The University of Kansas

Technical Report

Model Selection for Semi-Supervised Learning with Limited Labeled Data

Brian Quanz and Jun Huan

ITTC-FY2013-TR-65071-01

July 2012

Project Sponsor:
National Science Foundation

Model Selection for Semi-Supervised Learning with Limited Labeled Data

Brian Quanz and Jun Huan*

July 30th, 2012

Abstract

An important component for making semi-supervised learning applicable to real world data is the task of model selection. For the case of very limited labeled data, for which semi-supervised learning algorithms have the greatest potential to offer improvement in estimating predictive models, model selection is a significant challenge, a key open problem, and often avoided entirely in previous work. While previous work has demonstrated the benefit of semi-supervised learning in cases of very limited labeled training data, in order for such results to be achievable in practice, some effective method of selecting the hyper-parameters for these methods is necessary. In general, existing approaches rely heavily in some way on the labeled data directly for estimating either performance (e.g., error), some key characteristics of the model, or likelihood, and so can suffer when there is not much labeled data. Instead we propose an alternative, sampling approach in order to estimate model performance. The main idea is to evaluate the models on a large number of generated similar data sets, and to prefer those models that perform well on average across the data sets. New training and unlabeled/test data are generated by sampling from the large amount of unlabeled data and estimated conditional probabilities for the labels. Since these data sets are complete with labels, models can then be evaluated using the generated labels for the much larger set of unlabeled/test data. Using a variety of data sets we demonstrate the effectiveness of our approach, and, for small amounts of labeled data, large improvement over traditional methods like cross-validation, as well as better performance on average than the state-of-the-art for semi-supervised model selection.

1 Introduction

For every semi-supervised learning algorithm, in practice it is necessary to select the specific tuning or hyper parameters for the method, using the available training data. This process is called model selection. Model selection is a major issue for semi-supervised learning problems involving very limited labeled data, since the small amount of labeled data makes it difficult to reliably estimate predictive performance of a model.

However, in the work on semi-supervised learning the issue of model selection is often avoided [4, 6, 3, 22, 27, 21, 23, 7, 17, 1, 39, 11, 34, 19], for example, by reporting the best results found over a grid of hyper-parameters, the idea being that this is the best performance a particular method could achieve if there were some way to select that best model. However this best performance is meaningless for real wold applications if there is not some way to select the model. In general, existing model selection approaches rely heavily in some way on the labeled data directly for estimating either performance (e.g., error), some key characteristics of the model, or likelihood, and so can suffer when there is not much labeled data. For the case of extremely limited training data, the performance of common, general approaches to model selection like cross-validation deteriorates, and in addition many semi-supervised model selection methods are only designed to work for specific semi-supervised learning methods and so are not generally applicable. A recent survey article lists model selection for semi-supervised learning as one of five open problems in model selection: "Very little has been done for model selection in semi-supervised learning problems, in which only some training instances come with target values. Semi-supervised tasks can be challenging for traditional model selection methods, such as cross-validation, because the number of labeled data is often very small" [13].

This analysis leads us to propose an alternative, general approach to model selection for semi-supervised learning with extremely limited labeled data. Like cross-validation the approach is based on re-sampling and re-training, and also like cross-validation is already parallelized and thus can be efficiently carried out with modern parallel computing resources. The basic idea is to generate many data sets that are similar to the target one, by re-sampling the labeled and unlabeled data from the given data. In each case labels for the data are sampled using estimated conditional distributions derived by averaging the predictions on each data instance of all models in the set of models under consideration. In this way, if most models agree on a prediction for a label, then that label will consistently be generated, but if models largely disagree on a

*Information and Telecommunication Technology Center, Department of Electrical Engineering and Computer Science, University of Kansas, {bquanz,jhuan}@ittc.ku.edu

label, then that label will vary more across the generated data sets. Additionally the prior weights given to the models can be iteratively updated, with the goal of making the generation distribution more similar to the target data distribution. By estimating the average performance for each model across the generated similar data sets, this provides a rough estimate of its performance on the target data. This approach can also be seen as an alternative way of estimating the stability of a model by evaluating its performance on many different but similar data sets. If the model does not provide stable estimates, then its performance may vary greatly for slight changes in the data set, and this will be captured by a larger averaged test error over the generated data sets.

We evaluate our similar data sampling approach on four data sets with different amounts of labeled data, and, for the case of data with few labels, demonstrate significant improvement over existing model selection approaches, including a state-of-the-art semi-supervised model selection method, discussed in the next section. Our experimental results demonstrate the efficacy of the proposed approach.

2 Related Work

One of the most commonly used approaches to model selection is cross-validation. In k-fold cross-validation, the labeled training data is partitioned into k roughly equal sized sets. Then each of the k sets takes a turn as being the held-out set used for testing, and the remaining $k-1$ sets are used for training each model to be evaluated. The average performance on the held-out sets is used to estimate the models' performance, and this can be repeated and averaged over multiple random partitions. Cross-validation is one model selection approach commonly used for general semi-supervised learning methods [8, 38], however it has been found that its performance can suffer when only small amounts of labeled data are available [32]. Aside from standard supervised model selection methods like cross-validation, we can roughly break related work into two categories: work that avoids full model selection in some way, and work either focused on the problem of semi-supervised model selection or that uses some form of semi-supervised model selection.

2.1 Avoiding the Model Selection Issue

A large amount of the work on semi-supervised leading in the literature avoids the model selection issue in some way. Therefore we briefly mention some common approaches used that essentially avoid model selection, before discussing specific model selection approaches. We can further break this category down into two sub-categories.

2.1.1 Reporting the Performance for Fixed Values or Best Over Hyper-parameter Grids

The work in this category trains the methods used either by arbitrarily picking fixed values for some or all hyper-parameters or by using default or heuristic hyper-parameter values or by training the methods over hyper-parameter grids, sets of different hyper-parameter combinations, and returns the best results on the test error found, sometimes with hyper-parameter sensitivity results as well [4, 6, 3, 22, 27, 21, 23, 7, 17, 1, 39, 11, 34, 19].

2.1.2 Selecting using a validation set typically only available for model selection

The work in this category uses a separate validation set for model selection and selects the best model according to performance on the validation set, e.g., [35, 26, 9, 10]. Note that this is also an artificial scenario, since if extra labeled data were available for model selection it could also be used for model estimation, likely making the semi-supervised learning approaches unnecessary or at least reducing the benefit they offer and most likely changing the best model as well. For example, in one work [35], for one data set, the semi-supervised learning method has access to only 2 labeled instances, but 250 are used for validation - if these had been available for training after validation, supervised learning would most likely have been sufficient.

2.2 Model Selection Approaches

Various approaches do exist that address the model selection issue partially or fully for the case of semi-supervised data. However most are method-dependent - specific to the probabilistic models and frameworks proposed for the particular learning algorithm. Here we discuss such approaches as well as general semi-supervised approaches.

2.2.1 Approaches that are restricted to certain model classes

One common category of methods is the approach of estimating the marginal likelihood also called maximum likelihood type II approaches [36] or evidence-based model selection [32]. Given specific probabilistic models, the model parameters are approximately integrated out of the data likelihood equation leaving the marginal likelihood as a function of the hyper-parameters. The hyper-parameters maximizing this marginal likelihood are then typically chosen. However this requires assuming a particular probabilistic model for the different components of the model and the data, and is thus not applicable to general semi-supervised learning methods,

for instance co-training with arbitrary base classifiers in each view. Additionally, depending on the model, this approach could suffer from over-fitting with limited labeled data. A different type of marginalization strategy in which some of the hyper-parameters are marginalized when estimating the model parameters has also been proposed [17]. In this case, for hyper-parameter selection, some hyper-parameters are arbitrarily fixed, and the remaining hyper-parameters are treated as missing values. The conditional probability distributions defined by the model are then used with the expectation-maximization algorithm to fit the model parameters, essentially integrating out these specific hyper-parameters. Similar to the marginal likelihood approaches with gaussian processes, Zhu, Ghahramani and Lafferty proposed a Gaussian random field model with a label entropy model selection approach used for learning some hyper-parameters [41].

Another approach is to use information criteria. For instance, Culp, Michailidis and Johnson propose a generalized additive model with transductive smoothers for multi-view semi-supervised learning [12]. The associated proposed model scoring uses the likelihood or error penalty on the labeled data in combination with an estimate of degrees of freedom for the linear smoothers, which corresponds to the trace of a smoother matrix. In addition to being method-dependent, this approach only considers the performance on the labeled data with the unlabeled data effecting only the trace of the smoother matrix. For very limited labeled training data, this could result in poor solutions since many models could fit the labeled data very well so that the estimated degrees of freedom is the determining factor, potentially resulting in over-fitting for cases of many hyper-parameters.

2.2.2 General Approaches Several general approaches also exist for semi-supervised model selection. An interesting state-of-the-art approach for semi-supervised model selection is metric-based model selection [28, 29, 30], which was generally found to out-perform previous model selection methods including cross-validation and various information criteria. The first approach in this category uses estimated distances between hypotheses in different classes and the target hypothesis and tests a sequence of hypothesis classes in order until the triangle inequality is violated with some previous hypothesis class. Since the sequence traversal can be terminated early at a sub-optimal model, a second approach with an adjusted distance estimate was proposed using ratios of function distance estimates to score models. Bengio and Chapados consider metric-based model selection extensions to time series data, cases without unlabeled data, and a hybrid with cross-validation [2]. However, a major limitation for the metric-based model selection occurs with extremely limited labeled training data since many or all hypothesis classes considered could all achieve perfect training error. This means if the first approach is used, the sequence traversal is terminated immediately, and if the second approach is used, all methods have equal scores of zero, so there is no way to decide between them. Additionally this method requires a nested ordering of hypothesis classes, limiting its applicability for general learning methods, since the correct sequence of hypothesis classes which should be monotonic in terms of complexity is not always clear, particularly with multiple hyper-parameters. Schuurmans *et al.*addressed this issue by proposing a new model evaluator, called ADA, as the product of the training error and a function of the ratios of the distance between a learned function on the labeled and unlabeled data from a constant function, using Kullback-Leibler divergence for classification [30] since the original distance approach did not work well for classification. Collectively these metric-based model selection approaches were demonstrated to improve over the state-of-the-art in model selection, compared against a wide variety of model selection approaches. Like the proposed method of this thesis, this method can also be applied to a grid of hyper-parameters for model selection. However, if class conditional probabilities for any instances are zero the approach has a divide-by-zero problem, which can happen for some tasks with very small amounts of labeled training data. Additionally, small amounts of labeled data may not provide reliable enough information for the estimated labeled data function distances, and many semi-supervised learning methods already generally enforce similarity in the learned function evaluated on labeled and unlabeled data in some way so this method may not be as useful with semi-supervised learning algorithms. Furthermore, to our knowledge, this method has never been analyzed in conjunction with semi-supervised learning algorithms, which is part of what is provided here.

Madani, Pennock, and Flake proposed a co-validation approach [20] in which two functions are trained on different partitions of the labeled training data, and their disagreement is measured on the unlabeled data and used along with training error to estimate test error. However this approach requires enough labeled data to allow representative functions to be learned with half the amount of training data, making it an unsuitable choice for very small amounts of labeled training data. Additionally in their semi-supervised learning experiments the approach did not improve the model selection over cross-validation (though could be helpful for active and transfer learning). A similar approach is proposed in [16], in which cross validation is extended with a disagreement measure on the unlabeled data; also similarly the approach did not improve over cross-validation, but did offer more reliable generalization error guarantees. Another similar approach was proposed called stability selection, and extended these ideas to unlabeled data for the problem of estimating the number of clusters to use in a clustering model [18].

3 Methodology

We assume a set of labeled and unlabeled data instances $\mathcal{D}$ are generated by a fixed joint distribution P_{XY} over $\mathcal{X} \times \mathcal{Y}$. We further make the standard assumptions that the data $\{X_i\}$ is i.i.d. and that the distribution of Y_i depends only on X_i. Since typically for semi-supervised learning, a large amount of unlabeled data is available, this means the marginal distribution P_X is well characterized by the data sample. Sampling from the marginal distribution P_X can therefore be simply accomplished by re-sampling from the full set of labeled and unlabeled data. The intuition behind the proposed approach is then that, given marginal samples $\vec{x}$ that are close to the true distribution, if we can at least approximately sample associated labels, then we can come up with a way to sample data sets that are similar to the target data set. By training different models on these synthetically generated data sets, we can get an idea of how consistently they perform on similar tasks to the target one by averaging their performance over many randomly generated similar tasks. Since we have the ground truth labels for these similar data sets, we can directly evaluate each model for them. Intuitively, if a model works well for these similar data sets then we would expect it to work well for the target data set as well. We hypothesize that considering model performance across these similar data sets can result in better estimation of model performance than relying on one particular data sample (the target data sample) with only a small amount of labeled data to perform model selection, since with the proposed approach real performance is evaluated on similar data sets for which the labels are known and test error can be directly computed.

3.1 Estimating Expected Test Error by Re-sampling The goal can therefore be defined as estimating the expected test error for each model using a sampling approach, and a particular loss function $L(,)$. Typically $L(,)$ is taken to be the 0-1 loss, given by $L(a,b) = 1$ if $a \neq b$, 0 o.w. Specifically, $\text{Err} = E[L(Y, \hat{f}(X))|\mathcal{D}]$ where $\hat{f}()$ is the predictive function estimator using $\mathcal{D}$ corresponding to a particular model (i.e., set of hyper-parameters), and the expectation is over both the training data $\mathcal{D}$ of a particular size and the random variable X. Each model corresponds to a distinct estimator which maps a data sample $\mathcal{D}$ to a function from $\hat{f} : \mathcal{X} \rightarrow \mathcal{Y}$ and so $\hat{f}$ is a random variable. We assume the goal is to evaluate a finite set of models $\mathcal{M}$ of size k, with some initial prior distribution over the models P_M which would usually be taken to be uniform.

Since the expected test error is just an expectation over different data samples, we can approximate it via the law of large numbers as follows

$$\text{Err}_m \approx \frac{1}{d}\sum_{j=1}^{d}\frac{1}{t}\sum_{i=1}^{t} L(y_{j,i}, f_{m,j}(\vec{x}_{j,i})) \tag{3.1}$$

Here each $\mathcal{D}_k$, for $k = 1, \ldots, d$, is obtained by independently sampling a data set with the same number of labeled and unlabeled instances as $\mathcal{D}$ and t test instances by sampling $(\vec{x}_{j,i}, y_{j,i})$ from P_{XY}, and $f_{m,j}()$ is the predictive function learned for the particular model (i.e., set of hyper-parameters) m for training set $\mathcal{D}_j$. Note these training sets contain both labeled and unlabeled data. In the transductive setting, the unlabeled data is also the test data, so in this case t is the number of unlabeled instances.

Since sampling from P_X can be approximated by re-sampling (in our implementation we use without replacement so that we can partition the data) from the full set of labeled and unlabeled data, if the conditional distribution $P_{Y|X}$ were known at least at each data instance in the training data, then we could also sample Y given an X sample and thus sample from P_{XY}. Furthermore since the amount of unlabeled data is large we can estimate the test error using the generated (sampled) unlabeled data. If we assume $P_{Y|X}$ corresponds to a mixture of models in $\mathcal{M}$, which are in the form of the models that return probabilistic outputs, then $P_{Y|X=\vec{x},\mathcal{D}} \propto \sum P_{Y|X=\vec{x},M=m,\mathcal{D}} P_{M=m}$. Note also that this mixture could correspond to a single model. Therefore if the probability of each model, P_M, were known, we could generate data sets very similar to the target data set.

Therefore we propose the following iterative procedure to estimate average test error for a set of models, where we view the probabilities of the models as hidden variables. The procedure starts with an initial P_M usually taken to be uniform (i.e., $P_{M=m} = 1/k$), and a target training data set $\mathcal{D}$ with n labeled instances.

1. **Step 1:** For each $m \in \mathcal{M}$ compute $P_{Y|X,m,\mathcal{D}}$ and $f_m()$ by training the model on the target training data set $\mathcal{D}$. Average these conditional estimates together according to the current estimate for P_M. In particular, we define:

$$\hat{P}_{Y=y|X=\vec{x}} = \sum_m P_{Y=y|X=\vec{x},M=m,\mathcal{D}} P_{M=m} \tag{3.2}$$

2. **Step 2:** For each of some number d data sets, randomly sample without replacement n instances from D to use as

labeled training data, and use the remainder as both unlabeled training data and test data. To each instance, assign a label by sampling from conditional distribution estimates found in the previous step, $\hat{P}_{Y|\vec{x}_i}$ for each i.

3. **Step 3:** Estimate the average test error for each model m according to Equation 3.1.

4. (*Optional*) **Step 4:** Taking the likelihood for a given model to be the exponential of the negative test error, multiply these by the current probability for each model P_M and normalize across all models for each data set. Average the result across data sets to obtain the new estimates for the hidden variables P_M.

5. (*Optional*) **Step 5:** Repeat for several iterations, or until convergence.

If we stop after one iteration, then the average estimated test error is computed using a conditional distribution with equal weight for each model, i.e., the uniformly-weighted average, which might be preferable, in particular for the sake of computational efficiency. In practice we found this approach to be effective. Also note, for continuous Y, densities are used for its distribution.

3.2 Addressing Additional Issues One issue with computing the conditional distributions is that, even if all of the models agree in their label prediction, depending on the method used, the probability outputs might still be close to 0.5. In this case, the sampled data could still vary largely, with samples not too similar to the target data. Therefore we also use the average of predicted labels to estimate the conditional probabilities:

$$\hat{P}_{Y=y|X=\vec{x}} = \sum_m \mathbb{1}(y = f_m(\vec{x}))P_{M=m}, \tag{3.3}$$

where $\mathbb{1}(.)$ is the indicator function which returns 1 if its argument is true and 0 otherwise. Note that this definition assumes discrete labels. For other types of target variables Y, some modification is necessary. In particular, considering continuous Y and regression, conditional densities would be computed instead. In order to use the fixed output predictions in this case, a one-dimensional distribution can be fit to the set of model predictions of y for a given $\vec{x}$, using kernel density estimation [5].

Another issue arises with the combination of limited labeled data and imbalanced data. In this case, many instances might be predicted as belonging to the same class by most models. This can be an issue when sampling then, since the sampled label set might be all of one class - it might take a much larger set of sampled data sets to get a significant number that have labeled data from both classes. Therefore we also propose a balance modification in which we sample data sets until each has at least one labeled instance from each class.

3.3 Relationship to Expectation Maximization, Bootstrapping, and Stability Selection If the iterative re-weighting strategy is used, and we consider the missing labels to be hidden variables, this is in some ways similar to expectation-maximization-type approaches for learning with hidden variables - another category of semi-supervised learning algorithms [40]. However there are a few key differences. First the hidden variables are used mainly for evaluation of the models, as opposed to being an integral part of the models themselves. I.e., when training the models across the different random samples of the data and labels, instead of trying to incorporate all of the estimates for the labels of the unlabeled data in the training process, these estimated labels are mainly used in evaluating the trained model. The focus is on keeping the training conditions the same as for the actual training data. Second, maximization is not performed over the expectation, since each model is trained (maximized) over its local sample and then an expected value is taken. This emphasizes the key point that, instead of using this procedure to try to infer likely values for the hidden variables, i.e., the missing labels, which could be unreliable, or update a single model, the goal is to estimate the performance of the models. In this way, if most labels cannot be predicted with certainty, a model that most consistently achieves better performance, across all of these sample data sets, will be preferred, even if it is not the most likely (assuming there is even a fully defined probabilistic model). Therefore, this method may be more closely related to the metric-based and stability selection approaches mentioned in the related work (e.g., [30, 20]), but tries to estimate this stability by fully re-training models on similar data sets as opposed to either data subsets or a one-dimensional stability criterion of similarity between function values on labeled and unlabeled data.

The proposed resampling approach is also similar to bootstrapping [15], which samples from the labeled data with replacement to generate the similar data sets, and evaluates these on the held out data for each set. However, in this limited labeled data setting, there is only a small set of labeled data to resample from, e.g., in one experiment we only have 4 labeled points. In this case most of the data sets will not be very different - there is a limited number of unique data sets that can

be sampled. Furthermore there is a risk that a sample will contain only a single class, in which case the algorithms being evaluated might not even be applicable. If stratification is used to avoid this issue, the possible samples are reduced further. Enumerating possible train/hold-out combinations with stratification essentially amounts to nearly the same approach as cross-validation so this approach would suffer the same limitations as mentioned for cross-validation. Also for this reason in our experiments we only compare with cross validation as it is more widely used in this setting. Therefore another way of looking at our approach is that it extends a bootstrapping approach by using estimated labels with the unlabeled data instead. Data sets are resampled each time, but from the entire set of data with our approach, which includes the large set of unlabeled data allowing more possible data sets, and evaluation is performed on the large set of unlabeled data as well for each sample, as opposed to a very small hold out set.

4 Experimental Study

Here we provide experimental study of various model selection approaches for different semi-supervised learning (SSL) algorithms evaluated on 4 different data sets.

4.1 Data Sets We evaluated the model selection approaches with four different data sets. The first data set is a synthetic 2-dimensional data set, the second is a webpage classification data set, the third a document classification data set, and the fourth an image classification one. Below we describe these data sets, and their characteristics are summarized in Table 1. Three of the four data sets fit the scenario of *multi-view* semi-supervised learning, where two distinct sets of features called "views" are available, so we use multi-view semi-supervised learning algorithms for these. The other data set fits the manifold learning scenario, so we use a manifold-based semi-supervised learning algorithm for this one.

Table 1: Data sets, characteristics, and semi-supervised learning algorithm used.

Data Set	Num. Labeled	Num. Unlabeled	Num. View 1 Features	Num. View 2 Features	Class Ratio num. pos. / num. neg.	SSL Method Used
Synth	4	400	2	2	1.000	Co-Regularization [36]
WebKB	12	1039	2168	338	0.280	Co-Training [4]
Citeseer	24	1164	3703	1164	0.236	Co-Training [4]
Coil	20, 40	1420, 1400	1024	n/a	0.818	Manifold Co-Regularization [35]

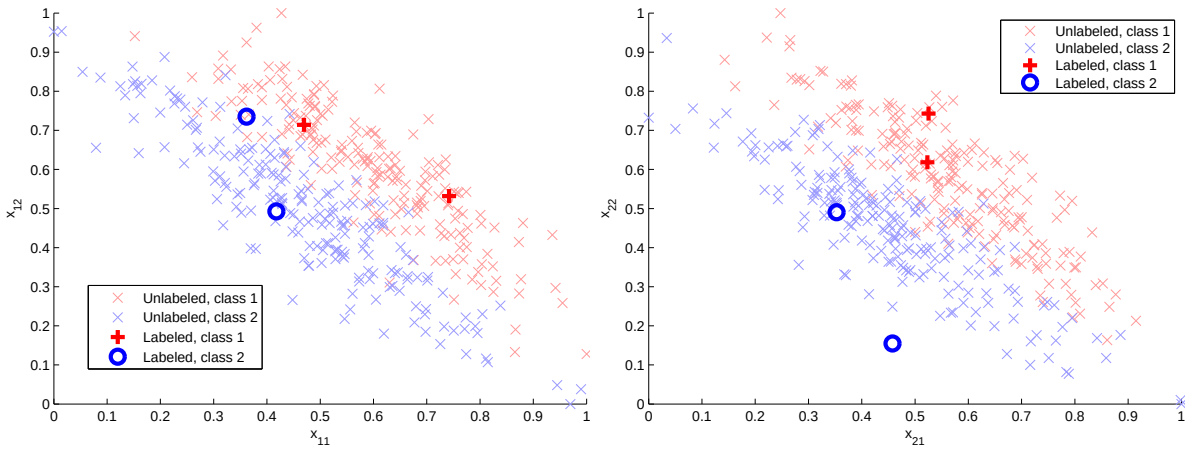
4.1.1 Synthetic Data Set The synthetic data in each view was generated from two slightly overlapping 2D Gaussian distributions, with the same pair of distributions used for both views. Specifically, for each class data was sampled from a zero-mean Gaussian distribution in two-dimensions with covariance $\{\{\ 16,\ 0\ \},\{\ 0,\ 1\ \}\}$, and was then transformed with the rotation matrix $\{\{\ \cos(\frac{\pi}{4}),\ -\sin(\frac{\pi}{4})\ \},\{\ \sin(\frac{\pi}{4}),\ \cos(\frac{\pi}{4})\ \}\}$; then the offset of $\{1, 1\}$ was added for the positive class, and $\{-1, -1\}$ for the negative class. To generate the data each instance was sampled from the distribution for one of the classes in view 1, and independently from the same class distribution in view 2. This way the two views are conditionally independent given the class label - an ideal scenario for multi-view semi-supervised learning algorithms. The data in each view was normalized to have minimum 0 and maximum 1 after the sampling. For each trial, 2 labeled training points and 200 unlabeled points, were generated for each class. Figure 1 shows a sample of the generated data in each view.

4.1.2 WebKB Course Data Set The WebKB Course data set is a collection of 1051 websites from four universities, belonging to two categories: course websites or non-course websites. There are 230 websites in the course category, and 821 in the non-course category. The first view consists of text on the webpage itself, the second view consists of the link text of links from other webpages linking to the webpage.

We obtained the webpage and link text data[1] then applied standard text pre-processing using Weka [14] to obtain 2,168 features in the text view and 338 features in the link view. As in [4], for each experiment iteration we randomly sample 3 course and 9 non-course instances for labeled training. The remaining instances were used for the unlabeled data.

4.1.3 Citeseer Data Set The Citeseer data set is a collection of scientific articles split into six categories ("Agents", "AI", "DB", "IR", "ML", and "HCI"). The first view consists of the text from the abstract of each article, and the second view is

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



(a) View 1 (b) View 2

Figure 1: Sample of two views of data generated for 2D test case

the citation profile, the list of papers a given paper is cited by or cites in the database. We obtained a version of the data set[2] with binary vectors for each article indicating if a word is present or not in that article, and built binary citation vectors for each with a 1-entry for a feature indicating the paper cites or is cited by the other paper given that corresponding index. We removed all papers with fewer than five other papers in the collection that cite or are cited by the paper. This resulting data set contains 1164 documents with 3703 features in view 1 and 1164 features in view 2.

As in [37] we take the largest class ("DB") as the positive class and the remainder as the negative class, resulting in 222 instances in class 1 and 942 instances in class 2. Also as in [37], for each experiment iteration we use 4 randomly sampled class 1 instances and 20 randomly sampled class 2 instances to make up the labeled training set, and the rest for the unlabeled data.

4.1.4 Coil Data Set The Coil data set is built from the Coil20 data set, commonly used as a benchmark data set for manifold-based approaches to semi-supervised learning. The data set consists of 1440 32x32 pixel greyscale images of 20 different objects, taken at various angles. We obtained the data set from the website[3] of the first author of a previous work on manifold regularization [33]. We created a binary classification task by splitting the objects into the categories of "toys," corresponding to 9 objects, and "other household objects," corresponding to the remaining 11 objects. We followed the same approach as [35] for computing the fixed kernels and adjacency matrices for the data (using 1-nearest-neighbor and fixed kernel width). To form the labeled and unlabeled sets we also followed the approach of [35], using stratified sampling to sample 2 images from each category to form the labeled set and used the rest as unlabeled data, for each experiment trial.

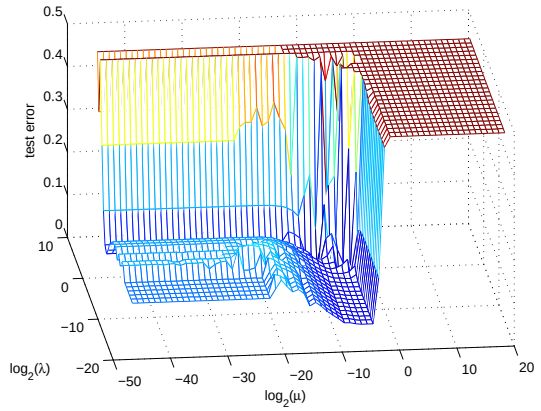
4.2 Preliminary Synthetic Data Study We first performed some preliminary study with the synthetic data to get an idea of the effect of updating the weights in the resampling approach, and to generate plots for qualitative comparison of the model selection methods showing how the estimated scores compare to ground truth.

For the preliminary synthetic we used the Synthetic, two Gaussians data set described in the previous section, and co-regularized logistic regression. The figure showing a sample of the data in both views is re-produced here for convenience in Figure 1. Here the L_1 regularization hyper-parameters for each view are fixed to be equal, so that results can be displayed in 3-D plots. Therefore, there are two regularization hyper-parameters, the L_1 regularization hyper-parameter, λ, and the co-regularization hyper-parameter μ. The set of models to evaluate are taken to be a grid of combinations of these two hyper-parameters, with λ ranging from 2^{-20} to 2^3 by incremental powers of 2, and μ similarly ranging from 2^{-40} to 2^{25}, but multiplied by a starting value equal to the number of labeled instances over the number of unlabeled instances.

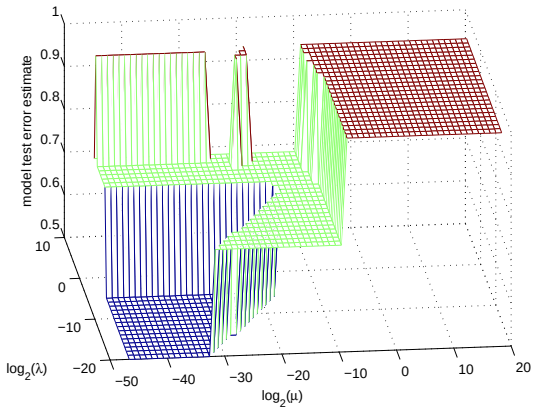
Since other approaches are method-dependent, the state-of-the-art metric-based model selection approach (ADA) [30] is taken as the main competitor to the proposed model selection approach, with cross-validation used as a baseline. When

[2] Available here: http://www.cs.umd.edu/ sen/lbc-proj/LBC.html
[3] Available here: http://vikas.sindhwani.org/manifoldregularization.html

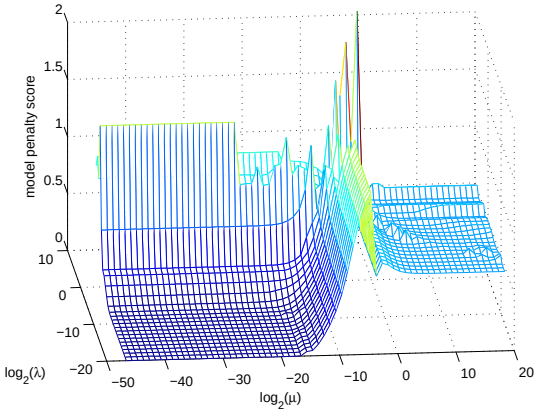
computing the ADA evaluator to avoid divide by 0 scenarios a small amount is added to probabilities of zero in the experiments.



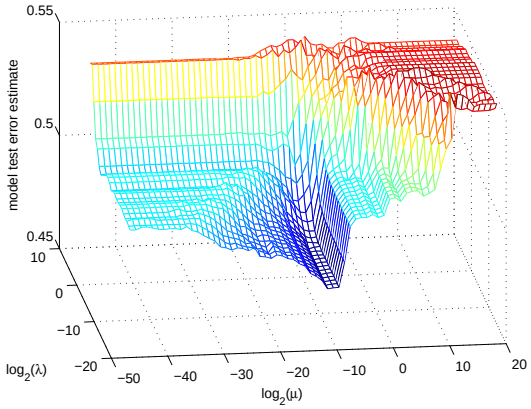
(a) Ground truth test error



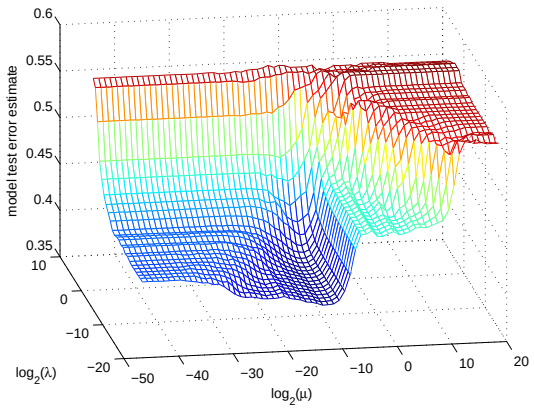
(b) Cross-validation estimate



(c) Metric-based approach estimate (ADA)



(d) Proposed re-sampling approach estimate, 5 updates



(e) Proposed re-sampling approach estimate, 20 updates

Figure 2: Ground truth and estimated test error (z-axis) vs pairs of hyper-parameters for different model selection methods

Results are shown in Figure 2 for a particular training data set, as 3D meshes showing the estimated model score for each pair of hyper-parameters. The results are shown for both 5 and 20 updates of the conditional probability estimates for

the proposed re-sampling approach. The figure shows the average score or test error estimates for each method evaluated over the grid of hyper-parameters, with the hyper-parameters on the x and y axes and the z axis corresponding to the estimate. The true test error is also shown in Figure 2(a). In this case, both the cross-validation and metric-based model selection approaches estimate their best scores for models in a sub-optimal region of the joint hyper-parameter space; the estimates corresponding to these methods are not accurate for this limited amount of labeled data in this case, and choosing the models with the best estimated performance would result in selecting sub-optimal models. The proposed re-sampling method however is able to come quite close to correctly estimating the low test error regions of the joint hyper-parameter space, and models with the best estimated performance result in lower test error.

We found applying the re-weighting updates generally smoothed-out the model score plots but did not have too much of effect on which models achieved the best scores. Therefore for the following experiments, for simplicity, we did not perform any weight updates for our resampling method.

4.3 Experiment Procedure As mentioned in the previous section, the state-of-the-art metric-based model selection approach (ADA) [30] is taken as the main competitor to the proposed model selection approach, which we denote **ADA**. We denote cross-validation [15] as **CV**. For the cross-validation approach we use leave-one-out cross-validation as the size of the labeled data is small. We compare with the .632+ bootstrap estimator [15] as well, and denote this method **.632+**. We also compare with the model selection approach of maximum marginal likelihood (also referred to as ML type II or evidence-based approach) [36] when possible. In order to be able to compute a marginal likelihood we use the Gaussian process co-regularization model (GPCR) [36] for any data set for which we use co-regularization. This model is a Bayesian probabilistic model and allows for approximate computation of the marginal likelihood, see [36] and [25] for details. We implemented this method with the "Gaussian Processes for Machine Learning Toolbox" version 3.1 [24]. For the Coil data set, we used the manifold co-regularization approach described in [35] to compute the kernels and used the GPCR method with the computed kernel matrices so that we could obtain marginal likelihoods. We denote the maximum marginal likelihood method as **MML**. Finally we denote the proposed Similar Data Sampling approach as **SDS**, and as mentioned we do not update the weights for each model - i.e., we use uniform weighting. Furthermore, we compare with the version of SDS that uses the average of predicted outputs for each model (Equation 3.3) as opposed to probabilistic outputs, which we denote **SDS-L**. Additionally for the Citeseer data set, since it is highly imbalanced and the lowest achievable test error is not close to 0, we use MCC as opposed to test error with the SDS methods for scoring models. However, for CV we still used test error, as using MCC causes significantly worse performance, due to having only a single test instance, i.e., using MCC with CV is not really an option for limited labeled data. In addition we test the combination of the state-of-the-art method ADA with our method and denote this combination **SDS+ADA**. This combination is accomplished by ranking the models with each selection method then adding the ranks to obtain new scores for each model - the model with the lowest score is then selected. There are two more baselines we provide as well. First, a non-semi-supervised learning approach using the Gaussian process classifier with both views if available. For the data sets we tested each view individually, stacking the views together to form a single view, and taking the average of classifiers trained on each view separately. Of the three, the averaging approach gave the best results, or not significantly different from the best, on all data sets, so we report these results. We denote this method as **GP - No SSL**. The final baseline reported is the result obtained when fixing the hyper-parameters across trials to the best set of fixed hyper-parameters from the grid of hyper-parameters (the hyper-parameter combination that gives the lowest test error averaged across all of the trials). This is the ideal result obtained if we had a model selection method capable of exactly determining the average performance of each model. We denote this method as **Best Fixed**. For each of the methods that use sampled sets (i.e., .632+, SDS, SDS+ADA, SDS-L, and SDS-LB) we sample 100 sets. The methods used are summarized in Table 2.

We chose a semi-supervised learning algorithm that had the potential to work well for each data set, so that some model under consideration could achieve good performance and model selection could have a noticeable effect. These choices are shown in Table 1. For all methods, we use the same logistic loss model. We use logistic likelihood models in GPCR and in a Gaussian process classifier for the non-semi-supervised learning baseline. For the co-training algorithm we use L_1 regularized logistic regression classifiers as the base models.

The hyper-parameter grids used for model selection are as follows. GPCR has two hyper-parameters, σ_1 and σ_2 [36]. For the synthetic data σ_1 and σ_2 were varied on a grid of values $\{10^2, 10^1, \ldots, 10^{-5}\}$, resulting in 64 different models to choose from. For the Coil data set we follow the approach of [35] and vary σ_1 and σ_2 over $\{10^6, 10^4, 10^2, 10^0, 10^{-1}, 10^{-2}\}$, resulting in 36 models to choose from. For the co-training method, there are 3 hyper-parameters to select. The first is the ratio of the number of positive to number of negative estimated confident points to update the labeled set with at each iteration. We varied this ratio over $\{$1:1, 1:2, 1:3, 1:4, 1:5$\}$, as in all training sets the ratio for the labeled data indicates

Table 2: Model selection methods used.

GP - No SSL	Gaussian process classifier [25] that does not do semi-supervised learning
CV	Leave-one-out cross-validation [15]
.632+	The .632+ bootstrap estimator [15]
MML	Maximum marginal likelihood approach [36]
ADA	State-of-the-art metric-based approach [30]
SDS	The proposed Similar Data Sampling approach
SDS+ADA	SDS combined with ADA by adding model ranks given by the two approaches to obtain new scores
SDS-L	SDS using the average of the predicted labels with each model (Equation 3.3) as the class conditional probability as opposed to averaging probabilistic outputs. Also excludes samples with labeled data having only one class.
Best Fixed	The fixed model corresponding to the set of hyper-parameters that gave the lowest test error averaged across all trials.

imbalance with fewer positive instances than negative. The other hyper-parameters are L_1 regularization hyper-parameters for view 1 and 2, λ_1 and λ_2, respectively. We varied these over $\{10^0, 10^{-1}, \ldots, 10^{-4}\}$. This resulted in 125 different models for the co-training method.

Since GPCR is transductive, we used a transductive approach for the experiments - that is for each trial, the data was randomly partitioned into num. labeled and num. unlabeled data instances as described in Table 1, and the unlabeled data is also used as the testing data for evaluating performance. We report results averaged over 100 random trials for each data set.

We report test error, Matthews Correlation Coefficient (MCC) - which is the correlation of the predicted labels with the actual labels, and F1 Score for each data set, described below. Let tp denote the number of true positive predictions, fp the number of false positives, fn false negatives, and tn true negatives.

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

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

4.4 Experiment Results The experimental results are summarized in Table 3. Additional significance testing is provided in Table 4, comparing the SDS method to other methods for each data set, in Table 5 for the SDS+ADA method, and in Table 6 for SDS-L. The testing is performed with respect to the test error for all data sets but the Citeseer data set, in which MCC is used instead as the data set is highly imbalanced and test error close to 0 is not achievable.

For the synthetic data, we found that the GPCR method was more sensitive to the hyper-parameters than the co-regularized logistic regression approach used in the preliminary study, which is likely part of the reason why most of the methods had higher variance and were farther from performing as well as the best fixed model. In order to take into account the sensitivity of the methods to the hyper-parameters for each data set, as well as how difficult the selection task is for a given data set, we also report how many models out of the total number considered are close to the best model in the set of all models (i.e., the hyper-parameter grid), including the best. Specifically we report the fraction of models in the set considered that give test error within 0.025 of the Best Fixed Hyper-Parameters model. This corresponds to the last row of the table, with the entry **"Frac. Close"**.

Across the first four tasks, those with the smallest amount of labeled data, the SDS method either achieves comparable or significantly better performance than the other methods, and is only out-performed by ADA on the Coil data set. Cross-validation (CV), the .632+ bootstrap estimator (.632+), and maximum marginal likelihood (MML) clearly suffer performance deterioration for very small amounts of labeled data. ADA remained competitive, but SDS-L obtained better scores on three out of the four data sets, with ADA still giving the best results for Coil even when reducing the number of labeled data instances to 20, though this did narrow the gap between the two methods. The combination SDS+ADA sometimes offered an improvement over SDS and ADA, but this was usually not very significant. SDS-L had the best

Table 3: Mean $\pm$ std. dev. of MCC, F1 score, and test error over 100 trials for each data set for the different model selection approaches, with best scores shown in bold. The data sets are ordered by increasing amount of labeled data.

		GP - No SSL	**CV**	**.632+**	**MML**	**ADA**	**SDS**	**SDS +ADA**	**SDS-L**	***Best Fixed***	***Frac. Close***
Synth (num. lab.=4)	Test Error	0.298 $\pm$0.107	0.217 $\pm$0.154	0.191 $\pm$0.138	0.294 $\pm$0.139	0.168 $\pm$0.128	0.171 $\pm$0.127	0.164 $\pm$0.112	**0.040** $\pm$0.050	0.030 $\pm$0.014	0.078 (5/64)
	MCC	0.403 $\pm$0.214	0.566 $\pm$0.308	0.619 $\pm$0.277	0.412 $\pm$0.278	0.664 $\pm$0.255	0.659 $\pm$0.253	0.671 $\pm$0.224	**0.921** $\pm$0.100	0.941 $\pm$0.027	
	F1 Score	0.701 $\pm$0.107	0.783 $\pm$0.154	0.809 $\pm$0.139	0.706 $\pm$0.139	0.832 $\pm$0.128	0.829 $\pm$0.127	0.835 $\pm$0.113	**0.960** $\pm$0.050	0.970 $\pm$0.013	
WebKB (num. lab.=12)	Test Error	0.216 $\pm$0.060	0.048 $\pm$0.043	0.057 $\pm$0.054	n/a	0.038 $\pm$0.021	**0.028** $\pm$0.017	0.029 $\pm$0.008	0.031 $\pm$0.008	0.017 $\pm$0.003	0.352 (44/125)
	MCC	0.559 $\pm$0.086	0.881 $\pm$0.082	0.840 $\pm$0.180	n/a	0.891 $\pm$0.047	**0.917** $\pm$0.061	0.914 $\pm$0.024	0.909 $\pm$0.026	0.950 $\pm$0.010	
	F1 Score	0.651 $\pm$0.065	0.905 $\pm$0.069	0.860 $\pm$0.189	n/a	0.911 $\pm$0.040	**0.932** $\pm$0.070	0.931 $\pm$0.020	0.927 $\pm$0.022	0.961 $\pm$0.008	
Coil (num. lab.=20)	Test Error	0.410 $\pm$0.009	0.075 $\pm$0.047	0.081 $\pm$0.057	0.095 $\pm$0.066	**0.047** $\pm$0.010	0.068 $\pm$0.034	0.060 $\pm$0.024	0.055 $\pm$0.010	0.047 $\pm$0.010	0.444 (16/36)
	MCC	0.225 $\pm$0.028	0.859 $\pm$0.082	0.848 $\pm$0.100	0.823 $\pm$0.116	**0.909** $\pm$0.019	0.870 $\pm$0.060	0.885 $\pm$0.043	0.893 $\pm$0.019	0.909 $\pm$0.019	
	F1 Score	0.164 $\pm$0.034	0.906 $\pm$0.073	0.895 $\pm$0.090	0.874 $\pm$0.105	**0.945** $\pm$0.012	0.916 $\pm$0.049	0.928 $\pm$0.033	0.935 $\pm$0.013	0.945 $\pm$0.012	
Citeseer (num. lab.=24)	Test Error	0.435 $\pm$0.029	0.140 $\pm$0.063	0.258 $\pm$0.089	n/a	0.149 $\pm$0.083	0.140 $\pm$0.087	0.137 $\pm$0.094	**0.135** $\pm$0.090	0.117 $\pm$0.070	0.184 (23/125)
	MCC	0.267 $\pm$0.052	0.501 $\pm$0.264	0.365 $\pm$0.196	n/a	0.576 $\pm$0.212	0.585 $\pm$0.226	0.595 $\pm$0.234	**0.605** $\pm$0.213	0.602 $\pm$0.240	
	F1 Score	0.423 $\pm$0.027	0.556 $\pm$0.271	0.456 $\pm$0.202	n/a	0.659 $\pm$0.164	0.664 $\pm$0.180	0.674 $\pm$0.183	**0.681** $\pm$0.167	0.669 $\pm$0.197	
Coil (num. lab.=40)	Test Error	0.375 $\pm$0.010	0.033 $\pm$0.016	0.031 $\pm$0.016	**0.024** $\pm$0.016	**0.024** $\pm$0.016	0.035 $\pm$0.018	0.030 $\pm$0.017	0.031 $\pm$0.014	0.024 $\pm$0.016	0.500 (18/36)
	MCC	0.314 $\pm$0.024	0.935 $\pm$0.031	0.940 $\pm$0.031	**0.954** $\pm$0.031	**0.954** $\pm$0.031	0.932 $\pm$0.035	0.942 $\pm$0.033	0.939 $\pm$0.027	0.954 $\pm$0.031	
	F1 Score	0.287 $\pm$0.033	0.961 $\pm$0.019	0.964 $\pm$0.020	**0.973** $\pm$0.019	**0.973** $\pm$0.019	0.960 $\pm$0.022	0.966 $\pm$0.021	0.964 $\pm$0.017	0.973 $\pm$0.019	
Average	Test Error	0.347	0.103	0.123	n/a	0.085	0.088	0.084	**0.058**	0.047	n/a
	MCC	0.354	0.749	0.722	n/a	0.799	0.793	0.802	**0.853**	0.871	
	F1 Score	0.445	0.822	0.797	n/a	0.864	0.860	0.867	**0.893**	0.903	

Table 4: Significance testing results at the 5 percent level for paired t-tests between the proposed approach, SDS, and other model selection approaches for MCC on the Citeseer data set and test error on the rest. A "1" indicates a significant difference in means, "0" not significant, and a "+" indicates SDS did better, "-" worse.

	Synth	**WebKB**	**Coil (n=20)**	**Citeseer**	**Coil (n=40)**
GP- No SSL	+1	+1	+1	+1	+1
CV	+1	+1	0	+1	0
.632+	0	+1	0	+1	-1
MML	+1	n/a	+1	n/a	-1
ADA	0	+1	-1	0	-1

Table 5: Significance testing results at the 5 percent level for paired t-tests between the rank sum combined approach, SDS+ADA, and other model selection approaches for MCC on the Citeseer data set and test error on the rest. A "1" indicates a significant difference in means, "0" not significant, and a "+" indicates SDS+ADA did better, "-" worse.

	Synth	**WebKB**	**Coil (n=20)**	**Citeseer**	**Coil (n=40)**
GP- No SSL	+1	+1	+1	+1	+1
CV	+1	+1	+1	+1	+1
.632+	0	+1	+1	+1	0
MML	+1	n/a	+1	n/a	-1
ADA	0	+1	-1	0	-1
SDS	0	0	+1	0	+1

average performance, that is the performance averaged across all of the tasks (corresponding to the bottom row of Table 3). The most drastic difference is seen for the smallest amount of labeled data, i.e., for the Synth data SDS-L was able to attain mean test error of 0.040, close to the mean test error of the best single model, 0.030, as compared to 0.217 for CV, 0.294 for MML, and 0.168 for ADA.

Additionally, all of the model selection methods performed well on the Coil data set with 40 labeled instances - coming close to achieving the same performance as the best fixed model. This data set was particularly easy for model selection,

Table 6: Significance testing results at the 5 percent level for paired t-tests between SDS using label outputs, SDS-L, and other model selection approaches for MCC on the Citeseer data set and test error on the rest. A "1" indicates a significant difference in means, "0" not significant, and a "+" indicates SDS-L did better, "-" worse.

	Synth	WebKB	Coil (n=20)	Citeseer	Coil (n=40)
GP- No SSL	+1	+1	+1	+1	+1
CV	+1	+1	+1	+1	+1
.632+	+1	+1	+1	+1	0
MML	+1	n/a	+1	n/a	-1
ADA	+1	+1	-1	0	-1
SDS	+1	0	+1	0	+1
SDS+ADA	+1	0	0	0	0

which is also indicated to some extent by "Frac. Close" of 0.5 meaning half of the models to select from had performance close to the best fixed model.

A key observation is that using averaging with labels to estimate class probabilities (SDS-L), as opposed to probabilities (SDS) generally worked better, since even if most trained models agree on the labels exactly, the models themselves might output class probabilities close to 0.5, so that the generated data sets would still have high variation. This could cause the SDS method to perform more poorly when the probabilistic models are not well-calibrated, but offer some improvement when they are. Checking the average test errors computed by the SDS method for the Coil data set, we found they did not vary far from 0.5 (the minimum was 0.485 and the maximum 0.498), even though the majority of models actually had low test error. A similar issue occurred with the Synth data. SDS-L avoids this issue and also allows the similar data sampling approach to be used with models that do not have probabilistic outputs.

5 Conclusion and Future Work

We have proposed a new approach to model selection for semi-supervised learning algorithms, based on estimating performance by re-training and evaluating each model on many generated, similar data sets, which we called SDS (Similar Data Sampling) for short. In the experimental study on four data sets, we found the version of SDS using the average of label predictions to estimate conditional distributions (SDS-L) to improve over the widely used cross-validation approach and the Bayesian approach of maximum marginal (type II) likelihood, for smaller amounts of labeled data, i.e., for the tasks in our experiments with less than 40 labeled instances. We also compared with a state-of-the-art metric-based approach to semi-supervised model selection, ADA, which to our knowledge, has not yet been evaluated for the case of model selection for semi-supervised learning algorithms, and found SDS-L achieved better performance on three of the four data sets.

A key area of future work is to apply SDS to a broader range of learning scenarios where effective model selection methods are lacking. Indeed this unique broad applicability is a key advantage of SDS - this approach can be applied to scenarios where traditional model selection methods cannot, because complete data sets are sampled. A particular example of interest is the active learning scenario [31], in which an algorithm iteratively selects which instances to obtain information about from an oracle, e.g., labels for unlabeled data. If the active selection algorithm has tuning parameters that must be set, there is no way to do this with traditional model selection approaches since it requires estimating the performance of the selection strategy as labels are obtained for the unlabeled data. However this is easily accomplished with SDS as the entire active acquisition process can be simulated using the complete data sets generated (for each actual iteration). The main challenge for future work is extending the similar data sampling/generation approach to handle these different scenarios, including, for example, active view completion (see Chapter **??** for details on this scenario). Another key scenario for future work, which is also an open problem for model selection [13], is transfer learning. This line of future work involves applying the SDS strategy to transfer learning problems which can have no labeled target data at all.

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

References

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

[2] Y. Bengio and N. Chapados. Extensions to metric based model selection. *The Journal of Machine Learning Research*, 3:1209–1227, 2003.

[3] S. Bickel and T. Scheffer. Estimation of mixture models using co-em. In *Proceedings of the European Conference on Machine Learning*, pages 35–46. Springer, 2005.

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

[5] Z. Botev, J. Grotowski, and D. Kroese. Kernel density estimation via diffusion. *The Annals of Statistics*, 38(5):2916–2957, 2010.

[6] U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel. Efficient co-regularised least squares regression. In *Proceedings of the 23rd international conference on Machine learning*, pages 137–144, 2006.

[7] U. Brefeld and T. Scheffer. Co-EM support vector learning. In *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004.

[8] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Proceedings of the tenth international workshop on artificial intelligence and statistics*, volume 2005. Citeseer, 2005.

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

[10] R. Collobert, F. Sinz, J. Weston, L. Bottou, and T. Joachims. Large scale transductive svms. *Journal of Machine Learning Research*, 7, 2006.

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

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

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

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

[15] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2009.

[16] M. Kaariainen. Semi-supervised model selection based on cross-validation. In *International Joint Conference on Neural Networks*, pages 1894 –1899, 2006.

[17] B. Krishnapuram, D. Williams, Y. Xue, A. Hartemink, L. Carin, and M. Figueiredo. On semi-supervised classification. *Advances in neural information processing systems*, 17:721–728, 2004.

[18] T. Lange, M. Braun, V. Roth, and J. Buhmann. Stability-based model selection. In *In Advances in Neural Information Processing Systems*, 2002.

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

[20] O. Madani, D. Pennock, and G. Flake. Co-validation: Using model disagreement on unlabeled data to validate classification algorithms. In *Proceedings of NIPS*. Citeseer, 2004.

[21] C. Müller, S. Rapp, and M. Strube. Applying co-training to reference resolution. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 352–359. Association for Computational Linguistics, 2002.

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

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

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

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

[26] D. S. Rosenberg. *Semi-supervised learning with multiple views*. PhD thesis, University of California, Berkely, 2008.

[27] A. Sarkar. Applying co-training methods to statistical parsing. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics, 2001.

[28] D. Schuurmans. A new metric-based approach to model selection. In *In Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, pages 552–558, 1997.

[29] D. Schuurmans and F. Southey. Metric-based methods for adaptive model selection and regularization. *Machine Learning*, 48(1):51–84, 2002.

[30] D. Schuurmans, F. Southey, D. Wilkinson, and Y. Guo. Metric-based approaches for semi-supervised regression and classification. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*, pages 421–451. The MIT Press, 2006.

[31] B. Settles. Active learning literature survey. Technical Report 1648, Department of Computer Science, University of Wisconsin-Madison, 2010.

[32] V. Sindhwani, W. Chu, and S. Keerthi. Semi-supervised gaussian process classifiers. In *Proceedings of the 20th International Joint*

Conference on Artificial Intelligence, pages 1059–1064, 2007.

[33] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 824–831. ACM, 2005.

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

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

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

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

[38] X. Zhang and W. S. Lee. Hyperparameter Learning for Graph Based Semi-supervised Learning Algorithms. In *Advances in Neural Information Processing Systems 19*, 2007.

[39] Z. Zhou, D. Zhan, and Q. Yang. Semi-supervised learning with very few labeled training examples. In *Twenty-Second AAAI Conference on Artificial Intelligence*, pages 675–680, 2007.

[40] X. Zhu. Semi-supervised learning literature survey. Technical report, Department of Computer Science, University of Wisconsin, Madison, 2008.

[41] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the twentieth international conference on Machine learning*, 2003.