Multi-View Dimensionality Reduction via Canonical Correlation Analysis

Dean P. Foster, University of Pennsylvania
Rie Johnson, RJ Research Consulting
Sham M. Kakade, University of Pennsylvania
Tong Zhang, Rutgers University

Abstract

We analyze the multi-view regression problem where we have two views $X = (X^{(1)}, X^{(2)})$ of the input data and a target variable Y of interest. We provide sufficient conditions under which we can reduce the dimensionality of X (via a projection) without loosing predictive power of Y. Crucially, this projection can be computed via a Canonical Correlation Analysis only on the unlabeled data. The algorithmic template is as follows: with unlabeled data, perform CCA and construct a certain projection; with the labeled data, do least squares regression in this lower dimensional space. We show how, under certain natural assumptions, the number of labeled samples could be significantly reduced (in comparison to the single view setting) — in particular, we show how this dimensionality reduction does not loose predictive power of Y (thus it only introduces little bias but could drastically reduce the variance).

We explore two separate assumptions under which this is possible and show how, under either assumption alone, dimensionality reduction could reduce the labeled sample complexity. The two assumptions we consider are a *conditional independence* assumption and a *redundancy* assumption. The typical conditional independence assumption is that conditioned on Y the views $X^{(1)}$ and $X^{(2)}$ are independent — we relax this assumption to: conditioned on some hidden state H the views $X^{(1)}$ and $X^{(2)}$ are independent. Under the redundancy assumption, we show that the best predictor from each view is roughly as good as the best predictor using both views.

1 Introduction

In recent years, the "multi-view" approach has been receiving increasing attention as a paradigm for semi-supervised learning. In the two view setting, there are views (sometimes in a rather abstract sense) $X^{(1)}$ and $X^{(2)}$ of the data, which co-occur, and there is a target variable Y of interest. Throughout the paper, we consider $X^{(1)}$ and $X^{(2)}$ as random vectors, and Y as a real valued random variable. The setting is one where it is easy to obtain unlabeled samples $(X^{(1)}, X^{(2)})$ but the labeled samples $(X^{(1)}, X^{(2)}, Y)$ are more scarce. The goal is to implicitly learn about the target Y via the relationship between $X^{(1)}$ and $X^{(2)}$.

We work in a setting where we have a joint distribution over $(X^{(1)}, X^{(2)}, Y)$, where all $X^{(1)}$ and $X^{(2)}$ are vectors (of arbitrarily large dimension) and $Y \in \mathbb{R}$.

This work focuses on the underlying assumptions in the multi-view setting and provides an algorithm which exploits these assumptions. We *separately* consider two natural assumptions, a conditional independence assumption and a redundancy assumption. Our work here builds upon the work in Ando and Zhang [2007] and Kakade and Foster [2007], summarizing the close connections between the two.

Ando and Zhang [2007] provide an analysis under only a conditional independence assumption — where $X^{(1)}$ and $X^{(2)}$ are conditionally independent of Y (in a multi-class setting, where Y is one of k outcomes). The common criticism of these conditional independence assumption is that it is far too stringent to assume that $X^{(1)}$ and $X^{(2)}$ are independent just conditioned on a the rather low dimensional target variable Y. We relax this assumption by only requiring that $X^{(1)}$, $X^{(2)}$, and Y all be independent conditioned on some hidden state H. Roughly speaking, we think of H as being the augmented information required to make $X^{(1)}$, $X^{(2)}$, and Y conditionally independent.

The other assumption we consider (and we consider it separately from the previous one) is based on redundancy, as in Kakade and Foster [2007]. Here, we assume that the best linear predictor from each view is roughly as good as

the best linear predictor based on both views. This assumption is weak in the sense that it only requires, on average, the optimal linear predictors from each view to agree.

There are many natural applications for which either of these underlying assumptions are applicable. For example, consider a setting where it is easy to obtain pictures of objects from different camera angles and say our supervised task is one of object recognition. Here, the first assumption holds, and, intuitively, we can think of unlabeled data as providing examples of viewpoint invariance. If there is no occlusion, then we expect our second assumption to hold as well. One can even consider multi-modal views, with one view being a video stream and the other an audio stream, and the task might be to identify properties of the speaker (e.g. recognition) — here conditioned on the speaker identity, the views may be uncorrelated. In NLP, an example would be a paired document corpus, consisting of a document and its translation into another language, and the supervised task could be understanding some high level property of the document (here both assumptions may hold). The motivating example in Blum and Mitchell [1998] is a webpage classification task, where one view was the text in the page and the other was the hyperlink structure.

It turns out that under *either* assumption, Canonical Correlation Analysis (CCA) provides a dimensionality reduction method, appropriate for use in a regression algorithm (see Hardoon et al. [2004] for a review of CCA with applications to machine learning). In particular, the semi-supervised algorithm is:

1. Using unlabeled data $\{(X^{(1)}, X^{(2)})\}$, perform a CCA.
2. Construct a projection Π that projects $(X^{(1)}, X^{(2)})$ to the most correlated lower dimensional subspace (as specified in Theorems 3 and 5).
3. With a labeled dataset $\{(X^{(1)}, X^{(2)}, Y)\}$, do a least squares regression (with MLE estimates) in this lower dimensional subspace, i.e. regress Y with $(\Pi X^{(1)}, \Pi X^{(2)})$.

Algorithm 1: Regression in a CCA Subspace

Our main results show that (under either assumption) we lose little predictive information by using this lower dimensional CCA subspace – the gain is that our regression problem has a lower sample complexity due to the lower dimensionality. These results are stated for linear predictors. For notational simplicity, we do not include intercept in linear models considered in this paper. For example, we consider linear predictor of the form $\beta^\top X^{(1)}$ (where β is a vector) instead of the form $\beta^\top X^{(1)} + b$ (with a real valued intercept b). Note that adding intercept is equivalent to appending a constant feature of 1 to $X^{(1)}$, and then ignoring the intercept. This means that our results apply without modifications for linear models with intercept. We will mention explicitly when subtle difference exists in the interpretation of the results.

1.1 Related Work

Both assumptions mentioned in the introduction have been considered together in the co-training framework of Blum and Mitchell [1998] (in a rather strong sense).

In Ando and Zhang [2007], the conditional independence assumption was with respect to a multi-class setting, where Y is discrete, i.e. $Y \in [k]$. In our generalization, we let Y be real valued and we relax the assumption in that we need only independence with respect to some hidden state. Our proof is similar in spirit to that in Ando and Zhang [2007]. However, instead of assuming conditional independence, we make a weaker assumption on conditional uncorrelatedness.

The redundancy assumption we consider here is from Kakade and Foster [2007], where two algorithms were proposed: one based on "shrinkage" (a form of regularization) and one based on dimensionality reduction. The results were stronger for the shrinkage based algorithm. Here, we show that the dimensionality reduction based algorithm works just as well as the proposed "shrinkage" algorithm.

We shall also mention that in the more traditional single-view regression setting, many dimensionality reduction models that are different from this work were proposed. An example is the sliced inverse regression model Li [1992]. Although these models are different from what we study in this paper, some issues concerning the suitability of the underlying assumptions are related. For example, in the context of sliced inverse regression, the suitability of the

linearity assumption has been discussed in Cook and Weisberg [1991]. This means that for real applications, we have to carefully examine the validity of our assumptions. Therefore in the experiment section of this paper, we have made some special efforts to check the suitability of multi-view assumptions introduced in Section 2.

2 Multi-View Assumptions

All vectors in our setting are column vectors. We slightly abuse notation and write $(X^{(1)}, X^{(2)})$, which really denotes the column vector of $X^{(1)}$ concatenated with $X^{(2)}$.

Let $\beta \cdot X$ be the best linear prediction of random variable Y using random vector X, where we use the notation $\beta \cdot X = \beta^\top X$ to denote the dot-product of vectors β and X. Note that we use the notations $\beta \cdot X$ and $\beta^\top X$ interchangeably in this paper. That is, the vector β is the least squares solution

$$\beta = (\mathbb{E}[XX^\top])^{-1}\mathbb{E}[XY]$$

that minimizes the mean squared loss

$$\text{loss}(\tilde{\beta}) = \mathbb{E}(Y - \tilde{\beta} \cdot X)^2 \tag{1}$$

among all vectors $\tilde{\beta}$. We can now introduce the definition of R^2, the coefficient of determination between Y and X, as follows:

$$R^2_{X,Y} := \text{correlation}(\beta \cdot X, Y)^2 := \frac{[\mathbb{E}(\beta \cdot XY)]^2}{\mathbb{E}[(\beta \cdot X)^2]\mathbb{E}[Y^2]}. \tag{2}$$

In other words, $R^2_{Y,X}$ is the proportion of variability in Y that is accounted for by the best linear prediction with X, i.e.

$$R^2_{X,Y} = 1 - \frac{\text{loss}(\beta)}{\mathbb{E}[Y^2]}, \tag{3}$$

and loss is the square loss defined in (1). This equation means that $R^2_{X,Y} > 0$ as long as $\text{loss}(\beta) < \mathbb{E}[Y^2]$. That is, X can *non-trivially predict* Y in the sense that $\text{loss}(\beta) < \mathbb{E}[Y^2] = \text{loss}(0)$.

Note that these definitions are useful when we want to show the effectiveness of dimension reduction without including an intercept into the linear predictor. In order to handle an intercept, we can simply change $\mathbb{E}[Y^2]$ to $\text{var}(Y)$ in (3), and include the intercept in the definition of loss in (1). In this case, the definition of correlation in (2) should also be changed to subtracting the means for each random variable.

2.1 Independence and Predictability of Hidden States

First, let us present the definition of a hidden state H. Intuitively, we think of hidden states as those which imply certain independence properties with respect to our observed random variables.

Definition We say that a random vector H is a *hidden state* for $X^{(1)}$, $X^{(2)}$ and Y if, conditioned on H, we have that $X^{(1)}$, $X^{(2)}$, and Y are all uncorrelated.

Note there always exists an H which satisfies this uncorrelated property. We say H is a *linear hidden state* if we also have that $\mathbb{E}[X^{(1)}|H]$, $\mathbb{E}[X^{(2)}|H]$, and $\mathbb{E}[Y|H]$ are linear in H.

Instead of dealing with independence with respect to Y (which is typically far too stringent), our assumption will be with respect to H, which always exists. Also note that the above definition only requires the weaker concept of uncorrelatedness rather than independence. This means if $X^{(1)}$, $X^{(2)}$, and Y are conditionally independent given H, then they are also uncorrelated given H. Therefore under the conditional independence assumption, H satisfies the definition of hidden state in Definition 2.1.

Assumption 1. (Hidden State Predictability) *Let H be a linear hidden state such that both $X^{(1)}$ and $X^{(2)}$ are non-trivially predictive of H. More precisely, assume that for all directions $w \in \mathbb{R}^{dim(H)}$:*

$$R^2_{X^{(1)},w\cdot H} > 0,\ R^2_{X^{(2)},w\cdot H} > 0$$

Intuitively, this multi-view assumption is that both $X^{(1)}$ and $X^{(2)}$ are informative of the hidden state. However, they need not be good predictors.

2.2 Redundancy

The other assumption we consider is one based on redundancy.

Assumption 2. (ϵ-Redundancy) *Assume that the best linear predictor from each view is roughly as good as the best linear predictor based on both views. More precisely, we have:*

$$R^2_{X^{(1)},Y} \geq R^2_{X,Y} - \epsilon$$
$$R^2_{X^{(2)},Y} \geq R^2_{X,Y} - \epsilon,$$

where $\epsilon > 0$ measures the degree of redundancy.

Note this is equivalent to:

$$\text{loss}(\beta_1) - \text{loss}(\beta) \leq \epsilon\, \mathbb{E}(Y^2)$$
$$\text{loss}(\beta_2) - \text{loss}(\beta) \leq \epsilon\, \mathbb{E}(Y^2)$$

where β_1, β_2 and β are the best linear predictors with $X^{(1)}$, $X^{(2)}$, and X, respectively. This is the form of the assumption stated in Kakade and Foster [2007].

3 CCA and Projections

We say that $\{U_i\}_i$ and $\{V_i\}_i$ are canonical coordinate systems for $X^{(1)}$ and $X^{(2)}$ if they are bases for each view and they satisfy

$$\text{correlation}(U_i \cdot X^{(1)}, V_j \cdot X^{(2)}) = \begin{cases} \lambda_i & \text{if } i = j \\ 0 & \text{else} \end{cases},$$

where similarly to (2), we define correlation as

$$\text{correlation}(U_i \cdot X^{(1)}, V_j \cdot X^{(2)}) := \frac{\mathbb{E}[(U_i \cdot X^{(1)})(V_j \cdot X^{(2)})]}{\sqrt{\mathbb{E}(U_i \cdot X^{(1)})^2}\sqrt{\mathbb{E}(V_j \cdot X^{(2)})^2}},$$

without subtracting the mean of each random variable. CCA finds such a basis (which always exists). Without loss of generality, assume that:

$$1 \geq \lambda_1 \geq \lambda_2 \geq \lambda_3 \ldots . \geq 0$$

We refer to U_i and V_i as the i-th canonical directions and λ_i as the i-th canonical value. Note that U_i (or V_i) are generally not orthogonal vectors with respect to the Euclidean distance. However, if we let

$$\Sigma_{11} = \mathbb{E}[X^{(1)}(X^{(1)})^\top], \quad \Sigma_{22} = \mathbb{E}[X^{(2)}(X^{(2)})^\top],$$

then $\Sigma_{11}^{1/2} U_i$ are orthogonal vectors (with respect to the Euclidean distance), and $\Sigma_{22}^{1/2} V_i$ are orthogonal vectors (with respect to the Euclidean distance).

Let $k = \max\{i : \lambda_i > 0\}$. Then we can define $\Pi_{\text{CCA}} X^{(1)}$ and $\Pi_{\text{CCA}} X^{(2)}$ to be the representations of $X^{(1)}$ and $X^{(2)}$ in $\mathbb{R}^k$, defined respectively as:

$$\Pi_{\text{CCA}} X^{(1)} = [U_i \cdot X^{(1)}]_{i=1,\ldots,k}, \quad \Pi_{\text{CCA}} X^{(2)} = [V_i \cdot X^{(2)}]_{i=1,\ldots,k}.$$

These projections represent each view by a point in the subspace that is positively correlated with the other view. Moreover, these projections are invariant under (full-rank) linear transformation of $X^{(1)}$ and $X^{(2)}$. Therefore we may assume that Σ_{11} and Σ_{22} are identity matrices in our theoretical analysis.

Similarly, we can define $\Pi_\lambda X^{(1)}$ and $\Pi_\lambda X^{(2)}$ as the projection representations of $X^{(1)}$ and $X^{(2)}$ in subspaces with correlation (to the other view) no less than λ. More precisely, we let

$$\Pi_\lambda X^{(1)} = [U_i \cdot X^{(1)}]_{i:\lambda_i \geq \lambda}, \quad \Pi_\lambda X^{(2)} = [V_i \cdot X^{(2)}]_{i:\lambda_i \geq \lambda}.$$

This projection Π_λ is useful since sometimes we deal with subspaces that are sufficiently correlated (to the extent that λ is small).

In our practical implementation of CCA, we employ a regularized formulation (14), where a small regularization term κI is added to Σ_{11} and Σ_{22} to avoid singularity. Note also that any (X independent) linear scaling of a CCA direction U_i (or V_i) is also a CCA direction. Therefore in practice, we need to choose appropriate normalizations for U_i and V_i. Since our theoretical results only concern with linear functions in terms of the CCA projections, the choice of normalization (or more generally when the projections undergo any X independent linear transform) does not affect the analysis. Moreover, in practice, it may be useful (and more standard) to transform the data by subtracting the mean of $X^{(1)}$ and $X^{(2)}$ before applying the CCA formulation considered here. Since this more standard definition (with mean subtracted from each view) is equivalent to the definition here after the above mentioned data transformation, we will not treat it separately in the paper.

4 Dimensionality Reduction Under Conditional Independence

We now present our main theorems, under our Hidden State Predictability Assumption. These theorems show that after dimensionality reduction (via CCA), we have not lost any predictive power of our target variable. The proofs are left to Section 6.

Results in this section also apply to the auxiliary problem-based method in Ando and Zhang [2007] that finds the same subspace as that of CCA (a brief description of the auxiliary problem based approach can be found in Section 7). However, the behavior of these methods may be different when noise is present. Since our theory does not claim which is better when assumptions in this section only hold approximately, for comparison purposes, experiments in Section 7 include both methods.

Theorem 3. *Suppose that Assumption 1 holds and that the dimension of H is k. Then Π_{CCA} is projection into a subspace of dimension precisely k and the following three statements hold:*

1. *The best linear predictor of Y with $X^{(1)}$ is equal to the best linear predictor of Y with $\Pi_{CCA}X^{(1)}$.*
2. *The best linear predictor of Y with $X^{(2)}$ is equal to the best linear predictor of Y with $\Pi_{CCA}X^{(2)}$.*
3. *The best linear predictor of Y with $(X^{(1)}, X^{(2)})$ is equal to the best linear predictor of Y with $\Pi_{CCA}X = (\Pi_{CCA}X^{(1)}, \Pi_{CCA}X^{(2)})$.*

where the best linear predictor is measured with respect to the square loss (1).

This theorem shows that we need only concern ourselves with a k dimensional regression problem, after the CCA projection, and we have not lost any predictive power. Note that the prediction error in each of these cases need *not* be the same. In particular, with both views, one could potentially obtain significantly lower error. Moreover, as explained in Section 4.1, in order to achieve the best linear prediction with the concatenated feature vector $(X^{(1)}, X^{(2)})$, it is necessary to reduce the dimensionality to $2k$ (instead of k, which would appear intuitive based on the more traditional view of CCA Tripathi et al. [2008]). This means that the result of Theorem 3 achieves optimal dimensionality reduction, and cannot be improved without additional assumptions. Moreover, it also means that the theoretical analysis in this paper provides important new insights that are not clear from more traditional view of CCA such as Tripathi et al. [2008].

In addition to direct CCA reduction, one may derive a similar result using bilinear functions of $X^{(1)}$ and $X^{(2)}$. Let d_1 be the dimension of $X^{(1)}$ and d_2 be the dimension of $X^{(2)}$. We define the tensor product of two vectors $X^{(1)} \in \mathbb{R}^{d_1}$ and $X^{(2)} \in \mathbb{R}^{d_2}$ as the $d_1 \times d_2$ dimensional vector:

$$X^{(1)} \circ X^{(2)} := [X^{(1)}_i X^{(2)}_j]_{i,j:(i=1,\ldots,d_1;j=1,\ldots,d_2)} \in \mathbb{R}^{d_1 d_2}.$$

In such case, one needs k^2 instead of $2k$ dimensionality reduction.

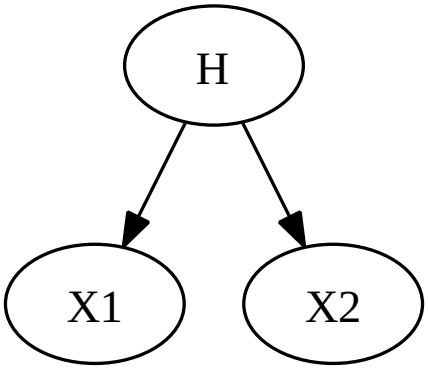


Figure 1: Two View Conditional Independence Model

Theorem 4. *Suppose that Assumption 1 holds and that the dimension of H is k. Let $Z = X^{(1)} \circ X^{(2)}$. Then the best linear predictor of Y with Z is equal to the best linear predictor of Y with the following k^2 projected variables*

$$Z^\top (\mathbb{E}ZZ^\top)^{-1}((\mathbb{E}X^{(1)}{X^{(1)}}^\top U_i) \circ (\mathbb{E}X^{(2)}{X^{(2)}}^\top V_j)),$$

where $U_i \in \mathcal{U}$ and $V_j \in \mathcal{V}$ are CCA basis vectors for the two views respectively, as defined in Section 3 (where $i, j = 1, \ldots, k$).

Note that $\mathbb{E}ZZ^\top$, $\mathbb{E}X^{(1)}{X^{(1)}}^\top$, and $\mathbb{E}X^{(2)}{X^{(2)}}^\top$ can be computed from unlabeled data. Therefore Theorem 4 says that we can compute a k^2 dimensional subspace that contains the best linear predictor using the tensor product of the two views. If the representation for each view contains a complete basis that is dense with respect to measurable functions defined on the corresponding view, then the tensor product gives a complete basis for a function class that is dense with respect to measurable functions that depend on both views. In such a case, Theorem 4 implies consistency; that is, the optimal measurable (linear or nonlinear) predictor using $[X^{(1)}, X^{(2)}]$ is achieved by the best linear predictor with the k^2 projections given by the theorem.

Section 4.1 contains two examples, showing that for some models, Theorem 3 is sufficient, while for other models, it is necessary to apply Theorem 4. Moreover, the discussion also shows that the reduced dimensionality k^2 in Theorem 4 cannot be improved without additional assumptions, which means that CCA achieves optimal dimensionality reduction. In summary, the theoretical justification of CCA in this section, which shows that CCA achieves optimal dimensionality reduction, complements the more traditional view of CCA, and shows explicitly why this method works in multi-view regression.

4.1 Examples: Hidden State Prediction Models

We consider two concrete conditional independence probability models where CCA can be applied. The general graphical model representation is given in Figure 1, which implies that $P(X^{(1)}, X^{(2)}|H) = P(X^{(1)}|H)P(X^{(2)}|H)$. Let us also assume that Y is contained in H (e.g. say Y is the first coordinate H_1). The first model is a two view Gaussian model, similar to that of Bach and Jordan [2005], and the second is a discrete model, similar to Ando and Zhang [2007]. The optimal predictor of Y (using both views) is linear in the first model, and thus the CCA reduction in Theorem 3 is sufficient. In the second model, the optimal predictor of Y is linear in the tensor product of $X^{(1)}$ and $X^{(2)}$, and thus Theorem 4 is needed.

4.1.1 Two view Gaussian model

We consider the following model, similar to Bach and Jordan [2005], but with a more general Gaussian prior on H:

$$\begin{aligned} P(X^{(\ell)}|H) =& N(W_\ell^\top H, \Sigma_\ell) \qquad (\ell \in \{1, 2\}), \\ P(H) =& N(\mu_0, \Sigma_0), \end{aligned}$$

where μ_0, W_ℓ, and Σ_ℓ are unknowns.

Since $\mathbb{E}[X^{(\ell)}|H] = W_\ell^\top H$ ($\ell \in \{1,2\}$), the result of Section 4 can be applied. In this model, we have

$$P(H|X^{(1)}, X^{(2)}) \propto \exp\left[-\frac{1}{2}\sum_{\ell=1}^{2}(X^{(\ell)} - W_\ell^\top H)^\top \Sigma_\ell^{-1}(X^{(\ell)} - W_\ell^\top H) - \frac{1}{2}(H-\mu_0)^\top \Sigma_0^{-1}(H-\mu_0)\right],$$

which is Gaussian in H. Moreover, the optimal prediction of H based on $(X^{(1)}, X^{(2)})$ is the conditional posterior mean $\mathbb{E}[H|X^{(1)}, X^{(2)}]$, which is clearly linear in $(X^{(1)}, X^{(2)})$. Therefore Theorem 3 implies that the Bayes optimal prediction rule is a linear predictor with $2k$ dimensional CCA projections of $X^{(1)}$ and $X^{(2)}$. Importantly, note that we do not have to estimate the model parameters W_ℓ, Σ_ℓ, and μ_0, which could be rather high dimensional quantities.

It is important to mention that the reduction to $2k$ dimension cannot be further improved. In particular, it is not possible to improve Theorem 3 by reducing dimension to k. This is at first counter-intuitive considering that the traditional view on CCA would imply that it might be sufficient to reduce the dimensionality to the k instead. The discrete example in Section 4.1.2 makes this point clear, and it shows why is necessary to develop solid theoretical understanding of CCA through rigorous analysis as we do in this work.

4.1.2 Two view discrete probability model

We consider the following model, which is a simplified case of Ando and Zhang [2007]. Each view $X^{(\ell)}$ represents a discrete observation in a finite set Ω_ℓ, where $|\Omega_\ell| = d_\ell$ (where $\ell \in \{1,2\}$). We may encode each view as a d_ℓ dimensional 0-1 valued indicator vector: $X^{(\ell)} \in \{0,1\}^{d_\ell}$, where only one component has value 1 (which indicates the index of the value in Ω_ℓ being observed), and the others have values 0. Similarly, assume the hidden state variable H is discrete and takes on one of k values, and we represent this by a length k binary vector (with the a-th entry being 1 iff $H = a$). Each hidden state induces a probability distribution over Ω_ℓ for view ℓ. That is, the conditional probability model is given by

$$P([X^{(\ell)}]_i = 1|H) = [W_\ell^\top H]_i \qquad (\ell \in \{1,2\}).$$

i.e. each row a of W_ℓ is the probability vector for $X^{(\ell)}$ conditioned on the underlying discrete hidden state being a. Hence, $\mathbb{E}[X^{(\ell)}|H] = W_\ell^\top H$, so H is a linear hidden state. Moreover, since the two views are discrete, the vector $(X^{(1)}, X^{(2)})$ is uniquely identified with $X^{(1)} \circ X^{(2)} \in \mathbb{R}^{d_1 \times d_2}$ that contains only one nonzero component, and hence an arbitrary function of $(X^{(1)}, X^{(2)})$ is trivially a linear function of $X^{(1)} \circ X^{(2)}$. This means that Theorem 4 can be applied to reduce the overall dimension to k^2 and that the Bayes optimal predictor is linear in this reduced k^2 dimensional space. Moreover, in this case, it can be shown that the reduced dimensions are given by the tensor products of the CCA basis for the two views.

This example also indicates that the number k^2 in Theorem 4 cannot be further improved. To see this, we consider the special case that $d_1 = d_2 = k$ (and hence CCA does not reduce dimensionality further). The hidden state $H = h \in \{1, \ldots, k\}$ induces a distribution on $X^{(\ell)}$ ($\ell = 1, 2$) with probability 0.9 at $X^{(\ell)} = \mathbf{e}_h$, and probability 0.1 as uniform random distribution over $X^{(\ell)} = \mathbf{e}_j$ with $j \neq h$. Here we use $\mathbf{e}_j$ to denote the k-dimensional vector of zeros except for the j-th component being one. In this case, both views are non-trivially predictive of H because both views are highly correlated with H where each $X^{(\ell)}$ has a high probability of being $\mathbf{e}_h$ given the hidden state h. Therefore Assumption 1 holds. We note that a linear function in the tensor product $X^{(1)} \circ X^{(2)}$ can take arbitrarily assigned values $f_{a,b}$ at $X^{(1)} \circ X^{(2)} = \mathbf{e}_a \circ \mathbf{e}_b$ for each element (a,b) in $\{(a,b) : 1 \leq a, b \leq k\}$. Moreover, since there is a non-zero probability to reach any $X^{(1)} \circ X^{(2)} = \mathbf{e}_a \circ \mathbf{e}_b$ based on our construction, it means that in order to match an optimal linear function in the tensor product of $X^{(1)}$ and $X^{(2)}$, it is necessary to match all function values $f_{a,b}$ at every $\mathbf{e}_a \circ \mathbf{e}_b$ in the example we constructed above. Now consider any embedding of the points $\{\mathbf{e}_a \circ \mathbf{e}_b : 1 \leq a, b \leq k\}$ as k^2 vectors in an M-dimensional space $\mathbb{R}^M$ with $M < k^2$, then these embedded k^2 vectors are linearly dependent. Therefore the function values of any linear function in $\mathbb{R}^M$ (evaluated at these embedded k^2 vectors) have to satisfy the same linear relationship, and thus cannot match arbitrarily assigned k^2 values $f_{a,b}$. This means that it is not possible to reduce dimensionality to less than k^2 projected variables, so that the optimal linear function with the projected variables match an arbitrary linear function using $X^{(1)} \circ X^{(2)}$. Therefore the number k^2 in Theorem 4 cannot be further improved without imposing additional assumptions.

The same reasoning also shows that the number $2k$ in Theorem 3 cannot be further improved, because in the above example, the class of linear functions of the form $\beta_1 \cdot X^{(1)} + \beta_2 \cdot X^{(2)}$ has $2k$ degrees of freedom, uniquely determined

by its function values (which may take arbitrary values) at the points $(\mathbf{e}_a, 0)$ and $(0, \mathbf{e}_b)$ ($1 \leq a, b \leq k$). Therefore any embedding of $(\mathbf{e}_a, \mathbf{e}_b)$ into an M-dimensional space $\mathbb{R}^M$ with $M < 2k$ will introduce a linear dependency among the embedded points $(\mathbf{e}_a, 0)$ and $(0, \mathbf{e}_b)$. It means that the class of linear functions in $\mathbb{R}^M$ cannot match arbitrary function values at the $2k$ points $(\mathbf{e}_a, 0)$ and $(0, \mathbf{e}_b)$ ($a, b, = 1, \ldots, k$), and thus cannot reproduce all linear functions of the form $\beta_1 \cdot X^{(1)} + \beta_2 \cdot X^{(2)}$. Therefore the number $2k$ in Theorem 3 cannot be further improved without additional assumptions.

We shall mention that the optimality of $2k$ in Theorem 3 and k^2 in Theorem 4 applies for linear predictors without intercept as considered throughout this paper. If intercept is allowed, then we only need $2k - 1$ dimensions in Theorem 3 and $k^2 - 1$ dimensions in Theorem 4. This is due to the extra degree of freedom coming from the intercept. That is, if we count the intercept as the corresponding coefficient of an extra feature (or dimension) of constant 1, then the needed dimensions are still $2k$ and k^2. Under this view (where we treat this constant 1 as one reduced dimension), our discussions apply without modification.

5 Dimensionality Reduction Under Redundancy

In this Section, we assume that Y is a scalar. We also use the projection Π_λ, which projects to the subspace which has correlation at least λ (recall the definition of Π_λ from Section 3). Results in this section apply to CCA but do not apply to the auxiliary problem-based method in Ando and Zhang [2007]. The following theorem shows that using $\Pi_\lambda X^{(1)}$ instead of $X^{(1)}$ for linear prediction does *not* significantly degrade performance. The proof is left to Section 6.

Theorem 5. *Suppose that Assumption 2 holds (recall, ϵ is defined in this assumption) and that $Y \in \mathbb{R}$. For all $0 \leq \lambda \leq 1$, we have that:*

$$R^2_{\Pi_\lambda X^{(1)}, Y} \geq R^2_{X^{(1)}, Y} - \frac{4\epsilon}{1-\lambda}$$
$$R^2_{\Pi_\lambda X^{(2)}, Y} \geq R^2_{X^{(2)}, Y} - \frac{4\epsilon}{1-\lambda}$$

Note that this implies that these R^2's are also close to $R^2_{X,Y}$.

Clearly, if we chose $\lambda = \frac{1}{2}$, then our loss in error (compared to the best linear prediction) is at most 8ϵ. However, we now only have to deal with estimation in the subspace spanned by $\{U_i : \lambda_i \geq \frac{1}{2}\}_i$, a potentially much lower dimensional space. In fact, the following corollary bounds the dimension of this space in terms of the spectrum.

Corollary 6. *Assume that we choose $\lambda = \frac{1}{2}$. Let d be the dimension of the space that Π_λ projects to, i.e. d is the number of i such that $\lambda_i \geq \frac{1}{2}$. For all $\alpha > 0$, we have:*

$$d \leq 2^\alpha \sum_i \lambda_i^\alpha$$

In particular, this implies that:

$$d \leq 2 \sum_i \lambda_i \text{ and } d \leq 4 \sum_i \lambda_i^2$$

Note that unlike the previous setting, this spectrum need not ever have a finite number of nonzero entries, so we may require a larger power of α to make the sum finite.

Proof. Using that $\lambda_i \geq \frac{1}{2}$, we have:

$$d = \sum_{i=1}^{d} 1 = \sum_{i=1}^{d} \frac{\lambda_i^\alpha}{\lambda_i^\alpha} \leq 2^\alpha \sum_{i=1}^{d} \lambda_i^\alpha \leq 2^\alpha \sum_{i=1}^{\infty} \lambda_i^\alpha$$

where the second to last step follows from the fact that $\lambda_i \leq \lambda$ by definition of d. □

6 Proofs

Let us denote:

$$\begin{aligned}
\Sigma_{11} &= \mathbb{E}[X^{(1)}(X^{(1)})^\top] \\
\Sigma_{22} &= \mathbb{E}[X^{(2)}(X^{(2)})^\top] \\
\Sigma_{12} &= \mathbb{E}[X^{(1)}(X^{(2)})^\top] \\
\Sigma_{HH} &= \mathbb{E}[HH^\top] \\
\Sigma_{1H} &= \mathbb{E}[X^{(1)}H^\top] \\
\Sigma_{2H} &= \mathbb{E}[X^{(2)}H^\top] \\
\Sigma_{1Y} &= \mathbb{E}[X^{(1)}Y^\top] \\
\Sigma_{2Y} &= \mathbb{E}[X^{(2)}Y^\top]
\end{aligned}$$

and $\Sigma_{12}^\top = \Sigma_{21}$, $\Sigma_{H1} = \Sigma_{1H}^\top$, etc.

Without loss of generality, we assume that we have the following isotropic conditions:

$$\Sigma_{11} = \text{Identity}, \Sigma_{22} = \text{Identity}, \Sigma_{HH} = \text{Identity}, \mathbb{E}[Y^2] = 1.$$

This is without loss of generality as our algorithm does not make use of any particular coordinate system (the algorithm is only concerned with the subspaces themselves). This choice of coordinate system eases the notational burden in our proofs. Furthermore, note that we can still have $H_1 = Y$ in this coordinate system.

Under these conditions, CCA corresponds to an SVD of Σ_{12}. Let the SVD decomposition of Σ_{12} be:

$$\Sigma_{12} = UDV^\top$$

where U and V are orthogonal and D is diagonal. Let

$$D = \text{diag}(\lambda_1, \lambda_2, \ldots)$$

Without loss of generality, assume that the SVD is ordered such that:

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \ldots .$$

Here, the column vectors of U and V form the CCA basis, and note that for a column U_i of U and V_j of V

$$\mathbb{E}[(U_i \cdot X^{(1)})(V_j \cdot X^{(2)})] = \begin{cases} \lambda_i & \text{if } i = j \\ 0 & \text{else} \end{cases} \tag{4}$$

which implies that

$$0 \leq \lambda_i \leq 1$$

since we are working in the coordinate system where $X^{(1)}$ and $X^{(2)}$ are isotropic.

Moreover, we let $U_{1:k}$ be the matrix with k columns, where the i-th column is U_i. Similarly, let $V_{1:k}$ be the matrix with k columns, where the i-th column is V_i. Define $D_{1:k} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_k)$. In this representation, assuming that the rank of Σ_{12} is k (which is a consequence of Lemma 7), then we can use the following more convenient notation in later part of the proofs:

$$\Pi_{\text{CCA}} X^{(1)} = U_{1:k}^\top X^{(1)}, \quad \Pi_{\text{CCA}} X^{(2)} = V_{1:k}^\top X^{(2)}, \quad \Sigma_{12} = U_{1:k} D_{1:k} V_{1:k}^\top.$$

6.1 Proof of Theorem 3

Throughout this subsection, we let $\beta_1 \cdot X^{(1)}$ be the best linear prediction of H with $X^{(1)}$ (which minimizes the square loss), let $\beta_2 \cdot X^{(2)}$ be the best linear prediction of H with $X^{(2)}$, and let $\beta \cdot (X^{(1)}, X^{(2)})$ be the best linear prediction of H with both $(X^{(1)}, X^{(2)})$.

With the aforementioned isotropic conditions:

$$\beta_1 = \Sigma_{1H},\ \beta_2 = \Sigma_{2H},\ \beta = (\mathbb{E}[XX^\top])^{-1}\mathbb{E}[XH^\top] \tag{5}$$

which follows directly from the least squares solution. Note that $\mathbb{E}[XX^\top]$ is not diagonal.

Our proof consists of showing that the best linear prediction of H with X is equal to the best linear prediction of H with $\Pi_{\text{CCA}}X$. This implies that the best linear prediction of Y with X is equal to the best linear prediction of Y with $\Pi_{\text{CCA}}X$, by the following argument. Since $\mathbb{E}[Y|H]$ is linear in H (by assumption), we can do a linear transformation of H such that $\mathbb{E}[Y|H] = H_1$ (where H_1 is the first coordinate of H). By Assumption 1, it follows that for all $\beta \in \mathbb{R}^{\dim(X)}$,

$$\mathbb{E}(Y - \beta \cdot X)^2 = \mathbb{E}(Y - H_1)^2 + \mathbb{E}(H_1 - \beta \cdot X)^2 \,.$$

Hence, our proof only needs to consider the linear prediction of H.

The following lemma shows the imposed structure on the covariance matrix Σ_{12} for any linear hidden state.

Lemma 7. *If H is a linear hidden state, then we have that:*

$$\Sigma_{12} = \Sigma_{1H}\Sigma_{H2}$$

which implies that the rank of Σ_{12} is at most k.

Proof. By the linear mean assumption we have that:

$$\mathbb{E}[X^{(1)}|H] = \Sigma_{1H}H$$
$$\mathbb{E}[X^{(2)}|H] = \Sigma_{2H}H$$

which follows from the fact that $\Sigma_{1H}H$ is the least squares prediction of $X^{(1)}$ with H (and this least squares prediction is the expectation, by assumption).

Recall, we are working with H in an isotropic coordinate system. Hence,

$$\begin{aligned}
\Sigma_{12} &= \mathbb{E}[X^{(1)}(X^{(2)})^\top] \\
&= \mathbb{E}_H[\mathbb{E}[X^{(1)}(X^{(2)})^\top|H]] \\
&= \mathbb{E}_H[\mathbb{E}[X^{(1)}|H]\mathbb{E}[(X^{(2)})^\top|H]] \\
&= \mathbb{E}_H[\Sigma_{1H}HH^\top\Sigma_{H2}] \\
&= \Sigma_{1H}\Sigma_{H2}
\end{aligned}$$

which completes the proof. □

Now we are ready to complete the proof of Theorem 3.

Proof. Now Assumption 1 implies that both Σ_{1H} and Σ_{2H} are rank k. This implies that Σ_{12} is also rank k by the previous lemma. Hence, we have the equality

$$\Sigma_{1H} = \Sigma_{12}(\Sigma_{H2})^{-1}$$

(where the inverse is the pseudo-inverse). Now, the optimal linear predictor $\beta_1 = \Sigma_{1H}$, so we have

$$\beta_1 = \Sigma_{12}(\Sigma_{H2})^{-1} \,.$$

Hence,

$$\begin{aligned}
\beta_1 \cdot X^{(1)} &= (\Sigma_{2H})^{-1}\Sigma_{21}X^{(1)} \\
&= (\Sigma_{2H})^{-1}V_{1:k}D_{1:k}U_{1:k}^\top X^{(1)} \\
&= (\Sigma_{2H})^{-1}V_{1:k}D_{1:k}\Pi_{\text{CCA}}X^{(1)} \\
&= \beta_1' \cdot \Pi_{\text{CCA}}X^{(1)} \,,
\end{aligned}$$

where $\beta_1' = [(\Sigma_{2H})^{-1} V_{1:k} D_{1:k}]^\top$. This completes the proof of the first claim, and the proof of the second claim is analogous.

Now we prove the third claim. Let $\tilde{\beta}$ be the weights for the best linear prediction of H with $\Pi_{\text{CCA}} X := (\Pi_{\text{CCA}} X^{(1)}, \Pi_{\text{CCA}} X^{(2)})$. The optimality (derivative) conditions on $\tilde{\beta}$ imply that:

$$\mathbb{E}[(H - \tilde{\beta}^\top \Pi_{\text{CCA}} \cdot X)(\Pi_{\text{CCA}} X)^\top] = 0 \tag{6}$$

In the following we can regard Π_{CCA} as a matrix with $2k$ rows, where its first k rows are $(U_i, 0)$ ($i = 1, \ldots, k$), and its second k rows are $(0, V_i)$ ($i = 1, \ldots, k$). If we show that:

$$\mathbb{E}[(H - \tilde{\beta}^\top \Pi_{\text{CCA}} \cdot X) X^\top] = 0$$

then this proves the result (as the derivative conditions for $\tilde{\beta}^\top \Pi_{\text{CCA}}$ being optimal for minimizing $\mathbb{E}(H - \tilde{\beta}^\top \Pi_{\text{CCA}} \cdot X)^2$ are satisfied). To prove the above, it is sufficient to show that, for all vectors α

$$\mathbb{E}[(H - \tilde{\beta}^\top \Pi_{\text{CCA}} \cdot X)(\alpha \cdot X)] = 0 \tag{7}$$

Let us decompose α as $\alpha = (u + u_\perp, v + v_\perp)$, where u can be expressed as $u = U_{1:k} u'$, and $u_\perp$ satisfies $U_{1:k}^\top u_\perp = 0$. Clearly u and $u_\perp$ are orthogonal. Similarly, define v and $v_\perp$. Since

$$\alpha \cdot X = u \cdot X^{(1)} + u_\perp \cdot X^{(1)} + v \cdot X^{(2)} + v_\perp \cdot X^{(2)} \,,$$

to prove Equation 7, it is sufficient to show that:

$$\mathbb{E}[(H - \tilde{\beta}^\top \Pi_{\text{CCA}} \cdot X)(u \cdot X^{(1)})] = 0 \tag{8}$$
$$\mathbb{E}[(H - \tilde{\beta}^\top \Pi_{\text{CCA}} \cdot X)(v \cdot X^{(2)})] = 0 \tag{9}$$

and

$$\mathbb{E}[(H - \tilde{\beta}^\top \Pi_{\text{CCA}} \cdot X)(u_\perp \cdot X^{(1)})] = 0 \tag{10}$$
$$\mathbb{E}[(H - \tilde{\beta}^\top \Pi_{\text{CCA}} \cdot X)(v_\perp \cdot X^{(2)})] = 0 \tag{11}$$

To prove Equation 8, simply note that $u = U_{1:k} u'$ for some u'. Hence $u \cdot X^{(1)} = u' \cdot U_{1:k}^\top X^{(1)} = u' \cdot \Pi_{\text{CCA}} X^{(1)}$. Therefore the result follows from Equation 6. Equation 9 is proven identically.

Now we prove Equation 10. First note that:

$$\mathbb{E}[\Pi_{\text{CCA}} X^{(1)} (u_\perp \cdot X^{(1)})] = \mathbb{E}[U_{1:k}^\top X^{(1)} (u_\perp \cdot X^{(1)})] = U_{1:k}^\top \Sigma_{11} u_\perp = 0$$

by our isotropic assumption and since $u_\perp$ is orthogonal to $U_{1:k}$. Also,

$$\mathbb{E}[\Pi_{\text{CCA}} X^{(2)} (u_\perp \cdot X^{(1)})] \mathbb{E}[V_{1:k}^\top X^{(2)} (u_\perp \cdot X^{(1)})] = V_{1:k}^\top \Sigma_{21} u_\perp = 0$$

from Equation 4 and by construction of $u_\perp$. These two imply that:

$$\mathbb{E}[\Pi_{\text{CCA}} X (u_\perp \cdot X^{(1)})] = 0$$

We also have that:

$$\begin{aligned}
\mathbb{E}[H(u_\perp \cdot X^{(1)})] &= \mathbb{E}[H (X^{(1)})^\top] u_\perp \\
&= \Sigma_{H1} u_\perp \\
&= (\Sigma_{2H})^{-1} \Sigma_{21} u_\perp \\
&= (\Sigma_{2H})^{-1} V_{1:k} D_{1:k} U_{1:k}^\top u_\perp \\
&= 0
\end{aligned}$$

where we have used the full rank condition on Σ_{12} in the third to last step. An identical argument proves Equation 11. This completes the proof. □

6.2 Proof of Theorem 4

The best linear predictor of Y with $Z = X^{(1)} \circ X^{(2)}$ is given by $\beta_*^\top Z$, where

$$\beta_* = \arg\min_\beta \mathbb{E}_{Z,Y}(\beta^\top Z - Y)^2.$$

That is,

$$\beta_*^\top Z = Z^\top (\mathbb{E}_Z Z Z^\top)^{-1} \mathbb{E}_{Z,Y}(ZY). \tag{12}$$

Now, for each index i, j and $Z_{i,j} = X_i^{(1)} X_j^{(2)}$, there exist $\alpha_i^{(1)} = [\alpha_{i,1}^{(1)}, \ldots, \alpha_{i,k}^{(1)}]$ and $\alpha_j^{(2)} = [\alpha_{j,1}^{(2)}, \ldots, \alpha_{j,k}^{(2)}]$ such that

$$\mathbb{E}[X_i^{(1)}|H] = H^\top \alpha_i^{(1)}, \quad \mathbb{E}[X_j^{(2)}|H] = H^\top \alpha_j^{(2)}$$

by assumption. Therefore, taking expectations over Z and Y,

$$\begin{aligned}
\mathbb{E}[Y Z_{i,j}] =& \mathbb{E}_H\, \mathbb{E}[Y X_i^{(1)} X_j^{(2)}|H] \\
=& \mathbb{E}_H\, [\mathbb{E}[Y|H]\, \mathbb{E}[X_i^{(1)}|H]\, \mathbb{E}[X_j^{(2)}|H]] \\
=& \mathbb{E}_H\, [\mathbb{E}[Y|H]\, (H^\top \alpha_i^{(1)})(H^\top \alpha_j^{(2)})] \\
=& {\alpha_i^{(1)}}^\top Q \alpha_j^{(2)} = \sum_{a=1}^k \sum_{b=1}^k Q_{a,b} \alpha_{i,a}^{(1)} \alpha_{j,b}^{(2)},
\end{aligned}$$

where $Q = \mathbb{E}_H[Y H H^\top]$. Let $\alpha_{\cdot,a}^{(1)} = [\alpha_{i,a}^{(1)}]_i$ and $\alpha_{\cdot,b}^{(2)} = [\alpha_{j,b}^{(2)}]_j$, then

$$\mathbb{E}_{Z,Y}[ZY] = \sum_{a=1}^k \sum_{b=1}^k Q_{a,b} \alpha_{\cdot,a}^{(1)} \circ \alpha_{\cdot,b}^{(2)}.$$

Since we are working with certain isotropic coordinates (see the beginning of Section 6), each $\alpha_{\cdot,a}^{(1)}$ is a linear combination of the CCA basis U_i ($i = 1, \ldots, k$) and each $\alpha_{\cdot,b}^{(2)}$ is a linear combination of the CCA basis V_j ($j = 1, \ldots, k$). Therefore we can find $Q'_{i,j}$ such that

$$\mathbb{E}_{Z,Y}[ZY] = \sum_{i=1}^k \sum_{j=1}^k Q'_{i,j} U_i \circ V_j.$$

From (12), we obtain that

$$\beta_*^\top Z = \sum_{i=1}^k \sum_{j=1}^k Q'_{i,j} Z^\top (\mathbb{E}_Z Z Z^\top)^{-1} U_i \circ V_j.$$

By changing to an arbitrary basis in $X^{(1)}$ and in $X^{(2)}$, we obtain the desired formula.

6.3 Proof of Theorem 5

Here, β_1, β_2 and β are the best linear predictors of Y with $X^{(1)}$, $X^{(2)}$, and X, respectively. Also, in our isotropic coordinates, $\mathrm{var}(Y) = 1$, so we will prove the claim in terms of the loss, which implies that statements about R^2.

The following lemma is useful to prove Theorem 5.

Lemma 8. *Assumption 2 implies that*

$$\sum_i (1 - \lambda_i)(\beta_\nu \cdot U_i)^2 \leq 4\epsilon$$

for $\nu \in \{1, 2\}$.

With this lemma, the proof of our Theorem follows.

Proof of Theorem 5. Let $\beta^{\text{CCA}} = [\beta_1^{\text{CCA}}, \ldots, \beta_d^{\text{CCA}}]$ be the weights of the best linear predictor using only $\Pi_\lambda X^{(1)}$, where d is the dimension of $\Pi_\lambda X^{(1)}$ (i.e. the number of $\lambda_i \geq \lambda$). Since $X^{(1)}$ is isotropic, it follows that $\beta_i^{\text{CCA}} = \beta_1 \cdot U_i$ for $1 \leq i \leq d$. First, note that since the norm of a vector is unaltered by a rotation, we have:

$$\text{loss}(\beta^{\text{CCA}}) - \text{loss}(\beta_1) = \sum_i (\beta_i^{\text{CCA}} - \beta_1 \cdot U_i)^2$$

because U is a rotation matrix. Note that with a slight abuse of notation, we assume that $\beta_i^{\text{CCA}} = 0$ when $i > d$ (that is $\lambda_i < \lambda$). Hence, we have that

$$\begin{aligned}
\text{loss}(\beta^{\text{CCA}}) - \text{loss}(\beta_1) &= \sum_i (\beta_i^{\text{CCA}} - \beta_1 \cdot U_i)^2 \\
&= \sum_{i:\lambda_i < \lambda} (\beta_i^{\text{CCA}} - \beta_1 \cdot U_i)^2 \\
&= \sum_{i:\lambda_i < \lambda} \frac{1-\lambda_i}{1-\lambda_i} (\beta_1 \cdot U_i)^2 \\
&\leq \frac{1}{1-\lambda} \sum_{i:\lambda_i < \lambda} (1-\lambda_i)(\beta_1 \cdot U_i)^2 \\
&\leq \frac{4\epsilon}{1-\lambda}
\end{aligned}$$

where the first line follows from algebraic manipulations for the square loss. □

The following lemma is useful for proving Lemma 8:

Lemma 9. *Assumption 2 implies that*

$$\mathbb{E}[(\beta_1 \cdot X^{(1)} - \beta_2 \cdot X^{(2)})^2] \leq 4\epsilon\,.$$

Proof. Let β be the best linear weights using $X = (X^{(1)}, X^{(2)})$. By Assumption 2

$$\begin{aligned}
\epsilon &\geq \mathbb{E}(\beta_1 \cdot X^{(1)} - Y)^2 - \mathbb{E}(\beta \cdot X - Y)^2 \\
&= \mathbb{E}(\beta_1 \cdot X^{(1)} - \beta \cdot X + \beta \cdot X - Y)^2 - \mathbb{E}(\beta \cdot X - Y)^2 \\
&= \mathbb{E}(\beta_1 \cdot X^{(1)} - \beta \cdot X)^2 - 2\mathbb{E}[(\beta_1 \cdot X^{(1)} - \beta \cdot X)(\beta \cdot X - Y)]
\end{aligned}$$

Now the first derivative conditions for the optimal linear predictor β imply that:

$$\mathbb{E}[X(\beta \cdot X - Y)] = 0$$

which implies that:

$$\begin{aligned}
\mathbb{E}[\beta \cdot X(\beta \cdot X - Y)] = 0 \\
\mathbb{E}[\beta_1 \cdot X^{(1)}(\beta \cdot X - Y)] = 0
\end{aligned}$$

Hence,

$$\mathbb{E}[(\beta_1 \cdot X^{(1)} - \beta \cdot X)(\beta \cdot X - Y)] = 0$$

A similar argument proves the identical statement for β_2.

We have shown that:

$$\begin{aligned}
\mathbb{E}(\beta_1 \cdot X^{(1)} - \beta \cdot X)^2 \leq \epsilon \\
\mathbb{E}(\beta_2 \cdot X^{(2)} - \beta \cdot X)^2 \leq \epsilon
\end{aligned}$$

The triangle inequality states that:

$$\begin{aligned} & \mathbb{E}(\beta_1 \cdot X^{(1)} - \beta_2 \cdot X^{(2)})^2 \\ \leq & \left(\sqrt{\mathbb{E}(\beta_1 \cdot X^{(1)} - \beta \cdot X)^2} + \sqrt{\mathbb{E}(\beta_2 \cdot X^{(2)} - \beta \cdot X)^2}\right)^2 \\ \leq & (2\sqrt{\epsilon})^2 \end{aligned}$$

which completes the proof. □

Now we prove Lemma 8.

Proof of Lemma 8. Let us write $[\beta_1]_i = \beta_1 \cdot U_i$ and $[\beta_2]_i = \beta_2 \cdot V_i$. From Lemma 9, we have:

$$\begin{aligned} 4\epsilon \geq & \mathbb{E}\left[(\beta_1 \cdot X^{(1)} - \beta_2 \cdot X^{(2)})^2\right] \\ = & \sum_i \left(([\beta_1]_i)^2 + ([\beta_2]_i)^2 - 2\lambda_i [\beta_1]_i [\beta_2]_i\right) \\ = & \sum_i \left((1-\lambda_i)([\beta_1]_i)^2 + (1-\lambda_i)([\beta_2]_i)^2 + \lambda_i(([\beta_1]_i)^2 + ([\beta_2]_i)^2 - 2[\beta_1]_i[\beta_2]_i)\right) \\ = & \sum_i \left((1-\lambda_i)([\beta_1]_i)^2 + (1-\lambda_i)([\beta_2]_i)^2 + \lambda_i([\beta_1]_i - [\beta_2]_i)^2\right) \\ \geq & \sum_i \left((1-\lambda_i)([\beta_1]_i)^2 + (1-\lambda_i)([\beta_2]_i)^2\right) \\ \geq & \sum_i (1-\lambda_i)([\beta_\nu]_i)^2 \end{aligned}$$

where the last step holds for either $\nu = 1$ or $\nu = 2$.

□

7 Experiments

In this section we report experiments to show how the methods suggested by our theoretical findings behave under various conditions in comparison with alternatives. As pointed out earlier, results in Section 4 apply both to CCA and to the auxiliary problem-based method (abbreviated as AUX). Therefore our experiments include both methods to examine their behavior on real data, when the idealized assumptions in Section 4 can only be satisfied approximately.

7.1 Tested methods

This section describes implementation of AUX, CCA, and the baseline methods tested for comparison.

7.1.1 Auxiliary problem-based method (AUX)

The auxiliary-problem based method we experiment with is closely related to the methods in Ando and Zhang [2005] (SVD-based ASO) and in Ando and Zhang [2007]. These methods generate auxiliary classification problems that predict some functions of one view (auxiliary labels) based on another view, which would reveal predictive structures when learned on unlabeled data. For example, auxiliary problems on the part-of-speech tagging task could be to predict whether the left context contains certain words, and this prediction could be done based on the right context. In the subspace-based configuration, additional features are generated by projecting the original features onto a subspace, which is spanned by the most significant left singular vectors of a matrix of auxiliary weight vectors obtained by training for auxiliary problems on unlabeled data.

In this work, inspired by CCA, we define auxiliary problems to be regression problems that predict the feature components of one view directly (instead of setting up classification problems) based on another view. The rest is similar to Ando and Zhang [2007]. On the unlabeled data, we obtain the regularized least squares solutions to the auxiliary problems and apply SVD to the matrix of the solutions (i.e., weight vectors). More precisely, let $\mathbf{X}^{(i)}$ ($i \in \{1,2\}$) be a matrix whose columns are the view-i portions of n unlabeled data points, and let d_i be the dimensionality of view-i. (That is, $\mathbf{X}^{(i)} \in R^{d_i \times n}$.) We have d_2 auxiliary problems to predict each of the d_2 feature components of view-2 based on view-1 features. Regularized least squares 'training' of the auxiliary predictors on the unlabeled data leads to the following optimization problems for $i_2 = 1, \ldots, d_2$:

$$\operatorname*{argmin}_{\mathbf{w}} \left(\frac{1}{n} \sum_{i_1=1}^{n} \left(\mathbf{w}^\top \mathbf{X}^{(1)}_{i_1} - \mathbf{X}^{(2)}[i_2, i_1] \right)^2 + \kappa \|\mathbf{w}\|_2^2 \right)$$

where κ is a regularization parameter and $\mathbf{X}^{(1)}_{i_1}$ denotes the i_1-th column of $\mathbf{X}^{(1)}$. The analytical solutions are given by the columns of $\mathbf{W}^{(1)}$ in the following:

$$\mathbf{W}^{(1)} = \left(\mathbf{X}^{(1)} {\mathbf{X}^{(1)}}^\top + n\kappa \mathbf{I} \right)^{-1} \mathbf{X}^{(1)} {\mathbf{X}^{(2)}}^\top . \tag{13}$$

AUX computes several left singular vectors (corresponding to the largest singular values) of this weight matrix $\mathbf{W}^{(i)}$. Our new feature vectors associated with view-1 are projections of the original view-1 feature vector onto the subspace spanned by these left singular vectors. The new feature vectors associated with view-2 are obtained by exchanging the roles of the two views.

To make a contrast with CCA, we note that the left singular vectors computed in this way are the solutions to the following eigenproblems:

$$\begin{aligned} \mathbf{W}^{(1)} {\mathbf{W}^{(1)}}^\top \mathbf{u} &= \alpha\, \mathbf{u}\,, \\ \mathbf{W}^{(2)} {\mathbf{W}^{(2)}}^\top \mathbf{v} &= \beta\, \mathbf{v}\,. \end{aligned}$$

It is worth noting that for the fixed dimensionality the obtained subspace produces the best approximation of $\mathbf{W}^{(i)}$ (whose columns are the weight vectors for the auxiliary problems), in the sense that their projections onto the subspace give the least square errors. As a slight extension of AUX, one can length-normalize the columns of $\mathbf{W}^{(i)}$ into unit vectors before computing SVD. The intuition behind is that all the auxiliary problems should be regarded as equally influential in the process of finding the subspace that gives the best approximation. Though this length-normalization step is optional, it often improves performance, and all the experiments reported in this paper are done with it.

The number of the singular vectors to retain (or the dimensionality of the subspace) is a parameter. The theoretical analysis suggests that the optimum dimensionality is the dimensionality of the hidden state, but in practice, the dimensionality of the hidden state is unknown. One way to choose dimensionality is by cross validation on the labeled training data as in our real-world data experiments discussed later.

Computation time for going through n unlabeled data points is in $O(n \cdot d_i^2)$, and computing $\mathbf{W}^{(i)}$ and solving SVD are both in $O(d_i^3)$. Throughout the experiments, LAPACK++[1] was used for linear algebra computation.

7.1.2 CCA

Using the notation above, CCA finds, on the unlabeled data, vectors $\mathbf{u}$'s and $\mathbf{v}$'s that maximize:

$$\frac{\mathbf{u}^\top \mathbf{X}^{(1)} {\mathbf{X}^{(2)}}^\top \mathbf{v}}{\sqrt{\mathbf{u}^\top \left(\mathbf{X}^{(1)} {\mathbf{X}^{(1)}}^\top / n + \kappa \mathbf{I} \right) \mathbf{u}} \sqrt{\mathbf{v}^\top \left(\mathbf{X}^{(2)} {\mathbf{X}^{(2)}}^\top / n + \kappa \mathbf{I} \right) \mathbf{v}}}\,, \tag{14}$$

[1] Available at http://sourceforge.net/projects/lapackpp

where κ is a regularization parameter. The solution to the CCA optimization problem leads to the following eigenproblems:

$$\begin{aligned} \mathbf{W}^{(1)}\mathbf{W}^{(2)}\mathbf{u} &= \alpha\,\mathbf{u}\,, \\ \mathbf{W}^{(2)}\mathbf{W}^{(1)}\mathbf{v} &= \beta\,\mathbf{v}\,, \end{aligned}$$

which show an interesting similarity and contrast to the eigenproblems for AUX above. We compute several eigenvectors (corresponding to canonical directions) associated with the largest eigenvalues, and the rest follows Section 3. As in AUX, the number of eigenvectors retained (or dimensionality of the subspace) is a parameter and needs to be determined by cross validation.

Computation time for solving the eigenproblems is in $O(\max_i d_i^3)$.

7.1.3 Baseline methods

For comparison, we also test co-training, naive Bayes EM, and dimension reduction by PCA.

Co-training Our implementation of co-training follows the original work in Balcan and Blum [2005]. The two classifiers (employing distinct views) are trained with labeled data. We maintain a pool of q unlabeled data points by random selection. The classifier proposes labels for the data points in this pool. We choose s data points for each classifier with high confidence while preserving the class distribution observed in the initial labeled data, and add them to the labeled data. The process is then repeated, and the final classifier is trained using both views. It is generally known that for co-training to perform well, two assumptions need to be satisfied: class-conditional independence of views and view redundancy (i.e., each view alone is almost sufficient for classification). The pool size q and increment size s were fixed to 10000 and 500 unless otherwise specified.

Naive Bayes EM Naive Bayes EM (NB-EM) is a traditional generative framework for unsupervised and semi-supervised learning. Its effectiveness on the tasks such as text categorization is well known. Our implementation follows Nigam et al. [2000]. A high-level summary is as follows. The model parameters (estimates of $P(\mathrm{class})$ and $P(f|\mathrm{class})$ where f is a feature component) are initialized by the labeled data, and probabilistic labels (soft labels) are assigned to the unlabeled data points according to the model parameters. Using the assigned soft labels, the model parameters are updated based on the naive Bayes assumption and the process repeats. Note that the naive Bayes assumption here is feature component-wise (independence of all the feature components given classes), which is more restrictive than the view independence assumption of co-training.

PCA PCA is a traditional dimension reduction technique. We apply SVD to the unlabeled data after subtracting the mean and retain several left singular vectors corresponding to the largest singular values. The new feature vectors are obtained by projection onto the subspace after subtracting the mean.

7.1.4 Base learner

The experiments use regularized least squares as the training objective whenever training is done with labeled data – which includes the supervised baseline, co-training, and the final training after deriving new features from unlabeled data by AUX, CCA, or PCA. More specifically, given n labeled data points $\{(\mathbf{x}_i, y_i)\}$ $(i = 1, ..., n)$, we find the weight vector $\hat{\mathbf{w}}$ that satisfies: $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}^\top \mathbf{x}_i - y_i)^2 + \gamma\|\mathbf{w}\|_2^2$ where γ is a regularization parameter. In all the methods but NB-EM, we use the 'one vs. all' training scheme when there are more than two classes.

7.2 Synthesized data experiments

The first suite of experiments use synthesized data sets in order to investigate the empirical behavior of the methods under various conditions in relation to our theoretical findings.

7.2.1 Data generation

The data sets we synthesize are all labeled for 10-class classification and consist of two views, each of which is 2000-dimensional. The data sets we generate are *indicator data sets* in which each view has exactly one feature component set to one while the other components are set to zero. The hidden state variable takes one of k values in $\{1, \cdots, k\}$ and is represented by a k-dimensional vector in which only one entry corresponding to its value has one and the other entries have 0. For simplicity, we slightly abuse terminology and sometimes use "state" for the value of the state variable in the following.

The data sets are generated by the following procedure based on pre-defined probability distributions:

- Draw a state h according to $P(\mathrm{state})$.
- Draw a class according to $P(\mathrm{class}|h)$.
- For each view, draw an index of feature component that should be set to one according to $P(\mathrm{index}|h)$.

On the data sets generated in this way, class labels and features are conditionally uncorrelated (and also independent) given states, and Assumption 1 (linear hidden state predictability) is satisfied.

The probabilities $P(\mathrm{index}|\mathrm{state})$ are defined as follows. For each state h, we define a set of feature component indices S_h (which is analogous to a vocabulary set associated with each state). Given h, the feature component (of view-1 or view-2) that is to be set to one is chosen from S_h with probability uniformly distributed over the members of S_h (i.e., $P(\mathrm{feature}_i|h) = 1/|S_h|$ if $i \in S_h$; 0 otherwise). In this simple model, one can adjust predictability (how easy it is to predict the hidden state from the features) by changing how much the index sets S_h's overlap with one another. Prediction will be the easiest if the index sets are disjoint. States are mapped to class labels without uncertainty with $P(\mathrm{class}_i|h) = 1$ if $i = (h \mod 10)$ or 0 otherwise.

For each data set, we make three disjoint sets: an unlabeled set of 200K data points, a test set of 10K data points, and a labeled training set of 90K data points from which various amounts of labeled data are drawn. Labeled data are drawn so that there is at least one positive example for every class but randomly otherwise.

7.2.2 Parameter setting

In the synthesized data experiments, regularization parameters for both supervised learning (γ in Section 7.1.4) and unsupervised learning (κ in Equation (13)) when applicable are fixed to 0.0001. The dimensionality (of each view) for AUX and CCA is set to the one which the theoretical analysis suggests (i.e., the dimensionality of the hidden state vector, which is the number of states k in our setup).

For the baseline methods, we show the performance essentially in their best configuration to show their potential irrespective of parameter selection methods. That is, for PCA, we show the performance at the best dimensionality chosen from $\{10, 20, \ldots, 200, 250, \ldots, 1000\}$. For NB-EM and co-training, the best performance among all the iterations is shown. (In theory, NB-EM should be iterated until convergence, but empirically, an early stop often improves performance significantly.)

7.2.3 With respect to the independence theorem

For each configuration, we perform 10 runs (five runs on each of two data sets generated by the same model with different random seeds) and report the average and standard deviation (vertical bars in the figures) of classification accuracy.

The first type of data set ('easy' data sets) was generated so that all the feature components are conditionally independent given classes, and that the optimum predictor employing one view only (as well as the one with both views) would achieve 100% classification accuracy. These data sets satisfy the assumptions of NB-EM and co-training. Data generation was done by making the index sets S_h's all disjoint and of the same size. The results in Figure 2 (a) shows that on the 'easy' data sets, all the semi-supervised methods perform well compared with the supervised baseline. (NB-EM slightly underperforms others, but with more labeled data, it also achieves 100%.)

The next type of data set was generated to test the methods on more difficult data sets. On these 'harder' data sets, the class-conditional independence assumptions for NB-EM and co-training no longer hold; also, some of the feature

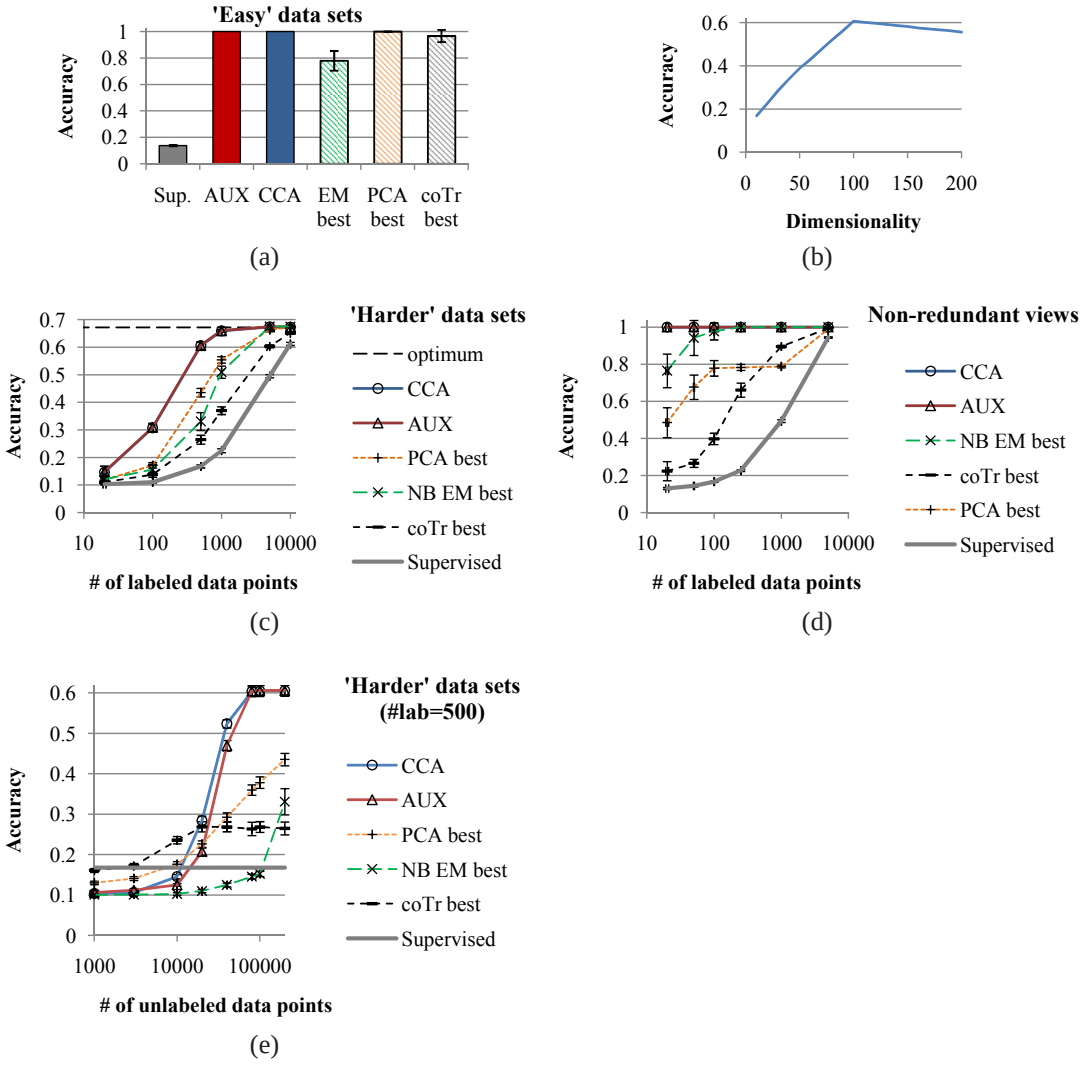


Figure 2: (a) 'Easy' data sets; 20 labeled examples. (b) CCA and dimensionality of each view on 'harder' data sets (#hidden states=100); 500 labeled examples. (c) 'Harder' data sets. (d) With non-redundant views. In (c) and (d), the lines for AUX and CCA are overlapping. (e) In relation to amounts of unlabeled data on 'harder' data sets; 500 labeled examples.

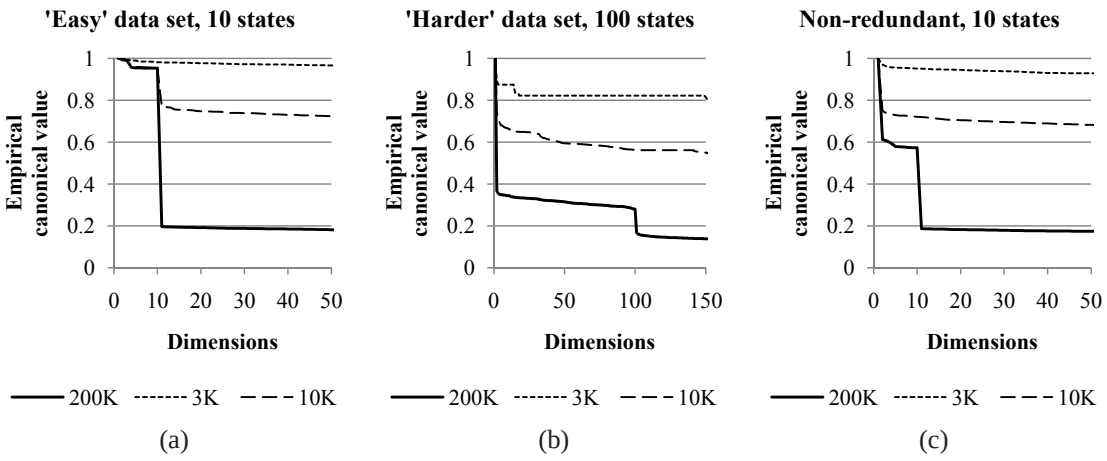


Figure 3: Empirical estimates of canonical values (λ_i in Section 3). Computed by taking square roots of eigenvalues of CCA computation using 3K, 10K, and 200K data points. The values are scaled so that the first canonical value equals to 1. (a) 'Easy' data set, (b) 'Hard' data set, (c) Non-redundant view data set.

components are not predictive. To violate the class-independence assumption, the data sets were generated from 100 states (larger than the number of classes), and to simulate irrelevant features, we let the index sets S_h's overlap with each other so that 30 indices are shared by all. Note that although the class-conditional independence no longer holds, the views are still conditionally independent given states.

Figure 2 (c) shows that on the 'harder' data sets, AUX and CCA (whose performances are almost identical) outperform the others. The broken horizontal line labeled as "optimum" represents an approximation of the optimum predictor's performance, obtained by training a supervised classifier with a very large amount of labeled data. AUX and CCA come very close to the optimum performance with 500 labeled data points and reach the optimum with 1000 labeled data points; by contrast, the supervised performance is still about 10% below the optimum performance even with 10000 labeled data points. It shows that the optimum performance can be achieved with the reduced dimensionality when trained with a sufficient amount of labeled data; thus, we observe that, as is theoretically predicted, dimension reduction by AUX and CCA with appropriate dimensionality do not lose anything required for prediction.

Subsequent experiments use data sets with non-redundant views in the sense that one of the views is far from being sufficient for prediction by itself, which violates one of the co-training assumptions. On these data sets, the optimum predictor employing view-1 achieves 100% accuracy, but the optimum predictor employing view-2 can achieve about 50% accuracy. The data sets were generated with 10 states so that they satisfy the class-independence assumptions of NB-EM and co-training. View-1 was generated without any irrelevant features (i.e., S_h are all disjoint), and view-2 was generated with 250 irrelevant features (shared by all S_h's), which makes prediction using view-2 alone difficult. As shown in Figure 2 (d), not surprisingly, these data sets are particularly hard for co-training. AUX and CCA (again whose performance is almost identical with each other's) reach the optimum predictor's performance (100%) with 50 labeled data points. NB-EM, whose assumption holds on these data sets, also performs well.

Figure 2 (b) shows CCA's performance dependency on the dimensionality when 500 labeled examples are used on the 'harder' data sets (used in (c)). The results indicate that the best performance is achieved when the dimensionality of each view is the number of states (100), which matches our theoretical prediction. The same phenomena are observed also with AUX, on other synthesized data sets, and with various amounts of labeled data.

It appears that irrelevant features raise PCA's best dimensionality. PCA's performance often peaks with 600–800 dimensions on the non-redundant data sets and with 400–600 dimensions on the 'harder' data sets. In contrast, CCA and AUX require only 2×10 dimensions for the former ("$2\times$" for two views) and 2×100 dimensions for the latter to reach the optimum performance. This explains why on these data sets PCA underperforms CCA when the amount of labeled data is small.

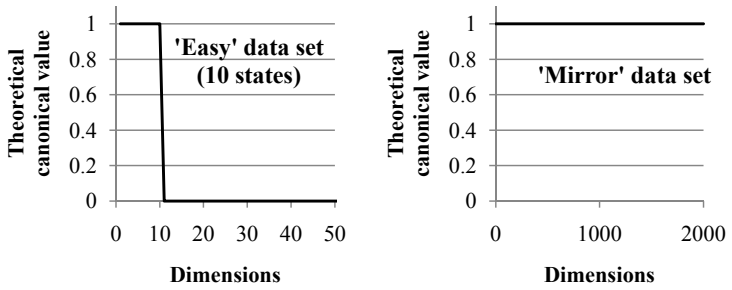


Figure 4: Theoretical canonical values. (a) 'Easy' data set. (b) 'Mirror' data set, in which the first view is the same as 'Easy' data set and the second view mirrors the first view.

7.2.4 On the amount of unlabeled data

Figure 2 (e) shows performance dependency of all the tested methods on amounts of unlabeled data when 500 labeled examples are used on the 'harder' data sets. When the amount of unlabeled data is relatively small, use of unlabeled data with any of the tested methods rather degrades performance. Among them, co-training appears to be less vulnerable to the smallness of unlabeled data, starting to improve performance over the supervised baseline when the amount of unlabeled data is as small as 3K while 10K of unlabeled data points still hurt all other methods. However, co-training performance stops increasing when the number of unlabeled points reaches 12K while the other methods are still improving; as a result, when the maximum amount (200K points) of unlabeled data is used, co-training turns out to be the worst performer. Based on this result, we note that performance comparison with only one fixed amount of unlabeled data may not be sufficient for fully understanding the characteristics of the methods and could be misleading, though sometimes it may be inevitable due to limited availability of data.

Recall that unlike co-training, our theoretical justification of CCA (and AUX) builds on expected values such as $\mathbb{E}[X^{(1)}(X^{(2)})^\top]$, which is assumed to be reliably estimated from large amounts of unlabeled data in practice. Therefore, it is plausible that when the amount of unlabeled data is too small to give good estimates, CCA (and AUX) dimensionality reduction cannot perform well. The same can be said for EM, which explicitly estimates probabilities from unlabeled data, and also PCA (with the Gaussian interpretation).

Figure 3 plots empirical estimates of *canonical values* λ_i (in Section 3) on the synthesized data sets, which were computed by taking square roots of eigenvalues of the eigenproblems for CCA computation. Recall that essentially the i-th canonical value represents correlation between two views in the i-th canonical direction. On all the three data sets, when computed using 200K unlabeled data points, the values show a prominent difference between the k-th and $(k+1)$-th dimensions where k is the number of states we generated the data from, and this is close to how theoretical canonical values (expectations) should look. (Theoretically, there should be precisely k non-zero canonical values for the data with a k-dimensional linear hidden state.) However, when computed from small amounts of unlabeled data (3K or 10K points), the curves become quite different from what they should be, which is consistent with CCA's relatively poor performance with small amounts of unlabeled data.

In an ideal situation, one could choose the optimum dimensionality by looking for the gap in empirical canonical values or eigenvalues. But in reality it is unlikely the situation would be precisely ideal. The amount of unlabeled data may not be sufficient, or the predictive hidden state assumption may not precisely hold. In Figure 6 we show empirical canonical values of two real-world data sets that we experiment with in the next section. There is no clear gap in the canonical values in the figure, which indicates that the situation is not ideal.

7.2.5 With respect to the redundancy theorem

So far we have confirmed that the experimental results are consistent with Theorem 3 (the independence theorem), whose assumptions include uncorrelatedness of two views conditioned on a linear hidden state. Now we turn to

Data sets	Accuracy X	Accuracy $X^{(1)}$	Accuracy $X^{(2)}$	Empirical $R^2_{X,Y}$	Empirical $R^2_{X^{(1)},Y}$	Empirical $R^2_{X^{(2)},Y}$	min ϵ to satisfy Assumption 2
Easy	100.0	100.0	100.0	1.00	1.00	1.00	0.00
Harder	67.3	45.8	45.7	0.84	0.78	0.78	0.06
Non-redundant	100.0	100.0	48.3	1.00	1.00	0.75	0.25

Figure 5: The synthesized data sets and Assumption 2. The R^2 values were computed on one class (out of 10 classes) by regarding the predictor trained with 200K labeled data points as the optimum predictor β and applying it to the 100K data points of the test set. The accuracy values (%) are for 10-class categorization obtained by the same training/test split.

Theorem 5 (the redundancy theorem). Recall that the theorem shows that the possible loss of predictive power of each view caused by CCA's dimensionality reduction is no greater than $(4\epsilon)/(1-\lambda)$. λ is a canonical value threshold which determines dimensionality. ϵ, from Assumption 2, quantifies the difference in predictive power (in terms of R^2) between the optimum predictor employing both views and the one employing a single view. A smaller ϵ indicates more redundant views.

When the dimensionality of the hidden state vector is k, in theory only the first k canonical values should be positive, as illustrated in Figure 4 (a). Then the redundancy theorem suggests that dimensionality k is the best choice, which is consistent with the independence theorem and consistent with our experimental results. Dimensionality k is suggested because reducing dimensionality to k would lose predictive power up to nearly 4ϵ since the threshold λ can be set to an infinitesimally small positive value, and this much loss is much better than that of dimensionality $k-1$ and nearly the same as that of any dimensionality larger than k. Estimates of ϵ on the synthesized data sets are shown in Figure 5. On the last data set in the table, ϵ is relatively large (indicating the views are not so redundant), which makes the absolute value of the bound too loose. Nevertheless, relative comparison of the bounds among dimensionalities is still useful for choosing the right dimensionality as described above.

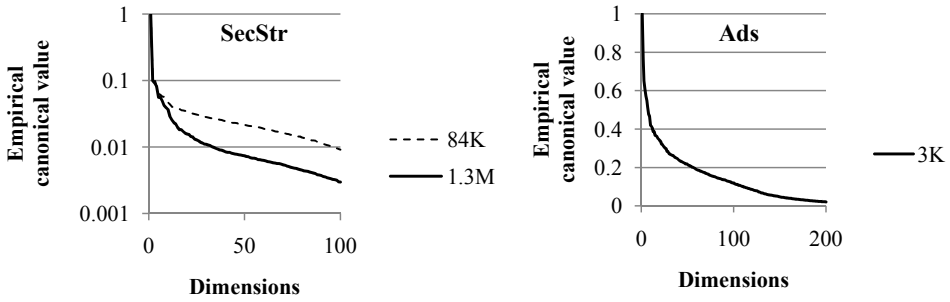
Given the two theorems, a natural question would be: what happens when the views are not independent but redundant to some degree. An extreme end of this direction is a data set in which two views are exactly the same and therefore totally redundant ($\epsilon = 0$) and dependent on each other. Consider what we call 'mirror data set', whose first view is generated as in the 'easy' data set above, and whose second view is exactly the same as the first view (i.e., the second view mirrors the first view). It is easy to verify that theoretically the canonical values of the 'mirror' data set is 1 for all the 2000 dimensions as in Figure 4 (b). Then the redundancy theorem suggests that on the 'mirror' data set, there is no guarantee that the predictive power would not be degraded too much by CCA dimensionality reduction, since whatever dimensionality we choose, λ in the theorem becomes 1.

In order to apply the independence theorem, one should realize that the 'mirror' data set can be interpreted as being generated from 2000 states, each of which is associated with each component of view-1 and view-2. With this interpretation, the independence theorem suggests that again we should not reduce dimensionality by CCA as the suggested optimum dimensionality is 2000, the same as the original dimensionality. The same can be said about AUX. Although the 'mirror' data set is an extreme example, things (such as λ_i's and the optimum dimensionality) continuously change when we gradually change the data set by increasing dependency between views from the 'easy' data set to the 'mirror' data set.

That is, even though the redundancy theorem is free of the independence assumption, it still has some connection to the hidden state assumption through λ_i or how fast λ_i decays. In order to benefit from CCA dimensionality reduction, the views still need to be to some extent uncorrelated, conditioned on a relatively low-dimensional and predictive hidden state.

7.3 Real-world data experiments

In order to study the behavior of the methods on real-world tasks, we experiment with two real-world data sets. One is the SecStr data set used in the benchmarking of a number of semi-supervised methods in Chapelle et al. [2006], whose associated task is to detect the secondary structures of amino acids. The other is the Ads data set for detecting web



(a) (b)

Figure 6: Empirical canonical values of real-world data sets. (a) SecStr, (b) Ads.

links to advertisements.

7.3.1 Implementation details

Since co-training is the only method that outperformed CCA and AUX and successfully benefited from unlabeled data in some settings in the synthesized data experiments, we focus on co-training and supervised classifiers as baseline methods.

For all the methods, feature vectors (including CCA and AUX's features derived from unlabeled data) are length-normalized so that the portion corresponding to each view becomes a unit vector, which narrows the range of appropriate regularization parameters and helps to simplify experimental setup as described below.

Selection of regularization parameters (γ in Section 7.1.4) based on a small amount of labeled data is known to be difficult as is often mentioned (e.g., Chapelle et al. [2006]). To separate out the issue of regularization parameter selection, which is shared by the baseline methods as well, for all the methods, we report performance obtained by γ fixed to a constant, which was 0.01 on both data sets. Although fine tuning of γ by cross validation for each run might improve the performance (or rather degrade when the number of labeled data is small), setting $\gamma = 0.01$ appears to be a good choice in most cases, and it is sufficient for the purpose of studying the behavior of the methods.

7.3.2 SecStr

In Chapelle et al. [2006], a number of semi-supervised systems were tested on several benchmark data sets[2]. Among them, we chose SecStr, because it was found to be particularly difficult for the systems participating in benchmarking, and because multiple views of features are provided.

According to Chapelle et al. [2006], the task associated with the SecStr data set is "to predict the secondary structure of a given amino acid in a protein based on a sequence window centered around that amino acid". The main purpose of this data set in the original benchmarking event was to test on a large-scale application. The data set consists of 84K data points, of which either 100 or 1000 data points are used as labeled examples while the rest serve as test data. In addition, 1200K extra unlabeled data points are provided, but none of the benchmarking participants used them.

The provided feature vectors are 315-dimensional, which represent 15 categorical attributes (each of which ranges over 21 categories) generated from 15 sequence windows of amino acids at the positions in [-7,+7]. We divide the provided features into two views as follows:

- View-1: Attributes based on the left context (positions in [-7,-1]). 147 dimensions.
- View-2: Attributes based on the current position and right context (positions in [0,7]). 168 dimensions.

[2]Downloadable at http://www.kyb.tuebingen.mpg.de/ssl-book.

We used the provided ten training/test splits and the provided feature vectors with length normalization mentioned above. Since the original benchmarking was designed to be in transductive learning settings (i.e., test data is available at the time of training), our experiments were also done in transductive learning settings – using all the data points as unlabeled data points.

In relation to dimensionality The first series of experiments on SecStr study the performance of the methods in relation to dimensionality (of each view) and the number of co-training iterations. The solid and dotted lines of the leftmost charts ((a) and (d)) of Figures 7 and 8 are CCA's accuracy averaged over 10 runs for each dimensionality in $\{2, 3, \cdots, 20, 25, \cdots, 100\}$. The solid lines labeled as CCA1 are the results obtained by setting $\kappa = 0.01$ ($= \gamma$; a natural choice), and the dotted lines labeled as CCA2 are the results obtained by setting $\kappa = 0.001$ to see sensitivity to changes in κ. (As a reminder, κ is a regularization parameter for solving least squares on unlabeled data; see Section 7.1.1.) The circles with horizontal bars represent the average of the accuracy at the best dimensionality of each run, and the horizontal bars represent the standard deviation of the best dimensionality. (Note that since the best dimensionality varies among the 10 runs, the average of the accuracy at the best dimensionality is typically higher than the peak of the line, which averages over 10 runs at a fixed dimensionality.) Shorter horizontal bars indicate that the best dimensionalities of the 10 runs are close to each other, which could make dimensionality selection easier. Similarly, the center charts ((b) and (e)) show AUX's performance, and the right-most charts ((c) and (f)) show co-training performance in relation to the number of iterations of co-training.

Comparison of (a)-(c) with (d)-(f) in both figures shows that use of a larger amount of unlabeled data improves accuracy for both CCA and AUX but not for co-training, which is consistent with our synthesized data experiments. The CCA and AUX's average performance exceeds the supervised baseline (solid gray lines) at wide ranges of dimensionality except for extremely low ones. The 'long dash dot' lines in the figures represent the previous best results (which will be described below). When both the numbers of labeled and unlabeled data points are relatively small (100 and 84K, respectively), the range of dimensionality that produce higher accuracy than the previous best results is narrow (roughly 2-20 dimensions). In the other three settings, both CCA and AUX exceed the previous best results unless the dimensionality is very low.

Choosing dimensionality by cross validation Although the performance results observed so far look promising, they also show that performance critically depends on choice of dimensionality. In Figure 9, we show CCA and AUX's performance when the regularization parameters γ and κ were fixed to 0.01 and dimensionality was chosen from $\{2, 3, \cdots, 50\}$ by 5-fold cross validation on the labeled data points. We regard these as 'semi-practical' performance since the regularization parameters were fixed to an okay value instead of finding a value from scratch by cross validation. Fine tuning of κ and γ by cross validation for each run might improve performance (or rather degrade performance with a small number of labeled instances), but it would also depend on details of cross validation such as tie-breaking mechanism or the number of folds, which we do not study in this paper. For comparison, the last two rows show the unrealistic best potential performance, which could be obtained if we knew the optimum dimensionality and κ in the range of $\{0.1, 0.01, 0.001, 0.0001\}$ (γ was still fixed to 0.01). Comparison of the semi-practical performance with the best potential performance reveals that as expected, dimensionality selection based on only 100 labeled data points is difficult, losing to the oracle potential performance by roughly 3–5%. With 1000 labeled data points, the performance differences become smaller (less than 1% in most cases).

The first rows of Figure 9 show Chapelle et al. [2006]'s benchmarking results using cluster kernels and graph-based learning methods such as Laplacian SVM. Essentially, these methods modify/improve kernels using unlabeled data based on certain assumptions (e.g., Laplacian-based methods assume that output values associated with nodes change smoothly over the graph). The results for Laplacian methods (numbers in parentheses) are potential performance obtained by selecting the best model based on the performance on test data. Chapelle et al. [2006] suggests that the relatively poor performance of the participating systems might be because only relatively simple strategies were feasible given the relatively large size of the unlabeled set, even though participants only used the smaller unlabeled set (84K).

Mann and McCallum [2007] proposed Expectation Regularization (XR) which performs regularization so that predictions on unlabeled data match certain expectations given as input from the user. In their experiments, the expectations were given in the form of "the true label priors" estimated from the data including the test data, which

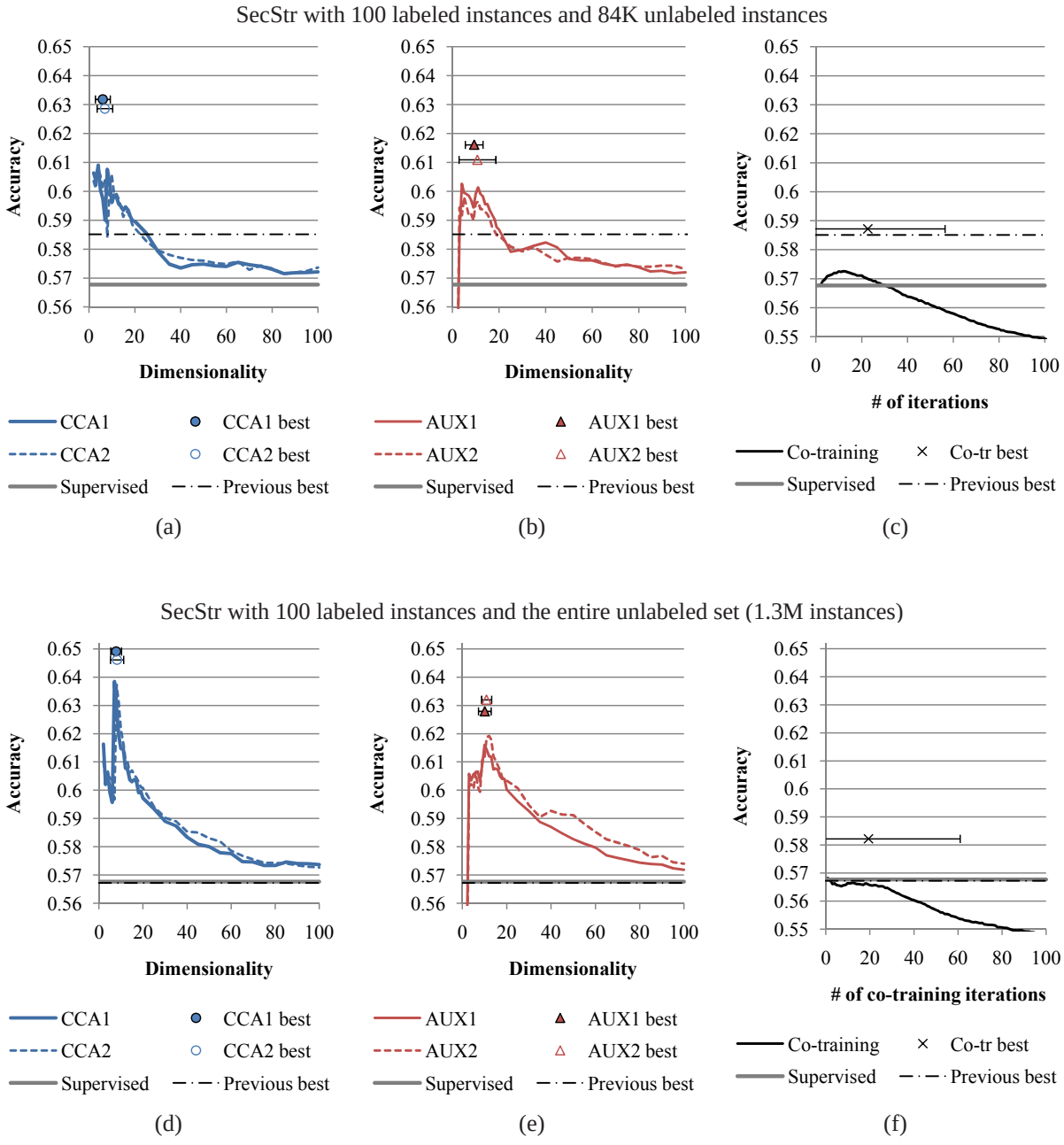


Figure 7: SecStr with 100 labeled instances; accuracy in relation to dimensionality and the number of co-training iterations; average of 10 runs. CCA1/AUX1: $\kappa = 0.01 (= \gamma)$. CCA2/AUX2: $\kappa = 0.001$. "... best" indicates the average accuracy at the best dimensionality/# of iterations. The horizontal bars of the "... best" are the standard deviations of the best dimensionality or the number of co-training iterations.

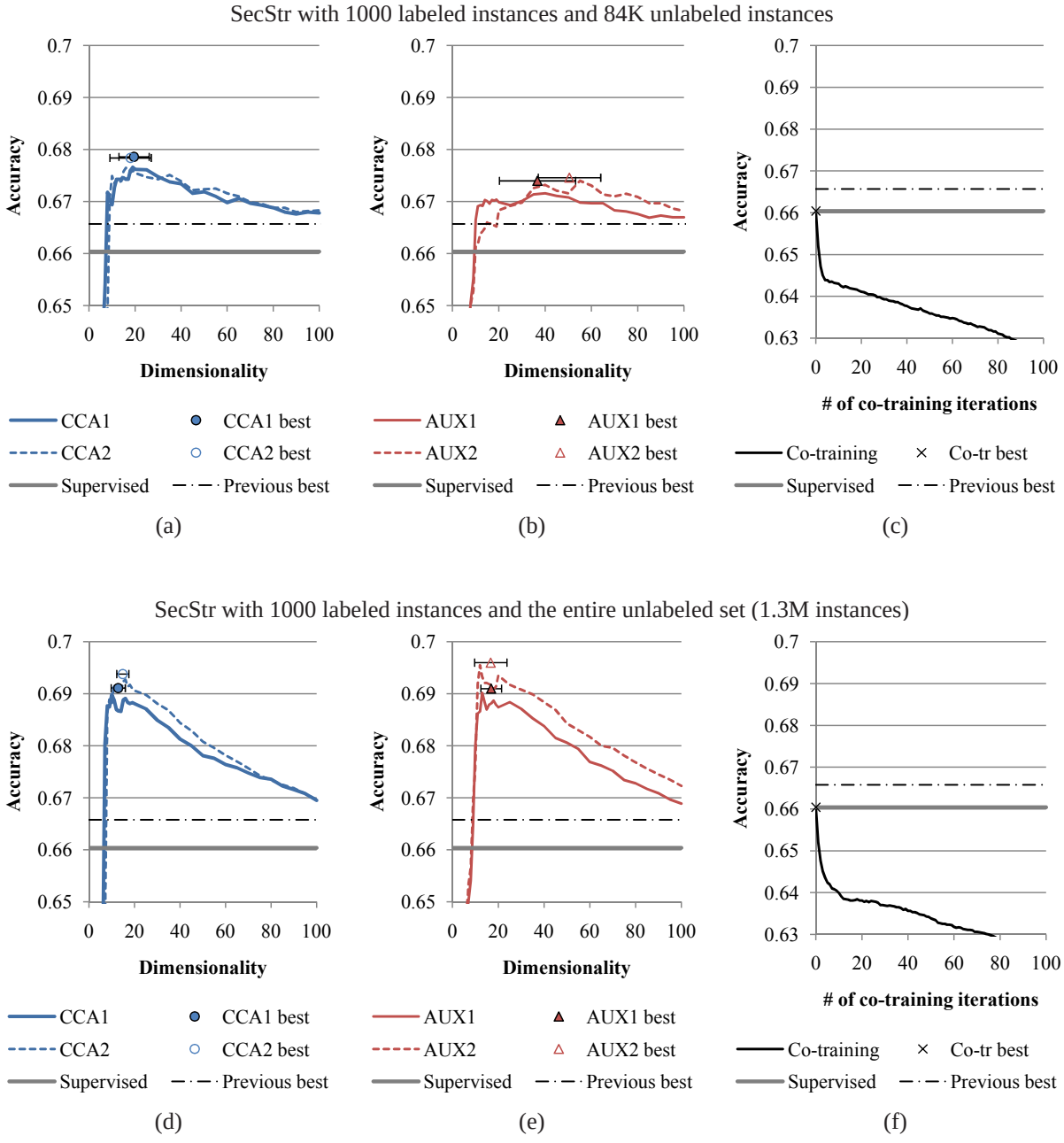


Figure 8: SecStr with 1000 labeled instances. The settings are the same as Figure 7 except for the number of labeled instances.

Previous results (average)		#labeled=100		#labeled=1000	
		#unlab=84K	#unlab=1.3M	#unlab=84K	#unlab=1.3M
Chapelle et al. [2006]	SVM (supervised)	55.41		*66.29*	
	Cluster Kernel	57.05	No Attempt	65.97	No Attempt
	QC randsub (CMN)	57.68		59.16	
	QC smartonly (CMN)	*57.86*		59.29	
	QC smartsub (CMN)	57.74		59.16	
	Laplacian RLS	(57.41)		(65.83)	
	Laplacian SVM	(56.58)		(66.04)	
Mann and McCallum [2007]	NB (supervised)	57.12		64.47	
	MaxEnt (supervised)	56.74		65.43	
	NB EM	57.34	No Attempt	57.60	No Attempt
	MaxEnt+Ent.Min.	54.45		58.28	
	MaxEnt+XR	*58.51*		*65.44*	
Loeff et al. [2008]	Tree-ManifoldBoost	*57.30*	*56.72*	*66.57*	*66.58*

Our experiments (average$\pm$standard deviation)

	Supervised RLS	56.77$\pm$2.65		66.04$\pm$0.68	
Fix γ and κ to 0.01 and choose dim by cross validation	AUX	*58.77*$\pm$3.72	60.09$\pm$4.25	66.93$\pm$0.57	*68.64*$\pm$0.41
	CCA	58.76$\pm$4.23	*60.83*$\pm$4.75	*67.53*$\pm$0.47	68.49$\pm$0.66
With the best dim and κ (unrealistic potential perf.)	AUX	(62.19$\pm$2.40)	(63.59$\pm$3.21)	(67.59$\pm$0.53)	(*69.65*$\pm$0.33)
	CCA	(*63.43*$\pm$2.03)	(*65.38*$\pm$2.00)	(*68.00*$\pm$0.34)	(69.54$\pm$0.40)

Figure 9: Accuracy (%) on SecStr. The best results in each group of rows are highlighted. The parenthesized numbers represent the best possible accuracy of the method (instead of practical performance) produced by choosing the models by observing the performance on the test data.

simulated the application setting where the user possesses the knowledge of true priors. In their results, XR is shown to be effective with 100 labeled examples, but it appears to have almost no merit over the supervised baseline when 1000 labeled examples were used. They did not use the extra unlabeled data points.

Loeff et al. [2008] used the extra unlabeled data of SecStr in their experiments with Tree-ManifoldBoost and reported that, however, the use of additional unlabeled data rather degraded performance (with 100 labeled examples) or produced a very small improvement (0.01% with 1000 labeled examples).

The computation of CCA and AUX is linear in the number of unlabeled data points. Solving the eigenproblems could be time consuming, but its computation depends on the feature dimensions (computing eigenvectors of a $|\text{view1}| \times |\text{view2}|$ matrix for CCA and a $|\text{view}i| \times |\text{view}i|$ matrix for AUX), not the number of unlabeled data points. This is in contrast to the graph-based methods, which are quadratic in the number of unlabeled data points. Since SecStr has relatively low-dimensional features, the CCA and AUX computation with the entire unlabeled set (1.3M points) takes only a few minutes. Compared with the previous results, CCA (and also AUX) show good potential on this data set especially in their ability to efficiently make use of and actually benefit from relatively large amounts of unlabeled data. A challenge in practical applications would be, like many other semi-supervised methods, parameter selection based on a small number of labeled examples, though.

7.3.3 Ads (the Internet Advertisement data set)

The next real-world data set we experiment with is Ads (the internet advertisement data set)[3] from the UCI Machine Learning Repository. The data set consists of 3279 instances, each of which represents a hyperlink associated with an image on a web page, and the task is to predict whether the link points to an advertisement or not based on the size of the image, terms in the caption, URL pointed by the link, and so on. Details of data creation and the feature design are described in Kushmerick [1999]. The data set has been used for studying multi-view learning methods in Muslea et al. [2002].

[3]The data set was downloaded from: http://archive.ics.uci.edu/ml/datasets/Internet+Advertisements .

Most of the provided features are binary (1/0) indicating the presence/absence of corresponding terms. The only changes we made to the provided features are to convert the continuous values of the width (and height) of the image into seven binary features by asking whether the value is smaller than $20, 40, 80, 160, 320,$ and 640, and also apply view length-normalization as described above. Our supervised RLS performance using these features is similar to but slightly higher than C4.5's performance in Kushmerick [1999] using the provided features in similar training/test splits.

Following Muslea et al. [2002], we define two views as follows:

- View-1: features based on the image itself such as its width, height, terms in the alt and the caption. 604 dimensions.
- View-2: features based on the other information such as terms in the anchor URL and the URL of the current site. 967 dimensions.

Since the data set contains only 3279 instances, there would not be much unlabeled data or test data if we made them disjoint. We used the entire data set as an unlabeled data set as in a transductive setting. Since there is no official training/test split provided for this data set, following Muslea et al. [2002], we randomly choose 100 instances as labeled training instances, and evaluate accuracy on the rest (3179 instances). We report the average of 10 runs.

Figure 10 shows accuracy averaged over 10 runs in relation to the dimensionality of CCA and AUX or the number of co-training iterations. As in the SecStr results, the points with horizontal bars represent the average of the performance at the best dimensionality (or the best number of iterations), and the horizontal bars represent the standard deviation of the best dimensionality (or the number of iterations).

CCA shows a clear performance peak around 15 dimensions, and the short horizontal bars of the circles indicate that all the 10 runs' performance peak concentrate around that dimensionality. AUX's result is fuzzier. For co-training, in addition to 'coTr1' which lets $\gamma = 0.01$ as in all other settings, the results obtained by setting $\gamma = 0.001$ labeled as 'coTr2' are also shown, as they are clearly better than 'coTr1'. The pool size and increment size for co-training were set to 1000 and 100 since 10000 and 500 (used for all other co-training experiments) are too large for this small data set. All the tested methods generally exceed the supervised performance.

AUX shows sensitivity to κ and generally underperforms CCA. Such discrepancy between CCA and AUX might be due to the smallness of unlabeled data. Figure 10 (d) and (e) show CCA and AUX's performance when the original features are concatenated with the features derived from unlabeled data. Inclusion of the original features in this way on this data set improved performance particularly with AUX. Though we did not analyze this configuration in this work, Ando and Zhang [2005] did. Intuitively, the original features could supplement predictive information that dimensionality reduction might lose, so inclusion of the original features could improve performance in some cases especially when a relatively large amount of labeled data is available.

Figure 11 shows Muslea et al. [2002]'s results using unlabeled data on this data set in a comparable setting. Compared with our co-training results, their co-training accuracy seems low, presumably due to the difference in base classifiers. If dimensionality is chosen adequately, CCA will produce higher accuracy than the previous results.

7.4 Computation time

As shown in Sections 7.1.1 and 7.1.2, computation of AUX and CCA consists of three parts. Let d_i be the dimensionality of the i-th view. The first part goes through n unlabeled data points and computes $\mathbf{X}^{(i)}(\mathbf{X}^{(i)})^\top$ and $\mathbf{X}^{(i)}(\mathbf{X}^{(j)})^\top$, which is in $O(n \cdot \max_i d_i^2)$. The second part, which inverts a d_i-by-d_i matrix to compute $\mathbf{W}^{(i)}$, and the third part, which solves the eigenproblems, are both in $O(\max_i d_i^3)$. For comparison, computation time of co-training in our experiments is in $O(n \cdot \max_i d_i^2)$ when iterated until the n unlabeled data points are exhausted (d_i^2 is for solving the least square optimization of the base learner). Let $d = d_1 + d_2$. Computation of NB-EM is $O(n \cdot d)$ with a constant number of iterations. Assuming $n >> d$, the PCA computation is in $O(n \cdot d^2)$ to compute $\mathbf{X}\mathbf{X}^\top$ and in $O(d^3)$ to solve the eigenproblem.

In our experiments, d_i is at most 2000, and the number of unlabeled data points varies from about 3000 to over one million. With this range of d_i and n and given the fact that most of the feature components are zeroes, computation of AUX and CCA is relatively fast. Throughout the experiments, the first part and the second part in total took a few minutes at most, and the third part took from several seconds to 10 minutes, using an ordinary desktop computer.

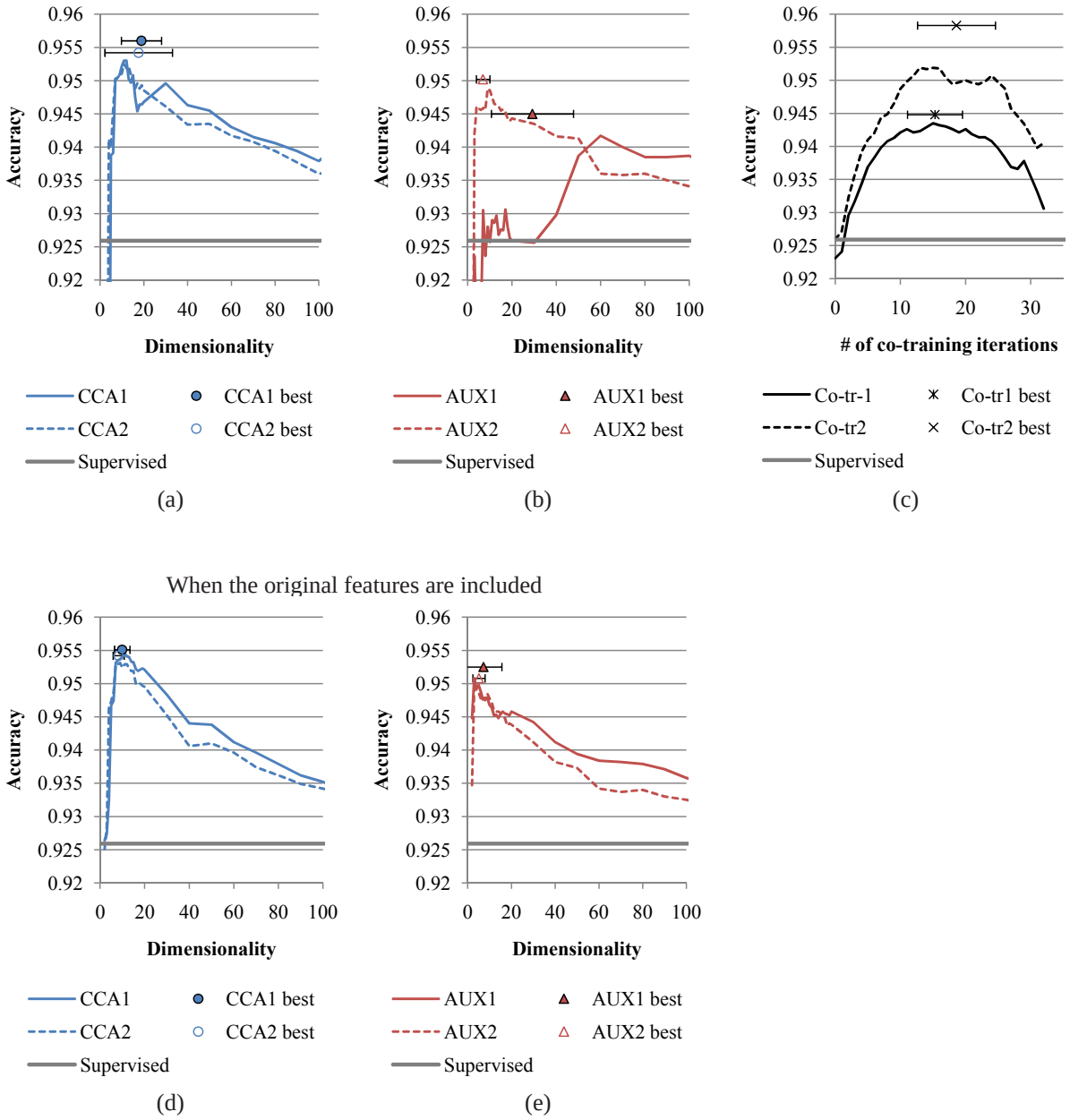


Figure 10: Advertisement data set; 100 labeled instances; accuracy in relation to dimensionality and # of co-training iterations; average of 10 runs. CCA1/AUX1: $\kappa = 0.01 (= \gamma)$; CCA2/AUX2: $\kappa = 0.1$; coTr1: $\gamma = 0.01$, coTr2: $\gamma = 0.001$. "... best" indicates the average accuracy at the best dimensionality/# of iterations. The horizontal bars of the "... best" are the standard deviations of the best dimensionality/# of iterations.

Muslea et al. [2002]	co-EMT (multi-view active learning)	94.25
	co-EM	92.20
	EM	91.45
	co-training (naive Bayes-based)	92.46

Figure 11: Previous results in similar training/test splits on Ads.

Data sets	Accuracy X	Accuracy $X^{(1)}$	Accuracy $X^{(2)}$	Empirical $R^2_{X,Y}$	Empirical $R^2_{X^{(1)},Y}$	Empirical $R^2_{X^{(2)},Y}$	min ϵ to satisfy Assumption 2
SecStr	72.4	63.6	69.6	0.27	0.10	0.20	0.17
Ads	98.2	97.9	82.6	0.93	0.92	0.76	0.17

Figure 12: The real-world data sets and Assumption 2. The R^2 values were computed by regarding the predictor trained with 90% of the data points as the optimum predictor β and applying it to the 10% of the data points. These may not be good estimates since the number of labeled data points used for training may not be large enough to approximate the optimum predictor.

However, if the original feature space is very high-dimensional, the second and third parts in $O(\max_i d_i^3)$ could become impractically expensive. In that case, AUX and CCA computation needs to be approximately solved.

7.5 Discussion

We have presented experiments using synthesized data and real-world data. The synthesized data experiments confirmed that the best performance is obtained when the dimensionality is set to the dimensionality of the hidden state vector, and the methods of dimensionality reduction do not lose anything required for prediction. We also showed that the dimensionality suggested by Theorem 5 (the redundancy theorem) is consistent with that suggested by Theorem 3 (the independence theorem) even though these two theorems are based on distinct assumptions. The real-world data experiments demonstrated that there exist real-world data sets with low-dimensional structures, which can be exploited by the multi-view dimensionality reduction methods. Note that it is not possible to directly examine whether Assumption 1 (the linear hidden state assumption) holds on the real-world data sets since the hidden state is unknown. The degree of view redundancy can be estimated to some extent as in Figure 12, but they may not be good estimates due to the limitation to the availability of labeled data.

In the ideal setting where the assumptions of Section 4 are completely satisfied and the amounts of unlabeled data are large enough to produce accurate estimates of the expected values, AUX and CCA would produce the same results. This is in fact what we observed on the synthesized data experiments. However, AUX and CCA produced distinct results on all the real-world data sets we experimented with. The indication is that either the amount of unlabeled data available to us are not large enough or the assumption in Section 4 do not accurately hold. Therefore, in practice, the two methods should be considered as two options to choose from depending on the specific task and the data at hand. One may also note that results in Section 5 apply only to CCA but not to AUX. This means these methods can be different under different assumptions. It will therefore be interesting to further characterize the difference between AUX and CCA theoretically.

References

Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6:1817–1853, 2005. ISSN 1533-7928.

Rie Kubota Ando and Tong Zhang. Two-view feature generation model for semi-supervised learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 25–32, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. doi: http://doi.acm.org/10.1145/1273496.1273500.

Francis R. Bach and Michael I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, U.C. Berkeley, 2005.

M. Balcan and A. Blum. A pac-style model for learning from labeled and unlabeled data. pages 111–126, 2005.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, pages 92–100, 1998.

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. URL `http://www.kyb.tuebingen.mpg.de/ssl-book`.

R. Dennis Cook and Sanford Weisberg. Sliced inverse regression for dimension reduction: Comment. *Journal of the American Statistical Association*, 86(414):328–332, 1991.

David R. Hardoon, Sandor R. Szedmak, and John R. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Comput.*, 16(12):2639–2664, 2004. ISSN 0899-7667.

Sham M. Kakade and Dean P. Foster. Multi-view regression via canonical correlation analysis. In Nader H. Bshouty and Claudio Gentile, editors, *COLT*, volume 4539 of *Lecture Notes in Computer Science*, pages 82–96. Springer, 2007.

Nicholas Kushmerick. Learning to remove Internet advertisements. In *Proceedings of 3rd Intertaional Conference on Autonomous Agents*, 1999.

Ker-Chau Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1992.

Nicolas Loeff, David Forsyth, and Deepak Ramachandran. ManifoldBoost: Stagewise function approximation for fully-, semi- and un-supervised learning. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 600–607, 2008.

Gideon S. Mann and Andrew McCallum. Simple, robust, scalable semi-supervised learning via expectation regularization. In *Proceedings of International Conference on Machine Learning (ICML)*, 2007.

Ion Muslea, Steve Minton, and Craig Knoblock. Active + semi-supervised learning = robust multi-view learning. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 435–442, 2002.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, Special issue on information retrieval:103–134, 2000.

Abhishek Tripathi, Arto Klami, and Samuel Kaski. Simple integrative preprocessing preserves what is shared in data sources. *BMC Bioinformatics*, 9(111), 2008.