AN ABSTRACT OF THE THESIS OF

Gaole Jin for the degree of Master of Science in Electrical and Computer Engineering presented on December 3, 2012.

Title: On Surrogate Supervision Multi-view Learning

Abstract approved: _____

Raviv Raich

Data can be represented in multiple views. Traditional multi-view learning methods (i.e., co-training, multi-task learning) focus on improving learning performance using information from the auxiliary view, although information from the target view is sufficient for learning task. However, this work addresses a semi-supervised case of multi-view learning, the surrogate supervision multi-view learning, where labels are available on limited views and a classifier is obtained on the target view where labels are missing. In surrogate multi-view learning, one cannot obtain a classifier without information from the auxiliary view. To solve this challenging problem, we propose discriminative and generative approaches.

On Surrogate Supervision Multi-view Learning

by

Gaole Jin

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented December 3, 2012
Commencement June 2013

Master of Science thesis of Gaole Jin presented on December 3, 2012.

APPROVED:

Major Professor, representing Electrical and Computer Engineering

---

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Gaole Jin, Author

ACKNOWLEDGEMENTS

My most sincere thanks go to my advisor, Dr. Raviv Raich. I thank Dr. Raich for his expertise, time, and patience in assisting me finishing my Master of Science program and researches. Dr. Raich's support and encouragement are very important for me to finish this work. I also would like to thank Dr. Xiaoli Fern, Dr. Sinisa Todorovic, Dr. Bill Smart for serving as committee members in my defence., and Dr. Yevgeniy Kovchegov for attending my defence.

I thank my lab mates: Qi, Behrouz, Greg, Zeyu, Evgenia, Balaji, and Deepthi. Thank them for creating a great atmosphere in the lab. I also thank all my friends in China and in USA. I enjoyed the time I spent with all my friends.

Finally, I would like to thank my parents. Their support for me is the greatest gift anyone has ever given to me. I would never make it without their love and support.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

Chapter 1: INTRODUCTION

1.1 Background

It is common to learn a classifier from a set of examples of the form (feature vector, label). Feature vectors can be represented as a vector the form $[x_1, \cdots, x_d]^T$. However, in multi-view learning we consider subsets of the feature vectors $[x_1, \cdots, x_s]^T$ and $[x_{s+1}, \cdots, x_d]^T$ where $s < d$ as multiple views, and are able to learn separate classifiers on each view. For example, in a lip-reading task the sets of data from visual and audio information are naturally considered as two subsets of the feature vectors (Fig.1.1). Another example, one can consider representations of the same documents in different languages as multiple views (Fig.1.2).

Co-training is a semi-supervised multi-view learning technique aiming at improving performance of a learning algorithm by expanding labeled training data using information from multiple views [1]. For example, in [2] a set of labeled two-view examples $\{(x, z, y)\}$ and a set of unlabeled two-view examples $\{(x, z)\}$ are available. An assumption in [2] is that data from each of the views is sufficient for training an accurate classifier if labeled data are sufficient on both views. In [2], two classifiers are iteratively trained on two sets of pairs $\{(x, y)\}_i$ and $\{(z, y)\}_i$ respectively which come from the available triplets $\{(x, z, y)\}_i$ to label the unlabeled examples with the most confident classifier.

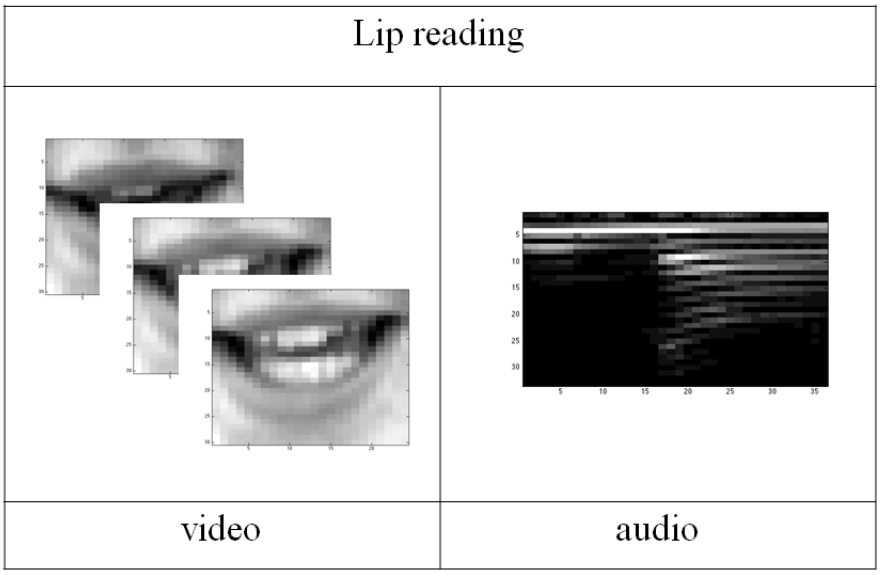


Figure 1.1: Audio and video in a lip reading task can be considered as two views.

The goal of transfer learning is to improve the learning performance on the target view using information from the auxiliary view [3]. Regularized multi-task learning is a supervised learning example of transfer learning, where labeled data $\{(x, y)\}_i$ and $\{(z, y)\}_i$ are available on both views [4]. Regularized multi-task learning algorithm transfers information from auxiliary view to the target view to improve classification performance on the target view [4]. Instead of improving classification performance, self-taught clustering aims at clustering a small collection of target unlabeled data with the help of a large amount of auxiliary unlabeled data [5]. Self-taught clustering algorithm clusters the target and auxiliary data simultaneously to allow the feature representation from the auxiliary data to influence the target data through a common set of features [5].

In this work, we focus on a special case of multi-view learning, for which we in-

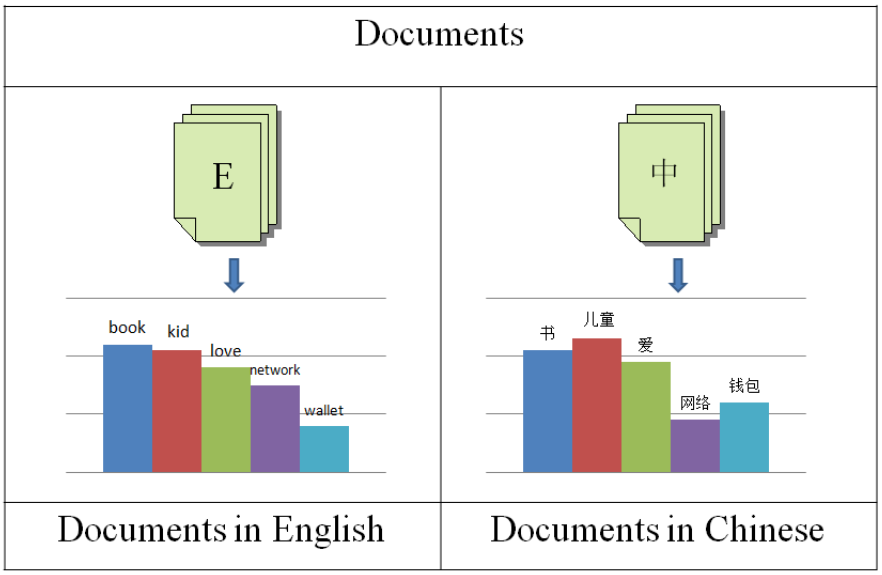


Figure 1.2: An English version and a Chinese version of the same document can be considered as two views.

troduce the term surrogate supervision multi-view learning, which aims at performing classification task on target view where labels are unavailable. In surrogate supervision learning, two mutually exclusive sets of pairs $\{(x_i, z_i)\}_{i \in S_1}$, $\{(x_i, y_i)\}_{i \in S_2}$ are available in the training data. However, the goal of surrogate supervision multi-view learning is to learn a classifier to predict y given z. The surrogate supervision multi-view learning scenario is different from the scenarios of [6] and [2] in that labeled examples of the desired view are unavailable in the training data. In other words, it is impossible to obtain a classifier that predicts labels for data from $\mathcal{Z}$ without using information from $\mathcal{X}$. Details of the surrogate supervision multi-view learning setting are introduced in Chapter 3. In this work, we solve the surrogate supervision multi-view learning problem using discriminative generative models.

1.2 Outline of Thesis

In this chapter, we introduce the background of surrogate supervision multi-view learning and show the roadmap of this thesis. In Chapter 2, we look at relevant literature on multi-view learning.

Chapter 3 describes the details of the problem of surrogate supervision multi-view learning. Two alternative solutions of surrogate supervision multi-view learning, the CCA+SVM approach and the label-transferred learning approach, are explained in Chapter 4.

Chapters 5, 6, and 7 are the core contributions of this thesis. Chapters 5 and 6 propose two solutions based on discriminative model to the surrogate supervision multi-view learning. Chapter 5 proposes the C^4A algorithm which combines the relationship learning (between views) stage and classifier training stage into a single stage. Chapter 6 proposes a hinge loss upper bound and the SSM-SVM algorithm based on that bound. Chapter 7 proposes a generative model solution to the surrogate supervision multi-view learning problem.

In Chapter 8, we apply the proposed algorithms in this work to a real world problem – speech recognition using both audio and video information. We also test the proposed algorithms in synthetic data. In Chapter 9, the thesis is concluded.

Chapter 2: LITERATURE REVIEW

Learning from multiple views is used to improve performance in learning tasks. In [5], Dat et al. propose the self-taught clustering algorithm that clusters points on the target view with help from the auxiliary view by learning a common space shared by the two views. Multi-task learning performs classification task one the target space by incorporating information from other view [6, 7, 4]. Co-training is a semi-supervised learning method in multi-view scenario. For example, in [8] one makes the assumptions that 1) the two views $\mathcal{X}$ and $\mathcal{Z}$ are conditionally independent, and that 2) either view $\mathcal{X}$ or $\mathcal{Z}$ is sufficient to predict $y \in \mathcal{Y}$ which is the label. In [8], the proposed algorithm first obtains two independent classifier from $\mathcal{X}$ and $\mathcal{Z}$ respectively. Then, the labels are generated by the most confident predictions of the two classifiers. In [2], the authors use co-training to classify web pages by topics. Co-training is also applied in [9] for cross-lingual sentiment classification and in [10] for email classification to improve performance of classification.

The surrogate supervision multi-view learning seeks a classifier on the space where no labled examples are available with help from a space where labeled examples are available. An intuitive solution to surrogate supervision multi-view learning problem is to transfer a classifier learned on the view $\mathcal{X}$ where labeled data is available. This naturally leads us to use canonical correlation analysis to analyze the relationship between the two views $\mathcal{X}$ and $\mathcal{Z}$. The CCA algorithm

maps data from two views $\mathcal{X}$ and $\mathcal{Z}$ to a common space. The CCA algorithm can be formulated as follows:

$$\begin{aligned} &\min_{a,b} \frac{1}{n} \sum_{i=1}^{n} ||a^T x_i - b^T z_i||_2^2 \\ &\text{subject to} \quad a^T R_X a = 1,\ b^T R_Z b = 1, \end{aligned} \tag{2.1}$$

where $R_X = \frac{1}{n} \sum_i x_i x_i^T$ and $R_Z = \frac{1}{n} \sum_i z_i z_i^T$. For simplicity, we assume that both the x_is and the z_is are zero mean. For details of canonical correlation analysis, the reader can refer to Appendix A.

The CCA algorithm has been widely applied in many areas. In [11], CCA is used in object recognition task. In [12], the relationship between different sets of data from two sonar is analyzed by CCA for the undersea targets classification. The CCA algorithm is applied in [13] to find relationship between audio and video features in speaker recognition task. The CCA is also used for clustering in [14]. The canonical correlation analysis (CCA) technique can be used to obtain mapping from both views to a common representation space [15, 16]. The kernel trick is embedded in CCA to solve non-linear data problem [17]. For example, the kernelized version of CCA (KCCA) is used in [18] and [19] to find the relationship between the same documents represented by different languages. In [20], the KCCA is used to find the matching between texts in two languages.

However, the components that are most correlated across views found by CCA are not necessarily optimal for classification. In [21], the SVM-2k algorithm combining the relationship (between views) learning stage and the classifier training

stage into one is proposed. The SVM-2k algorithm gives the following optimization:

$$\min L = \frac{1}{2}||W_A||^2 + \frac{1}{2}||W_B||^2 + C^A \sum_{i=1}^{l} \xi_i^A + C^B \sum_{i=1}^{l} \xi_i^B + D \sum_{i=1}^{l} \eta_i \tag{2.2}$$

such that

$$| < W_A, \phi_A(x_i) > + b_A - < W_B, \phi_B(x_i) > - b_B| \leq \eta_i + \epsilon \tag{2.3}$$

$$y_i(< W_A, \phi_A(x_i) > + b_A) \geq 1 - \xi_i^A \tag{2.4}$$

$$y_i(< W_B, \phi_B(x_i) > + b_B) \geq 1 - \xi_i^B \tag{2.5}$$

$$\xi_i^A \geq 0, \xi_i^B \geq 0, \eta_i \geq 0 \; all \; for \; 1 \leq i \leq l. \tag{2.6}$$

The experimental results show that the SVM-2k algorithm outperforms the KCCA + SVM method. Counter to the SSML setting, [21] assumes that the labeled data are available on both views. In addition, [21] does not give a solution to the multi-label classification problem.

Chapter 3: PROBLEM FORMULATION

We consider data from two views: $x \in \mathcal{X}$, $z \in \mathcal{Z}$ and the labels $y \in \mathcal{Y}$. In surrogate supervision multi-view learning, we assume that the label $y \in \mathcal{Y}$ is only available on one view $\mathcal{X}$; the lables y are never directly provided on $\mathcal{Z}$. However, the unlabelled paired examples (x, z) are provided in both views.

To be specific, in a two-view learning setting, data can be represented as a set of triplets: $\{(x_i, z_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathcal{X}$, $z_i \in \mathcal{Z}$, and $y_i \in \mathcal{Y} = \{1, \ldots, K\}$. However, in surrogate supervision multi-view learning, we are given two sets of data: paired examples $\{(x_i, z_i)\}_{i=1}^{m}$ from $\mathcal{X}$ and $\mathcal{Z}$, and labeled examples $\{(x_i, y_i)\}_{i=m+1}^{n}$ from $\mathcal{X}$. We are interested in learning a classifier for y given z. This formulation is illustrated in Fig. 3.1. The main challenge of the setting is to obtain the mapping form $\mathcal{Z}$ to $\mathcal{Y}$ without a single example of the form (z_i, y_i). Table 3.1 provides descriptions of the symbols here.

Figure 3.2 illustrates the data formulation we used in our simulation. As in Fig.3.2, the data is separated to training data and testing data. In training data, we consider two groups: one is set of examples from $\mathcal{X}$ with accordingly labels from $\mathcal{Y}$; the other is set of unlabeled pairs from both $\mathcal{X}$ and $\mathcal{Z}$. In testing data, we want to obtain labels $y \in \mathcal{Y}$ of examples from $\mathcal{Z}$.

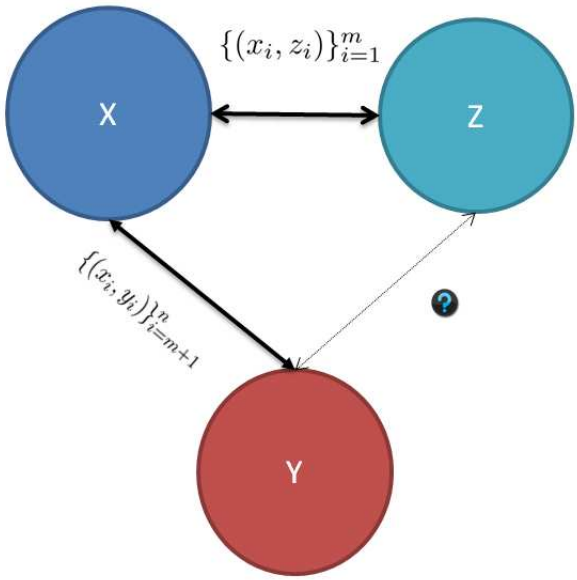


Figure 3.1: In surrogate supervision multi-view learning, we are given two sets of data: paired examples $\{(x_i, z_i)\}_{i=1}^m$ from $\mathcal{X}$ and $\mathcal{Z}$, and labeled examples $\{(x_i, y_i)\}_{i=m+1}^n$ from $\mathcal{X}$. We are interested in learning a classifier for y given z.

	Training data		Testing data
	I	II	
$\mathcal{X}$	◯	◯	
$\mathcal{Z}$		◯	◯
$\mathcal{Y}$	◯		?

Figure 3.2: The data is separated to training data and testing data. In training data, we consider two groups: one are examples from $\mathcal{X}$ with accordingly labels from $\mathcal{Y}$; the other are unlabeled pairs from both $\mathcal{X}$ and $\mathcal{Z}$. In testing data, we wan to obtain labels $y \in \mathcal{Y}$ of examples from $\mathcal{Z}$. Note that "◯" indicates the availability of data.

Symbol	Description
$\mathcal{X}$	The domain where labeled examples are available
$\mathcal{Z}$	The domain where only unlabeled examples are available
$\mathcal{Y}$	Category set
n	Total number of labeled triplets and unlabelled pairs
m	Total number of unlabelled pairs
K	Number of categories

Table 3.1: Descriptions of the symbols.

Chapter 4: ALTERNATIVE SOLUTIONS

4.1 CCA + SVM

One solution to the SSML problem is: first, using CCA to find the mapping from $\mathcal{X}$ to $\mathcal{Z}$. Then a SVM classifier can be trained in $\mathcal{X}$ and mapped to $\mathcal{Z}$. We refer this method to CCA + SVM. The CCA algorithm maps $\mathcal{X}$ and $\mathcal{Z}$ to a common space $\mathcal{R}$.

$$\begin{aligned} &\min_{a,b} \frac{1}{n} \sum_{i=1}^{n} ||a^T x_i - b^T z_i||_2^2 \\ &\text{subject to} \quad a^T R_X a = 1,\ b^T R_Z b = 1, \end{aligned} \tag{4.1}$$

where $R_X = \frac{1}{n} \sum_i x_i x_i^T$ and $R_Z = \frac{1}{n} \sum_i z_i z_i^T$. For simplicity, we assume that both the x_is and the z_is are zero mean.

Note that CCA performs a similar function to that of the second term of the RHS of (6.3). Next, an SVM classifier $f(\cdot) : \mathcal{R} \to \mathcal{Y}$ is obtained. However, the CCA algorithm does not guarantee that the dimensions found to maximize the correlation across views are optimal for training a discriminative classifier.

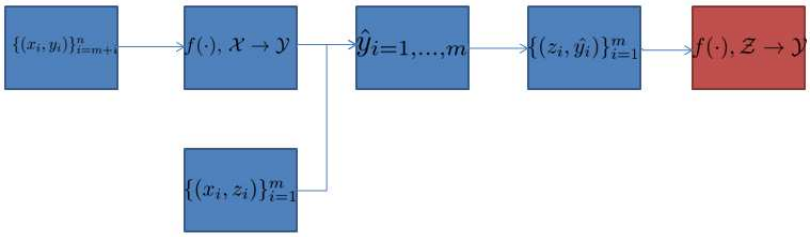


Figure 4.1: A classifier $f(\cdot) : \mathcal{X} \to \mathcal{Y}$ is first learned based on the training examples $\{(x_i, y_i)\}_{i=m+1}^{n}$. Then, based on the two-view pair examples $\{(x_i, z_i)\}_{i=1}^{n}$ the following training examples $\{(z_i, \hat{y})\}_{i=1}^{m}$ are formed.

4.2 Label-transferred Learning

Another solution is focused on estimating y for $\mathcal{Z}$ using traditional classification technique (i.e., SVM). Consider the approach where a classifier $f(\cdot) : \mathcal{X} \to \mathcal{Y}$ is first learned based on the training examples $\{(x_i, y_i)\}_{i=m+1}^{n}$. Then, based on the two-view pair examples $\{(x_i, z_i)\}_{i=1}^{n}$ the following training examples $\{(z_i, f(x_i))\}_{i=1}^{m}$ are formed. Note that since the classifier f can map example x to a label, the two-view pairs (x, z) can be modified to a estimated labeled examples $(z, \hat{y})$ where $\hat{y} = f(x)$. We refer to this approach as label-transferred learning. Figure 4.1 explains the label-transferred learning method.

Algorithm 1 Label-transferred Learning Method

we have $\{(x_i, y_i)\}_{i=m+1}^{n}$ and $\{(x_i, z_i)\}_{i=1}^{n}$
obtain $f(\cdot) : \mathcal{X} \to \mathcal{Y}$ based on $\{(x_i, y_i)\}_{i=m+1}^{n}$
$\{\hat{y}_i\}_{i=1}^{n} \leftarrow f(\{x_i\}_{i=1}^{n}) : \mathcal{X} \to \mathcal{Y}$
obtain $f(\cdot) : \mathcal{Z} \to \mathcal{Y}$ based on $\{(z_i, f(x_i))\}_{i=1}^{m}$

Chapter 5: C^4A ALGORITHM

[1]

To improve the method in Section 4.1, one can combine the relationship learning stage and the classifier training stage into one. For example, in [21], the authors proposes the SVM-2k algorithm though [21] assumes that labeled data are available on both views. The C^4A algorithm deals with the SSML problem by merging the relationship learning and the classifier training into one stage. In the following sub sections, we first look at the C^4S algorithm in two-class case, then the multi-class version. Finally, the sub-gradient descent implementation of the C^4A algorithm is given. The numerical evaluation of the C^4A algorithm is discussed in Chapter 8.

5.1 Two-class Case

As we propose to jointly learn mappings to the shared representation space and a maximum margin classifier, we consider the following convex formulation:

$$\min_{a,b} \frac{\gamma}{m} \sum_{i=1}^{m} \|a^T x_i - b^T z_i\|_2^2 + \frac{1}{n-m} \sum_{i=m+1}^{n} (1 - y_j a^T x_j)_+, \tag{5.1}$$

[1] This work was submitted to the IEEE International Workshop on Machine Learning for Signal Processing (MLSP) with Prof. Raviv Raich of Oregon State University.

where $(\alpha)_+$ is α for $\alpha > 0$ and 0 otherwise. Note that the first term is used to learn the most correlated components between $\mathcal{X}$ and $\mathcal{Z}$, which coincides the objective function in A.1. The second term ensures that the resulting components posses a predictive power, i.e., for each of the labeled examples these components can predict the sign of the label. In this subsection, with some abuse of notations we consider $\mathcal{Y} = \{+1, -1\}$. This coincides the objective function of binary SVM algorithm:

$$\min_a \frac{1}{p} \sum_{i=1}^{p} (1 - y_j a^T x_j)_+ + \frac{\gamma}{2} ||a||^2. \tag{5.2}$$

Note that this proposed objective (while not identical) is similar to the objective in SVM-2k [21] with omission of the term that relates to labeled examples for z_i and omission of regularization terms of the form $\|a\|^2$ or $\|b\|^2$.

The classification rule for y given x is given by

$$f_a(x) = \text{sgn}(a^T x) \tag{5.3}$$

and the classification rule for y given z is

$$f_b(z) = \text{sgn}(b^T z), \tag{5.4}$$

where $\text{sgn}(x)$ is the sign function yielding 1 for $x > 0$ and -1 for $x < 0$.

5.2 Multi-class Case

We follow the approach of extending SVM to the multiclass case taken in [31]. We start by defining the kth class score function for example x as $a_k^T x$, and similarly the kth class score function for example z as $b_k^T z$. We are interested in maximizing the correlation between the x based score functions and the z based score functions by minimizing $\sum_{l=1}^{K} \sum_{i=1}^{m} \|a_k^T x_i - b_k^T z_i\|_2^2$. Similarly, we would like ensure that the score function $a_k^T x$ for example (x_i, y_i) is highest when $k = y_i$. Following the soft-margin approach, we require that $a_{y_i}^T x_i \geq a_k^T x_i + 2 - \xi_{ik}$ for all $k \neq y_i$ [32]. Consequently, we define the following constrained convex problem formulation:

$$\min_{A,B} \frac{\gamma}{2mK} \sum_{k=1}^{K} \sum_{i=1}^{m} \|a_k^T x_i - b_k^T z_i\|_2^2 + \frac{1}{2(K-1)(n-m)} \sum_{i=m+1}^{n} \sum_{\substack{k=1,\\ k\neq y_i}}^{K} \xi_{ik} \tag{5.5}$$

subject to:

$$(a_{y_i} - a_k)^T x_i \geq 2 - \xi_{ik}, \quad k \in \mathcal{Y} \backslash y_i, \ i = m+1, \ldots, n$$

$$\xi_{ik} \geq 0, \quad k \in \mathcal{Y}, \ i = m+1, \ldots, n,$$

where $A = [a_1, a_2, \ldots, a_K]$, and $B = [b_1, b_2, \ldots, b_K]$. Note that the above constrained convex optimization can be reformulated as an unconstrained problem by solving for ξ_{ik}: $\xi_{ik}^* = (a_k^T x_i - a_{y_i}^T x_i + 2)_+$ and substituting the optimal ξ_{ik}^* back into

the objective. The resulting reformulation is:

$$\min_{A,B} \frac{\gamma}{2mK} \sum_{k=1}^{K} \sum_{i=1}^{m} \|a_k^T x_i - b_k^T z_i\|_2^2 + \tag{5.6}$$
$$\frac{1}{2(K-1)(n-m)} \sum_{i=m+1}^{n} \sum_{\substack{k=1,\\ k\neq y_i}}^{K} (a_k^T x_i - a_{y_i}^T x_i + 2)_+.$$

This objective coincides with (5.1), for the two class case. Due to space limitation, we omit the proof. Instead we point out that a key to the proof is selecting $a = (a_1 - a_2)/2$ and $b = (b_1 - b_2)/2$.

The classification rule for y given x is:

$$f_a(x) = \arg \max_{k\in\mathcal{Y}} a_k^T x, \tag{5.7}$$

and the classification rule for y given z is:

$$f_b(z) = \arg \max_{k\in\mathcal{Y}} b_k^T z. \tag{5.8}$$

5.3 Sub-Gradient Descent Implementation

Implementation of sub-gradient for (5.6) is given below.

Note that $\bigtriangledown_{a_k} f(a_k^t, b_k^t)$ and $\bigtriangledown_{b_k} f(a_k^t, b_k^t)$ are the sub-gradients with respect to

Algorithm 2 Sub-Gradient Descent Implementation of C^4A Algorithm

$t \leftarrow 1$
$a_k^1 \leftarrow 0$
$b_k^1 \leftarrow 0$
while $t \leq N$ **do**
  $a_k^{t+1} \leftarrow a_k^t - \alpha_1^{(t)} \bigtriangledown_{a_k} f(a_k^t, b_k^t)$
  $b_k^{t+1} \leftarrow b_k^t - \alpha_2^{(t)} \bigtriangledown_{b_k} f(a_k^t, b_k^t)$
  $t \leftarrow t + 1$
end while

a_k and b_k, and α_i^t is the step size. The gradient with respect to a_k is:

$$\begin{aligned}\bigtriangledown_{a_k} f(a_k^t, b_k^t) = \frac{\gamma}{K}(R_X a_k - R_{XZ} b_k) + \frac{1}{2(n-m)(K-1)} \\ \big(\sum_{i=1}^{n-m} x_i I(y_i \neq k) I((2 - (a_{y_i} - a_k)^T x_i) > 0) \\ - \sum_{i=1}^{n-m} \sum_{\substack{l=1, \\ l \neq k}}^{K} x_i I((2 - (a_{y_i} - a_l)^T x_i) > 0)\big).\end{aligned}$$

The gradient with respect to b_k is

$$\bigtriangledown_{b_k} f(a_k^t, b_k^t) = \frac{\gamma}{K}(R_Z b_k - R_{ZX} a_k), \tag{5.9}$$

where $R_X = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T$, and $R_{XZ} = \frac{1}{n} \sum_{i=1}^{n} x_i z_i^T$, $R_{ZX} = \frac{1}{n} \sum_{i=1}^{n} z_i x_i^T$.

Chapter 6: HINGE LOSS UPPER BOUND FOR SSML

[1]

In this chapter, we derive an upper bound for the hinge loss for the SSML and propose the SSM-SVM algorithm. The numerical evaluation of the SSM-SVM algorithm is presented in Chapter 8.

6.1 Two-class Case

We start with the binary-class case. In classification, the goal is to minimize the following classification error objective with respect to $g(\cdot)$:

$$E_{z,y}[\frac{1}{2}|g(z) - y|], \tag{6.1}$$

where $g(\cdot) : \mathcal{X} \to \mathcal{Y}$ is a decision function mapping feature space $\mathcal{X}$ to a label in $\mathcal{Y} = \{1, -1\}$. A common approach (e.g., in SVMs) is to replace the 0-1 loss in (6.1) with a hinge loss:

$$E_{z,y}[(1 - g(z)y)_+], \tag{6.2}$$

[1] This work was submitted to Pattern Recognition Letter with Prof. Raviv Raich of the Oregon State University.

where $(t)_+ = \max\{0, t\}$. In SVM, a classifier is obtained by minimizing the regularize sample based objective: $\frac{1}{n}\sum_{i=1}^{n}[(1 - g(z_i)y_i)_+] + Pen(g)$, where $Pen(g)$ denotes a regularization term. For example, in a linear SVM $g(z) = w^T z$, the regularization term is $Pen(g) = \frac{\lambda}{2}\|w\|^2$. In the SSML scenario, labeled examples are only available in $\mathcal{X}$. In the absence of examples of the type (z_i, y_i), one cannot compute directly the classifier which minimizes (6.2) or its regularized sample-based alternative.

Naturally, in SSML, we can only deal with objectives that are based on samples of the type (x_i, y_i) and (x_i, z_i) or equivalently objectives that require the joint distributions of (x, y) and (x, z). This leads us to considering an upper bound approach to designing the surrogate objective. Consider the following upper bound to (6.2):

$$E_{z,y}[(1 - g(z)y)_+] \leq E_{z,y}[(1 - h(x)y)_+] + E_{x,z}[|h(x) - g(z)|], \tag{6.3}$$

where $h(\cdot) : \mathcal{X} \to \mathcal{Y}$ is a classifier mapping feature space $\mathcal{X}$ to a surrogate objective on $\mathcal{Y}$. For a proof of (6.3), we refer the reader to C. The RHS of (6.3) consists of two terms. The first is a hinge-loss for the classifier $h(\cdot)$ measuring how well $h(\cdot)$ can predict the label y, while the second term measures how close are the predictions of the two classifiers $g(\cdot)$ and $h(\cdot)$. In other words, the objective on the RHS of (6.3), promotes a classifier on $\mathcal{X}$ $h(\cdot)$ which can simultaneously predict y and can be well-approximated by a classifier on $\mathcal{Z}$ $g(\cdot)$. Note that since the bound holds for any $h(\cdot)$, the bound can be tightened by minimizing the RHS w.r.t. $h(\cdot)$.

This formal bound suggests the replacement of the hinge loss in one view with the hinge loss in the other view plus a multi-view classifier mismatch term. In the following, we present a generalization of this bound to the multi-class case.

6.2 Multi-class Case

Inspired by multi-class SVM [31], we consider the multi-class objective

$$\frac{1}{2}\sum_{k\neq l} E_{z,y}[(2-(g_l(z)-g_k(z))_+ I(y=l)], \tag{6.4}$$

where $I(\Omega)$ is a indicator function such that: $I(\Omega)=1$ when Ω is achieved and otherwise $I(\Omega)=0$. In this generalization, for each class $k\in\mathcal{Y}$ a score functions $g_k(\cdot)$ are sought after. Ideally, given example z with label y, the score function $g_y(z)$ should be larger than $g_l(z)$ for any $l\neq y$ with some margin. Here a margin of 2 is selected. This choice can be better understood by examining the equivalence between the multi-class objective in (6.4) and the binary-class objective in (6.2) in the two class case. The loss function of binary-class case in (6.2) can be expanded as $(1-g(\cdot))_+ I(y=1)+(1+g(\cdot))_+ I(y=-1)$ whereas the multi-class objective (when $K=2$) can be written as $\frac{1}{2}((2-(g_2(\cdot)-g_1(\cdot))_+ I(y=2)+(2-(g_1(\cdot)-g_2(\cdot))_+ I(y=1)) = (1-g(\cdot))_+ I(y=2)+(1+g(\cdot))_+ I(y=1)$. Setting the binary class classifier to the weighted difference of the multi-class score functions $g(\cdot)=\frac{g_2(\cdot)-g_1(\cdot)}{2}$, one can show that the two objectives are equivalent.

As with the binary class problem, the objective relies on the joint distribution

of z and y for which no samples are available. Hence a surrogate in terms of the joint distributions of (x, y) and (x, z) is sought after. Generalizing the bounding technique used in (6.2), we bound the multi-class error objective in (6.4) as follows:

$$\begin{aligned}
&\sum_{k\neq l} E_{z,y}[(2-(g_l(z)-g_k(z)))_+ I(y=l)] \\
\leq &\sum_{k\neq l} E_{x,y}[(2-(h_l(x)-h_k(x)))_+ I(y=l)] + \sum_k E_{x,z}[|g_k(z)-h_k(x)|] \\
&+(K-2)\max_k E_{x,z}[|g_k(z)-h_k(x)|]. \qquad (6.5)
\end{aligned}$$

For a proof of (6.5), we refer the reader to C. Without loss of generality we omit the $\frac{1}{2}$ term. On RHS of (6.5), the first term measures how well $h(\cdot)$ can map $\mathcal{X}$ to $\mathcal{Y}$, while the other two terms measure how close are the predictions made by $h(\cdot)$ and by $g(\cdot)$. Compared to the binary-class case in (6.3), the upper bound in (6.5) minimizes a linear combination of the average score differences and the per-class maximum score. The bound in (6.5) is key to the SSM-SVM algorithm in Section 6.3.

6.3 SSM-SVM Algorithm

In SVM, a regularized version of the hinge loss in (6.2) is used as an objective function with sample average replacing the expectation. We follow a similar approach. Adding a regularizer $\frac{\lambda}{K}\sum_{k=1}^{K}||b_k||^2$ to (6.5), and replacing the expectations $E[g(x,y)]$ and $E[h(z,x)]$ with the sample averages $\frac{1}{n-m}\sum_{i=1}^{n-m} g(x_i, y_i)$ and

$\frac{1}{m}\sum_{i=n-m+1}^{n} h(z_i, x_i)$ respectively, we obtain the following optimization:

$$\min_{A,B} \frac{\lambda}{K}\sum_{k=1}^{K}||b_k||^2 + \frac{1}{(n-m)(K-1)}\sum_{k=1}^{K}\sum_{i=m+1}^{n}(2-(a_{y_i}-a_k)^T x_i)_+ I(y_i \neq k) + \frac{1}{m(K-1)}\sum_{k=1}^{K}\sum_{i=1}^{m}|b_k^T z_i - a_k^T x_i| + \frac{K-2}{m(K-1)}\sum_{i=1}^{m}\max_k |b_k^T z_i - a_k^T x_i| \quad (6.6)$$

where $A = [a_1, a_2, \ldots, a_K]$, $B = [b_1, b_2, \ldots, b_K]$, and λ is a tuning parameter that controls the weight of the regularizer. Additionally, to derive (6.6) we replace linear score functions in (6.5): $h_k(x) = a_k^T x$, $g_k(z) = b_k^T z$. The minimization of (6.6) constitutes a training phase in which the classifier parameters A and B are obtained. Based on the parameters found in the training phase, classification can be performed. The classification rule for y given x is:

$$f_a(x) = \arg \max_{k\in\mathcal{Y}} a_k^T x, \quad (6.7)$$

and the classification rule for y given z is:

$$f_b(z) = \arg \max_{k\in\mathcal{Y}} b_k^T z. \quad (6.8)$$

Note that the optimization function in6.6 basically consists of three items: a regularization item, a relationship analyzing item, and a classification item. The relationship analyzing item is similar to A.1. The classification item are similar to

the optimization function in SVM which can be written as:

$$\min_{a} \frac{\gamma}{2}||a||^2 + \frac{1}{p}\sum_{i=1}^{p}(1 - y_j a^T x_j)_+. \tag{6.9}$$

6.4 Sub-gradient Descent for SSM-SVM

To solve (6.6), we propose a simple sub-gradient descent approach. The implementation of SSM-SVM algorithm is given below.

Algorithm 3 Sub-Gradient Descent Implementation of SSM-SVM Algorithm

$t \leftarrow 1$
$a_k^1 \leftarrow 0$
$b_k^1 \leftarrow 0$
while $t \leq N$ **do**
  $a_k^{t+1} \leftarrow a_k^t - \alpha_1^{(t)} \bigtriangledown_{a_k} f(a_k^t, b_k^t)$
  $b_k^{t+1} \leftarrow b_k^t - \alpha_2^{(t)} \bigtriangledown_{b_k} f(a_k^t, b_k^t)$
  $t \leftarrow t + 1$
end while

Note that $\bigtriangledown_{a_k} f(a_k^t, b_k^t)$ and $\bigtriangledown_{b_k} f(a_k^t, b_k^t)$ are sub-gradients with respect to a_k and b_k respectively, and α_i^t is the step size. The sub-gradient with respect to a_k is:

$$\bigtriangledown_{a_k} f(a_k, b_k) = \frac{1}{(n-m)(K-1)} \sum_{i=m+1}^{n} \Big(x_i I(y_i \neq k) I\big((2 - (a_{y_i} - a_k)^T x_i) > 0\big)$$
$$- \sum_{\substack{l=1,\\ l\neq k}}^{K} x_i I\big((2 - (a_{y_i} - a_l)^T x_i) > 0\big) I(y_i = k) \Big) - \frac{1}{m(K-1)} \sum_{i=1}^{m} sgn(b_k^T z_i - a_k^T x_i) \cdot x_i$$
$$- \frac{K-2}{m(K-1)} \sum_{i=1}^{m} sgn(b_{k_i^*}^T z_i - a_{k_i^*}^T x_i) I(k_i^* = k) \cdot x_i, \tag{6.10}$$

where $k_i^* = \arg\max_k |b_k^T z_i - a_k^T x_i|$. The gradient with respect to b_k is:

$$\begin{aligned}\nabla_{b_k} f(a_k^t, b_k^t) = \frac{2\lambda}{K} b_k + \frac{1}{m(K-1)} \sum_{i=1}^{m} sgn(b_k^T z_i - a_k^T x_i) \cdot z_i \\ + \frac{K-2}{m(K-1)} \sum_{i=1}^{m} sgn(b_{k_i^*}^T z_i - a_{k_i^*}^T x_i) I(k_i^* = k) \cdot z_i. \end{aligned} \tag{6.11}$$

Chapter 7: GAUSSIAN MIXTURES

[1]

7.1 Introduction

We address learning a classifier to predict the class label $C \in \mathcal{C} = \{1, \ldots, N_c\}$ given a *multi-view* feature vector $\mathbf{X} = (X^{(1)}, \ldots, X^{(N_v)})$ [33], $X^{(i)}$ the feature (sub)-vector for the ith view. We focus on a challenging label-deficient scenario dubbed 'surrogate supervision multiview learning' (SSML), wherein there are *no* labeled training examples for some views, even though there are *unlabeled* training examples with multiple (perhaps all) views present. This scenario may occur, *e.g.*, when there is a new sensing modality or technology for an existing application domain. In such cases, a (legacy) labeled training set may already exist for the standard sensors. Moreover, one can take joint observation measurements using both the standard and new sensors, creating multi-view examples. However, ground-truth *labeling* these new examples may be both time-consuming and expensive. This scenario may also occur if, during the labeled training data acquisition process, some sensors were "censored" or suffered from equipment glitches. To fix our ideas and, we emphasize, without any loss of generality, we explicitly consider the two-view case

[1] This work was submitted to the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) with Dr. Raviv Raich of the Oregon State University and Dr. David Miller of the Pennsylvania State University.

here: $\mathbf{X} = (X, Z), X \in \mathcal{R}^{d_x}, Z \in \mathcal{R}^{d_z}$. Thus, we assume an unlabeled training data subset $\mathcal{X}_u = \{(x_i, z_i), i \in \mathcal{S}_u\}$ and a labeled training subset $\mathcal{X}_l = \{(x_i, c_i), i \in \mathcal{S}_l\}$, $\mathcal{S}_u = \{1, 2, \ldots, N_u\}$ and $\mathcal{S}_l = \{N_u + 1, N_u + 2, \ldots, N_u + N_l\}$. Several previous works have investigated this problem. In [19], a two-stage discriminative learning approach was proposed. Here, a classifier that treats X as the input feature vector is first designed in a supervised fashion based on $\mathcal{X}_l$. Next, this classifier is used to make class predictions on $\mathcal{X}_u$, thus creating surrogate (albeit noisy) labels that are then used to train a classifier that makes class inferences given Z. In [33], a single joint optimization technique was proposed, learning linear transformations that aim both to maximize the canonical correlations between X and Z *and* to act as a linear discriminant function, well-separating the data from the different classes. The learned linear transformations that map Z to the canonical coordinate space are used as a linear discriminant function, providing class inferences given Z. One limitation of both of these methods is that they are tailored for the two-view learning case. It is unclear whether they are readily extendible to handle more than two views, let alone many views (which may occur in some distributed sensor settings).

Here, alternatively, we develop a generative mixture model solution that readily handles multiple (even many) views, and with the capability to perform exact class inference given any *subset* of views observed (*i.e.*, given arbitrary patterns of missing views, both in testing as well as in the training phase). Our model is both a multi-view extension of the semi-supervised framework from [34] and a semi-supervised extension of mixture of factors analyzers (MFA) , with the MFA

approach used to parameterize the covariance matrices of the multivariate Gaussian mixture components, ensuring well-conditioned matrices with controllable model complexity, given limited training data [35].

7.2 Formulation

Suppose samples are generated i.i.d., with $(X_i, Z_i), i \in \mathcal{S}_u$ jointly generated according to a multivariate Gaussian mixture density (GMM) and with X_i and $C_i, i \in \mathcal{S}_l$ conditionally independent given the mixture component of origin, with X_i generated according to the same GMM (but marginalized over the missing random vector Z) and with C_i generated according to a component-conditional multinomial pmf. The associated incomplete data likelihood for our model is:

$$f_{\text{inc}}(\mathcal{X}_l, \mathcal{X}_u; \theta) = (\prod_{i \in S_l} \sum_j \phi(x_i; \mu_{xj}, A_{xj}A_{xj}^T + \sigma^2 I) B_{c_i j} \alpha_j) \cdot$$
$$(\prod_{i \in S_u} \sum_j \phi(x_i, z_i; \mu_j, A_j A_j^T + \sigma^2 I) \alpha_j). \tag{7.1}$$

Here, comprising the parameter set θ: $\{\alpha_j\}$ are the component masses, $\sum_j \alpha_j = 1$, $\alpha_j \geq 0 \forall j$; B is a matrix whose j-th row is the component-conditional class probability vector $B_{\cdot j} = [B_{1j} \ldots B_{Cj}]$ ($\sum_c B_{cj} = 1$ and $B_{cj} \geq 0$); $\mu_j = [\mu_{xj}^T, \mu_{zj}^T]^T$ is component j's mean vector; $A_j = [A_{xj}; A_{zj}]^T$ is a factor loading matrix [36], used to parameterize the covariance matrix for Gaussian component j (with the row sub-matrix A_{xj} used to parameterize the covariance matrix for modeling $X_i, i \in S_l$); and $\phi(\cdot)$ is the multivariate Gaussian density. Also, σ^2 will be treated as a

hyperparameter, chosen to ensure well-conditioned covariance matrices and held fixed during (EM) learning of all other parameters.

An EM algorithm for (locally) maximizing (7.1) is developed as follows. We naturally introduce as hidden data within the EM framework [37] the mixture component of origin for each sample, $J_i, i = 1, \ldots, N_u + N_l$. Also, since we are invoking a mixture of factors approach, we also treat as hidden data the *factor vector* $V_i \in \mathcal{R}^d$. As in the standard MFA approach, we assume $V_i \sim \mathcal{N}(0, I)$, with $X_i|v_i, j \sim \mathcal{N}(\mu_{xj} + A_{xj}v_i, \sigma^2 I), I \in S_l$ and with $[X_i, Z_i]^T|v_i, j \sim \mathcal{N}(\mu_j + A_j v_i, \sigma^2 I), I \in S_u$. These choices are consistent with the incomplete data likelihood form in (7.1). Let $\mathcal{V} = \{\mathcal{V}_l, \mathcal{V}_u\}$ and $\mathcal{J} = \{\mathcal{J}_l, \mathcal{J}_u\}$ denote the sets of hidden data. The *complete* data likelihood for the labeled subset is then:

$$f_c(\mathcal{X}_l, \mathcal{V}_l, \mathcal{J}_l|\theta) = \prod_{i \in S_l} f(x_i|v_i, j_i) f(v_i) P(c_i|j_i) P(j_i)$$
$$= \prod_{i \in S_l} \phi(x_i|A_{xj}v_i + \mu_{xj}, \sigma^2 I)\phi(v_i; 0, I) B_{c_i j_i} \alpha_{j_i}.$$

Likewise, the complete data likelihood for the unlabeled data subset is:

$$f_c(\mathcal{X}_u, \mathcal{V}_u, \mathcal{J}_u) = \prod_{i \in S_u} \phi(\begin{bmatrix} x_i \\ z_i \end{bmatrix}|A_j v_i + \mu_j, \sigma^2 I)\phi(v_i; 0, I)\alpha_{j_i}.$$

The EM auxiliary function for the log-likelihood [37] is given by

$$
\begin{aligned}
&Q(\theta;\theta^n) = E_{\mathcal{V},\mathcal{J}}[\log f(\mathcal{X}_l,\mathcal{X}_u,\mathcal{V},\mathcal{J})|\{x_i,c_i\}_{i\in S_l},\{x_i,z_i\}_{i\in S_u};\theta^n] \\
&\propto \sum_{i\in S_l} E_{v_i,j_i}[\log\phi(x_i|A_{xj}v_i+\mu_{xj},\sigma^2 I)|\{x_i,c_i\}_{i\in S_1};\theta^n]+ \\
&\sum_{i\in S_l} E_{j_i}[\log B_{c_i j_i}+\log\alpha_{j_i}|\{x_i,c_i\}_{i\in S_1};\theta^n] \\
&+\sum_{i\in S_u} E_{v_i,j_i}[\log\phi(\begin{bmatrix} x_i \\ z_i \end{bmatrix}|A_j v_i+\mu_j,\sigma^2 I)|\{x_i,z_i\}_{i\in S_u};\theta^n] \\
&+\sum_{i\in S_u} E_{j_i}[\log\alpha_{j_i}|\{x_i,z_i\}_{i\in S_u};\theta^n].
\end{aligned}
$$

Further, after applying the iterated expectation law, $E_{v_i,j_i}[\cdot] = E_{j_i}[E_{v_i|j_i}[\cdot]]$, and simplifying, we obtain

$$
\begin{aligned}
&-Q(\theta;\theta^n) \propto \\
&\frac{1}{2\sigma^2}\sum_{i\in S_l}\sum_j E_{x_i'}[\|x_i-(A_{xj}v_i+\mu_{xj})\|^2|x_i,c_i,j;\theta^n]P(j|x_i,c_i)- \\
&\sum_j\sum_c \log B_{cj}\sum_{i\in S_l:c_i=c} P(j|x_i,c_i)-\sum_j\log\alpha_j\sum_{i\in S_l}P(j|x_i,c_i)+ \\
&\frac{1}{2\sigma^2}\sum_{i\in S_u}\sum_j E[\|x_i-(A_{xj}v_i+\mu_{xj})\|^2|x_i,z_i,j;\theta^n]P(j|x_i,z_i)+ \\
&\frac{1}{2\sigma^2}\sum_{i\in S_u}\sum_j E[\|z_i-(A_{zj}v_i+\mu_{zj})\|^2|x_i,z_i,j;\theta^n]P(j|x_i,z_i)- \\
&\sum_j\log\alpha_j\sum_{i\in S_u}P(j|x_i,z_i).
\end{aligned}
$$

E-step:

The E-step computes the required expected hidden quantities in the above auxiliary function, given the model parameters held fixed at θ^n (superscripting parameters by 'n' is omitted for concision), *i.e.*

$$P(j|x_i, c_i) = \frac{\phi(x_i; \mu_{xj}, A_{xj}A_{xj}^T + \sigma^2 I)B_{c_i j}\alpha_j}{\sum_k \phi(x_i; \mu_{xk}, A_{xk}A_{xk}^T + \sigma^2 I)B_{c_i k}\alpha_k} \tag{7.2}$$

$$P(j|x_i, z_i) = \frac{\phi(\begin{bmatrix} x_i \\ z_i \end{bmatrix}; \mu_j, A_j A_j^T + \sigma^2 I)\alpha_j}{\sum_k \phi(x_i; \mu_k, A_k A_k^T + \sigma^2 I)\alpha_k} \tag{7.3}$$

$$E[v_i|x_i, z_i, j] = A_j^T(A_j A_j^T + \sigma^2 I)^{-1}(\begin{bmatrix} x_i \\ z_i \end{bmatrix} - \mu_j) \tag{7.4}$$

$$E[v_i|x_i, j] = A_{xj}^T(A_{xj}A_{xj}^T + \sigma^2 I)^{-1}(x_i - \mu_{xj}) \tag{7.5}$$

$$E[v_i v_i^T|x_i, z_i, j] = I - A_j^T(A_j A_j^T + \sigma^2 I)^{-1}A_j + A_j^T \cdot \tag{7.6}$$

$$(A_j A_j^T + \sigma^2 I)^{-1}(\begin{bmatrix} x_i \\ z_i \end{bmatrix} - \mu_j)(\begin{bmatrix} x_i \\ z_i \end{bmatrix} - \mu_j)^T(A_j A_j^T + \sigma^2 I)^{-1}A_j$$

$$E[v_i v_i^T|x_i, j] = I - A_{xj}^T(A_{xj}A_{xj}^T + \sigma^2 I)^{-1}A_{xj} + A_{xj}^T \cdot \tag{7.7}$$

$$(A_{xj}A_{xj}^T + \sigma^2 I)^{-1}(x_i - \mu_{xj})(x_i - \mu_{xj})^T(A_{xj}A_{xj}^T + \sigma^2 I)^{-1}A_{xj}.$$

We further note that the above E-step computations involving matrix inversion can be simplified and (for $d \ll d_x, d_z$ greatly) reduced by invoking the matrix inversion lemma, replacing the inversion of a $(d_x + d_z) \times (d_x + d_z)$ matrix or a

$d_x \times d_x$ matrix with inversion of a $d \times d$ matrix, as follows:

$$(Q_j Q_j^T + \sigma^2 I)^{-1} = \frac{1}{\sigma^2} I - \frac{1}{\sigma^2} Q_j (\sigma^2 I + Q_j^T Q_j)^{-1} Q_j^T. \tag{7.8}$$

This can be applied, respectively, for $Q_j = A_j$ in (7.4) and (7.6) and for $Q_j = A_{xj}$ in (7.5) and (7.7). Furthermore, letting $M_j = A_j^T A_j$ and $M_{xj} = A_{xj}^T A_{xj}$, using the result that $\frac{1}{\sigma^2}(I - M_j(\sigma^2 I + M_j)^{-1}) = (\sigma^2 I + M_j)^{-1}$, and after several simplifying steps which exploit the similarity transformation of a matrix, we obtain final, compact E-step expressions as follows:

$$E[v_i|x_i, z_i, j] = (\sigma^2 I + M_j)^{-1} A_j^T (\begin{bmatrix} x_i \\ z_i \end{bmatrix} - \mu_j) \tag{7.9}$$

$$E[v_i|x_i, j] = (\sigma^2 I + M_j)^{-1} A_{xj}^T (x_i - \mu_{xj}) \tag{7.10}$$

$$E[v_i v_i^T|x_i, z_i, j] = \sigma^2(\sigma^2 I + M_j)^{-1} + \tag{7.11}$$

$$(\sigma^2 I + M_j)^{-1} A_j^T (\begin{bmatrix} x_i \\ z_i \end{bmatrix} - \mu_j)(\begin{bmatrix} x_i \\ z_i \end{bmatrix} - \mu_j)^T A_j (\sigma^2 I + M_j)^{-1}$$

$$E[v_i v_i^T|x_i, j] = \sigma^2(\sigma^2 I + M_{xj})^{-1} + \tag{7.12}$$

$$(\sigma^2 I + M_{xj})^{-1} A_{xj}^T (x_i - \mu_{xj})(x_i - \mu_{xj})^T A_{xj} (\sigma^2 I + M_{xj})^{-1}.$$

Note that this simplification of the E-step, without any approximation, can also be applied to reduce complexity of the E-step in the standard, original EM algorithm formulation for mixtures of factors analyzers [36].

M-step:

Solving the minimization of $-Q$ subject to $\sum_j \alpha_j = 1$ and $\sum_c B_{cj} = 1 \forall j$, yields the following M-step update of θ:

$$\alpha_j^{(n+1)} = \frac{\sum_{i \in S_l} P(j|x_i, c_i) + \sum_{i \in S_u} P(j|x_i, z_i)}{N_l + N_u} \tag{7.13}$$

$$B_{cj}^{(n+1)} = \frac{\sum_{i \in S_l : c_i = c} P(j|x_i, c_i)}{\sum_{i \in S_l} P(j|x_i, c_i)} \tag{7.14}$$

$$\begin{aligned}[A_{xj}\ \mu_{xj}]^{(n+1)} = &(\sum_{i \in S_l} x_i E[[v_i; 1]|x_i, j]^T P(j|x_i, c_i) + \\ &\sum_{i \in S_u} x_i E[[v_i; 1]|x_i, z_i, j]^T P(j|x_i, z_i)) \cdot \\ &(\sum_{i \in S_l} E[[v_i; 1][v_i; 1]^T|x_i, j] P(j|x_i, c_i) + \\ &\sum_{i \in S_u} E[[v_i; 1][v_i; 1]^T|x_i, z_i, j] P(j|x_i, z_i))^{-1}\end{aligned} \tag{7.15}$$

$$\begin{aligned}[A_{zj}\ \mu_{zj}]^{(n+1)} = &(\sum_{i \in S_u} z_i E[[v_i; 1]|x_i, z_i, j]^T P(j|x_i, z_i)) \cdot \\ &(\sum_{i \in S_u} E[[v_i; 1][v_i; 1]^T|x_i, z_i, j] P(j|x_i, z_i))^{-1}\end{aligned} \tag{7.16}$$

Missing Views and Missing Labels in the General Multi-View Case:

While the above EM formulation only explicitly considers the two-view case, it is straightforward to extend our approach for the case of more than two views, with arbitrary patterns of missing views, with missing individual *features* for particular

views, as well as with missing class labels for the views (and individual features) that are observed for a given training example. This general applicability of our framework stems from the fact that each row of the factor loading matrix is used to generate an individual feature. Thus, the factor loading matrix A_j (and the mean vector μ_j) can be arbitrarily row-partitioned, *as needed*, to model via the GMM an individual training example with missing views and missing features for observed views (i.e., an arbitrary sub-vector of the full multi-view observation vector).

Class Inferences:

Class decisionmaking is based on the maximum *a posteriori* (MAP) rule:

$$P(c|q) = \frac{\sum_j f(q|j)P(c|j)P(j)}{\sum_j f(q|j)P(j)} = \frac{\sum_j \phi(q; \mu_{qj}, A_{qj}A_{qj}^T + \sigma^2 I)B_{cj}\alpha_j}{\sum_j \phi(q; \mu_{qj}, A_{qj}A_{qj}^T + \sigma^2 I)\alpha_j}, \tag{7.17}$$

where we may have $q = z$, $q = x$, or $q = [xz]^T$, where, for the latter case, $A_{qj} = A_j$, the full factor loading matrix. More generally, when there are more than two views, by suitable row-partitioning of the factor loading matrices and mean vectors, as discussed above, our MFA model can be used to make exact class posterior inferences given arbitrary patterns of missing views and arbitrary patterns of missing features for observed individual views.

Chapter 8: SIMULATIONS

In the simulation chapter, we first test the C^4A and SSM-SVM algorithms in a set of synthetic data and compare them with alternative methods (such as CCA + SVM, and label-transferred learning). Then, we apply the proposed algorithms to lip-reading task.

8.1 Test of the Proposed Discriminative Approach

8.1.1 Test on Synthetic Data

In this experiment, we construct multi-view data from the UCI machine learning repository [38] and compare the C^4A, SSM-SVM algorithms with the CCA + SVM method, and the label-transferred learning method in terms of classification accuracy.

8.1.1.1 Experimental Setting

A common method to construct a multi-view data from a single-view is to extract a subset features from the latter. For examples, we have $\mathcal{D} = \{d_1, d_2, \cdots, d_n\}$, then the feature subsets of $\mathcal{D}$: $\mathcal{X} = \{d_1, \cdots, d_u\}$ and $\mathcal{Z} = \{d_{u+1}, \cdots, d_v\}$ where $u < v$ can be considered as two views from $\mathcal{D}$.

We construct multi-view data from six popular datasets in the UCI machine learning repository. In each of the datasets, we first normalize all the samples and randomly choose half of the features as one view and the other half as the other view. Then, the data is divided into training and testing data in a size ratio of $7:1$. In the training data, we consider the same number of samples from (x_i, y_i) and (x_i, z_i). Finally, the classifier is trained on the training data and tested on the testing data.

8.1.1.2 Results

Table 8.1 shows the experimental results for the lung-cancer, spect, wine, hill-valley, ionosphere, and glass datasets from UCI machine learning repository. The proposed algorithms outperform the CCA+SVM and the label-transferred learning in terms of accuracy on all of the datasets. We would like to point out that the accuracies provided in this experiment may appear lower than those obtained in a typical classification experiment (e.g., see hill-valley). However, this experiment presents two challenges: first, only a subset of he features are used for classification. Moreover, no direct label is available in the desired view due to the SSML setting. Nonetheless, the proposed algorithms still outperforms the other two methods.

Table 8.1: Classification accuracies achieved by C^4A, SSM-SVM, CCA + SVM, and label-transferred learning in lung-cancer, spect, wine, hill-valley, ionosphere, glass datasets. Note that the classification is performed on $\mathcal{Z}$ where the labeled samples are not available and that the linear SVM is applied in CCA + SVM and in label-transferred learning.

	SSM-SVM	CCA+SVM	label-transferred learning
lung-cancer	$73.33\% \pm 14.91$	$60.00\% \pm 14.91$	$46.67\% \pm 17.21$
spect	$82.00\% \pm 9.19$	$76.00\% \pm 14.29$	$81.00\% \pm 11.97$
wine	$95.45\% \pm 3.21$	$89.54\% \pm 4.81$	$93.93\% \pm 6.94$
hill-valley	$57.63\% \pm 3.41$	$54.47\% \pm 1.50$	$54.45\% \pm 6.22$
ionosphere	$78.18\% \pm 5.23$	$76.82\% \pm 2.96$	$76.04\% \pm 2.68$
glass	$55.56\% \pm 5.86$	$44.44\% \pm 11.11$	$47.41\% \pm 11.23$

8.1.2 Application to Lip-reading

In this section, we test C^4A, and SSM-SVM algorithms, and compare them with alternative approaches in terms of accuracy on an audiovisual data set (Grid Corpus). We first explain the preprocessing of the raw audiovisual data. Then in Section 8.1.2.4, the experimental setup is described. Finally results are analyzed in Section 8.1.3.

8.1.2.1 Data Preprocessing

The Grip Corpus data consists of both audio and video recordings of simple-structured sentences spoken by 34 talkers. Each sentence is of the form "[command] [color] [preposition] [letter] [digit] [adverb]", for example,"place blue at F 9 now"[39]. In our experiment, we only consider the classification of digits. In the following subsections, we will describe our approach processing the raw audiovisual

data.

8.1.2.2 Face and Lip Detection

Since the audio recording is mostly relevant to the lip movements in the video, we restrict our attention to the lip region. The face and lip detection technique in this experiment is from [23]. In [23], the authors propose a fast algorithm to detect face and lips based on color information. This algorithm is suited for our experiment considering the large amount and high quality data we processed.

A face image is first converted from RGB color space to chromatic color space. The chromatic color space is defined as $r = R/(R+G+B)$ and $g = G/(R+G+B)$ where R, G, B are the intensity of pixel in RGB color space.

In a face image the g value of skin pixels which lay on a compact region over the $r - g$ plane can be bounded by two polynomials [23]:

$$f_{upper}(r) = -1.3767r^2 + 1.0743r + 0.1452 \tag{8.1}$$

$$f_{lower}(r) = -0.776r^2 + 0.5601r + 0.1766. \tag{8.2}$$

The lip can be detected within the face region by set an upper bound for g:

$$f'_{upper}(r) = -0.776r^2 + 0.5601r + 0.2123 \tag{8.3}$$

and a lower bound as in [8.2].

Since the distance between the speaker and the camera varies for different videos, we resize each lip image according to the size of the face to make all the lip images consistent.

Feature	Accuracy
Audio	90.28%
Video	72.83%

Table 8.2: The classification accuracies achieved by trained linear SVM classifier on audio and on video respectively.

8.1.2.3 Feature Extraction

Each video of speaking a sentence consists of 72 frames. In each video sample, the segment that contains 8 consecutive frames corresponding to saying the digit is used for our classification task. For each frame, the lip region is extracted and converted to a gray scale image as shown in fig. 8.1. Then the 8 frames are stacked together and reduced to a 100 dimension feature vector by the kernelized version of principal component analysis (KPCA). We use KPCA instead of linear PCA to map the original data to a non-linear space.

The spectrograms of audio recordings are used as features (8.2). Each audio sample is down-sampled from 50kHz to 5kHz. The audio segment is converted to spectrogram with Hamming window of 64 samples duration and 3/4 overlap. Each spectrogram is reduced to an 100 dimension vector by KPCA.

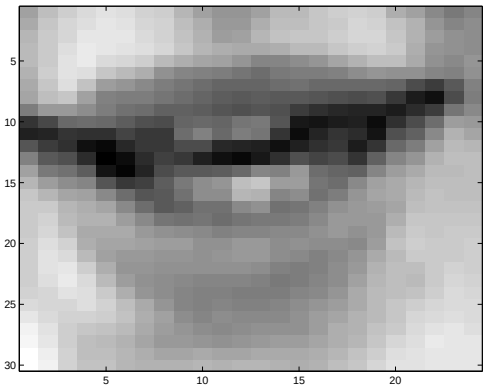


Figure 8.1: A gray scale image of the lip region.

Feature		Acc. predicting on Z	Acc. predicting on X
Video (X)	Audio (Z)	78.30%	73.27%
Video (Z)	Audio (X)	64.15%	91.82%

Table 8.3: The highest classification accuracy achieved by the C^4A algorithm among different combinations of r and γ.

8.1.2.4 Experimental Setting

In our experiment, we consider the video as the view in which both labeled and unlabeled examples are available, and audio as the view in which only unlabeled examples are available, and vice versa. The labels $\mathcal{Y} \in \{0, 1, \ldots, 9\}$ are the ten digits. To test the statistical quality of the audiovisual data, we train and test a linear SVM classifier in audio and in video independently. We use the linear SVM instead of a kernalized SVM since (i) the original data is already processed by KPCA, and (ii) the linear SVM is comparable with our linear algorithm. Classification results

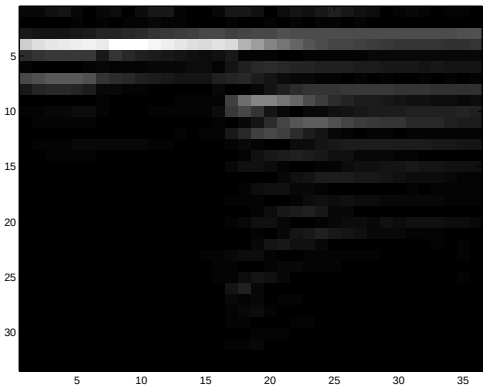


Figure 8.2: A spectrogram of the audio.

Feature		Acc. predicting on Z	Acc. predicting on X
Video (X)	Audio (Z)	77.04%	74.53%
Video (Z)	Audio (X)	63.52%	90.88%

Table 8.4: The highest classification accuracy achieved by the SSM-SVM algorithm among different combinations of r and γ.

in table 8.1.2.2 show that the audiovisual data is statistically asymmetric across views, namely, the classes of audio data are more separable than of the video data.

In Section 8.1.3.1, we first compare our algorithms with the CCA+SVM algorithm and the label-transferred learning method. Then an empirical assessment in a single-view learning case is performed to compare with our algorithm in Section 8.1.3.2.

8.1.3 Testing Result and Analysis

8.1.3.1 Comparison with Alternative Methods

We compare the proposed algorithms with the CCA + SVM algorithm and the label-transferred learning method in terms of predicting accuracy. For the first two approaches, the relationship between the two views are statistically learned, however in the label-transferred learning method, the estimated labels of the paired unlabeled training examples are directly produced by the classifier trained based on the labeled examples. Intuitively, the ratio $r = \frac{m}{n-m}$ between the data size $n-m$ of the labeled examples, and m of the unlabeled examples from the training data is a crucial factor that influences prediction accuracy. The total number of training examples is $n = 740$ in our experiment. In the experiment, we test the prediction accuracy of each approach for different $r \in \{\frac{1}{10}, \frac{1}{6}, \frac{1}{4}, \frac{1}{3}, 1, 5, 9, 12, 14\}$. Note that each of the three approaches yields two classifiers $f_a(x)$ and $f_b(z)$ that are able to classify examples from either view. Accordingly, we provide the classification results for both $f_a(x)$ and $f_b(z)$ from each approach.

Figures 8.3 and 8.4 compare the C^4A algorithm, the SSM-SVM algorithm, the CCA+SVM approach, and the label-transferred learning method in terms of prediction accuracy. As shown in Fig. 8.3(a) and Fig. 8.4(a), the C^4A algorithm achieves the highest classification accuracy among the three algorithms in predicting new examples from the view where no labeled training data is available. Especially, as shown in Fig. 8.3(a) when labeled examples are available on video the C^4A algorithm achieves over 15% accuracy more than the other two. The

label-transferred learning method achieves a comparable performance whenever the paired unlabeled examples are abundant.

As shown in Fig. 8.7(c) and Fig. 8.4(b), when predicting new examples from the view where the labeled examples are available, the performance of the proposed algorithms are comparable to that of label-transferred learning method. The latter even outperforms the C^4A algorithm and the CCA + SVM method when r is small. This is because a semi-supervised multi-view learning method takes care of learning relationship between $\mathcal{X}$ and $\mathcal{Z}$, and between $\mathcal{X}$ and $\mathcal{Y}$, in contrast, the label-transferred learning method only focuses on learning the relationship between $\mathcal{X}$ and $\mathcal{Y}$ or $\mathcal{Z}$ and $\hat{\mathcal{Y}}$.

8.1.3.2 Evaluation with Learning in Single-view

In the single-view learning setting where labeled examples are available, the performance of the trained classifier is greatly affected by the size of the training data. In this experiment we evaluate the change of classification accuracy by increasing training data size in a single-view scenario and compare it to the classification accuracy in the semi-supervised multi-view setting. In the semi-supervised multi-view setting, we choose the classification accuracy for comparison when $r = 1$ for which the proposed algorithms achieve the highest classification accuracy in both predicting audio and video. Evaluations on both audio and video individually are performed.

Figures 8.5(a) and 8.5(b) show that the accuracy when predicting labels on

audio and on video, respectively, improves as the training sample size increases, and compare it to the accuracies achieved by a C^{4}A classifier and by SSM-SVM classifier when $r = 1$. In Fig. 8.5(b), when the prediction is performed on video in which classes are not easily separable the C^{4}A algorithm only uses 370 labeled examples while in the single-view supervised learning it uses around 500 labelled examples.

8.2 Test of the Proposed Generative Approach

In this section, we evaluate our approach and compare with an approach which should upper-bound its performance, 'direct supervision multiview learning' (DSML), wherein a mixture model on (Z, C) is directly learned given a labeled (Z_i, C_i) pair training set.

8.2.1 Test on Synthetic Data

We first consider a 2-class, 3-component, 2-dimensional synthetic example, shown in Fig. 8.6, which represents a formidable challenge for SSML. The ground truth model parameters, based on (7.1), are: $B = \begin{pmatrix} 0.9 & 0.2 & 0.1 \\ 0.1 & 0.8 & 0.9 \end{pmatrix}$, $\underline{\alpha} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]^T$, $N_l = N_u = 437$, A_j a 2×2 identity matrix, $[\mu_{x1}, \mu_{x2}, \mu_{x3}] = [5, 5, 0]$, $[\mu_{z1}, \mu_{z2}, \mu_{z3}] = [10, 5, 0]$, and $\sigma^2 = 1$. Clearly, based on Fig. 1, SSML cannot perform well on this example, since labeled examples are only available for X, yet with X uninformative for discriminating the two components centered at (5,5) and (5, 10). Without

labeled examples for Z drawn from these two components, it is not possible to accurately estimate the B matrix columns for these two components. While overall SSML performance is thus expected to be poor, experiments on this example give some interesting, non-obvious results that are particularly illustrative of the *discrete* nature of the performance (accuracy) sensitivity of the model to random parameter initializations for EM.

We considered a number of experimental trials wherein, to *combat* sensitivity to parameter initialization, for each trial, the EM algorithm was run starting from 20 random parameter initializations and the solution with greatest likelihood (7.1) was chosen. In Fig. 8.7, we plot the average accuracy of these best models across trials, as well as its standard deviation, for both the SSML and DSML scenarios. Each plot shows performance as a function of one of the three parameters σ^2, d, and J, with the other two fixed to the true values. We also estimated the Bayes correct decision rate as 84.4%, based on plugging the true parameter values into (7.17).

There are both expected as well as unexpected observations to make on the results in Fig. 8.7. First, as expected, accuracy is greater under the DSML scenario than under SSML. Second, SSML and DSML achieve the same accuracy when $J = 1$. In fact, in this case, the two (single-component) models make the same predictions for all data points, assigning all points to the majority class, and thus achieving accuracy of $(0.1+0.8+0.9)/3 = 0.6$. More interestingly, we note that the standard deviation on prediction accuracy for SSML is much greater than that for DSML when $J > 2$. The 2-dimensional synthetic data in Fig. 8.6 is still largely sep-

arable to three components after projection onto Z. This leads to relatively little variation in learned models across trials in DSML, and to good accuracy. However, the two components with $[B_{11}, B_{21}]^T = [0.9, 0.1]^T$ and $[B_{12}, B_{22}]^T = [0.2, 0.8]^T$ are totally overlapped after the synthetic data is projected to X, which makes it hard for EM under the SSML scenario to distinguish the two overlapped components and utterly infeasible to accurately estimate the associated true columns of B. One might, accordingly, imagine that the classification accuracy would be uniformly poor, and without large variation, across the experimental trials. However, looking at Fig. 8.7, this is not the case – there is large variation in accuracy with, moreover, quite unexpectedly *good* accuracy over some trials. This phenomenon can be well understood as follows. Note that, even though the inference rule (7.17) sums contributions over all components, if the components are sufficiently well-separated, then one component (e.g. j^*) will dominate the sum, with the MAP decision then reducing to $c^* = \text{argmax}_c B_{cj^*}$. In such case, the correct decision will be made for an example from class k so long as B_{kj^*} is the largest probability, irrespective even of *gross* inaccuracy in the estimated B matrix. By the same token, an incorrect decision will be made if B_{kj^*} is not the largest probability. Thus, for the example in Fig. 8.6, random initialization induces a discrete random effect on classification accuracy, involving the cases where i) $\hat{B}_{11} > \hat{B}_{21}$ and $\hat{B}_{22} > \hat{B}_{12}$ (estimates have same ordering as true values, resulting (surprisingly) in high accuracy); ii) $\hat{B}_{11} < \hat{B}_{21}$ and $\hat{B}_{22} < \hat{B}_{12}$ (estimates do not have same ordering as true values for both components, resulting in grossly poor accuracy); iii) ordering is correct for one component and incorrect for the other (resulting in accuracy

between these two extremes). To more quantitatively analyze this phenomenon, we considered the following idealization of the effects of random initialization on parameter learning for the example in Fig. 8.6. Assume that $J = 3$ and, for the two overlapped components, that the estimated parameter values are $B_{11} = p$, $B_{12} = q$, where $p + q = 1.1$. Depending on the learned model's (p, q) realization, there are three possible prediction accuracies in SSML: (1) when $0.6 < p < 1$ and $0.1 < q < 0.5$, $P_1 = (0.9 + 0.8 + 0.9)/3 = 0.87$; (2) when $0.5 \leq p \leq 0.6$ and $0.5 \leq q \leq 0.6$, $P_2 = (0.9 + 0.2 + 0.9)/3 = 0.67$; (3) when $0.1 < p < 0.5$ and $0.6 < q < 1$, $P_3 = (0.1 + 0.2 + 0.9)/3 = 0.4$. Assuming $p, q \sim \mathcal{U}[0.1, 1]$, *average* prediction accuracy is $P_{avg} = 0.87*4/9+0.67*1/9+0.4*4/9 = 0.64$ with standard deviation of 0.217. Note that these two statistics, under this idealized modeling, are in reasonable agreement with the results shown in Fig. 8.7 for $J = 3$.

8.2.2 Test on an Audiovisual Task

In this section, we apply the proposed algorithm to a lip-reading task. In lip-reading, audio and video are considered as separate views. The data used in our simulation is from [39]. In section 8.2.2.1, we explain the experimental setting. The simulation results are given in section 8.2.2.2.

8.2.2.1 Experimental Setting

In preprocessing the audiovisual data, we follow the same method as in [33]. The audio data and the video data extracted from Grid Corpus are considered as separate views $\mathcal{X}$ and $\mathcal{Z}$, respectively. The training data consists of examples of the form (X_i, Z_i) and (X_i, C_i). Note that in training data $N_l = N_u = 370$. The testing data consists of examples of the form (Z_i, C_i). The labels $\mathcal{C} \in \{0, 1, \ldots, 9\}$ are the ten digits.

	$J = 40$	$J = 60$	$J = 80$	$J = 100$
$\sigma^2 = 0.8$	$55.7\% \pm 7.2$	$54.9\% \pm 5.5$	$53.0\% \pm 3.8$	$50.2\% \pm 3.8$
$\sigma^2 = 1$	$49.1\% \pm 4.5$	$57.4\% \pm 5.8$	$53.6\% \pm 3.5$	$53.4\% \pm 3.2$
$\sigma^2 = 1.2$	$53.4\% \pm 6.2$	$55.3\% \pm 4.5$	$52.5\% \pm 1.0$	$49.1\% \pm 5.2$

Table 8.5: Digit prediction accuracies with inferences made solely using Z as input, for varying σ^2 and J, using the proposed model.

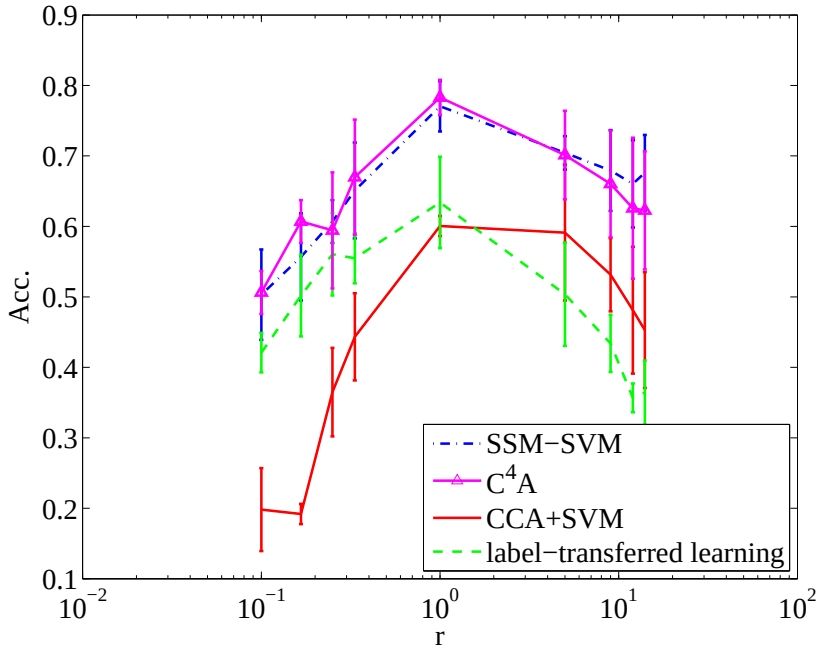
	$J = 40$	$J = 60$	$J = 80$	$J = 100$
$\sigma^2 = 0.8$	$70.1\% \pm 6.0$	$69.1\% \pm 1.8$	$73.8\% \pm 5.8$	$70.4\% \pm 7.1$
$\sigma^2 = 1$	$68.3\% \pm 5.7$	$68.7\% \pm 3.4$	$72.8\% \pm 3.4$	$69.4\% \pm 2.9$
$\sigma^2 = 1.2$	$70.8\% \pm 2.8$	$68.5\% \pm 4.3$	$69.4\% \pm 5.0$	$68.7\% \pm 2.2$

Table 8.6: Prediction accuracies for inference based on $\mathcal{Z}$ with varying σ^2 and J, using mixtures trained based on supervised (Z_i, C_i) pairs.

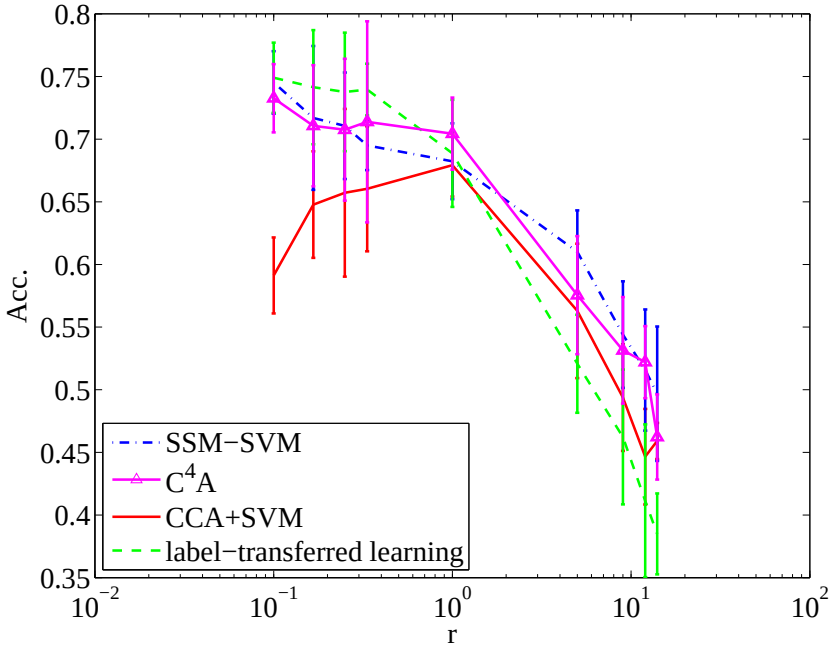
8.2.2.2 Experimental Results

In table 8.5, we present prediction accuracies achieved by our proposed method, in making digit predictions using only Z for different J and σ^2 when $d = 10$ in the audiovisual data. Note that the highest prediction accuracy was achieved when $d = 10$.

The results in table 8.6 show that the highest accuracy achieved by a mixture learned in a supervised fashion given labeled pairs (Z_i, C_i) is 73.8%, which is comparable to the 72.83% accuracy achieved by a discriminative model, as reported in [33]. From tables 8.5 and 8.6, we observe that the highest prediction accuracy achieved by our proposed multi-view model, which learns without any labeled examples involving $\mathcal{Z}$, is 57.4%. As expected, there is reduction in accuracy, compared with a classifier learned in a standard supervised fashion. However, 57.4% accuracy still represents a substantial prediction capability on this ten-class problem space.

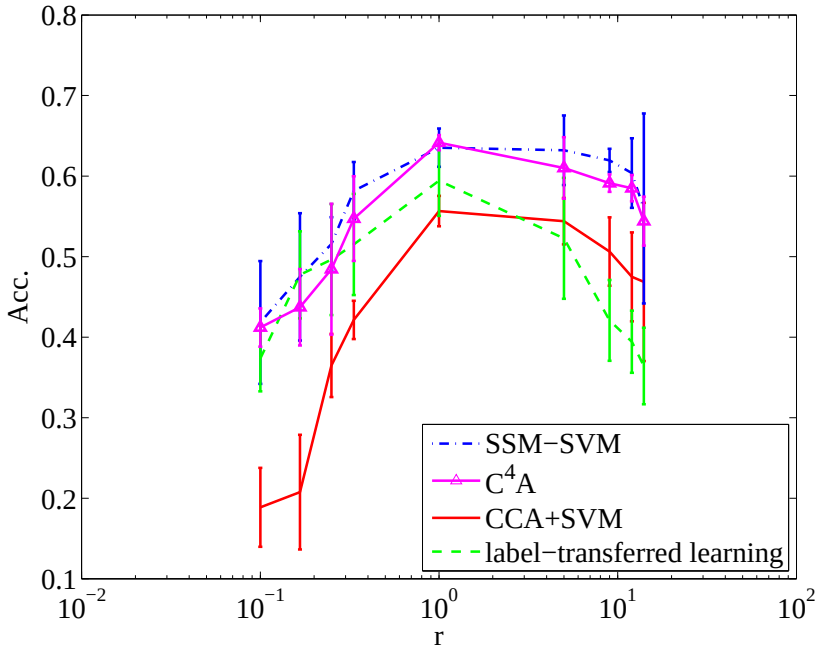


(a) The prediction is performed on label of audio.

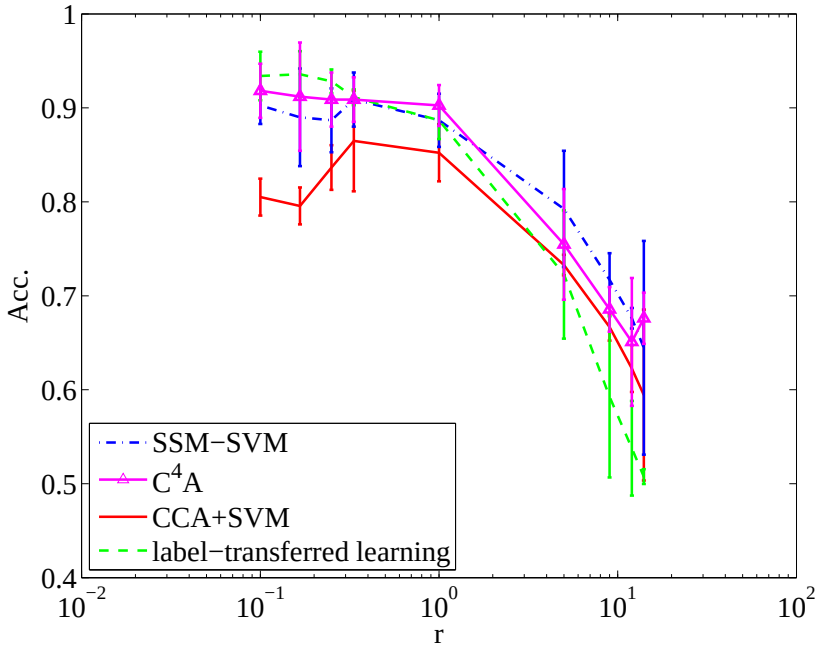


(b) The prediction is performed on label of video.

Figure 8.3: Comparison of the proposed algorithms with the CCA + SVM approach and the label-transferred learning approach in terms of prediction accuracy when the labled examples are only available on video whilst paired unlabeled examples are available on both audio and video.

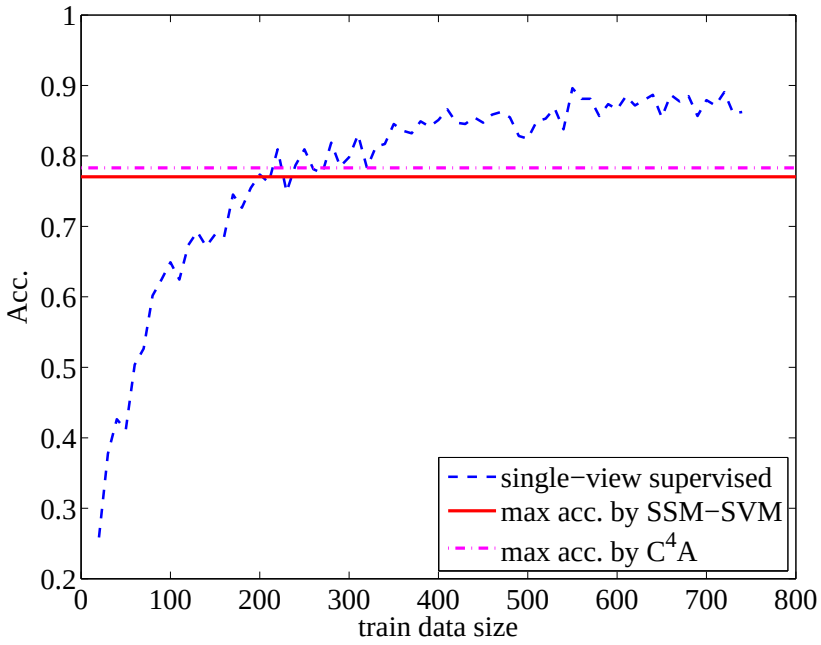


(a) The prediction is performed on video.

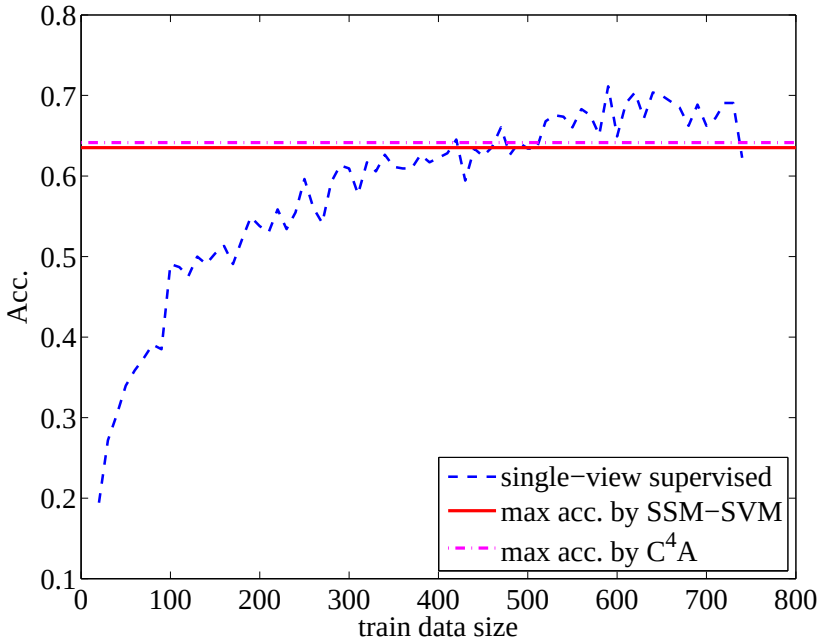


(b) The prediction is performed on audio.

Figure 8.4: Comparison of the proposed algorithms with the CCA + SVM approach and the label-transferred learning approach in terms of prediction accuracy when the labled examples are available on audio whilst paired unlabeled examples are available on both audio and video.



(a) The classification is performed on audio.



(b) The classification is performed on video.

Figure 8.5: The accuracies achieved for different training sample sizes under single-view supervised learning are compared to the best accuracy achieved by the C^{4}A algorithm when $r = 1$. The linear SVM technique is applied in the previous scenario. The total training size is 740.

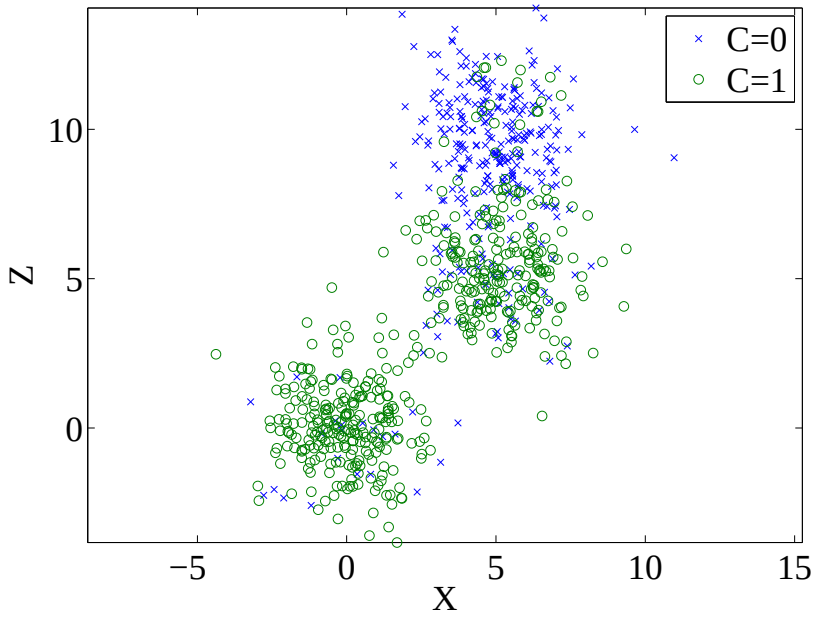
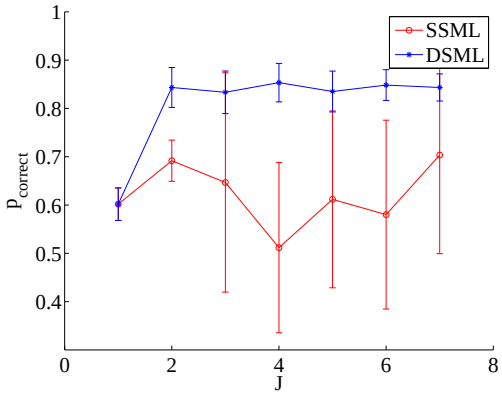
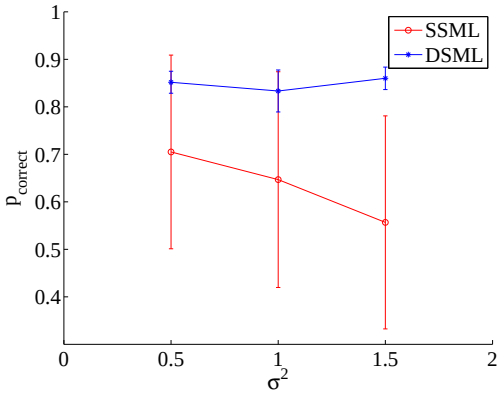


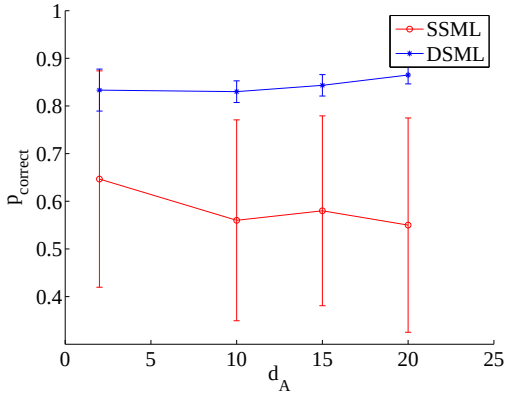
Figure 8.6: The synthetic data set, with two one-dimensional (X and Z) views.



(a) Accuracies of prediction on $\mathcal{Z}$ versus J when other parameters are fixed to true values: $\sigma^2 = 1$, and $d = 2$.



(b) Accuracies of prediction on $\mathcal{Z}$ versus σ^2 when other parameters are fixed to true values: $d = 2$, and $J = 3$.



(c) Accuracies of prediction on $\mathcal{Z}$ versus d when other parameters are fixed to true values: $\sigma^2 = 1$, and $J = 3$.

Figure 8.7: Plot of prediction accuracy versus one of the three parameters σ^2, d_A, and J while the other two are fixed to the true values. An upper bound of the prediction accuracy achieved by the proposed EM algorithm by calculating Bayes error rate is 84.4%.

Chapter 9: CONCLUSION

9.1 Summary

In this work, we introduced the surrogate supervision multi-view learning, which is a semi-supervised multi-view learning problem. To solve the surrogate supervision multi-view learning problem, we proposed both discriminative model approach and generative model approach. For the discriminative model approach, we proposed the C^4A algorithm and the SSM-SVM algorithm which combine the data mapping stage (between views) and classifier training stage instead of separating them. For the generative model approach, we proposed a semi-supervised Gaussian model. In the simulations, we showed that the proposed C^4A and SSM-SVM algorithms perform better than the alternative methods in terms of prediction accuracy. Especially, the simulation showed that the C^4A and SSM-SVM algorithm achieve higher prediction than the CCA + SVM method which separates the mapping stage and classier training stage. For the generative model approach, We developed a novel EM algorithm, with a reduced-complexity E-step, to estimate the proposed mixtures. The E-step formulation given here can also be used to reduce complexity of the E-step in the standard EM algorithm for MFA.

9.2 Contributions

1. We introduced surrogate supervision multi-view learning which is a special case of multi-view learning. In surrogate supervision multi-view learning, labels are provided on limited views. The goal of surrogate supervision multi-view learning is to obtain labels on the view where labels are missing.

2. We solved the surrogate supervision multi-view learning using generative approach. We proposed the C^4A algorithm and the SSM-SVM algorithm which combine the relationship learning stage and the classifier training stage into a single stage.

3. We proposed a generative, semi-supervised mixtures of factors analyzers model to solve the surrogate supervision multi-view learning. We developed a novel EM algorithm, with a reduced-complexity E-step, to estimate the proposed mixtures.

9.3 Publications

Here is a list of publications based on the work in this thesis:

1. G. Jin, R. Raich,"Hinge Loss Bound Approach for Surrogate Supervision Multi-view Learning", submitted to Pattern Recognition Letter for review

2. G. Jin, R. Raich,"On Surrogate Supervision Multi-view Learning", in *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on.*

IEEE, 2012, pp.1-6

3. G. Jin, R. Raich, D. J. Miller, "A Generative Semi-supervised Model for Multi-view Learning When Some Views Are Label-free", in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* IEEE, submitted for review

9.4 Future Work

Although intensive experiments have been conducted to certify our proposed Gaussian mixtures for SSML, we still lack a theoretical proof of the algorithm. It would be also appealing to look at the dependency between two views in SSML as we still do not understand how the dependency between two views influences the performances of our proposed algorithms. Finally, a potential research direction is that we can consider the SSML in multi-instance learning scenario.

Bibliography

[1] 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*. ACM, 2000, pp. 86–93.

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

[3] S.J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, 2010.

[4] T. Evgeniou, C.A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *Journal of Machine Learning Research*, vol. 6, no. 1, pp. 615, 2006.

[5] W. Dai, Q. Yang, G.R. Xue, and Y. Yu, "Self-taught clustering," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 200–207.

[6] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 109–117.

[7] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.

[8] M.A. Krogel and T. Scheffer, "Multi-relational learning, text mining, and semi-supervised learning for functional genomics," *Machine Learning*, vol. 57, no. 1, pp. 61–81, 2004.

[9] X. Wan, "Co-training for cross-lingual sentiment classification," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 235–243.

[10] S. Kiritchenko and S. Matwin, "Email classification with co-training," in *Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*. Citeseer, 2001, p. 8.

[11] H. Meng, D.R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Generic object recognition by combining distinct features in machine learning," 2005.

[12] A. Pezeshki, M.R. Azimi-Sadjadi, and L.L. Scharf, "Undersea target classification using canonical correlation analysis," *Oceanic Engineering, IEEE Journal of*, vol. 32, no. 4, pp. 948–955, 2007.

[13] K. Livescu and M. Stoehr, "Multi-view learning of acoustic features for speaker recognition," in *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE, 2009, pp. 82–86.

[14] K. Chaudhuri, S.M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 129–136.

[15] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A.Y. Ng, "Multimodal deep learning," in *Proc. ICML*, 2011.

[16] S. Kakade and D. Foster, "Multi-view regression via canonical correlation analysis," *Learning Theory*, pp. 82–96, 2007.

[17] B. Fortuna, "Kernel canonical correlation analysis with applications," 2004.

[18] A. Vinokourov, J. Shawe-Taylor, and N. Cristianini, "Inferring a semantic representation of text via cross-language correlation analysis," *Advances in neural information processing systems*, vol. 15, pp. 1473–1480, 2003.

[19] Y. Li and J. Shawe-Taylor, "Using KCCA for japanese–english cross-language information retrieval and document classification," *Journal of intelligent information systems*, vol. 27, no. 2, pp. 117–133, 2006.

[20] A. Tripathi, A. Klami, and S. Virpioja, "Bilingual sentence matching using kernel cca," in *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*. IEEE, 2010, pp. 130–135.

[21] J. Farquhar, D. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmak, "Two view learning: SVM-2K, theory and practice," in *Advances in Neural Information Processing Systems 19: Proceedings of the 2005 Conference*. The MIT Press, 2005, vol. 19, p. 355.

[22] M.H. Yang and N. Ahuja, "Detecting human faces in color images," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*. IEEE, 1998, vol. 1, pp. 127–130.

[23] C.C. Chiang, W.K. Tai, M.T. Yang, Y.T. Huang, and C.J. Huang, "A novel method for detecting lips, eyes and faces in real time," *Real-Time Imaging*, vol. 9, no. 4, pp. 277–287, 2003.

[24] H. Hongo, M. Ohya, M. Yasumoto, Y. Niwa, and K. Yamamoto, "Focus of attention for face and hand gesture recognition using multiple cameras," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*. IEEE, 2000, pp. 156–161.

[25] S. Kawato and J. Ohya, "Real-time detection of nodding and head-shaking by directly detecting and tracking the between-eyes," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*. IEEE, 2000, pp. 40–45.

[26] C.H. Lee, J.S. Kim, and K.H. Park, "Automatic human face location in a complex background using motion and color information," *Pattern recognition*, vol. 29, no. 11, pp. 1877–1889, 1996.

[27] Y. Dai and Y. Nakano, "Face-texture model based on sgld and its application in face detection in a color scene," *Pattern recognition*, vol. 29, no. 6, pp. 1007–1017, 1996.

[28] IH Ellis and JR Lishman, "Automatic extraction of face-feature," *Pattern Recognition Letters*, pp. 183–187, 1987.

[29] D. Maio and D. Maltoni, "Real-time face location on gray-scale static images," *Pattern Recognition*, vol. 33, no. 9, pp. 1525–1539, 2000.

[30] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[31] J. Weston and C. Watkins, "Multi-class support vector machines," Tech. Rep., Citeseer, 1998.

[32] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[33] G. Jin and R. Raich, "On surrogate supervision multiview learning," in *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*. IEEE, 2012, pp. 1–6.

[34] D.J. Miller and H.S. Uyar, "A mixture of experts classifier with learning based on both labelled and unlabelled data," *Advances in neural information processing systems*, pp. 571–577, 1997.

[35] G.J. McLachlan, RW Bean, and D. Peel, "A mixture model-based approach to the clustering of microarray expression data," *Bioinformatics*, vol. 18, no. 3, pp. 413–422, 2002.

[36] Z. Ghahramani, G.E. Hinton, et al., "The EM algorithm for mixtures of factor analyzers," Tech. Rep., Technical Report CRG-TR-96-1, University of Toronto, 1996.

[37] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[38] A. Frank and A. Asuncion, "UCI machine learning repository," 2010.

[39] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 120, pp. 2421, 2006.

[40] D.R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

APPENDICES

Appendix A: Canonical Correlation Analysis

In multi-view learning, we are interested in learning the relationship between two views. The canonical correlation analysis (CCA) algorithm is a way to find a mapping between two views. Basically, the CCA algorithm maps two sets of data respectively from views $\mathcal{X}$ and $\mathcal{Z}$ into a common space where the projected data from the tow views are maximally correlated.

The CCA algorithm can be formulated as follows:

$$\begin{aligned} &\min_{a,b} \frac{1}{n} \sum_{i=1}^{n} ||a^T x_i - b^T z_i||_2^2 \\ &\text{subject to} \quad a^T R_X a = 1,\ b^T R_Z b = 1, \end{aligned} \tag{A.1}$$

where $R_X = \frac{1}{n} \sum_i x_i x_i^T$ and $R_Z = \frac{1}{n} \sum_i z_i z_i^T$. For simplicity, we assume that both the x_is and the z_is are zero mean. This is equivalent to seeking a and b such that the correlation of $Q = a^T X$ and $P = b^T Z$:

$$\rho(Q, P) = \frac{a^T \Sigma_{XZ} b}{\sqrt{a^T \Sigma_{XX} a} \sqrt{b^T \Sigma_{ZZ} b}} \tag{A.2}$$

is maximized [40], where $\Sigma_{XX} = cov(X, X)$. Let $c = \sqrt{\Sigma_{XX}} \cdot a$, $d = \sqrt{\Sigma_{ZZ}} \cdot b$, (A.2) becomes

$$\rho(Q, P) = \frac{c^T \Sigma_{XX}^{-\frac{1}{2}} \Sigma_{XZ} \Sigma_{YY}^{-\frac{1}{2}} d}{\sqrt{c^T c} \sqrt{d^T d}}. \tag{A.3}$$

According to the Cauchy-Schwarz inequality, an upper bound can be obtained for nominator of RHS of (A.3) as following:

$$c^T \Sigma_{XX}^{-\frac{1}{2}} \Sigma_{XZ} \Sigma_{ZZ}^{-\frac{1}{2}} d \;\; \leq \;\; (c^T \Sigma_{XX}^{-\frac{1}{2}} \Sigma_{XZ} \Sigma_{ZZ}^{-\frac{1}{2}} \Sigma_{ZZ}^{-\frac{1}{2}} \Sigma_{ZX} \Sigma_{XX}^{-\frac{1}{2}} c)^{\frac{1}{2}} (d^T d)^{\frac{1}{2}}. \tag{A.4}$$

Applying (A.4) to (A.3), we have

$$\rho(Q, P) \leq \frac{(c^T \Sigma_{XX}^{-\frac{1}{2}} \Sigma_{XZ} \Sigma_{ZZ}^{-1} \Sigma_{ZX} \Sigma_{XX}^{-\frac{1}{2}} c)^{\frac{1}{2}}}{(c^T c)^{\frac{1}{2}}}. \tag{A.5}$$

The equality of (A.5) holds when c is the eigenvector of $\Sigma_{XX}^{-\frac{1}{2}} \Sigma_{XZ} \Sigma_{ZZ}^{-1} \Sigma_{ZX} \Sigma_{XX}^{-\frac{1}{2}}$ with the maximum eigenvalue.

Note that the CCA algorithm is used in one of the alternative solutions to the SSML problem (which is discussed in 4.1).

Appendix B: Principal Component Analysis

The principal component analysis (PCA) is a linear dimension reduction technique. The PCA algorithm seeks a few orthogonal linear combinations (principal components) of the original data with the largest variance. The first principal component (PC) s_1 is the linear combination with the largest variance. We have $s_1 = x^T w_1$, where

$$w_1 = \arg\max_{||w=1||} Var\{x^T w\}. \tag{B.1}$$

The second PC is the linear combination with the second largest variance and is orthogonal to the first PC, and so on. We can keep the a number of PCs with the largest variances and discard the rest. For finding the PCs, we can calculate the covariance matrix of the original data matrix

$$\Sigma = \frac{1}{n} X X^T. \tag{B.2}$$

We can rewrite Σ as

$$\Sigma = U \Lambda U^T, \tag{B.3}$$

where $\Lambda = diag(\lambda_1, ..., \lambda_q)$ is a diagonal matrix with eigenvalues $\lambda_1 \leq ... \leq \lambda_p$, and U is accordingly an eigenvector matrix. The the PCs can be given by S, where

$$S = U^T X. \tag{B.4}$$

Appendix C: Proof

For any $P, Q \in \mathcal{R}$, the following inequality holds:

$$(P+Q)_+ \leq P_+ + |Q|. \tag{C.1}$$

Proof: We examine the inequality in (C.1) in the following cases

- When $P+Q \leq 0$, $(P+Q)_+ = 0$. Since $P_+ \geq 0$ and $|Q| \geq 0$, $P_+ + |Q| \geq 0$. Thus (C.1) holds.
- When $P+Q > 0$, $(P+Q)_+ = P+Q$. Since $P_+ \geq P$ and $|Q| \geq Q$, (C.1) holds.

We would like to obtain an upper bound on $(1-g(z)y)_+$:

$$(1-g(z)y)_+ \leq (1-h(x)y)_+ + |h(x)-g(z)|. \tag{C.2}$$

We start by substituting $P = 1-h(x)y$ and $Q = (h(x)-g(z))y$ in (C.1):

$$(1-g(z)y)_+ \leq (1-h(x)y)_+ + |y| \cdot |h(x)-g(z)| \tag{C.3}$$

Since $y \in \{+1, -1\}$ and $|y| = 1$,

$$(1-g(z)y)_+ \leq (1-h(x)y)_+ + |h(x)-g(z)|. \tag{C.4}$$

Appendix D: Proof

D.1

In SSML, the following inequality holds:

$$\begin{aligned}&\sum_{k\neq l} E[(2-(g_l(z)-g_k(z)))_+ I(y=l)]\\ &\leq \sum_{k\neq l} E[(2-(h_l(x)-h_k(x)))_+ I(y=l)] + \sum_{k} E[|g_k(z)-h_k(x)|]\\ &\qquad +(K-2)\max_k E[|g_k(z)-h_k(x)|]. \end{aligned} \tag{D.1}$$

Proof: Substituting $P = (2-(h_l(x)-h_k(x)))_+ I(y=l)$, $Q = |(h_l(x)-g_l(z)) - (h_k(x)-g_k(x))|I(y=l)$ in (C.1), we obtain:

$$\begin{aligned}&\sum_{k\neq l} E[(2-(g_l(z)-g_k(z)))_+ I(y=l)]\\ &\qquad \leq \sum_{k\neq l} E[(2-(h_l(x)-h_k(x)))_+ I(y=l)]\\ &\qquad\quad + \sum_{k\neq l} E[|(h_l(x)-g_l(z)) - (h_k(x)-g_k(x))|I(y=l)]. \end{aligned} \tag{D.2}$$

Using triangle inequality $|(h_l(x) - g_l(z)) - (h_k(x) - g_k(z))| \leq |h_l(x) - g_l(z)| + |h_k(x) - g_k(z)|$, we can bound the second term on the RHS of (D.2) by

$$\sum_{k \neq l} E[|((h_l(x) - g_l(z)) - (h_k(x) - g_k(z)|I(y = l)] \\ \leq \sum_{k \neq l} E[|h_l(x) - g_l(z)|I(y = l)] + \sum_{k \neq l} E[|h_k(x) - g_k(z)|I(y = l)] \quad \text{(D.3)}$$

The two terms on the RHS of (D.3) can be simplified respectively as follows:

$$\sum_{k \neq l} E[|h_l(x) - g_l(z)|I(y = l)] = (K - 1) \sum_{l} E[|h_l(x) - g_l(z)|I(y = l)] \quad \text{(D.4)}$$

$$\sum_{k \neq l} E[|h_k(x) - g_k(z)|I(y = l)] = \sum_{k} E[|h_k(x) - g_k(z)|] \\ - \sum_{k} E[|h_k(x) - g_k(z)|I(y = k)] \quad \text{(D.5)}$$

Substituting (D.4) and (D.5) into the RHS of (D.3), we obtain

$$\sum_{k} E[|g_k(z) - h_k(x)|] + (K - 2) \sum_{k} E[|g_k(z) - h_k(x)|I(y = k)].$$

Since $\sum |g_k(z) - h_k(x)|I(y = k) \leq \max_k |g_k(z) - h_k(x)|$, we further bound the RHS of (D.3) using

$$\leq \sum_{k} E[|g_k(z) - h_k(x)|] + (K - 2) \max_{k} E[|g_k(z) - h_k(x)|]. \quad \text{(D.6)}$$

Substituting (D.6) into the RHS of (D.3) yields the desired bound in (D.1).