A GENERATIVE SEMI-SUPERVISED MODEL FOR MULTI-VIEW LEARNING WHEN SOME VIEWS ARE LABEL-FREE

Gaole Jin[1]*, Raviv Raich*[1]*, and David J. Miller*[2]

[1] School of EECS, Oregon State University, Corvallis, OR 97731-5501
[2] EE Dept., The Pennsylvania State University, Rm. 227-C EE West Bldg., University Park, PA 16802

ABSTRACT

We consider multi-view classification for the challenging scenario where, for some views, there are *no* labeled training examples. Several discriminative approaches have been recently proposed for special instances of this problem. Here, alternatively, we propose a generative semi-supervised mixture model across all views which, via marginalization, flexibly performs exact class inference, given any subset of available views. The proposed model is an extension of semi-supervised mixtures to a multi-view setting, as well as a semi-supervised extension of mixtures of factors analyzers (MFA)[1]. A novel EM algorithm with a computationally efficient E-step is derived for learning our multi-view model. Specialization of this formulation to the standard MFA problem also gives a reduced complexity E-step, compared to the original EM algorithm proposed for MFA. Our multi-view method is experimentally demonstrated on digit recognition using audio and lip video views, achieving competitive results with alternative, discriminative approaches.

Index Terms— multi-view learning, semi-supervised learning, mixture of factors analyzers, Expectation-Maximization

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 $\mathbf{X} = (X^{(1)}, \ldots, X^{(N_v)})$ [2], $X^{(i)}$ 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 here: $\mathbf{X} = \{X, Z\}, X \in \mathcal{R}^{d_x}, Z \in \mathcal{R}^{d_z}$, and labels $C \in \mathcal{C}$. 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 $\mathcal{S}_l = \{N_u + 1, N_u + 2, \ldots, N_u + N_l\}$. Note that the two sets do not share any common x_i. Several previous works have investigated this problem. In [3], 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 [2], 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 [4] 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 [5].

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 label $C_i, i \in \mathcal{S}_l$ 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. We assume the Gaussian components follow a factor analysis model, i.e., mixture component j is generated according to $[x_i, z_i]^T = [\mu_{xj}^T, \mu_{zj}^T]^T + [A_{xj}^T A_{zj}^T]^T v + n$ where $v \sim \mathcal{N}(0, I)$ and $n \sim \mathcal{N}(0, I)$. The associated incomplete data likelihood for our model is:

$$f_{\text{inc}}(\mathcal{X}_l, \mathcal{X}_u; \theta) = \Big(\prod_{i \in S_l} \sum_{j=1}^{J} \phi(x_i; \mu_{xj}, A_{xj} A_{xj}^T + \sigma^2 I) B_{c_i j} \alpha_j\Big) \cdot \Big(\prod_{i \in S_u} \sum_{j} \phi(x_i, z_i; \mu_j, A_j A_j^T + \sigma^2 I) \alpha_j\Big). \quad (1)$$

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 $B_{\cdot j} =$

$[B_{1j} \dots B_{Cj}]$ ($\sum_c B_{cj} = 1$ 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 [6], 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 *hyper-parameter*, chosen to ensure well-conditioned covariance matrices and held fixed during (EM) learning of all other parameters.

An EM algorithm for (locally) maximizing (1) is developed as follows. We naturally introduce as hidden data within the EM framework [7] the mixture component of origin for each sample, $J_i, i = 1, \dots, N_u + N_l$. 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 $V_i \sim \mathcal{N}(0, I_d)$, 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 (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, factor vector set and component of origin, respectively. 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:

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

The EM auxiliary function for the log-likelihood [7] is given by

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

where Δ corresponds to the terms that are constant with respect to θ such as $E_{v_i}[\log \phi(v_i; 0, I)]$. 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

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

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.*

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

$$P(j|x_i, z_i) = \frac{\phi([\tbinom{x_i}{z_i}]; \mu_j, A_j A_j^T + \sigma^2 I) \alpha_j}{\sum_k \phi(x_i; \mu_k, A_k A_k^T + \sigma^2 I) \alpha_k} \quad (3)$$

$$E[v_i|x_i, z_i, j] = A_j^T (A_j A_j^T + \sigma^2 I)^{-1} ([\tbinom{x_i}{z_i}] - \mu_j) \quad (4)$$

$$E[v_i|x_i, j] = A_{xj}^T (A_{xj} A_{xj}^T + \sigma^2 I)^{-1} (x_i - \mu_{xj}) \quad (5)$$

$$E[v_i v_i^T|x_i, z_i, j] = I - A_j^T (A_j A_j^T + \sigma^2 I)^{-1} A_j + A_j^T \cdot \quad (6)$$
$$(A_j A_j^T + \sigma^2 I)^{-1} ([\tbinom{x_i}{z_i}] - \mu_j)([\tbinom{x_i}{z_i}] - \mu_j)^T (A_j A_j^T + \sigma^2 I)^{-1} A_j$$

$$E[v_i v_i^T|x_i, j] = I - A_{xj}^T (A_{xj} A_{xj}^T + \sigma^2 I)^{-1} A_{xj} + A_{xj}^T \cdot \quad (7)$$
$$(A_{xj} A_{xj}^T + \sigma^2 I)^{-1} (x_i - \mu_{xj})(x_i - \mu_{xj})^T (A_{xj} A_{xj}^T + \sigma^2 I)^{-1} A_{xj}.$$

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:

$$(Q_j Q_j^T + \sigma^2 I)^{-1} = \frac{1}{\sigma^2} I - \frac{1}{\sigma^2} Q_j (\sigma^2 I + Q_j^T Q_j)^{-1} Q_j^T. \quad (8)$$

This can be applied, respectively, for $Q_j = A_j$ in (4) and (6) and for $Q_j = A_{xj}$ in (5) and (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:

$$E[v_i|x_i, z_i, j] = (\sigma^2 I + M_j)^{-1} A_j^T ([\tbinom{x_i}{z_i}] - \mu_j) \quad (9)$$

$$E[v_i|x_i, j] = (\sigma^2 I + M_j)^{-1} A_{xj}^T (x_i - \mu_{xj}) \quad (10)$$

$$E[v_i v_i^T|x_i, z_i, j] = \sigma^2 (\sigma^2 I + M_j)^{-1} + \quad (11)$$
$$(\sigma^2 I + M_j)^{-1} A_j^T ([\tbinom{x_i}{z_i}] - \mu_j)([\tbinom{x_i}{z_i}] - \mu_j)^T A_j (\sigma^2 I + M_j)^{-1}$$

$$E[v_i v_i^T|x_i, j] = \sigma^2 (\sigma^2 I + M_{xj})^{-1} + \quad (12)$$
$$(\sigma^2 I + M_{xj})^{-1} A_{xj}^T (x_i - \mu_{xj})(x_i - \mu_{xj})^T A_{xj} (\sigma^2 I + M_{xj})^{-1}.$$

Note that this simplification of the E-step, without any approximation, is to our knowledge novel, and can also be applied to reduce complexity of the E-step in the standard, original EM algorithm formulation for mixtures of factors analyzers [6].

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:

$$\alpha_j^{(n+1)} = \frac{\sum_{i \in S_l} P(j|x_i, c_i) + \sum_{i \in S_u} P(j|x_i, z_i)}{N_l + N_u} \quad (13)$$

$$B_{cj}^{(n+1)} = \frac{\sum_{i \in S_l : c_i = c} P(j|x_i, c_i)}{\sum_{i \in S_l} P(j|x_i, c_i)} \quad (14)$$

$$[A_{xj}\ \mu_{xj}]^{(n+1)} = (\sum_{i\in S_l} x_i E[[v_i;1]|x_i,j]^T P(j|x_i,c_i)+ \quad (15)$$
$$\sum_{i\in S_u} x_i E[[v_i;1]|x_i,z_i,j]^T P(j|x_i,z_i))\cdot$$
$$(\sum_{i\in S_l} E[[v_i;1][v_i;1]^T|x_i,j]P(j|x_i,c_i)+$$
$$\sum_{i\in S_u} E[[v_i;1][v_i;1]^T|x_i,z_i,j]P(j|x_i,z_i))^{-1}$$
$$[A_{zj}\ \mu_{zj}]^{(n+1)} = (\sum_{i\in S_u} z_i E[[v_i;1]|x_i,z_i,j]^T P(j|x_i,z_i))\cdot \quad (16)$$
$$(\sum_{i\in S_u} E[[v_i;1][v_i;1]^T|x_i,z_i,j]P(j|x_i,z_i))^{-1}$$

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:

$$P(c|q) = \frac{\sum_j f(q|j)P(c|j)P(j)}{\sum_j f(q|j)P(j)} = \quad (17)$$
$$\frac{\sum_j \phi(q;\mu_{qj},A_{qj}A_{qj}^T+\sigma^2 I)B_{cj}\alpha_j}{\sum_j \phi(q;\mu_{qj},A_{qj}A_{qj}^T+\sigma^2 I)\alpha_j},$$

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.

3. NUMERICAL EVALUATION

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.

3.1. Test on Synthetic Data

We first consider a 2-class, 3-component, 2-dimensional synthetic example, shown in Fig. 1, which represents a formidable challenge for SSML. The ground truth model parameters, based on (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.

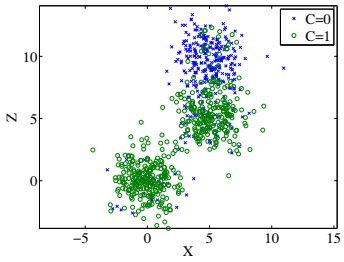
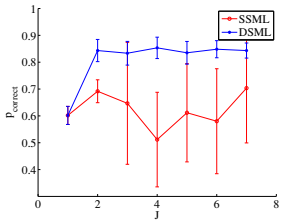


Fig. 1. The synthetic data set, with two one-dimensional (X and Z) views.

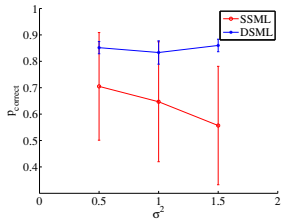
We considered a number of experimental trials wherein, to *combat* sensitivity to parameter initialization, for each trial, the EM algorithm was run starting from 20 random parameter initializations and the solution with greatest likelihood (1) was chosen. In Fig. 2, we plot the average accuracy of these 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 (17).

There are both expected as well as unexpected observations to make on the results in Fig. 2. 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. 1 is still largely separable 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. 2, 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 (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

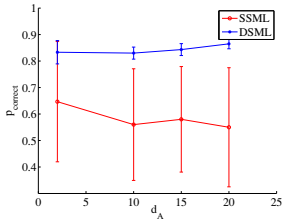
for an example from class k so long as B_{kj^*} is the largest probability, irrespective even of *gross* inaccuracy in the estimated B matrix. By the same token, an incorrect decision will be made if B_{kj^*} is not the largest probability. Thus, for the example in Fig. 1, random initialization induces 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. 1. 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. 2 for $J = 3$.



(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 $J = 3$.

Fig. 2. 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%.

3.2. 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 [8]. In section 3.2.1, we explain the experimental setting. The simulation results are given in section 3.2.2.

3.2.1. Experimental Setting

In preprocessing the audiovisual data, we follow the same method as in [2]. 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 domain of the labels C_i is $\mathcal{C} = \{0, 1, \ldots, 9\}$, i.e. the ten digits.

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

Table 1. 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%±6.0	69.1%±1.8	73.8%±5.8	70.4%±7.1
$\sigma^2 = 1$	68.3%±5.7	68.7%±3.4	72.8%±3.4	69.4%±2.9
$\sigma^2 = 1.2$	70.8%±2.8	68.5%±4.3	69.4%±5.0	68.7%±2.2

Table 2. 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.

3.2.2. Experimental Results

In table 1, 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 2 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 [2]. From tables 1 and 2, 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.

4. CONCLUSION

This paper proposed a generative, semi-supervised mixtures of factors analyzers model to solve the surrogate supervision multi-view learning problem. 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. We evaluated our method in comparison with a supervised learning approach (which serves as a performance upper bound target) both on synthetically generated mixtures and on a two-view lip-reading task. In future we may consider tasks involving many views, for which our model's exact inference capability could be advantageous.

5. REFERENCES

[1] M.E. Tipping and C.M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural computation*, vol. 11, no. 2, pp. 443–482, 1999.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.