incorporating Available Partial View Data

When another sufficient and contrasting view is known to exist, and is present in some cases, ideally the training for the feature generation model should take advantage of this available second view data, to help find a better feature generation function and ensure classification sufficiency of the generated view 2 data. The feature generation model should be biased toward a model that generates values close to the true second view values. This is easily accomplished in the proposed feature generation network model by incorporating an additional penalty term in the objective function. The penalty term is the sum of the square differences between the generated view 2 feature output and the true view 2 feature vector for an instance. Let $\mathcal{P}$ denote the index set of instances for which the second view is present, and $l = |\mathcal{P}|$. Then the basic objective function including available second view data is given by Equation 2, where $\vec{f}_{j,i}$ is the output of the j^{th} layer on an input to the network of $\vec{x}_i^1$ for i in a given index set and $j = 1, \ldots, K$, where λ_1 controls a trade-off between fitting the labeled data well and fitting the available second view data well.

$$\begin{aligned} \text{argmin.}_{W_j, \vec{b}_j, \forall j} \quad & \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i f_{K,i})) \\ & + \frac{\lambda_1}{l} \sum_{i \in \mathcal{P}} ||\vec{f}_{K-1,i} - \vec{x}_i^2||_2^2 \end{aligned} \tag{2}$$

The new term is differentiable so standard gradient descent approaches are still applicable, and gradient computations are accomplished succinctly with basic matrix operations.

4.4 Biasing the Model for Multi-View Semi-Supervised Learning

In order to incorporate the aforementioned differing profile idea in estimating the neural network model, an additional term is added to the objective function of Equation 2, given in Equation 3. This term biases the learning, forcing the generated view to differ more in its labeled data instances' distances to unlabeled data for larger values of the regularization parameter λ_2. The idea is that this in turn can help the subsequent application of MVSSL methods to be effective by providing the opportunity to utilize differing confidence in label predictions for unlabeled data for different views, e.g., by propagating confident label predictions between views, as explained in Section 3.2.

$$-\frac{\lambda_2}{nmp_2}\sum_{i=1}^{n}\sum_{j=n+1}^{n+m}(||\vec{x}_i^1-\vec{x}_j^1||_2^2-||\vec{f}_{K-1,i}-\vec{f}_{K-1,j}||_2^2)^2 \quad (3)$$

Again this term fits within the backpropagation framework and allows computation with basic matrix operations.

Additionally, for huge amounts of unlabeled data a stochastic gradient approach can be used in estimating the unlabeled data profile distances - a sample of the unlabeled data in such cases could be used to estimate the difference in profiles, and thus a random sample could be taken at each gradient update.

The basic training and testing procedures for multi-view semi-supervised learning approaches combined with the proposed feature generation approach are given by Algorithms 1 and 2, respectively.

Algorithm 1 Training with the Feature Generation Network

Input: A set of data $\mathcal{S}$ containing (view 1, view 2, label) triplets, in which view 2 and labels may be missing for a given instance, initial weights and offsets $W_j, \vec{b}_j, \forall j$, a multi-view semi-supervised learning algorithm A which outputs a predictive function $f_A(\mathcal{S}) : \mathcal{X}^1 \times \mathcal{X}^2 \rightarrow \mathcal{Y}$ given complete training data. Additional parameters for the feature generation network, λ_1, λ_2, number of backpropagation iterations T, and whether or not to use only the generated view 2 data.

Output: Final weights and biases for the network $W_j, \vec{b}_j, \forall j$, and the trained predictor f_A.

-Use T iterations of gradient descent to find an approximate local solution to Equation 2 with Equation 3 added to the objective.

-Use the learned network $(W_j, \vec{b}_j, \forall j)$ from the previous step to generate view 3 for all instances in $\mathcal{S}$. Normalize the generated view 3 data.

-Fill in any missing view 2 instances of $\mathcal{S}$ with those from the previous step, the generated view 3; optionally replace non-missing view 2 instances with the generated ones as well. Denote the completed data $\hat{\mathcal{S}}$.

-Apply algorithm A to the completed multi-view semi-supervised data $\hat{\mathcal{S}}$ to obtain f_A.

Algorithm 2 Testing using the Feature Generation Network

Input: A set of data $\mathcal{R}$ containing (view 1, view 2) pairs, in which view 2 and may be missing for a given instance, a trained feature generation network $(W_j, \vec{b}_j, \forall j)$, and a trained predictive function $f : \mathcal{X}^1 \times \mathcal{X}^2 \rightarrow \mathcal{Y}$, and whether or not to use only the generated view 2 data.

Output: Predictions $y \in \mathcal{Y}$ for each instance of R.

-Use the trained network $(W_j, \vec{b}_j, \forall j)$ to fill in any missing view instances of $\mathcal{R}$ and optionally replace the available second view data; denote the completed data $\hat{\mathcal{R}}$.

-Apply f to each instance in $\hat{\mathcal{R}}$ to obtain the predicted y for that instance.

4.5 Connections to Modern Deep Network Approaches

The recent resurgence in interest in neural networks in the machine learning and data mining communities is the result of different interpretations of / assumptions about the networks; the models along with these new interpretations/assumptions are often referred to as "deep belief networks" due to a different generative probabilistic (i.e., belief) perspective being assigned to the multi-layer networks [13, 15, 19, 20, 28, 30, 35]. In general most modern approaches keep the same layered structures, and in terms of predictions and network outputs, in general the same feed-forward approach is used to generate layer and label outputs. Additionally backpropagation is commonly still used to fit the net to the data after pre-training. The key difference of the modern approaches are the assumptions of the underlying probabilistic models which can result in different pre-training strategies [13], for example, using layer-wise contrastive divergence [19] to pre-train networks layer-by-layer with unlabeled data. A key practical difference between past neural network methods and modern ones is in how the networks are pre-trained or initialized. Also, even standard neural network methods that do not use pre-training and just use the backpropagation have still been used recently to achieve state of the art performance [44]. Although our approach is for generating an additional, complementary set of features as opposed to replacing an existing one, this view generation problem could offer a new direction for work on deep network architectures, and our regularization terms could be viewed as additional ways to prevent overfitting with such architectures. An important component of our work is testing the combination of the deep belief network approach with our method, through pre-training the feature generation network.

5. EXPERIMENTAL STUDY

We test our method with synthetic and real data. For each experiment we report results in terms of test error if the data is balanced, and also Matthews Correlation Coefficient (MCC) and F1 Score if the data is unbalanced. Let tp denote the number of true positive predictions, fp the number of false positives, fn false negatives, and tn true negatives.

- Test error is given by: $\frac{fp+fn}{tp+tn+fp+fn}$.
- MCC is given by: $\frac{(tp)(tn)-(fp)(fn)}{\sqrt{(tp+fp)(tp+fn)(tn+fp)(tn+fn)}}$.
- F1 Score is given by: $\frac{2tp}{2tp+fn+fp}$.

Note that MCC and F1 score attain their best values at 1, and test error at 0, and MCC takes into account both false positive and false negative rates whereas F1 score does not take into account the false negative rate.

We compare our method CoNet with two state-of-the-art methods. The first method has the claim of being the first approach to handle missing view data in the MVSSL setting, Gaussian process co-regularization (GPCR) [45]. The second is the most recent approach to applying MVSSL to the single view case (completely missing second view - i.e., whatever second view data is available is ignored) and reported state-of-the-art results - pseudo multi-view co-training (PMC) [7]. We obtained the code for PMC from the authors, and used the "Gaussian Processes for Machine Learning Toolbox" version 3.1 [32] to implement GPCR. Note that for our experiments in general we cannot apply basic multi-view semi-supervised learning methods not designed to handle missing view data, such as co-training, as baselines. This is because view 2 is missing at random and may not be present even in the labeled data, or if it is it may only be present for one class due to the often highly imbalanced nature of the data. Additionally we compare with the baseline of only using the single omnipresent (first) view, using a Gaussian process classifier with this view (View 1 GP) [33]. For all methods, we use the same logistic loss model for

fair comparison. PMC uses logistic regression models for the base classifiers, and we use logistic likelihood models in GPCR and in a Gaussian process classifier for the view 1 only baseline (View 1 GP). For the MVSSL algorithm used by CoNet we use either GPCR with logistic likelihood or co-training with L_1 regularized logistic regression classifiers as the base models. To simplify the experiments we choose either co-training or GPCR as the MVSSL algorithm used by CoNet based on which gave the best MCC when no second view data is available.

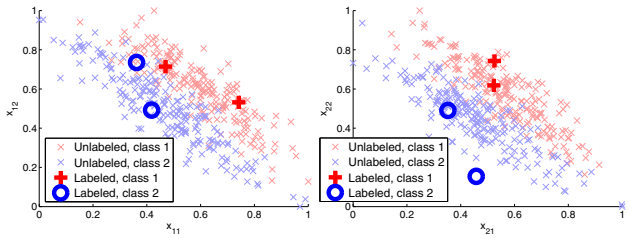
Additionally to allow straight-forward comparison with the GPCR method, all of our experiments are carried out in a transductive setting, i.e., the unlabeled data (or some portion of it) for a given trial also corresponds to the test data. Note that CoNet itself is not restricted to a transductive setting. For the real data experiments, we perform experiments for CoNet with both random initialization and the contrastive divergence pre-training and also both filling in ("fill") and not filling in ("no fill") the second view with the observed second view for instances when it is available (observed). For the CoNet methods we fix the number of backpropagation gradient descent iterations to 100. For all methods we report the results for the parameters giving the best average performance, where averages are taken across 100 or more random splits of the data, which essentially corresponds to reporting results of model selection if labels were available for some or all of the unlabeled data. Thus we avoid the model selection issue which is common practice in this type of scenario (e.g., [2, 5, 23, 37, 38]), and essentially shows the results achievable given an ideal model selection method for the scenario. Since there is usually a very limited amount of labeled training data in the MVSSL setting, standard model selection approaches like cross-validation often fail [36], so the common procedure of reporting subsequent performance after model selection would not be at all representative of the underlying methods' performances but rather of the (poor) performance of the model selection approach used. Model selection in this scenario is still an open problem [17]. We discuss the model selection issue in more detail and alternative model selection approaches in a technical report [29].

5.1 Synthetic Data Experiment

We present results for an illustrative 2D data experiment, for the task of learning a function to separate two overlapping sets of Gaussian-distributed data. Data for two views was generated independently from the same Gaussian distribution for each class. In this way the two views come from the same distribution, but are conditionally independent given the class label - an ideal scenario for multi-view semi-supervised learning algorithms. We vary the mean fraction of second view data available from 0% to the ideal case of 100%, by removing each data instance from the second view completely at random with fixed probability corresponding to each fraction. For each trial, 2 labeled training points and 200 unlabeled points, were generated for each class using the two Gaussian distributions. Figure 3 shows a sample of the generated data in each view.

This data set demonstrates a simple case where existing single-view approaches are generally not well-suited. In this case, feature-splitting cannot be effective since both features are needed for sufficiency; splitting the features would result in different data classes largely overlapping in both views. Additionally there are no clear clusters - the marginal distributions look similar to unimodal groupings of points.

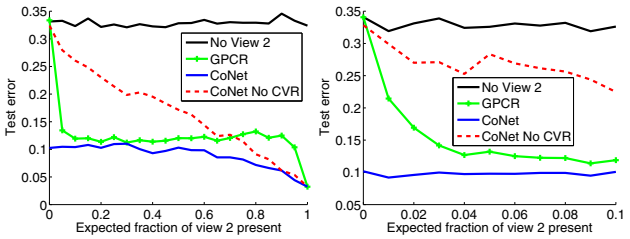
We choose the state-of-the-art Gaussian process co-regularization algorithm [45] as the base algorithm to be applied after filling in the missing views with our CoNet method. In addition we use the version of this algorithm that can handle missing views to compare our method with, as it is the state-of-the-art approach [45]. In addition we report results for comparing with a view-mapping approach - an approach that only directly tries to learn a mapping from view 1 to view 2 using the available data. This corresponds to using our same feature generation network approach to generate the second view, without using the proposed bias, corresponding to Equation 2.



(a) View 1 (b) View 2

Figure 3: Sample of two views of data generated for an ideal 2D test case

First we varied the mean fraction of second view data available from 0.0 to 1.0 in increments of 0.05. The experiment was repeated for 200 random samples of the data, and average test error and standard deviation is reported in Table 1 and Figures 4 and 4(b).



(a) From 0 (no view 2) to 1 (all) (b) Zoomed in (0.0 to 0.1)

Figure 4: Test error vs. mean fraction of view 2 present for the 2-Gaussian data set

Table 1: Mean ± std. dev. of test error from 200 trials for each method on the 2-Gaussian data, for 0% second view data available.

View1 GP	PMC	GPCR	CoNet
0.331±0.125	0.442±0.075	0.334±0.125	**0.103±0.058**

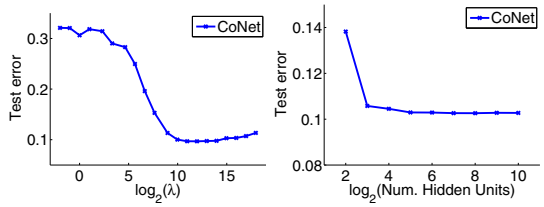


Figure 5: Performance criteria vs. contrasting view regularization parameter and vs. number of hidden units in hidden layer 1 for 0% second view data for the 2-Gaussian data set

The proposed feature generation approach was found to perform significantly better than using the same base classifier with a single

view of the data, or using the state-of-the-art GPCR method, especially in two extreme ranges of having very little view 2 data, and having close to the amount of view 2 data needed to achieve the best performance. Additionally without the contrasting view regularization (CVR) term, and with the exact same network structure and approach to initialization and training, the feature generation approach ("CoNet CVR") took much more view 2 data to come close to the same level of performance as CoNet. We also show the results of repeating the experiment zoomed in more closely on the beginning region, this time varying the mean fraction of view 2 data present from 0.0 to 0.1 in increments of 0.01. The results are shown in Figure 4.

Furthermore, the results for the single view case - i.e., no view 2 data available are shown in Table 1, here also compared with the state-of-the-art single view method, pseudo-multi-view co-training (PMC). In this case PMC fails because the features cannot be partitioned in such a way to form sufficient views - in this case both features are needed to separate the classes well. This highlights the need for a more complex mechanism to generate the new view from the existing ones, which CoNet provides.

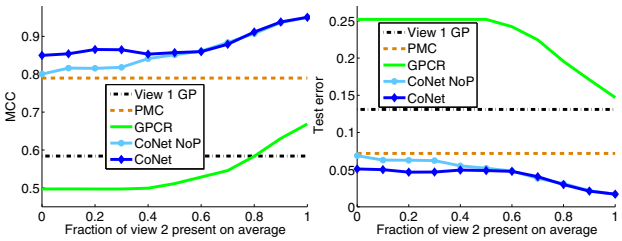
5.2 WebKB Course Data Experiment

The WebKB Course data set is a collection of 1051 websites from four universities, belonging to two categories: course websites or non-course websites. There are 230 websites in the course category, and 821 in the non-course category, making the data set unbalanced. The first view consists of text on the webpage itself, the second view consists of the link text of links from other webpages linking to the webpage. We use co-training as the base MVSSL algorithm to be used after filling in the missing views with CoNet for this data set.

We obtained the webpage and link text data[2] then applied standard text pre-processing using Weka [18] to obtain 2,168 features in the text view and 338 features in the link view. As in [5], for each experiment iteration we randomly sample 3 course and 9 non-course instances for labeled training. The remaining instances were used for the unlabeled data and also testing - a transductive setting so that we could compare with GPCR. We then varied the mean fraction of second view data available from 0.0 to 1.0 in increments of 0.1. Here the second view is missing completely at random - that is for a given fraction, each view 2 instance is present with probability given by that fraction. We repeated the experiment 100 times for each fraction value and report the mean results. For the base classifier for co-training we used $L1$ regularized logistic regression, with the regularization parameters set to 0.001 for view 1 and 0.01 for view 2 throughout since these worked well for basic co-training when view 2 was completely available - though as long as these values were not too large (less than 1) the performance stayed basically the same. For the comparison state-of-the-art methods GPCR and PMC we varied all of the parameters by powers of 10 and report the results for the best set of parameters in each case.

5.3 Results - WebKB Course

The overall results for the Course data are shown in Figure 6. This plot shows CoNet with pre-training (denoted as "CoNet") and without pre-training (denoted as "CoNet NoP") compared with the other methods for varying amounts of expected fraction of view 2 data present (observed), from no view 2 data (0.0) to all view 2 data (1.0). The results for F1 score follow the exact same trend as for MCC so we do not show them here. Again the other methods are the Gaussian process classifier with the single view ("View 1 GP") [33], the state-of-the-art Gaussian process co-regularization (GPCR) [45], and the state-of-the-art single view method, pseudo-multi-view co-training (PMC) [7]. GPCR required significantly more view 2 data to perform better than single view learning for this data. However CoNet was able to take advantage of the available second view data, obtaining the best performance. Also, in this case using pre-training resulted in a significant improvement for CoNet when limited view 2 data was available.



(a) MCC vs. fraction present (b) Test error vs. fraction present

Figure 6: MCC and Test error vs. mean fraction of view 2 present for the WebKB Course data set

In Table 2 we show the effect each component of CoNet has, and also the difference between filling in cases with available view 2 data (denoted "fill") and using only the generated view 2 data (denoted "no fill"). That is we correspondingly fix one or both of λ_1 and λ_2 to 0, i.e., "No Reg" corresponds to both fixed to 0, "VMR Only" to $\lambda_2 = 0$, and "CVR Only" to $\lambda_1 = 0$. We show results for the version of CoNet with pre-training and only for MCC, but the other performance criteria have similar trends, and the trends for no pre-training are also similar except that using the available view 2 data becomes the better strategy sooner, at the fraction of 0.5. Note that for fraction present equal to 0.0, the "fill" and "no-fill" results are the same since there are no available view 2 instances to fill in, and for 1.0 since view 2 is present for all instances all "fill" results are the same.

From these results we observe a general trend - at first, with less view 2 data available (observed), using the generated view 2 as opposed to filling in the real view is more effective, and further the contrasting view component is more important. As more view 2 data becomes available, so that a better mapping to view 2 can be learned, then filling in the available view 2 data becomes the better strategy, and the view-matching component becomes more important. Usually both components are needed for CoNet to achieve its best performance, and in most cases one or both components have a significant effect on performance. For the case of limited view 2 data one reason that filling in the available view 2 data does not help might be that the generated view 2 data is very different from the available view 2 data since there is not yet enough to learn a very accurate view mapping function. Another reason using the real view 2 where available becomes a better strategy as more view 2 data is observed is because the real view 2 data has built-in the desirable properties for MVSSL methods, e.g., of sufficiency for classification, whereas for the generated view we can only estimate these properties.

5.4 Chemical Toxicity Data Experiment

We next evaluated these methods on a chemical toxicity prediction task using a data set from the Environmental Protection Agency (EPA) TOXCAST program [21] (`http://www.epa.gov/ncct/toxcast/`) which includes experimental results con-

[2]Available here: `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-51/www/co-training/data/`

Table 2: Mean ± std. dev. of MCC from 100 trials for each method on the WebKB Course data, for varying amounts of average second view data available in fraction of all data instances. Comparison for the case of using pre-training and both the view-matching and contrasting view components ("CoNet") with neither component ("No Reg."), just the view-matching component ("VMR Only") and just the contrasting view component ("CVR Only"). The first half, "fill" corresponds to filling in cases with available view 2 data, i.e., using whatever view 2 data is available and "no fill" to using only the generated view 2 data.

		0.0	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
fill	**CoNet**	**0.850** ±0.126	0.791 ±0.093	0.761 ±0.102	0.747 ±0.133	0.717 ±0.159	0.764 ±0.165	0.811 ±0.137	**0.879** ±0.062	**0.912** ±0.019	**0.938** ±0.017	**0.950** 0.010
	No Reg.	0.832 ±0.092	0.685 ±0.169	0.648 ±0.176	0.654 ±0.188	0.655 ±0.176	0.630 ±0.186	0.629 ±0.203	0.643 ±0.182	0.680 ±0.175	0.805 ±0.083	**0.950** 0.010
	VMR Only	0.832 ±0.092	0.690 ±0.159	0.643 ±0.183	0.661 ±0.174	0.634 ±0.181	0.698 ±0.228	0.732 ±0.226	0.801 ±0.069	0.910 ±0.020	0.937 ±0.014	**0.950** 0.010
	CVR Only	**0.850** ±0.126	0.789 ±0.103	0.753 ±0.122	0.743 ±0.146	0.712 ±0.173	0.702 ±0.223	0.736 ±0.235	0.848 ±0.149	0.881 ±0.095	0.874 ±0.074	**0.950** 0.010
no fill	**CoNet**	**0.850** ±0.126	**0.854** ±0.111	**0.865** ±0.066	**0.865** ±0.045	**0.853** ±0.120	**0.857** ±0.104	**0.860** ±0.099	0.857 ±0.111	0.856 ±0.109	0.856 ±0.111	0.858 0.105
	No Reg.	0.832 ±0.092	0.838 ±0.068	0.837 ±0.110	0.834 ±0.092	0.835 ±0.091	0.834 ±0.092	0.834 ±0.090	0.832 ±0.090	0.834 ±0.089	0.832 ±0.092	0.835 0.093
	VMR Only	0.832 ±0.092	0.834 ±0.094	0.836 ±0.089	0.814 ±0.102	0.830 ±0.100	0.837 ±0.065	0.837 ±0.077	0.834 ±0.085	0.834 ±0.088	0.837 ±0.085	0.838 0.083
	CVR Only	**0.850** ±0.126	0.843 ±0.134	0.858 ±0.101	0.849 ±0.126	0.852 ±0.119	0.850 ±0.125	0.851 ±0.119	0.850 ±0.126	0.852 ±0.120	0.851 ±0.119	0.851 0.125

ducted on 309 unique chemical pesticides. In vitro tests were performed with 624 different assays - we take the results of these tests as the feature set for the second view. Since both the animal toxicity endpoints and the in vitro second view data are time consuming and expensive to obtain (e.g. the study cost millions of dollars and took more than a year), this data set fits the MVSSL with partially observed views scenario well. After basic pre-processing, e.g., removing duplicates and compounds with missing or inconclusive endpoint results, the data set consists of 225 chemical compounds with 597 view 2 features. For the class label we took the toxicity endpoint of "tumors on mouse liver", resulting in 68 positive and 157 negative instances so this data set is also imbalanced. To obtain a large set of related unlabeled data, we searched the PubChem database (`http://pubchem.ncbi.nlm.nih.gov/`) for all compounds with the keyword "pesticide" or "herbicide," resulting in an additional 1262 compounds added to the data set. To obtain the common, readily-available view 1, we extracted numerical chemical descriptors from the full set of compounds using the DRAGON software (version 5) [40] for the atom-centered fragment descriptors, resulting in a total of 103 features in view 1. For each trial, we randomly sampled half of the labeled data to be used as training data, and the other half to be included with all of the unlabeled data and for testing. Since only those data instances from the original TOXCAST collection have the second view available, the maximum obtainable fraction of view 2 data present is only approximately 0.15. Therefore for this data set we only tested two cases: no view 2 data (labeled fraction present of 0.0) and all available view 2 data (labeled fraction present of 0.15). For this data set we use GPCR as the MVSSL algorithm used by CoNet.

5.5 Results - Chemical Toxicity

The results for the chemical toxicity data are summarized in Table 3. For this data set, unlike the text data set, using pre-training for the network (denoted as "CoNet") was somewhat detrimental to performance compared to the randomly initialized net (CoNet NoP). Aside from the type of data (e.g., chemical descriptors as opposed to images or text), this may also be due to overfitting of the generative model since there are many more features in view 2 than view 1 in this case. Further improvement may be possible by more thorough experimentation with the pre-training approach used.

Although PMC achieves slightly lower test error than the CoNet methods, it has significantly worse scores under the balanced performance criteria (MCC and F1 score) which are more indicative of efficacy for this data. The results indicate that essentially the method cannot detect the positive cases well but still has low test error due to the highly imbalanced nature of the data. On the other hand CoNet scores highly under the more balanced performance criteria, and still manages to reach nearly the same test error in the case of the small amount of partial view data available. This is similar when CoNet (NoP - the no pre-training version - in particular) is compared with the other methods. With respect to MCC, arguably the most balanced criterion, CoNet obtains significantly better performance compared to all other methods. With respect to F1 score, the single view GP classifier has a slightly better score for the expected fraction of 0.0 view 2 data present and GPCR has a slightly better score for the fraction of 0.15. However these are not significantly different from the CoNet NoP scores. To give an idea of how the methods compare under the different criteria, we show the results of ANOVA with multi-comparison tests in Table 4. An entry of "1" indicates a significant difference in the means of the given performance criterion for the two methods at the five percent level.

Table 4: ANOVA multi-comparison test results for each of MCC, F1 score, and test error criteria on the Chemical Toxicity data, for 0.15 fraction of view 2 data present. A "1" indicates significant difference in mean between the two methods at the 5 percent level.

	MCC					**F1 Score**					**Test Error**				
	View 1 GP	**PMC**	**GPCR**	**CoNet NoP**	**CoNet**	**View 1 GP**	**PMC**	**GPCR**	**CoNet NoP**	**CoNet**	**View 1 GP**	**PMC**	**GPCR**	**CoNet NoP**	**CoNet**
View 1 GP	0	1	1	1	0	0	1	0	0	1	0	1	1	1	1
PMC	1	0	1	1	1	1	0	1	1	1	1	0	1	1	1
GPCR	1	1	0	1	1	0	1	0	0	1	1	1	0	1	1
CoNet NoP	1	1	1	0	1	0	1	0	0	1	1	1	1	0	1
CoNet	0	1	1	1	0	1	1	1	1	0	1	1	1	1	0

Table 5 shows the comparison between CoNet with no pre-training (NoP) with both view matching regularization (VMR) and contrasting view regularization (CVR) and with one or neither, corresponding to setting the appropriate parameter/s to 0. For this data

Table 3: Mean ± std. dev. of MCC, F1 score, and test error from 100 trials for each method on the Chemical Toxicity data, for varying amounts of average second view data available in fraction of all data instances.

	MCC		F1 Score		Test Error	
	0.0	**0.15**	**0.0**	**0.15**	**0.0**	**0.15**
View 1 GP	0.122±0.077	0.122±0.077	**0.456±0.041**	0.456±0.041	0.470±0.042	0.470±0.042
PMC	0.054±0.084	0.054±0.084	0.272±0.081	0.272±0.081	**0.359±0.032**	**0.359±0.032**
GPCR	0.113±0.078	0.159±0.085	0.417±0.055	**0.468±0.049**	0.415±0.035	0.433±0.041
CoNet NoP	**0.150±0.084**	**0.188±0.079**	0.440±0.057	0.463±0.055	0.396±0.040	0.378±0.038
CoNet	0.114±0.081	0.132±0.074	0.425±0.055	0.426±0.053	0.425±0.044	0.389±0.034

Table 5: Mean ± std. dev. of MCC, F1 score, and test error from 100 trials for the CoNet method on the chemical toxicity data. Comparison for the case of using no pre-training and both the view-matching and contrasting view components ("CoNet") with neither component ("No Reg."), just the view-matching component ("VMR Only") and just the contrasting view component ("CVR Only"). The first half, "fill" corresponds to filling in cases with available view 2 data, i.e., using whatever view 2 data is available and "no fill" to using only the generated view 2 data.

		MCC		F1 Score		Test Error	
		0.0	**0.15**	**0.0**	**0.15**	**0.0**	**0.15**
fill	**CoNet NoP**	**0.150±0.084**	**0.188±0.076**	**0.440±0.057**	**0.463±0.053**	**0.396±0.040**	**0.378±0.035**
	No Reg.	0.091±0.079	0.157±0.079	0.405±0.055	0.436±0.057	0.426±0.038	0.382±0.035
	VMR Only	0.091±0.079	0.157±0.080	0.405±0.055	0.436±0.057	0.426±0.038	0.382±0.036
	CVR Only	**0.150±0.084**	0.171±0.079	**0.440±0.057**	0.458±0.054	**0.396±0.040**	0.394±0.035
no fill	**CoNet NoP**	**0.150±0.084**	0.168±0.073	**0.440±0.057**	0.451±0.053	**0.396±0.040**	0.387±0.033
	No Reg.	0.091±0.079	0.147±0.086	0.405±0.055	0.434±0.060	0.426±0.038	0.392±0.039
	VMR Only	0.091±0.079	0.137±0.087	0.405±0.055	0.427±0.062	0.426±0.038	0.397±0.039
	CVR Only	**0.150±0.084**	0.148±0.078	**0.440±0.057**	0.446±0.053	**0.396±0.040**	0.407±0.035

including both components was necessary to achieve the best performance.

6. CONCLUSION

An obstacle for multi-view semi-supervised learning approaches when applied to real world data is the lack of complete multiple view data. For example, a common scenario is that one data view is readily and cheaply available, but additional views may only be available in some cases and may be costly to obtain. Current work to address such scenarios is limited and also each previous approach has some limitations. In summary, existing approaches either are not able to incorporate partial view information when available or are not applicable or effective with limited amounts of additional view data. Additionally, the previous works either make restrictive assumptions, are method-dependent, or fail to incorporate a way of enforcing the approach to be useful for subsequent application of multi-view semi-supervised learning algorithms. To address these limitations, we introduced a unified approach for multi-view semi-supervised learning with missing views that can be applied to the full range of problems with incomplete view information. We propose a feature-generation learning approach, based on fine-tuning random nonlinear feature functions, for learning a mapping to fill in missing views, with a particular bias incorporated that is motivated by theoretical results on multi-view semi-supervised learning. This is carried out using additional terms in the objective function of a feature generation network model that encourages the data instances in distinct views to be nearby different unlabeled instances. We demonstrated the efficacy of our method with synthetic and real data experiments and for these experiments our method achieved superior performance to two recent state-of-the-art approaches designed for the case of MVSSL with missing views.

7. ACKNOWLEDGMENTS

This work has been supported by the National Science Foundation under Grant No. 0845951 and a Graduate Research Fellowship award for B.Q.

8. REFERENCES

[1] M. Amini, N. Usunier, and C. Goutte. Learning from multiple partially observed views–an application to multilingual text categorization. *Advances in neural information processing systems*, 23, 2009.

[2] R. R. Ando and T. Zhang. Two-view feature generation model for semi-supervised learning. In *Proceedings of the 24th international conference on Machine learning*, pages 25–32. ACM, 2007.

[3] M. F. Balcan, A. Blum, and K. Yang. Co-training and expansion: Towards bridging theory and practice. *Advances in neural information processing systems*, pages 89–96, 2005.

[4] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399–2434, 2006.

[5] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[6] U. Brefeld, C. Büscher, and T. Scheffer. Multi-view discriminative sequential learning. *Proceedings of the European Conference on Machine Learning*, pages 60–71, 2005.

[7] M. Chen, K. Weinberger, and Y. Chen. Automatic feature decomposition for single view co-training. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 953–960, New York, NY, USA, June 2011. ACM.

[8] N. Chen, J. Zhu, and E. P. Xing. Predictive subspace learning for multi-view data: a large margin approach. In *Advances in neural information processing systems 24*, 2010.

[9] C. M. Christoudias, R. Urtasun, and T. Darrell. Multi-view learning in the presence of view disagreement. In *Proceedings of the 24th conference on Uncertainty in Artifical Intelligence*, 2008.

[10] A. Coates, H. Lee, and A. Ng. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[11] M. Culp and G. Michailidis. A co-training algorithm for multi-view data with applications in data fusion. *Journal of chemometrics*, 23(6):294–303, 2009.

[12] M. Culp, G. Michailidis, and K. Johnson. On multi-view learning with additive models. *The Annals of Applied Statistics*, 3(1):292–318, 2009.

[13] D. Erhan, Y. Bengio, A. Courville, P. Manzagol, P. Vincent, and

S. Bengio. Why does unsupervised pre-training help deep learning? *The Journal of Machine Learning Research*, 11:625–660, 2010.

[14] J. D. R. Farquhar, D. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmak. Two view learning: SVM-2K, theory and practice. *Advances in neural information processing systems*, 18:355, 2006.

[15] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, 2011.

[16] S. Goldman and Y. Zhou. Enhancing supervised learning with unlabeled data. In *Proceedings of the 17th International Conference on Machine Learning*, 2000.

[17] I. Guyon, A. Saffari, G. Dror, and G. Cawley. Model selection: Beyond the bayesian/frequentist divide. *The Journal of Machine Learning Research*, 11:61–87, 2010.

[18] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[19] G. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[20] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006.

[21] R. Judson, K. Houck, R. Kavlock, T. Knudsen, M. Martin, H. Mortensen, D. Reif, D. Rotroff, I. Shah, A. Richard, et al. In Vitro Screening of Environmental Chemicals for Targeted Testing Prioritization: The ToxCast Project. *Environmental health perspectives*, 118(4):485–492, 2010.

[22] Y. Kang and S. Choi. Restricted deep belief networks for multi-view learning. In *Proceedings of the ECML/PKDD 2011*, 2011.

[23] G. Li, S. C. H. Hoi, and K. Chang. Two-view transductive support vector machines. In *Proceedings of the SIAM International Conference on Data Mining*, 2010.

[24] B. M. Marlin. *Missing data problems in machine learning*. PhD thesis, University of Toronto, 2008.

[25] I. Muslea, S. Minton, and C. A. Knoblock. Active learning with multiple views. *Journal of Artificial Intelligence Research*, 27(1):203–233, 2006.

[26] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 689–696. ACM, June 2011.

[27] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 86–93, 2000.

[28] H. Poon and P. Domingos. Sum-product networks: A new deep architecture. In *Proc. 12th Conf. on Uncertainty in Artificial Intelligence*, pages 337–346, 2011.

[29] B. Quanz and J. Huan. Model selection for semi-supervised learning with limited labeled data. Technical Report ITTC-FY2013-TR-65071-01, Information Telecommunication and Technology Center, University of Kansas, Lawrence, KS, July 2012.

[30] M. Ranzato, J. Susskind, V. Mnih, and G. Hinton. On deep generative models with applications to recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2857–2864. IEEE, 2011.

[31] B. Raskutti, H. Ferrá, and A. Kowalczyk. Combining clustering and co-training to enhance text classification using unlabelled data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 620–625. ACM, 2002.

[32] C. E. Rasmussen and H. Nickisch. GPML: Gaussian processes for machine learning toolbox. `http://www.gaussianprocess.org/gpml/code/matlab/doc/`, 2010.

[33] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[34] D. Rosenberg and P. Bartlett. The rademacher complexity of co-regularized kernel classes. In *Proceedings of Artificial Intelligence & Statistics*, 2007.

[35] R. Salakhutdinov and G. Hinton. Deep Boltzmann machines. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 5, pages 448–455, 2009.

[36] V. Sindhwani, W. Chu, and S. Keerthi. Semi-supervised gaussian process classifiers. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1059–1064, 2007.

[37] V. Sindhwani, P. Niyogi, and M. Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *Workshop on Learning with Multiple Views, International Conference on Machine Learning*, 2005.

[38] V. Sindhwani and D. S. Rosenberg. An RKHS for multi-view learning and manifold co-regularization. In *Proceedings of the 25th international conference on Machine learning*, pages 976–983. ACM, 2008.

[39] S. Szedmak and J. Shawe-Taylor. Synthesis of maximum margin and multiview learning using unlabeled data. *Neurocomputing*, 70(7-9):1254–1264, 2007.

[40] Talete srl. *DRAGON (Software for Molecular Descriptor Caluclations)*. Talete srl, Milano, Italy, 2007. http://www.talete.mi.it/.

[41] W. Wang and Z. H. Zhou. Analyzing co-training style algorithms. In *Proceedings of the 18th European Conference on Machine Learning*, pages 454–465. Springer-Verlag New York Inc, 2007.

[42] W. Wang and Z. H. Zhou. Multi-view active learning in the non-realizable case. In *Neural Information Processing Systems*, 2010.

[43] W. Wang and Z. H. Zhou. A new analysis of co-training. In *Proceedings of the 27th international conference on Machine learning*, 2010.

[44] J. Weston, F. Ratle, and R. Collobert. Deep learning via semi-supervised embedding. In *Proceedings of the 25th international conference on Machine learning*, pages 1168–1175. ACM, 2008.

[45] S. Yu, B. Krishnapuram, R. Rosales, and R. B. Rao. Bayesian co-training. *Journal of Machine Learning Research*, 12:2649–2680, Sep. 2011.

[46] S. Yu, B. Krishnapuram, R. Rosales, H. Steck, and R. B. Rao. Bayesian co-training. *Advances in neural information processing systems*, 20:1665–1672, 2008.

[47] D. Zhou and C. J. C. Burges. Spectral clustering and transductive learning with multiple views. In *Proceedings of the 24th international conference on Machine learning*, pages 1159–1166. ACM, 2007.

[48] Z. H. Zhou and M. Li. Tri-training: exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, pages 1529–1541, 2005.

[49] Z.-H. Zhou and M. Li. Semi-supervised regression with co-training style algorithms. *IEEE Trans. on Knowl. and Data Eng.*, 19:1479–1493, November 2007.