# Optimal Reverse Prediction

## A Unified Perspective on Supervised, Unsupervised and Semi-supervised Learning

**Linli Xu**                                         linli@cs.ualberta.ca
**Martha White**                                    whitem@cs.ualberta.ca
**Dale Schuurmans**                                  dale@cs.ualberta.ca
University of Alberta, Department of Computing Science, Edmonton, AB T6G 2E8, Canada

## Abstract

Training principles for unsupervised learning are often derived from motivations that appear to be independent of supervised learning. In this paper we present a simple unification of several supervised and unsupervised training principles through the concept of *optimal reverse prediction*: predict the inputs from the target labels, optimizing both over model parameters and any missing labels. In particular, we show how supervised least squares, principal components analysis, k-means clustering and normalized graph-cut can all be expressed as instances of the same training principle. Natural forms of semi-supervised regression and classification are then automatically derived, yielding semi-supervised learning algorithms for regression and classification that, surprisingly, are novel and refine the state of the art. These algorithms can all be combined with standard regularizers and made non-linear via kernels.

## 1. Introduction

Unsupervised learning is one of the key foundational problems of machine learning and statistics, encompassing problems as diverse as clustering (MacQueen, 1967; Shi & Malik, 2000), dimensionality reduction (Mika et al., 1998), system identification (Katayama, 2005), and grammar induction (Klein, 2004). It is also one of the most studied problems in both fields. Yet, despite the long parallel history with supervised learning, the principles that underly unsupervised learning are often distinct from those underlying supervised learning. For example, classical methods such as prin-

cipal components analysis and k-means clustering are derived from principles for re-representing the input data, rather than minimizing prediction error on any associated output variables. Minimizing prediction error on associated outputs is obviously related to, but not directly determined by input reconstruction in any single, obvious way. Although some unification can be achieved between supervised and unsupervised learning in a pure probabilistic framework, here too it is not known which unifying principles are appropriate for discriminative models, and a similar diversity of learning principles exists (Smith & Eisner, 2005; Corduneanu & Jaakkola., 2006).

The lack of a unification between supervised and unsupervised learning might not be a hindrance if the two tasks are considered separately, but for *semi-supervised learning* one is forced to consider both together. Given both labeled and unlabeled training examples, we would like to know how to infer a better predictor than just using the labeled data alone. Unfortunately, the lack of a foundational connection has led to a proliferation of semi-supervised learning strategies (Zhu, 2005), while there are no guarantees that any of these methods will ensure improvements over using labeled data alone (Ben-David et al., 2008).

The dominant approaches to semi-supervised learning currently appear to be to use unsupervised loss as a regularizer for supervised training (Zhou & Schölkopf, 2006; Belkin et al., 2006; Corduneanu & Jaakkola., 2006); to combine self-supervised training on the unlabeled data with supervised training on the labeled data (Joachims, 1999; De Bie & Cristianini, 2003); to train a joint probability model generatively (Bishop, 2006); and to follow alternatives such as co-training (Blum & Mitchell, 1998). Unfortunately, the lack of a unified perspective has led to the slow development of supporting theory (although there has been continued progress (Balcan & Blum, 2005)). Instead, the literature relies on intuitions like the "cluster assumption" and the "manifold assumption" to provide help-

ful guidance (Belkin et al., 2006), but these have yet to lead to a general characterization of the potential and limits of semi-supervised learning.

In this paper we demonstrate a unification of several classical unsupervised and supervised training principles. In particular, we show how supervised least squares, principal components analysis, k-means clustering and normalized graph-cut can all be expressed as the same training principle. These methods differ only in the assumptions they make about the training labels (i.e. whether the labels are missing, continuous or discrete).[1] Interestingly, the unification is not based on predicting target labels from input descriptions (an approach that does not work), but rather the other way around: predicting input descriptions from associated labels. We will show that reverse prediction allows classical unsupervised training principles to be unified, while, importantly, it still allows standard forward regression to be recovered exactly. This unification encompasses both regularization and kernels.

Once this unification is established, we can then achieve our main result: a natural principle for semi-supervised learning. In particular, for least squares, we show that the reverse prediction loss decomposes into a sum of two independent losses: a loss defined on the labeled and unlabeled parts of the data, and another, orthogonal loss defined only on the unlabeled parts of the data. Given auxiliary unlabeled data, one can then reduce the variance of the latter loss estimate without affecting the former, hence achieving a strict variance reduction over using labeled data alone.

In the sequel, we first establish some preliminary foundations of forward least squares and then present the reverse prediction model, showing how the standard forward solution can be recovered even in the presence of ridge regularization and kernels. We then present the unification of supervised least squares with principal components analysis, k-means clustering and normalized graph-cut. With this unification, we demonstrate the reverse loss decomposition and present the new semi-supervised training principle. The paper concludes with an empirical evaluation on both regression and classification problems.

## 2. Preliminaries

Assume we are given input data in a $t \times n$ matrix $X$, with rows corresponding to instances and columns to features. For supervised learning, we assume we are given a $t \times k$ matrix of prediction targets $Y$. Regression and classification problems can be represented simi-

larly, where for classification one assumes the rows in $Y$ indicate the class label; that is, $Y \in \{0, 1\}^{t \times k}$ such that $Y\mathbf{1} = \mathbf{1}$ (a single 1 in each row).

**Notation**: We will use $\mathbf{1}$ to denote the vector of all ones, $\mathrm{tr}(\cdot)$ to denote matrix trace, $'$ to denote matrix transpose, and $\dagger$ to denote matrix pseudo-inverse.

For supervised learning, training typically consists of finding parameters $W$ for a model $f_W : X \mapsto Y$ that minimizes some loss with respect to the targets. We will focus on minimizing least squares loss. The following results are all standard, but specific variants our approach has been designed to handle are outlined.

For linear models, least squares training amounts to solving for an $n \times k$ matrix $W$ that minimizes

$$\min_W \ \mathrm{tr}\left((XW - Y)(XW - Y)'\right) \tag{1}$$

This convex minimization is easily solved to obtain the global minimizer $W = X^\dagger Y$. The result can then be used to make predictions on test data via $\hat{\mathbf{y}} = W'\mathbf{x}$ (thresholding $\hat{\mathbf{y}}$ in the case of classification).

Linear least squares can be trivially extended to incorporate regularization, kernels and instance weighting. For example, (ridge) regularization can be introduced

$$\min_W \ \mathrm{tr}\left((XW - Y)(XW - Y)'\right) + \alpha \, \mathrm{tr}\left(W'W\right) \tag{2}$$

yielding the altered solution $W = (X'X + \alpha I)^{-1}X'Y$.

A kernelized version of (2) can then be easily derived from the identity $(X'X + \alpha I)^{-1}X' = X'(XX' + \alpha I)^{-1}$, since this implies the solution of (2) can be expressed as $W = X'A$ for $A = (XX' + \alpha I)^{-1}Y$. Thus, the input data $X$ need only appear in the problem through the inner product matrix $XX'$. Once the input data appears only as inner products, positive definite kernels can be used to obtain non-linear prediction models (Schölkopf & Smola, 2002). For least squares, the kernelized training problem can then be expressed as

$$\min_A \ \mathrm{tr}\left((KA - Y)(KA - Y)'\right) + \alpha \, \mathrm{tr}\left(AA'K\right) \tag{3}$$

where $K$ corresponds to $XX'$ in some implicit feature representation. It is easy to verify that $A = (K + \alpha I)^{-1}Y$ is the global minimizer. Given this solution, test predictions can be made via $\hat{\mathbf{y}} = A'\mathbf{k}$ where $\mathbf{k}$ corresponds to the implicit inner products $X\mathbf{x}$.

Finally, we will need to make use of instance weighting in some cases below. To express weighting, let $\Lambda$ be a diagonal matrix of strictly positive instance weights. Then the previous training problem can be expressed

$$\min_A \ \mathrm{tr}\left(\Lambda(KA - Y)(KA - Y)'\right) + \alpha \, \mathrm{tr}\left(AA'K\right) \tag{4}$$

with the optimal solution $A = (\Lambda K + \alpha I)^{-1}\Lambda Y$.

---

[1]Normalized graph-cut also incorporates a re-weighting.

# 3. Reverse Prediction

Our first key observation is that the previous results can all be replicated in the reverse direction, where one attempts to predict inputs $X$ from targets $Y$. In particular, given optimal solutions to reverse prediction problems, the corresponding forward solutions can be recovered exactly. Although reverse prediction might seem counterintuitive, it will play a central role in unifying classical training principles in what follows.

For reverse linear least squares, we seek a $k \times n$ matrix $U$ that minimizes

$$\min_U \ \operatorname{tr}\left((X - YU)(X - YU)'\right) \tag{5}$$

This minimization can be easily solved to obtain the global solution $U = Y^\dagger X$. Interestingly, as long as $X$ has full rank, $n$, the forward solution to (1) can be recovered from the solution of (5).[2] In particular, from the solutions of (1) and (5) we obtain the identity that $X'XW = X'Y = U'Y'Y$, hence

$$W = (X'X)^{-1} U'Y'Y \tag{6}$$

For the other problems, invertibility is assured and the forward solution can always be recovered from the reverse without any additional assumptions. For example, the optimal solution to the regularized problem (2) can always be recovered from the solution to the plain reverse problem (5) by using the straightforward identity that relates their solutions, $(X'X + \alpha I)W = X'Y = U'Y'Y$, allowing one to conclude that

$$W = (X'X + \alpha I)^{-1} U'Y'Y \tag{7}$$

Extending reverse prediction to kernels on the input space is also particularly easy, since the reverse solutions always have the form $U = BX$ (where above we had $B = Y^\dagger$). Thus the kernelized training problem corresponding to (5) is given by

$$\min_B \ \operatorname{tr}\left((I - YB)K(I - YB)'\right) \tag{8}$$

where $K$ corresponds to $XX'$ in some feature representation. It is easy to verify that the global minimizer is given by $B = Y^\dagger$. Interestingly, the forward solution can again be recovered from the reverse solution. In particular, using the identity arising from the solutions of (3) and (8), $(K + \alpha I)A = Y = B'Y'Y$, we get

$$A = (K + \alpha I)^{-1} B'Y'Y \tag{9}$$

Finally, as in the forward case, we will need to make use of instance weighting. Given the diagonal weighting matrix $\Lambda$ the weighted problem is

$$\min_B \ \operatorname{tr}\left(\Lambda(I - YB)K(I - YB)'\right) \tag{10}$$

---

[2]If $X$ is not rank $n$, we can drop dependent columns.

where $B = (Y'\Lambda Y)^{-1}Y'\Lambda$ is the global minimizer. The forward solution can be recovered by

$$A = (\Lambda K + \alpha I)^{-1} B'Y'\Lambda Y \tag{11}$$

Therefore, for all the major variants of supervised least squares, one can solve the reverse problem and use the result to recover a solution to the forward problem.

# 4. Unsupervised Learning

Given the relation between forward and reverse prediction, we can now unify classical supervised with unsupervised learning principles. The key connection between supervised and unsupervised is a simple principle of optimism: if the training targets $Y$ are not given, optimize over guessed labels to achieve a best possible reconstruction of the input data. That is

$$\min_Z \min_U \ \operatorname{tr}\left((X - ZU)(X - ZU)'\right) \tag{12}$$

Before outlining specific unifications below, we first observe that the corresponding formulation of optimal forward prediction does not work. In fact, forward prediction fails to preserve useful structure in the unsupervised learning problem. To see why, note that optimistic forward training with missing labels is

$$0 = \min_Z \min_W \ \operatorname{tr}\left((XW - Z)(XW - Z)'\right) \tag{13}$$

Unfortunately, as long as $\operatorname{rank}(X) > k$, which we assume, this problem gives vacuous results. For *any* set of model parameters $W$ one can merely set the guessed labels to achieve $Z = XW$ and the prediction error is reduced to zero. Thus, the standard forward view has no ability to distinguish between alternative parameter matrices $W$ under optimistic guessing. However, given $\operatorname{rank}(X) > k$, optimal reverse prediction is **not** vacuous. In fact, it leads to several interesting results.

**Proposition 1** *Unconstrained reverse prediction*

$$\min_Z \min_U \ tr\left((X - ZU)(X - ZU)'\right) \tag{14}$$

*is equivalent to principal components analysis.*

**Proof:** Note that (14) is equivalent to

$$\min_Z \ \operatorname{tr}\left((I - ZZ^\dagger)XX'(I - ZZ^\dagger)'\right) \tag{15}$$

$$= \min_Z \ \operatorname{tr}\left((I - ZZ^\dagger)XX'\right) \tag{16}$$

The first equivalence follows from $U = Z^\dagger X$ by the solution of (5), and the second from the fact that $ZZ^\dagger$ is symmetric and $I - 2ZZ^\dagger + ZZ^\dagger ZZ^\dagger = I - ZZ^\dagger$. Clearly, (16) has the same optimizer as

$$\max_Z \ \operatorname{tr}\left(ZZ^\dagger XX'\right) = \max_{Z:Z'Z=I} \ \operatorname{tr}\left(ZZ'XX'\right) \tag{17}$$

The latter equality can be shown from the singular value decomposition of $Z$: if $Z = V\Sigma Q'$ for $V'V = I$, $Q'Q = I$ and $\Sigma$ diagonal, then $ZZ^\dagger = VV'$. Thus the solution is given by the top $k$ eigenvectors of $XX'$. ∎

This same observation has been made (surprisingly recently) in the statistics literature (Jong & Kotz, 1999), but can be extended.

**Corollary 1** *Kernelized reverse prediction*

$$\min_Z \min_B \; tr((I - ZB)K(I - ZB)') \qquad (18)$$

*is equivalent to kernel principal components analysis.*

Thus, least squares regression and principal components analysis can both be expressed by (14), where $Z$ is set to $Y$ if the labels are known.

A similar unification can be achieved for least squares classification. In classification, recall that the rows in the target label matrix, $Y$, indicate the class label of the corresponding instance. If the target labels are missing, we would like to guess a label matrix $Z$ that satisfies the same constraints, namely that $Z \in \{0,1\}^{t \times k}$ and $Z\mathbf{1} = \mathbf{1}$. Simply adding these constraints to (14) gives another interesting result.

**Proposition 2** *Constrained reverse prediction*

$$\min_{Z:Z\in\{0,1\}^{t\times k},\, Z1=1} \min_U \; tr((X - ZU)(X - ZU)') \quad (19)$$

*is equivalent to k-means clustering.*

**Proof:** Consider the equivalent objective (15) and notice that it is a sum of squares of the difference $(I - ZZ^\dagger)X = X - ZZ^\dagger X = X - Z(Z'Z)^{-1}Z'X$. To interpret this difference matrix, we exploit some observations from (Peng & Wei, 2007). First note that $Z'X$ is a $k \times n$ matrix where row $i$ is the sum of rows in $X$ that have class $i$ in $Z$ (that is, the sum of rows in $X$ for which the $i$th entry in the corresponding row of $Z$ is 1). Next, notice that $Z'Z$ is a diagonal matrix that contains the count of ones in each column of $Z$. Hence $(Z'Z)^{-1}$ is a diagonal matrix of reciprocals of column counts. Combining these facts shows that $U = (Z'Z)^{-1}Z'X$ is a $k \times n$ matrix whose row $i$ is the *mean* of the rows in $X$ that correspond to class $i$ in $Z$. Finally, note that $ZU = Z(Z'Z)^{-1}Z'X$ is a $t \times n$ matrix where row $i$ contains the mean corresponding to class $i$ in $Z$. Therefore, $X - Z(Z'Z)^{-1}Z'X$ is a $t \times n$ matrix containing rows from $X$ with the mean row for each corresponding class subtracted. The problem (19) can now be seen to be equivalent to assigning $k$ centers, encoded by $U = (Z'Z)^{-1}Z'X$, and assigning each row in $X$ to a center, encoded by $Z$, so as to minimize the sum of the squared distances between each row and its assigned center. ∎

**Corollary 2** *Constrained kernelized prediction*

$$\min_{Z:Z\in\{0,1\}^{t\times k},\, Z1=1} \min_B \; tr((I - ZB)K(I - ZB)') \quad (20)$$

*is equivalent to kernel k-means.*

This striking similarity between PCA and k-means clustering has been previously observed (Ding & He, 2004), but here we have shown that both use identical objectives to supervised (reverse) least squares.

Interestingly, even normalized graph-cut can be unified in a similar manner. Here we only need to introduce a weighting on the training instances. To show this connection, we first establish a preliminary result.

**Proposition 3** *For a nonnegative matrix $K$ and weighting $\Lambda = diag(K\mathbf{1})$, weighted reverse prediction*

$$\min_{Z:Z\in\{0,1\}^{t\times k},Z1=1} \min_B tr\left(\Lambda(\Lambda^{-1} - ZB)K(\Lambda^{-1} - ZB)'\right) \;(21)$$

*is equivalent to normalized graph-cut.*

**Proof:** For any $Z$, the inner minimization can be solved to obtain $B = (Z'\Lambda Z)^{-1}Z'$. Substituting back into the objective and reducing yields

$$\min_{Z:Z\in\{0,1\}^{t\times k},\, Z1=1} tr\left((\Lambda^{-1} - Z(Z'\Lambda Z)^{-1}Z')K\right) \quad (22)$$

The first term is constant and can be altered without affecting the minimizer, hence (22) is equivalent to

$$\min_{Z:Z\in\{0,1\}^{t\times k},\, Z1=1} tr\left(I\right) - tr\left(Z(Z'\Lambda Z)^{-1}Z'K\right) \quad (23)$$

$$= \min_{Z:Z\in\{0,1\}^{t\times k},\, Z1=1} tr\left((Z'\Lambda Z)^{-1}Z'(\Lambda - K)Z\right) \quad (24)$$

Xing & Jordan (2003) have shown that with $\Lambda = diag(K\mathbf{1})$, $\Lambda - K$ is the Laplacian and (24) is equivalent to normalized cut. ∎

From this result, we can now relate normalized graph-cut to the same reverse least squares formulation.

**Corollary 3** *The weighted least squares problem*

$$\min_{Z\in\{0,1\}^{t\times k},\, Z1=1} \min_U tr\left(\Lambda(\Lambda^{-1}X - ZU)(\Lambda^{-1}X - ZU)'\right) \;(25)$$

*is equivalent to norm-cut (21) on $K = XX'$ if $K \geq 0$.*

As far as we know, this connection has not been previously realized. These results simplify some of the connections observed in (Dhillon et al., 2004; Chen & Peng, 2008; Kulis et al., 2009) relating k-means to normalized graph-cut, but generalizes them to relate to supervised least squares. This generalized connection is crucial to obtaining simple and principled approaches to semi-supervised training.

## 5. Semi-supervised Learning

We have shown how the perspective of reverse prediction unifies classical supervised and unsupervised training principles. Both can be viewed as minimizing an identical least squared reconstruction cost, differing only in imposing constraints on the labels (and altering the instance weights in the case of normalized graph-cut). This connection between supervised and unsupervised learning provides several obvious and yet apparently new strategies for semi-supervised learning.

The first and simplest strategy we explore is based on the following objective

$$\min_{Z} \min_{U} \operatorname{tr}\left((X_L - Y_L U)(X_L - Y_L U)'\right)/t_L \quad (26)$$
$$+\mu \operatorname{tr}\left((X_U - ZU)(X_U - ZU)'\right)/t_U$$

Here $(X_L, Y_L)$ and $X_U$ denote the labeled and unlabeled data, $t_L$ and $t_U$ denote the respective number of examples, and the parameter $\mu$ trades off between the two losses (see below). The strategy proceeds by first solving for the optimal reverse model $U$ in (26), and then recovering the corresponding forward model. This basic procedure can be adapted to any of the reverse prediction models we have presented, including regularized, kernelized, and instance weighted versions of both regression and classification.

Although this particular strategy for combining supervised and unsupervised training is straightforward (and we show how to improve it below), it already exhibits some advantages over existing methods. For example, by using the normalized graph-cut formulation (25) one obtains a semi-supervised learning algorithm that is very similar to state of the art approaches (Zhou & Schölkopf, 2006; Belkin et al., 2006; Kulis et al., 2009). However, none of these previous approaches use the *reverse* loss on the supervised component. Instead, they couple the reverse loss on the unsupervised component with a *forward* prediction loss on the labeled data. Given the previous discussion, such a combination seems ad hoc. Below we show that by using reverse losses on both the supervised and unsupervised components beneficial results can be achieved. First, however, we can go deeper and derive a principled combination of supervised and unsupervised learning that achieves a strict variance reduction.

## 6. Reverse Loss Decomposition

A principled approach to semi-supervised training can be based on a decomposition of the reverse least squares loss that can be arrived at both geometrically and algebraically. The decomposition can be understood by considering Figures 1 and 2 respec-
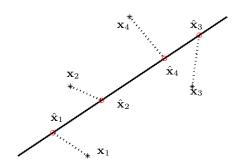


*Figure 1.* Supervised reverse training attempts to find a linear subspace (determined by the model) in the original data space so that reconstruction of the training examples $\mathbf{x}_1, ..., \mathbf{x}_4$ from the targets $\mathbf{y}_1, ..., \mathbf{y}_4$ minimizes the sum of squared reconstruction errors.
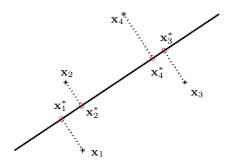


*Figure 2.* Unsupervised reverse training is the same as supervised, except that for any given linear model, the targets $\mathbf{z}_1, ..., \mathbf{z}_4$ are adjusted to ensure a least squares solution (given by the orthogonal projection of $\mathbf{x}_1, ..., \mathbf{x}_4$ onto the linear subspace). This leads to a Pythagorean decomposition of squared supervised error into squared unsupervised error plus the squared distance between the supervised $\hat{\mathbf{x}}_i$ and unsupervised $\mathbf{x}^*$ reconstructions.

tively. Assume we have a fixed model $U$ and a given set of data $X$. Let $\hat{X} = YU$ denote the supervised reconstruction of $X$ using given labels $Y$; see Figure 1. On the other hand, in the unsupervised case the optimal target labels $Z^*$ are chosen to minimize the sum of squared reconstruction error of the data $X$ under reverse model $U$. Let $X^* = Z^*U$ denote the optimistic unsupervised reconstruction, where $Z^* = \arg\min_Z \operatorname{tr}\left((X - ZU)(X - ZU)'\right)$; see Figure 2.

Now observe that $\hat{X}$ and $X^*$ must be members of the same linear subspace determined by $U$. In the unsupervised case, each data point is projected to its nearest point in the subspace (Figure 2). In the supervised case, each data point is reconstructed by a point in the subspace determined by its corresponding label in $Y$ (Figure 1). To minimize the sum of squared loss, the goal of both optimizations is to find a model $U$ that

maps reconstructions to a linear subspace that minimizes the squared lengths of the dotted reconstruction lines in the figures—supervised being constrained by the given labels and unsupervised free to project.

These two figures reveal the key relationship between the supervised and unsupervised losses: by the Pythagorean Theorem the squared supervised error equals the squared unsupervised error plus the squared distance between the supervised and unsupervised reconstructions in the linear subspace. That is, consider data point $\mathbf{x}_3$ in Figures 1 and 2. For this point, $\|\mathbf{x}_3 - \hat{\mathbf{x}}_3\|^2$ in Figure 1 equals $\|\mathbf{x}_3 - \mathbf{x}_3^*\|^2$ in Figure 2 plus $\|\hat{\mathbf{x}}_3 - \mathbf{x}_3^*\|^2$ from Figures 1 and 2 respectively. Algebraically, we have

**Proposition 4** *For any $X$, $Y$, and $U$*

$$tr((X - YU)(X - YU)) \qquad (27)$$
$$= tr((X - Z^*U)(X - Z^*U)) \qquad (28)$$
$$+ tr((Z^*U - YU)(Z^*U - YU)) \qquad (29)$$

*where $Z^* = \arg\min_Z tr((X - ZU)(X - ZU))$.*

This additive decomposition provides one of the key insights of this paper. It shows that the reverse prediction loss of any model $U$ decomposes into a sum of two losses, where one loss (28) depends only on the input data $X$, and the other loss (29) depends on both the target labels $Y$ and the inputs $X$ (through the dependence of $Z^*$ on $X$). This allows us to use auxiliary data to obtain an unbiased estimate of (27) that strictly reduces its variance.

**Corollary 4** *For any $U$*

$$E\left[tr((X_L - Y_LU)(X_L - Y_LU))/t_L\right] \qquad (30)$$
$$= E\left[tr((X - Z^*U)(X - Z^*U))/t_S\right] \qquad (31)$$
$$+ E\left[tr((Z_L^*U - Y_LU)(Z_L^*U - Y_LU)/t_L)\right] \quad (32)$$

*where $Z^* = \arg\min_Z tr((X - ZU)(X - ZU))$.*

Here, $X$ and $Z^*$ are defined over the union of labeled and unlabeled data, $Y_L$ is the supervised label matrix, $Z_L^*$ is the corresponding component of $Z^*$ on the labeled portion, and $t_L$ and $t_S$ are the number of labeled and labeled plus unlabeled examples respectively. Since the middle term is unbiased but based on a larger sample, it reduces the variance of the total (supervised) loss estimate for a given model $U$.

For large amounts of unlabeled data, the naive semi-supervised approach (26) closely approximates the unbiased principle (30), but introduces a bias due to double counting the unlabeled loss (31). One advantage of (26) over the approach based on (30), though, is that it admits more straightforward optimization procedures.

## 7. Regression Experiments

We implemented variants of semi-supervised regression based on (26). Although the supervised and unsupervised terms in the loss can be efficiently optimized in isolation, it is not clear how they can be efficiently minimized jointly. Therefore, we first solve the supervised training problem to obtain an initial $U$, then alternate between optimizing $Z$ and $U$ in the semi-supervised objective to reach a local solution. The forward model, $W$, can then be recovered from $U$.

To evaluate this approach, we compared against standard supervised learning methods and the transductive regression algorithm of (Cortes & Mohri, 2006)—which has been reported to outperform earlier transductive regression algorithms (Chapelle et al., 1999; Belkin et al., 2006). The latter algorithm has two steps: (1) locally estimate the unlabeled data by using the labeled points within $r$ of the unlabeled point; and (2) find a solution that best fits the labeled data and data estimated in Step 1. We used kernel ridge regression to estimate the unlabeled targets in Step 1, as suggested in (Cortes & Mohri, 2006).

Experiments were run on two datasets from (Cortes & Mohri, 2006), `kin-32fh` and `Cal-housing`, plus an additional dataset, `Pumadyn-32h`.[3] Performance was evaluated based on the average of 10 random splits of the data. For the kernel-based methods we used a Gaussian kernel with width parameter set to 1. For the transductive regression algorithm, we additionally selected the distance $r$ in Step 1 to include 2.5% of the labeled data, as in (Cortes & Mohri, 2006), and set the remaining parameters with 10-fold cross validation. For the semi-supervised approach based on (26), we set $\mu = 1$ and selected the regularization parameter $\alpha$ by 10-fold cross validation.

Table 1 summarizes the performance of three variants of the semi-supervised algorithm, compared to the supervised techniques and the transductive regression algorithm. Here we see that the straightforward semi-supervised approach we propose, with kernels and regularization, obtains competitive performance in each case. (On `Pumadyn-32h` the three kernel-based algorithms obtain the same error, likely due to the fact that all three found the same local minima.)

## 8. Classification Experiments

In addition to regression, we also considered classification problems and investigated the performance of semi-supervised learning based on (26) using kernel-

---

[3] http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html

*Table 1.* Forward error rates (average root mean squared error, ± standard deviations) for different regression algorithms on various data sets. The values of $(k, n; t_L, t_U)$ are indicated for each data set.

| | kin-32fh $(1, 32; 10, 1000)$ | Pumadyn-32h $(1, 32; 30, 3000)$ | Cal-housing $(1, 8; 10, 500)$ |
|---|---|---|---|
| SUP. LEASTSQUARES | $18.150 \pm 23.30$ | $0.577 \pm 0.190$ | $12.520 \pm 3.755$ |
| SUP. REGULARIZED | $0.408 \pm 0.039$ | $0.109 \pm 0.060$ | $2.083 \pm 2.210$ |
| SUP. REGKERNEL | $1.070 \pm 0.038$ | $0.030 \pm 0.001$ | $2.504 \pm 0.615$ |
| TRANSDUCTIVEKERNEL (CORTES & MOHRI, 2006) | $1.350 \pm 0.010$ | $0.030 \pm 0.001$ | $2.423 \pm 0.727$ |
| SEMI. LEASTSQUARES | $0.365 \pm 0.067$ | $0.043 \pm 0.014$ | $12.210 \pm 3.660$ |
| SEMI. REGULARIZED | $0.278 \pm 0.039$ | $0.050 \pm 0.060$ | $6.249 \pm 3.660$ |
| SEMI. REGKERNEL | $0.554 \pm 0.069$ | $0.030 \pm 0.001$ | $1.492 \pm 0.448$ |

ized k-means and normalized graph-cut. We compared the proposed algorithms to spectral graph based transduction (SGT) (Joachims, 2003), and the semi-supervised extensions of regularized least squares and support vector machines utilizing Laplacian regularization (LapRLS and LapSVM respectively) (Sindhwani et al., 2005). To enable these competitors, our experiments were conducted in a transductive setting; that is, given a partially labeled data set, we measured test error on the unlabeled data points. (Note, however, that the algorithms proposed in this paper are not limited to the transductive setting.)

Experiments were run on four well-investigated data sets from the semi-supervised learning literature: `g50c` is an artificial data set generated from two Gaussians in 50-dimensional space (known to have a Bayes misclassification error rate of 5%); `MNIST-069` is a sample of three digits (0, 6, and 9) from the MNIST digit data set; `mac-mswindows` is a binary classification task that involves two classes, `mac` and `mswindows`, from the 20Newsgroup data set respectively; and `faculty-course` is taken from the WebKB data set, comprising of the categories `faculty` and `course`.

Performance was evaluated based on the average of 10 random splits of the data. We used a Gaussian kernel and set the width by 10-fold cross validation for each algorithm. For the semi-supervised approaches based on (26), we set $\mu = 10$ and $\alpha = 0$. The remaining parameters for the competing methods, SGT, LapRLS and LapSVM, were set optimistically on the test set.

From Table 2 one can see that the semi-supervised k-means and normalized cut algorithms perform very well on the four data sets, competing with and sometimes surpassing the current state of the art. On data set `g50c`, the prediction errors of the proposed algorithms approach the Bayes optimal error. One can also observe that semi-supervised normalized cut is more stable and accurate than semi-supervised k-means, possibly due to the weighting effect.

## 9. Conclusion

The principled approach to semi-supervised learning derived in this paper depended heavily on using least squares. The main enabling property of least squares is that we were able to exactly recover forward from reverse solutions and obtain an additive decomposition of the reverse loss into supervised and unsupervised components. Changing the loss function, unfortunately, blocks both aspects. Nevertheless, the least squares decompositions can provide guidance for developing heuristic algorithms for alternative losses. Current self-supervised learning approaches, such as those based on hinge loss (De Bie & Cristianini, 2003; Xu et al., 2004), can be seen as a rough version of this heuristic. It remains to determine whether the observations of this paper can be used to further improve such methods.

## References

Balcan, M.-F., & Blum, A. (2005). A PAC-style model for learning from labeled and unlabeled data. *Conf. Comput. Learn. Theory (COLT)* (pp. 111–126).

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

Ben-David, S., Lu, T., & Pál, D. (2008). Does unlabeled data provably help? *Conf. Comput. Learn. Theory (COLT)* (pp. 33–44).

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *Conf. Comput. Learn. Theory (COLT)* (pp. 92–100).

Chapelle, O., Vapnik, V., & Weston, J. (1999). Transductive inference for estimating values of functions. *Adv. Neural Info. Proc. 12 (NIPS)* (pp. 421–427).

Table 2. Forward error rates (average misclassification error in percentages, $\pm$ standard deviations) for different classification algorithms on various data sets. The values of $(k, n; t_L, t_U)$ are indicated for each data set.

| | g50C (2, 50; 50, 500) | MNIST-069 (3, 784; 15, 900) | mac-mswindows (2, 7511; 50, 1896) | faculty-course (2, 40195; 22, 2031) |
|---|---|---|---|---|
| SUP. KMEANS | $8.20 \pm 1.14$ | $10.01 \pm 3.99$ | $18.54 \pm 5.54$ | $19.85 \pm 10.21$ |
| SUP. NCUT | $8.72 \pm 1.44$ | $8.94 \pm 3.13$ | $15.04 \pm 1.84$ | $16.97 \pm 9.16$ |
| SGT (JOACHIMS, 2003) | $7.02 \pm 1.38$ | $5.02 \pm 0.72$ | $8.69 \pm 1.62$ | $6.62 \pm 1.80$ |
| LAPSVM (SINDHWANI ET AL., 2005) | $5.44 \pm 0.61$ | $4.91 \pm 1.13$ | $10.27 \pm 1.19$ | $27.36 \pm 5.72$ |
| LAPRLS (SINDHWANI ET AL., 2005) | $5.18 \pm 0.71$ | $4.12 \pm 2.70$ | $9.94 \pm 1.30$ | $19.30 \pm 2.58$ |
| SEMI. KMEANS | $5.28 \pm 1.14$ | $5.32 \pm 1.11$ | $7.39 \pm 1.70$ | $22.22 \pm 19.33$ |
| SEMI. NCUT | $5.18 \pm 0.68$ | $4.89 \pm 0.72$ | $6.23 \pm 0.90$ | $5.45 \pm 0.92$ |

Chen, H.-R., & Peng, J. (2008). 0-1 semidefinite programming for graph-cut clustering. In *CRM Proceedings and Lecture Notes of the Amer. Math. Soc.*

Corduneanu, A., & Jaakkola., T. (2006). Data dependent regularization. In O. Chapelle, B. Scholköpf and A. Zien (Eds.), *Semi-supervised learning*, 163–182. MIT Press.

Cortes, C., & Mohri, M. (2006). On transductive regression. *Adv. Neural Info. Proc. Sys. 19 (NIPS)* (pp. 305–312).

De Bie, T., & Cristianini, N. (2003). Convex methods for transduction. *Adv. Neural Info. Proc. Sys. 16 (NIPS)* (pp. 73–80).

Dhillon, I. S., Guan, Y., & Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts. *Know. Disc. Data Mining (KDD)* (pp. 551–556).

Ding, C., & He, X. (2004). K-means clustering via principal component analysis. *Inter. Conf. Mach. Learn. (ICML)* (pp. 225–232).

Joachims, T. (1999). Transductive inference for text classification using support vector machines. *Proceed. Inter. Conf. on Machine Learning (ICML)*.

Joachims, T. (2003). Transductive learning via spectral graph partitioning. *Inter. Conf. Mach. Learn. (ICML)* (pp. 290–297).

Jong, J.-C., & Kotz, S. (1999). On a relation between principal components and regression analysis. *The American Statistician*, *53*, 349–351.

Katayama, T. (2005). *Subspace methods for system identification.* Springer.

Klein, D. (2004). *The unsupervised learning of natural language structure.* Doctoral dissertation, Stanford.

Kulis, B., Basu, S., & Dhillon, I. (2009). Semi-supervised graph clustering: A kernel approach. *Machine Learning*, *74*, 1–22.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Berkeley Symp. on Math. Stats. and Prob* (pp. 281–297).

Mika, S., Schölkopf, B., Smola, A., Müller, K.-R., Scholz, M., & Rätsch G. (1998). Kernel PCA and de-noising in feature spaces. *Adv. Neural Info. Proc. Sys. 11 (NIPS)* (pp. 536–542).

Peng, J., & Wei, Y. (2007). Approximating k-means-type clustering via semidefinite programming. *SIAM Journal on Optimization*, 186 – 205.

Schölkopf, B., & Smola, A. (2002). *Learning with kernels.* MIT Press.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. and Mach. Intell.*, *22*, 888–905.

Sindhwani, V., Niyogi, P., & Belkin, M. (2005). Beyond the point cloud: from transductive to semi-supervised learning. *Inter. Conf. Mach. Learn. (ICML)* (pp. 824–831).

Smith, N., & Eisner, J. (2005). Contrastive estimation: Training log-linear models on unlabeled data. *Conf. Assoc. Comput. Ling. (ACL)* (pp. 354–362).

Xing, E., & Jordan, M. (2003). On semidefinite relaxation for normalized k-cut and connections to spectral clustering. TR CSD-03-1265, Berkeley.

Xu, L., Neufeld, J., Larson, B., & Schuurmans, D. (2004). Maximum margin clustering. *Adv. Neural Info. Proc. Sys. 17 (NIPS)* (pp. 1537–1544).

Zhou, D., & Schölkopf, B. (2006). Discrete regularization. In O. Chapelle, B. Scholköpf and A. Zien (Eds.), *Semi-supervised learning*, 221–232. MIT Press.

Zhu, X. (2005). Semi-supervised learning literature survey. TR 1530, U. Wisconsin, CS Dept.