Semi-Supervised SVMs for Classification with Unknown Class Proportions and a Small Labeled Dataset

S Sathiya Keerthi
Yahoo! Labs
Santa Clara, California, U.S.A
selvarak@yahoo-inc.com

Bigyan Bhar
Indian Institute of Science
Bangalore, India
bigyan@csa.iisc.ernet.in

S Sundararajan
Yahoo! Labs
Bangalore, India
ssrajan@yahoo-inc.com

Shirish Shevade
Indian Institute of Science
Bangalore, India
shirish@csa.iisc.ernet.in

ABSTRACT

In the design of practical web page classification systems one often encounters a situation in which the labeled training set is created by choosing some examples from each class; but, the class proportions in this set are not the same as those in the test distribution to which the classifier will be actually applied. The problem is made worse when the amount of training data is also small. In this paper we explore and adapt binary SVM methods that make use of unlabeled data from the test distribution, viz., Transductive SVMs (TSVMs) and expectation regularization/constraint (ER/EC) methods to deal with this situation. We empirically show that when the labeled training data is small, TSVM designed using the class ratio tuned by minimizing the loss on the labeled set yields the best performance; its performance is good even when the deviation between the class ratios of the labeled training set and the test set is quite large. When the labeled training data is sufficiently large, an unsupervised Gaussian mixture model can be used to get a very good estimate of the class ratio in the test set; also, when this estimate is used, both TSVM and EC/ER give their best possible performance, with TSVM coming out superior. The ideas in the paper can be easily extended to multi-class SVMs and MaxEnt models.

Categories and Subject Descriptors: I.5.2 [Pattern Recognition] Design Methodology-Classifier design and evaluation

General Terms: Algorithms, Experimentation

Keywords: Transductive and Semi-supervised learning, Classification, Support Vector Machines

1. INTRODUCTION

The problem of classifying web pages into a given set of classes arises frequently in web applications. Linear classifiers such as support vector machines (SVMs) and maximum entropy (MaxEnt) models employing a rich feature representation (e.g., bag-of-words) are used successfully for this purpose. When labeling is expensive one is forced to work with a small set of labeled examples for training. In such cases a linear classifier trained using only labeled data does not give a good performance. Performance can be significantly boosted by employing semi-supervised methods that make effective use of unlabeled data, which is usually available in plenty. The transductive support vector machine (TSVM) [10] and MaxEnt models trained with expectation regularization (ER) [12] are good examples of such methods. Such semi-supervised methods assume the knowledge of auxiliary information about the underlying data such as class proportions.

The performance of semi-supervised methods such as ER and TSVM are sensitive to the class proportion values used to find their solution; see section 4.2 for a detailed empirical analysis. In many practical situations, class proportions are unknown. Often it also happens that the class proportions in the labeled training set are different from those in the actual distribution to which the classifier will be applied. A few examples of such scenarios are as follows.

- There are tens of thousands of city aggregator sites covering categories such as dining, attractions, night life etc. The content and vocabulary across these sites are very similar, but in each site the proportion of examples belonging to various classes can be quite different. So, building a separate classifier for each site is appropriate. Let us say that we are given a few global examples of dining and non-dining pages. The aim is to develop a classifier for each site to separate dining pages from others making use of unlabeled web pages in that site.
- Frequently, in classifier design, one would like to reuse labeled data from a previous run to develop a classifier for a new distribution (e.g., a new stream of news pages); the old labeled set is still good as the union of sets of samples from the individual classes, but class proportions are different.
- A user shows a few relevant and irrelevant example

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

web pages and asks for finding relevant pages from a fresh collection of web pages.

The main aim of this paper is to address the problem of mismatched class proportions mentioned above.

Let x denote a generic example web page and c denote a class. From a probabilistic viewpoint our problem consists of dealing with a situation in which: (i) $p(x|c)$ is same in the labeled data and the actual data of interest for all c; but, (ii) $\{p(c)\}$ is very different in the two sets. Coupled with labeled data being small the problem becomes difficult.

Note that, even if labeled data is not small, extension of methods such as TSVM and ER to deal with mismatched class proportions requires careful thought because labeled data does not have information on actual class proportions and unlabeled data has to be suitably brought in to estimate them. There have been some previous efforts to address this problem for TSVMs [7, 14]. But these methods have some basic issues: they make adhoc changes to the training process in an active self-learning mode and are not cleanly tied to a well-defined problem formulation; they significantly deviate from the TSVM formulation and hence perform quite worse than TSVMs when labeled data and the actual distribution are indeed well matched in terms of class proportions; and, they have not been demonstrated in difficult situations involving large distortions in class proportions.

In this paper we empirically analyze various issues related to the problem of mismatched class proportions, propose ways of estimating actual class proportions and demonstrate their usefulness. We only take up binary classification problems in this paper; but the ideas are quite general and they have the potential for extension to multi-class settings. Since binary classification involves only two classes (positive and negative), class proportion can be represented using a single quantity f, the fraction of positive examples.

1.1 Contributions of the paper

In a nutshell following are our main contributions, listed by order of treatment in the paper.

1. The ER method is introduced in the context of SVMs. In addition we introduce two related methods based on expectation constraints (EC) and simple threshold adjustment (SVMTh).

2. We empirically analyze all the methods (TSVM, ER, EC and SVMTh) in terms of their learning curves and their sensitivity to incorrect specification of f.

3. If labeled data has distorted class proportion (even severe), but the actual f happens to be known, we empirically show that the methods are pretty much unaffected.

4. To handle the crucial case in which the actual f is unknown we propose and empirically analyze two methods for estimating the actual f. The first estimation method is based on finding the f having the least labeled loss along a trajectory of solutions given by a method as f is varied. The second estimation method is based on fitting a mixture of two Gaussians to the output of the SVM trained using only labeled data. The first estimation method works well with TSVM and it is suited for the situation where labeled data is small while the second estimation method is powerful and works well with all semi-supervised methods when labeled data is large. Overall, TSVM combined with the two estimation methods (suitably switched depending on the amount of labeled data available) is the top performing method.

2. DESCRIPTION OF METHODS

In this section we describe all the methods that will be analyzed in this paper. These methods are meant for binary classification. Labeled data consists of l examples $\{x_i, y_i\}_{i=1}^l$ where the input patterns $x_i \in R^d$ are feature vectors representing web pages and the labels $y_i \in \{+1, -1\}$. Semi-supervised/transductive methods make use of unlabeled examples in addition; unlabeled data consists of u examples $\{x_i\}_{i=l+1}^{l+u}$. All the methods develop a linear classification function $w^T x$ with $\{x : w^T x > 0\}$ denoting the positive class region. Web page classification problems involve a large feature space (d is large); the input patterns are sparse, i.e., the fraction of non-zero elements in one x_i is small. Unlabeled data is always a random sample picked from the actual distribution to which the classifier will be applied; labeled data, on the other hand, may have a class proportion which is different from that in the actual distribution.

2.1 Supervised SVMs

Supervised SVMs make use of labeled data and optimize the regularized large margin loss function:

$$\min_w T_{\text{sup}} = \frac{\lambda}{2}\|w\|^2 + \frac{1}{l}\sum_{i=1}^{l} L(y_i, o_i) \qquad (1)$$

where $o_i = w^T x_i$. An example of the large margin loss function is the squared hinge loss $L(y, o) = \max(0, 1 - yo)^2$. Alternatively one can use the hinge loss: $L(y, o) = max(0, 1 - yo)$. All the experiments in this paper are done with the squared hinge loss. Irrespective of which loss is used there exist very efficient numerical methods [11, 13] for solving (1); the running time of these methods is linear in the number of examples.[1] On web page and text classification tasks the performance of the SVM classifier is quite steady over a large range of values of the regularization coefficient, λ. In all the experiments of the paper we use $\lambda = 1$.

2.2 Transductive SVMs

Transductive/Semi-Supervised SVMs (TSVMs) make effective use of unlabeled data to enhance the performance of SVMs. These methods perform very well on web page and text classification problems. Even with just a few labeled examples they can combine this labeled data with a large unlabeled set to attain a performance equal to that of a supervised SVM designed using a large labeled set. Through unlabeled examples many features which are even completely absent in the labeled set end up getting very good weights.

TSVMs are based on the *cluster (or, low density separation) assumption* which states that the decision boundary should not cross high density regions, but instead lie in low density regions. Joachims [10] gave the first effective TSVM formulation. A key ingredient of this formulation is that it

[1] Good software for supervised and transductive SVMs can be found in `http://svmlight.joachims.org/` and `http://vikas.sindhwani.org/svmlin.html`.

assumes f, the fraction of positive examples in the actual distribution is known and also the corresponding constraint is enforced in the formulation.[2] Joachims [10] solved the following problem in which, apart from w, the set of labels of unlabeled examples, $\{y_i\}_{i>l}$ are also variables:

$$\min_{w,\{y_i\}_{i>l}} T_{\text{sup}} + \frac{1}{u}\sum_{i>l} L(y_i, o_i) \quad s.t. \quad \frac{1}{u}\sum_{i>l}[y_i == 1] = f \tag{2}$$

where $[z]$ is 1 if z is true and 0 otherwise. Unlike (1), (2) is a not a convex minimization problem. In general, it is hard to solve. It has been pointed out [5] that the solution of (2) can get stuck in poor local minima. *Fortunately, this is not the case in linear classification problems involving large feature space, such as web page and text classification.* TSVMs are routinely used to solve applied problems [4, 3]. Alternative optimization iterations (fix w and optimize $\{y_i\}_{i>l}$, then fix $\{y_i\}_{i>l}$ and optimize w) are usually used to solve (2).

There also exist variations of the TSVM algorithm. For example the discrete variables in $\{y_i\}_{i>l}$ can be eliminated from (2) to get the following equivalent problem:

$$\min_w T_{\text{sup}}+\frac{1}{u}\sum_{i>l}\min\{L(1,o_i), L(-1,o_i)\} \; s.t. \; \frac{1}{u}\sum_{i>l}[o_i > 0] = f \tag{3}$$

Sigmoid smoothing or other methods can be applied to write $[o_i > 0]$ in a differentiable form. Then the solution can be approached via gradient based optimization of w. See [6, 13, 5] for methods of this kind. Most often this method yields a performance that is slightly better than (2).

2.3 Expectation Methods

Mann and McCallum [12] proposed the *Expectation Regularization (ER)* method in the context of MaxEnt models. The idea is to use expectation terms related to some domain knowledge to influence the training process. This can be easily done with SVM models too, like we do here. In our case the domain knowledge of interest is the fact that the fraction of positively classified examples in unlabeled data equals f. This fraction constraint can be used to influence the solution by including an additional regularization term in the objective function:

$$\min_w T_{\text{sup}} + \frac{\rho}{2}\left(\frac{1}{u}\sum_{i>l}[o_i > 0] - f\right)^2 \tag{4}$$

As mentioned earlier with respect to (3), sigmoid smoothing can be used to make the expectation regularization term to be differentiable so that gradient based numerical optimization techniques can be employed. In our implementation of ER we use $\rho = 50$ as the default value. Later we will also study the effect of varying ρ.

Note that in (4) the expectation constraint on the fraction of positive examples is only approximately enforced since it is only included as a regularizer term. If the domain knowledge says that the fraction constraint holds certainly then it may be better to force the constraint rather than adding a regularizer. This leads to a new method which we call as the *Expectation Constraints (EC)* method. The optimization problem to be solved is:

$$\min_w T_{\text{sup}} \quad s.t. \quad \frac{1}{u}\sum_{i>l}[o_i > 0] = f \tag{5}$$

Again, sigmoid smoothing can be used to write the fraction constraint function in a differentiable form. A suitable numerical method that can deal with equality constraints can be used to solve for w. In our implementation we use the Augmented Lagrangian method [1].

Let us introduce another expectation based baseline method that has been ignored in the literature. The method consists of taking the supervised SVM solution w (i.e., solution of (1)) and adding a threshold θ to the scoring function so that the fraction of unlabeled examples that satisfy $w^T x+\theta > 0$ equals f. The new classifier boundary is defined by $w^T x + \theta = 0$. We will refer to this method as SVMTh.

2.4 Relations between the methods

It is easy to see that ER and EC methods are closely related. In particular, when the parameter ρ in (4) is made very large then ER and EC are expected to show very similar behavior. SVMTh is quite different from EC although both methods enforce the fraction constraint; while EC tries to balance the minimization of T_{sup} with the fraction constraint, SVMTh simply adjusts the threshold to satisfy the fraction constraint without worrying about its effect on T_{sup}. If we compare (3) and (5) we see that TSVM has an extra term (loss on unlabeled data). This term turns out to be important as it helps TSVM to place the classifier boundary more precisely by making it pass through low density regions of unlabeled data; as we will see later the performance improvement due to it is large especially when the size of labeled data is small.

For MaxEnt models, Grandvalet and Bengio [9] suggested including an unlabeled loss term called *entropy regularization* that is similar in spirit to the unlabeled loss term in (3). However, their formulation does not include the fraction constraint. Mann and McCallum [12] conduct experiments to clearly point out the fact that without this constraint this method does not do well. They also show that, if the constraint is included (even as a regularizer term) then the performance is greatly enhanced.

3. DATASETS AND NOTATIONS

All the empirical analyses in this paper are done using the following five text classification datasets: `gcat`, `ccat`, `aut-avn`, `real-sim` and `earn`. The first four datasets are same as the ones used in [13]; `earn` is a binary dataset created from the *Reuters-21578* dataset[3] by taking examples in the *earn* class as positive and the remaining examples as negative. Key properties of the datasets are given in Table 1.

Each dataset is divided into three parts: labeled data, unlabeled data and test data. Unlabeled data and test data have the same class proportion; let f^{actual} denote the fraction of positive examples in test/unlabeled data. Labeled data can have a different class proportion depending on the experiment; note that the main aim of the paper is to find ways of dealing with a mismatch in class proportion between labeled data and the actual distribution. Let f^{lab} denote the fraction of positive examples in labeled data.

[2] We will refer to the constraint as the *fraction constraint.* Also, when applied in a discrete setting (e.g., the fraction constraint in (2)) it is assumed that appropriate rounding of f is allowed.

[3] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

Table 1: Properties of datasets. n **: number of examples,** d **: data dimensionality,** f^{actual} **: positive class ratio.**

Dataset	n	d	f^{actual}
gcat	23149	47236	0.30
ccat	23149	47236	0.46
aut-avn	71175	20707	0.65
real-sim	72309	20958	0.31
earn	9603	10783	0.30

Always f^{actual} is set to the fraction of positive examples in the original, full dataset. Given f^{lab} and the number of labeled examples, n^{lab}, we create a random division of the full dataset into test, labeled and unlabeled data as follows. First, 50% of the examples are randomly chosen in a class-stratified fashion and kept as testing data. Let $\mathcal{R}$ denote the set of remaining examples. We form labeled data by choosing $\lceil n^{lab} f^{lab} \rceil$ positive examples from $\mathcal{R}$ and $n^{lab} - \lceil n^{lab} f^{lab} \rceil$ negative examples from $\mathcal{R}$. Of the remaining examples, unlabeled data is randomly chosen to be the largest set obeying f^{actual}. Since the dataset sizes are large and n^{lab} is small, the size of unlabeled data is large and it is several times larger than n^{lab}.

Performance of the methods will be measured in terms of F score, which is the harmonic mean of precision and recall of the positive class.

To ease the understanding we use consistent notations in all the figures. The following colors will be used to denote the various methods:

- Black dotted: *LSVM* - SVM trained with labeled examples only;
- Red dotted: *FullSVM* - SVM trained using all unlabeled examples taking their labels as known;
- Blue: *TSVM* - Transductive SVM;
- Black: *SVMTh* - LSVM with threshold adjustment based on f;
- Green: *ER* - Expectation Regularization; and,
- Red: *EC* - Expectation Constraints.

In most figures f or f^{lab} is the horizontal axis. In such figures f^{actual} is shown as a vertical dotted line. In most figures the plots are averages over four random splits of a dataset into labeled, unlabeled and test data. Exceptions are Figures 3, 8, 9, and 10 where just one random split is used. In the figures *LabSize* and *LabFrac* denote, respectively, the size of labeled data, n^{lab} and the fraction of positive examples in labeled data, f^{lab}.

4. BASIC EXPERIMENTS

Before we go on to deal with differences in class proportions in labeled data and unlabeled/test data it is useful to give some basic experimental results that help understand the methods.

4.1 Learning Curves

Figure 1 shows the learning curves (variations of performance with n^{lab}) for various methods on four datasets. All methods use the value $f = f^{actual}$. TSVM gives a much stronger performance than other methods at small values of n^{lab}. Even at higher values of n^{lab} it is better than or competitive with others. Whenever TSVM does significantly better than SVMTh it is a clear indication of the important role played by the unlabeled loss term in (2). The magnitude of performance improvement given by TSVM over others depends on the dataset. On gcat and aut-avn TSVM is very much superior; on ccat it is quite better than others; and, on real-sim its performance matches that of SVMTh. After TSVM, SVMTh is the next best performer, followed by EC, ER and LSVM, in that order.

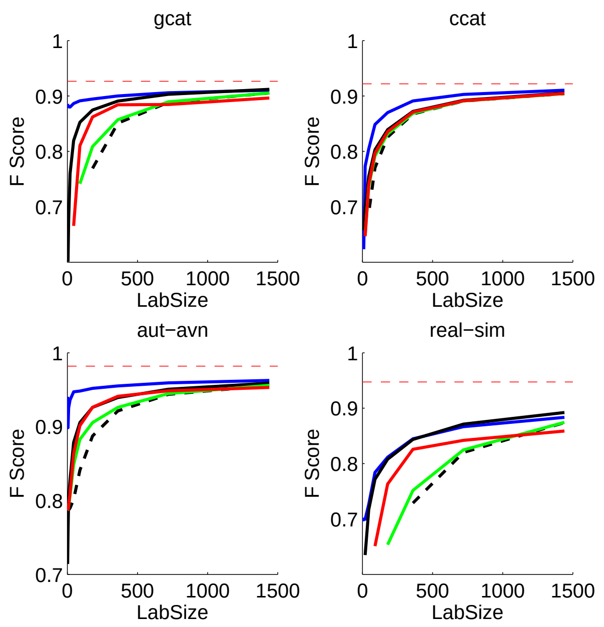

Figure 1: Learning curves for various datasets. All methods use $f = f^{actual}$.

4.2 Sensitivity Analysis

Since our aim is to deal with situations in which f^{actual} is unknown, it is useful to understand how the performance of each method is affected when f (the value of positive class fraction used in (1)-(5)) is changed away from f^{actual}. Figure 2 describes this sensitivity for $n^{lab} = 90$ and $n^{lab} = 1440$. We chose these n^{lab} values as they represent the rising and stable parts of the learning curves for most (method, dataset) combinations.

Let us first make some observations for $n^{lab} = 90$. All methods suffer when f is too much lower/higher than f^{actual}. If f is sufficiently far from f^{actual} then the performance can degrade to be even worse than that of LSVM. The fall in performance on the lower side is sharper since recall is badly affected when f is decreased from f^{actual} and this has an adverse effect on the F score.

TSVM and SVMTh attain their peak performance at f values that are close to f^{actual}. On the other hand, ER and EC attain their best performance at f values that are quite shifted away from f^{actual}. Consistently, this shifting away from f^{actual} is such that the fraction of the minority class is larger than its actual value in the unlabeled/test distribu-

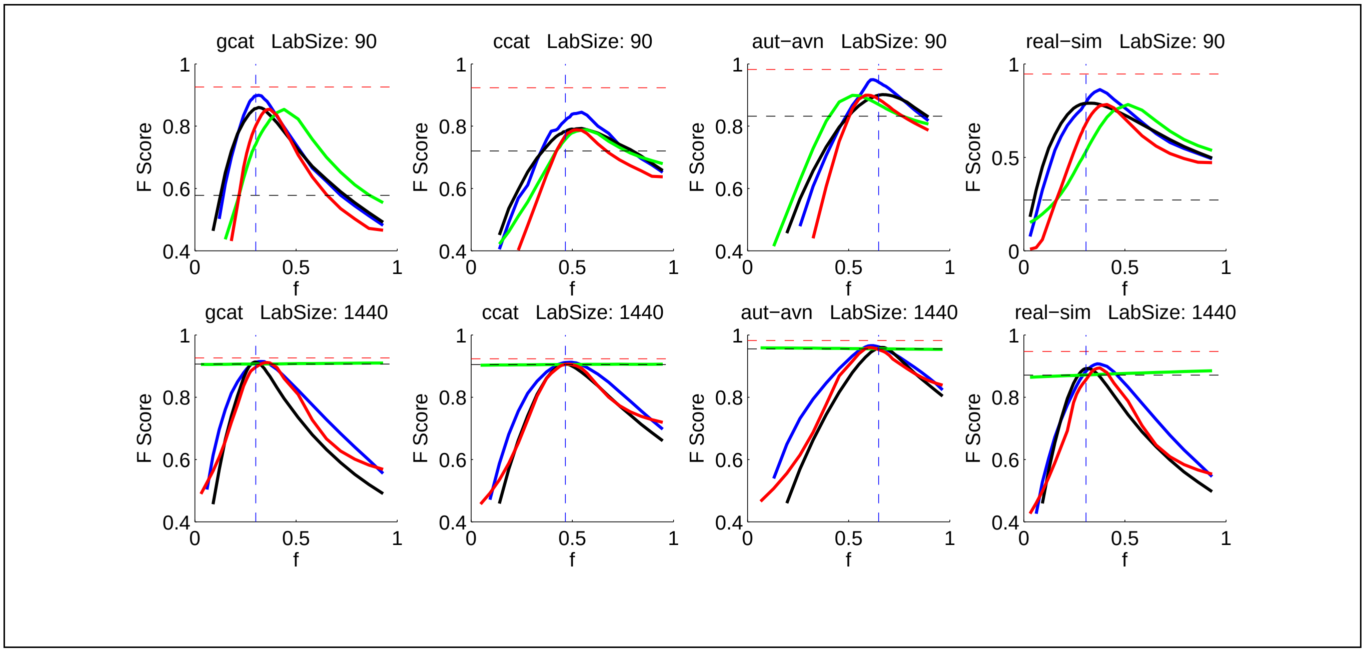

Figure 2: Sensitivity of methods with respect to f. The top row is for $n^{lab} = 90$ and the bottom row is for $n^{lab} = 1440$.

tion; in the top row of Figure 2 note the left shift of the peak for `aut-avn` and right shifts for all other datasets. This is due to two properties associated with LSVM, of which EC and ER are modifications. First, when n^{lab} is small the direction of the LSVM classifier makes a large angle with that of the ideal SVM classifier built using a large labeled training set. So the LSVM classifier boundary cuts through dense regions of the test set. Second, this cutting is more severe on the minority class examples due to lesser representation.[4] Thus, LSVM (which optimizes T_{sup}) ends up choosing a classifier boundary placement in which many test examples of the minority class are pushed to the wrong side. See the left bottom plot of Figure 8 (which comes later in the paper) to confirm this for the `gcat` dataset. Therefore, to pull the classifier towards satisfying the f^{actual} fraction and hence do well on F score, EC and ER methods need to use a value f (see (4) and (5)) that is higher (lower) than f^{actual} when the minority class is the positive (negative) class. The f value that is needed by ER to get the best F score on the test set is farther from f^{actual} than the f value needed for EC because ER only loosely enforces the f constraint. SVMTh behaves differently because it doesn't try to balance T_{sup} optimization and the fraction constraint; it simply enforces the fraction constraint in post-processing without worrying about what happens to T_{sup} in that step.

When n^{lab} is large the behavior is different. The bottom row of Figure 2 gives sensitivity for LabSize=1440. LSVM gives a very good performance since labeled data set is sufficiently large. The performance of ER is close to that of LSVM (see the green continuous and black dotted lines in the bottom row of Figure 2) because the ρ value (50) used in (4) is not strong enough to exert the fraction constraint. TSVM, SVMTh and EC suffer when f is moved well away from f^{actual} because they enforce the fraction constraint.

The observations made above imply that, for EC and ER it is a good idea to employ an f value that is different from f^{actual}, when n^{lab} is small. This observation is missing in previous works on ER [12]. If the class proportions in labeled, unlabeled, and test data are all matching and n^{lab} is not too small, it is then a good idea to use cross validation to choose f to optimize performance. If n^{lab} is not large the Leave-One-Out method can be used for cross validation. However we do not go ahead and demonstrate this in this paper since the main focus of the paper is to suggest ways of handling a mismatch between class proportions in labeled data and the actual distribution of interest.

Figure 3 gives the sensitivity plots for the `earn` dataset. The experimental setup is close to that used by Zhang and Oles [15] who used the setup to point out (wrongly) that TSVM does not work well. Zhang and Oles [15] possibly optimized the TSVM objective function in (2) without the fraction constraint. Figure 3 explains what can go wrong if that is done. From the right side plot of Figure 3 it is clear that the least value of the objective function occurs at $f = 0$ where the performance is very poor. This clearly shows the important role played by the fraction constraint in TSVM. Recall a similar comment that we made earlier at the end of subsection 2.4, with respect to MaxEnt models and entropy regularization.

5. DISTORTION IN CLASS PROPORTION

The results of section 4 concern the case in which labeled data and test/unlabeled data have the same class proportion. We now consider situations in which the class proportions in the two sets are different. Sometimes it happens that even though there is a difference in those class proportions,

[4] This can be explained as follows. Take the case where the positive class is the minority class. Since the positive class has a smaller number of loss terms (see (1)) the classifier boundary of LSVM is placed closer to the labeled examples of the positive class in order to reduce the total loss on the labeled examples of the negative class. Therefore the fraction of examples classified by LSVM as positive is less than f^{actual}.

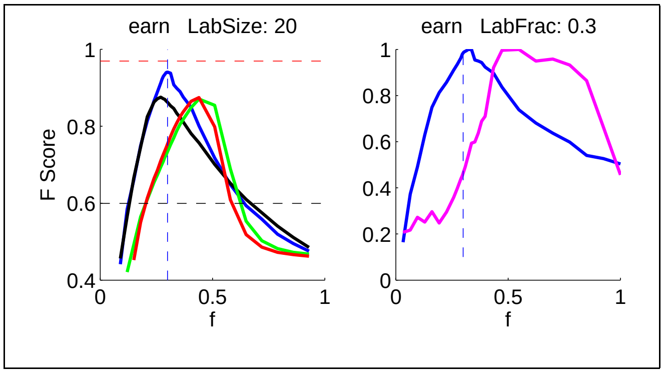

Figure 3: Earn dataset: sensitivities of the methods with respect to f with just 20 labeled examples. The left plot shows performance of various methods. The right plot shows TSVM performance (blue) and TSVM objective function in equation (2) (magenta) (both are normalized to max value of 1 to show them on the same plot). When f^{actual} is known TSVM gives an F score of 0.93.

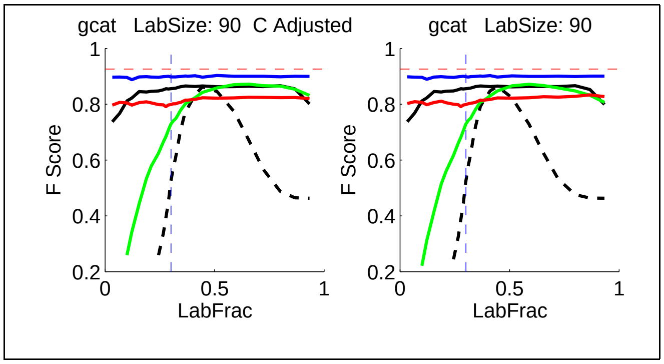

Figure 4: Demonstration that cost adjusted labeled loss does not change performance significantly.

f^{actual} may actually be known. Such knowledge helps the methods to achieve a good performance. In subsection 5.1 we take up this case. In subsection 5.2 we deal with the more difficult case in which f^{actual} is unknown.

5.1 Case 1: f^{actual} is Known

Since we know f^{actual} as well as the number of positive and negative labeled examples in the labeled set, it is appropriate to modify the SVM objective function $T_{\rm sup}$ so that the labeled loss term can be viewed as if it is the mean loss of labeled data picked (with replacement) from the actual distribution of interest. To do this we change $T_{\rm sup}$ to $T_{\rm sup}^{\rm Adj}$ by introducing γ, a relative weighting parameter for positive labeled examples. $T_{\rm sup}^{\rm Adj}$ is given by

$$T_{\rm sup}^{\rm Adj} = \frac{\lambda}{2}\|w\|^2 + \frac{1}{N}(\gamma \sum_{1\le i\le l, y_i=1} L(y_i, o_i) + \sum_{1\le i\le l, y_i=-1} L(y_i, o_i)) \tag{6}$$

where γ is chosen so that $\gamma\frac{f^{lab}}{(1-f^{lab})} = \frac{f^{actual}}{1-f^{actual}}$, $f^{lab} = \frac{n_p}{n_p+n_n}$, n_p and n_n are the number of positive and negative examples in labeled data, and $N = n_p\gamma + n_n$ is the normalizer chosen to make sure that the labeled loss term is the mean loss. Note that when $f^{lab} < f^{actual}$, γ is bigger than 1, and so the weight on the positive labeled examples is increased. When $f^{lab} > f^{actual}$, γ is smaller than 1. All the methods can be appropriately modified to use $T_{\rm sup}^{\rm Adj}$ instead of $T_{\rm sup}$. However, we found that the performance of the methods are little affected when $T_{\rm sup}^{\rm Adj}$ is used instead of $T_{\rm sup}$. Figure 4 compares the performance for one dataset, `gcat`. Similar behavior is seen on other datasets.

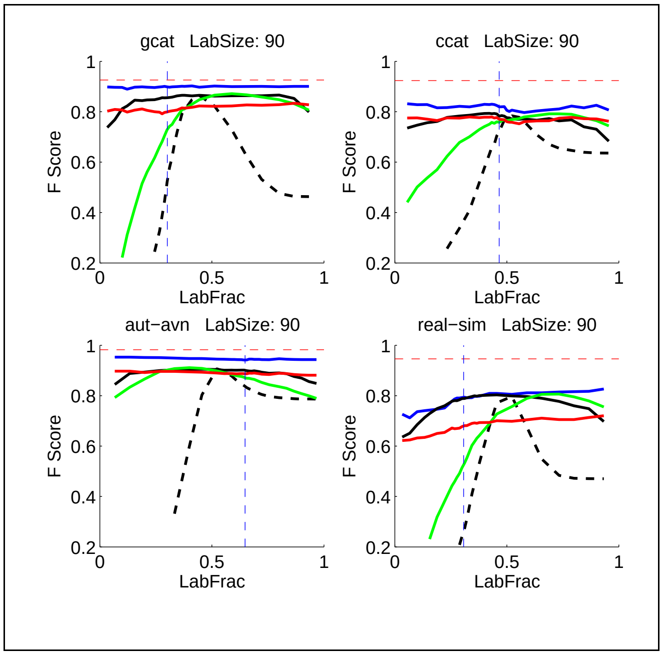

Figure 5: Variation of the performance of methods with f^{lab} (LabelFrac) when f^{actual} is known and the methods use $f = f^{actual}$.

Figure 5 compares all methods on four datasets when $f = f^{actual}$ is used by the methods and f^{lab} is varied over a range of values from 0 to 1. Interestingly, TSVM, SVMTh and EC are only mildly affected by changes in f^{lab}. This is very much due to f^{actual} being known and used well by the methods via the fraction constraint with $f = f^{actual}$. On the other hand LSVM is badly affected when f^{lab} is moved away from f^{actual}. ER suffers when $f^{lab} < f^{actual}$. The penalty weight, ρ used in (4) plays a key role in ER. Figure 6 shows the performance variation for a range of ρ values. When ρ is small the performance of ER is close to that of LSVM. For large ρ its performance is close to that of EC. Both these observations are along expected lines. For different values of f^{lab} the optimal ρ is different.

5.2 Case 2: f^{actual} is Unknown

We propose and explore some methods for estimating f^{actual} in the presence of large variations in f^{lab}. Note that, even when labeled data is large, it cannot be used alone to get an estimate of f^{actual} since the class proportion in that set are distorted. We explore three methods for estimating f^{actual}. Let us refer to an estimate of f^{actual} as f_{est}^{actual}.

Estimation Method 1. Simply take f_{est}^{actual} to be the fraction of positive examples in labeled data. This is just a baseline method.

The next two methods make use of unlabeled data together with labeled data.

Estimation Method 2. Given labeled data we can solve (1) and obtain the LSVM solution w. From the good performance of SVMTh over a large range of f^{lab} values in Figure

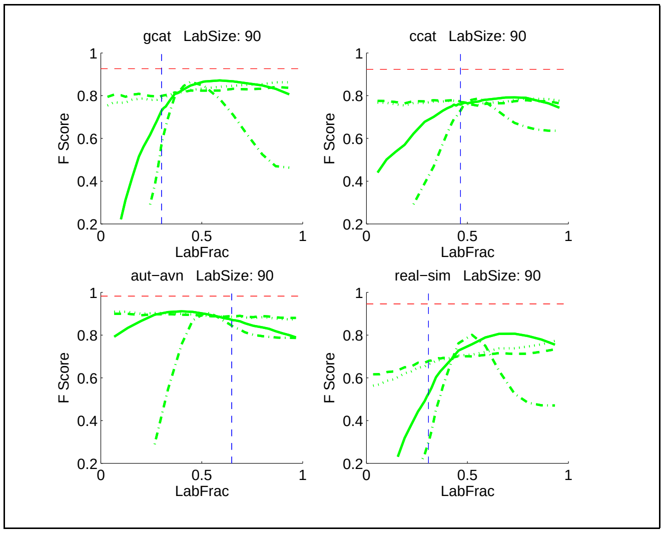

Figure 6: Variation of ER performance with f^{lab} (LabelFrac) for various ρ values. Dashdot: $\rho = 5$ (gives performance close to LSVM); Continuous: $\rho = 50$ (same as in Figure 5); Dotted: $\rho = 500$; Dashed: $\rho = 5000$. ER with $\rho \geq 500$ yields a performance close to that of EC.

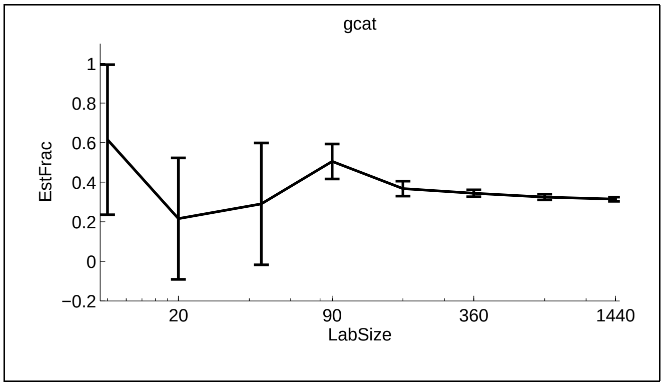

Figure 7: Estimation Method 2. Mean (with error bars) of f_{est}^{actual} versus n^{lab} for `gcat`. $f^{actual} = 0.3011$ for `gcat`.

5 we know that the classifier direction corresponding to w is good. It is appropriate to model the classifier scoring function, $o = w^T x$ applied on unlabeled data, $\{o_i = w^T x_i\}_{i>l}$ as a mixture of two Gaussians:

$$p(o_i) = \beta_1 g(o_i; \mu_1, \sigma_1) + \beta_2 g(o_i; \mu_2, \sigma_2) \qquad (7)$$

where $g(\cdot; \mu_k, \sigma_k)$ is the univariate Gaussian density function with mean μ_k and standard deviation σ_k, and β_1, β_2 are (non-negative) mixing proportion values satisfying $\beta_1 + \beta_2 = 1$. By fitting this model to the data $\{o_i\}_{i>l}$ to maximize likelihood we can get the parameter values β_1, β_2, μ_1, μ_2, σ_1 and σ_2. The Gaussian with the larger μ_k represents the positive class; so, the corresponding β_k can be taken as f_{est}^{actual}, an estimate of the positive class fraction.

Figure 8 helps us understand the usefulness of this method. When the number of labeled examples is small, w, the solution of (1), makes a large angle with $w^\star$, the solution associated with the ideal SVM classifier built using a large labeled training set having the actual class proportion; so the $\{o_i = w^T x_i\}_{i>l}$ distribution is not a clean mixture of two Gaussians, and the estimate f_{est}^{actual} is also poor. On the other hand, when the number of labeled examples is large (closer to the higher side of the learning curve) this estimation method performs much better.

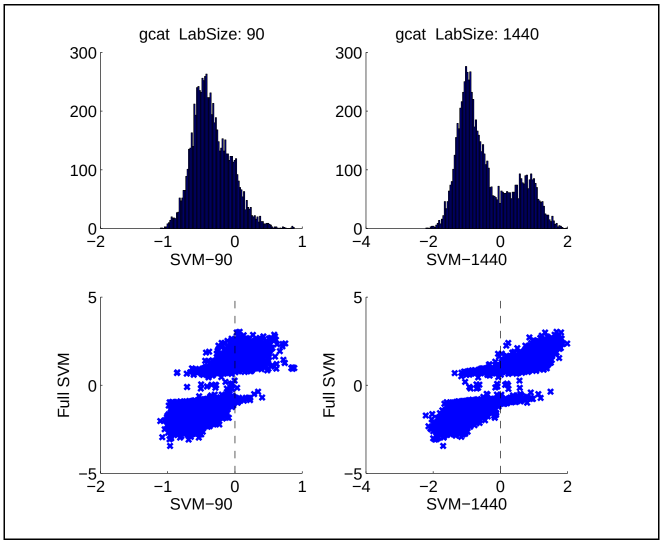

Figure 8: Understanding Estimation Method 2. SVM-N denotes LSVM output for N labeled examples and FullSVM is output of SVM corresponding to a very large labeled set.

Figure 7 plots the mean (with error bars) of f_{est}^{actual} (computed using 10 random splits of `gcat`) as n^{lab} is varied. Such a plot can help one decide when this estimation method is useful.

Estimation Method 3. In this method we explore to see if some property associated with the solution of a method can be used to find f_{est}^{actual}. Let us take TSVM. Figure 9 gives, for `gcat` and `ccat`, the variation of the TSVM objective function (see (2)), the Labeled Loss (the second term of T_{sup} in (1)), as well as the F score with respect to f used in (2). The behavior in other datasets is similar. Clearly the TSVM objective function is not useful for tuning f. Gärtner et al [8] suggest to minimize the TSVM objective function to tune class proportions; but Figure 9 clearly points out that it is not the right thing to do. The plots also show the goodness of minimizing the Labeled Loss as a way of obtaining f_{est}^{actual}.[5] A rough explanation for this goodness is as follows. Consider (2), the optimization problem associated with TSVM. End f values (near 0 and 1) put heavy pressure on containing all examples on one side of the classifier boundary. With such pressure Labeled Loss becomes large for such f values. Near $f = f^{actual}$ minimizing Labeled Loss, Unlabeled Loss and satisfying the fraction constraint are all consistent, so Labeled Loss achieves small values there.

We also tried the same estimation method (i.e., minimize labeled loss along the trajectory of solutions defined by varying f) on the expectation methods. But it did not work well. Figure 10 gives, for EC and `gcat` and `ccat`, the variation of labeled loss as well as the F score as a function of f. The minimizer of the labeled loss does not coincide quite well with the maximizer of F score. Similar behavior is observed

[5] To avoid confusion we note that (2) is always the optimization problem that is solved to get the TSVM solution; minimizing the Labeled Loss that we are doing here is for tuning f at a higher level treating f as a hyperparameter.

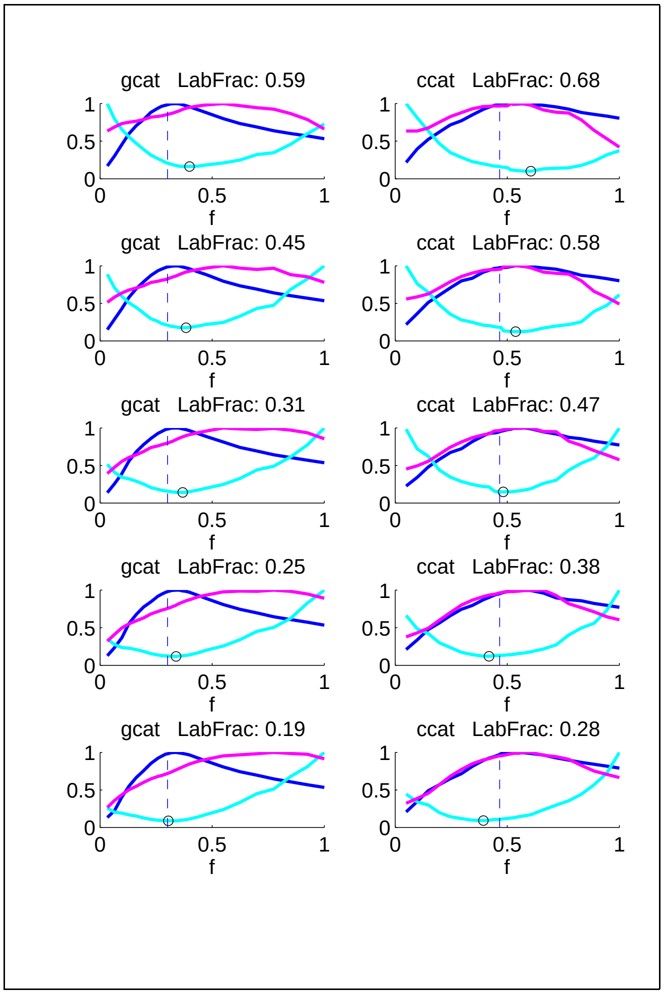

Figure 9: Variation of performance (blue), TSVM objective function in (2) (magenta) and Labeled Loss (the second term of T_{sup} in (1)) (cyan) with respect to f, for `gcat` and `ccat`. The minimizer of Labeled Loss is marked by a black circle. The rows correspond to different fraction distortions. The middle row corresponds to $f^{lab} = f^{actual}$. The bottom two rows correspond to $f^{lab} = 0.8f^{actual}$ and $f^{lab} = 0.6f^{actual}$. The top two rows correspond to $(1-f^{lab}) = 0.8(1-f^{actual})$ and $(1-f^{lab}) = 0.6(1-f^{actual})$.

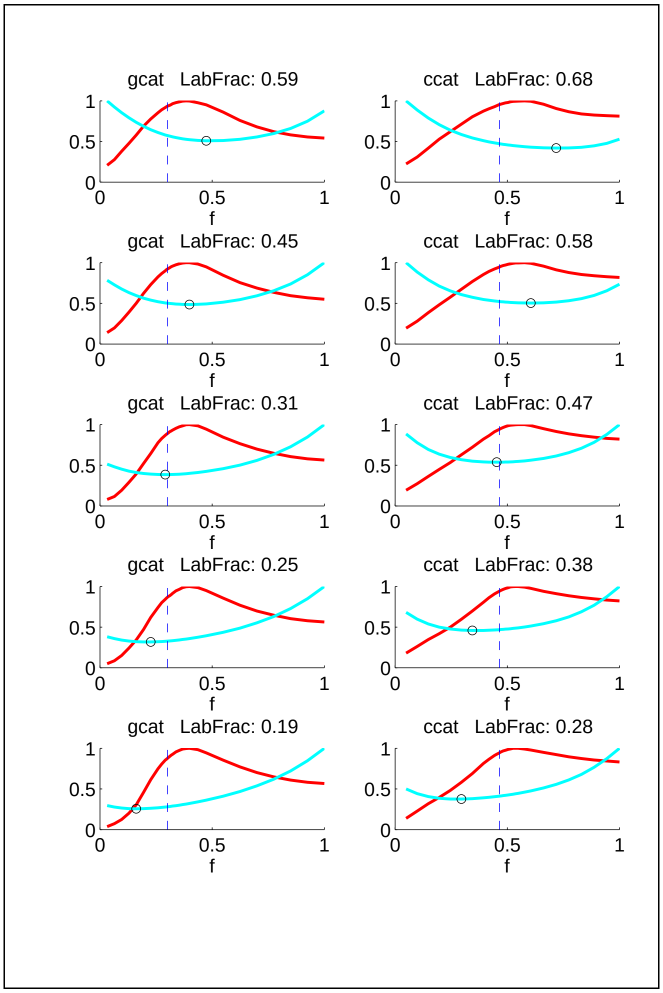

Figure 10: EC: Variation of performance (red) and Labeled Loss (the second term of T_{sup} in (1)) (cyan) with respect to f, for `gcat` and `ccat`. The minimizer of Labeled Loss is marked by a black circle. The rows correspond to different fraction distortions. The middle row corresponds to $f^{lab} = f^{actual}$. The bottom two rows correspond to $f^{lab} = 0.8f^{actual}$ and $f^{lab} = 0.6f^{actual}$. The top two rows correspond to $(1-f^{lab}) = 0.8(1-f^{actual})$ and $(1-f^{lab}) = 0.6(1-f^{actual})$.

for SVMTh and ER. For $n^{lab} = 90$ Figure 11 shows the performance of Estimation Method 3 with EC and SVMTh.

Figure 12 compares the three estimation methods as applied to TSVM. For $n^{lab} = 90$ Estimation Method 2 doesn't do even as well as Estimation Method 1. Estimation Method 3 is very good in a large range of f^{lab} values containing f^{actual}. It suffers only at extreme f^{lab} values. When n^{lab} is large, e.g., 1440, Estimation Method 2 performs very well. It is appropriate to switch between the two estimation methods depending on how large n^{lab} is. Going by what we saw in Figure 7, we can use the steadiness of f^{actual}_{est} given by Estimation Method 2 over a range of n^{lab} values to decide when to switch to it. Of course, n^{lab} has to be reasonably big to notice a clear steady pattern. Until that size is reached it is best to use Estimation Method 3.

When n^{lab} is large LSVM itself does quite well; since it is not dependent on f it becomes the preferred method for that case. Therefore, the value of a fraction estimation method should be determined by how well it performs when labeled data is small. From that viewpoint Estimation Method 3 (as applied to TSVM) is most valuable.

Of all (Method, Estimation Method) combinations (TSVM, Estimation Method 3) is the one that takes the most computing time; even for that combination, the computation time for one run on any of our five datasets is less than two minutes on a 3 GHz machine.

6. RELATED WORK

Since the first crucial paper of Joachims [10] several works (for a sample see [15, 7, 14, 6, 13, 5]) have been published extending and exploring various aspects of the first TSVM model. Of these, the works of Chen, Wang and Dong [7] and Wang and Huang [14] are the most relevant to our work since they address the problem of mismatch in class proportions. The main drawback of these methods is that they make adhoc labeling of the unlabeled examples during the training process in an active self-learning mode and are not cleanly tied to a well-defined problem formulation. Due to this they significantly deviate from the TSVM formulation and hence perform quite worse than TSVMs for the normal situation in which labeled data and the actual distribution are indeed well matched in terms of class proportions. Also, the methods have been demonstrated only in situations where LSVM itself gives decent performance, which is not true when there are large distortions in class proportions.

Mann and McCallum [12] empirically analyze ER in binary, multi-class and structured output settings in the context of MaxEnt models. ER's working and performance in SVMs and MaxEnt models are similar. Mann and McCallum [12] do a brief study of the sensitivity of ER's performance with respect to class proportions. But they only focus on the case of distortion towards uniform class proportions; in the binary case this corresponds to changing f from f^{actual} to 0.5. Unfortunately their study is done only on one multi-class dataset. As we saw in section 4.2, ER shows interesting behavior in the binary case; in fact ER improves in performance when f is moved towards 0.5. SVMTh is an important baseline method that is completely missed in [12].

There is a building literature on dealing with differences in train and test distributions referred to as *covariate shift* (see [2] for instance), but such papers introduce and analyze new formulations and do not help to modify TSVM or the expectation methods while keeping their spirit in tact.

7. CONCLUSION

The main contributions of this paper are: (i) empirical analysis of TSVM and expectation methods and their sensitivity with respect to label proportions; and, (ii) proposal and evaluation of new methods for dealing with mismatches in label proportions between labeled and test sets. We have also done preliminary experiments to verify the ideas on Least squares and MaxEnt models of binary classification and observed behavior similar to SVM loss. With some care all the ideas can also be extended to the multi-class setting. The ideas and results of this paper are mainly for web page and text classification. More work is needed to see if they hold on other types of problems.

8. REFERENCES

[1] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[2] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *ICML*, pages 81–88, 2007.

[3] L. Bruzzone, M. Chi, and M. Marconcini. A novel transductive SVM for semisupervised classification of remote-sensing images. *IEEE-GRSS*, 3363–3373, 2006.

[4] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2006.

[5] O. Chapelle, V. Sindhwani, and S. S. Keerthi. Optimization techniques for semi-supervised support vector machines. *JMLR*, 9:203–233, 2008.

[6] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *AISTATS*, pages 57–64, 2005.

[7] Y. Chen, G. Wang, and S. Dong. Learning with progressive transductive support vector machine. *Pattern Recognition Letters*, 24:1845–1855, 2003.

[8] T. Gärtner, Q. V. Le, S. Burton, A. J. Smola, and S. V. N. Vishwanathan. Large-scale multiclass transduction. In *NIPS*, 2005.

[9] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *NIPS*, 2004.

[10] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, 1999.

[11] T. Joachims. Training linear SVMs in linear time. In *KDD*, 2006.

[12] G. S. Mann and A. McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *JMLR*, 11:955–984, 2010.

[13] V. Sindhwani and S. S. Keerthi. Large scale semi-supervised linear SVMs. In *SIGIR*, pages 477–484, 2006.

[14] Y. Wang and S.-T. Huang. Training TSVM with the proper number of positive samples. *Pattern Recognition Letters*, 26:2187–2194, 2005.

[15] T. Zhang and F. J. Oles. A probability analysis on the value of unlabeled data for classification problems. In *ICML*, pages 1191–1198, 2000.

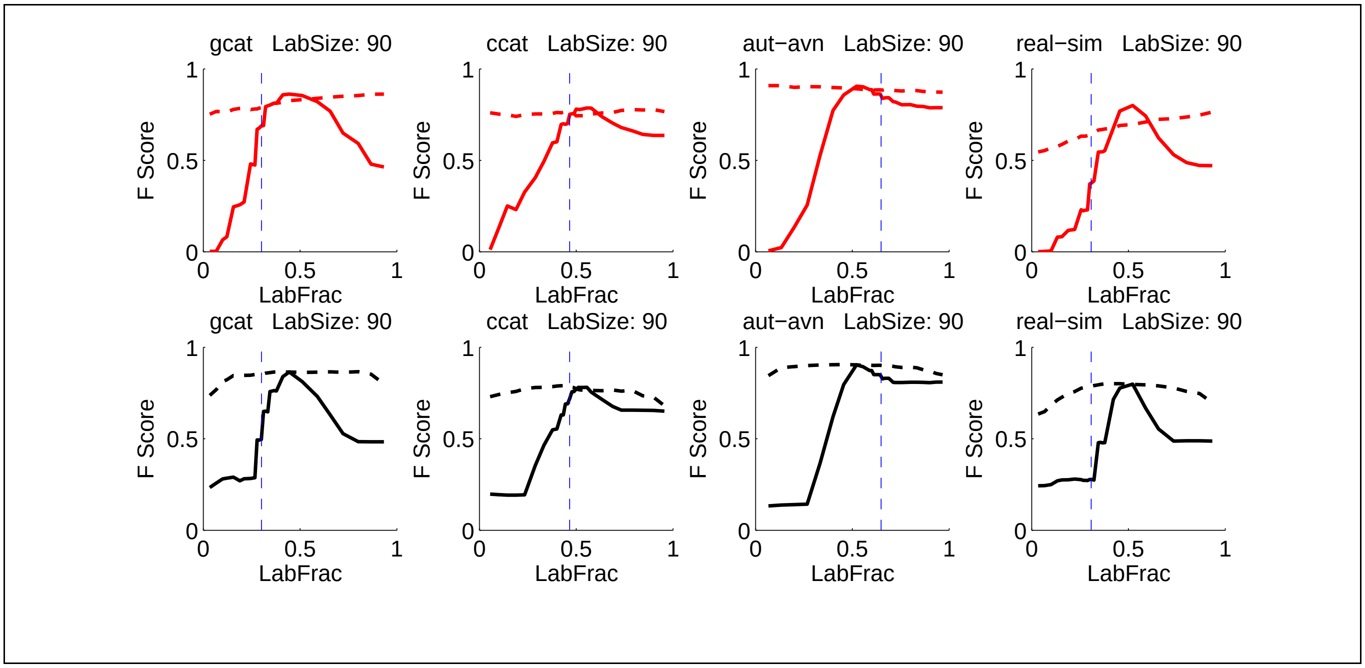

Figure 11: Performance of Estimation method 3 as applied to EC (top row) and SVMTh (bottom row). Estimation Method 3: Continuous. Dashed: Use $f = f^{actual}$ (Upper baseline)

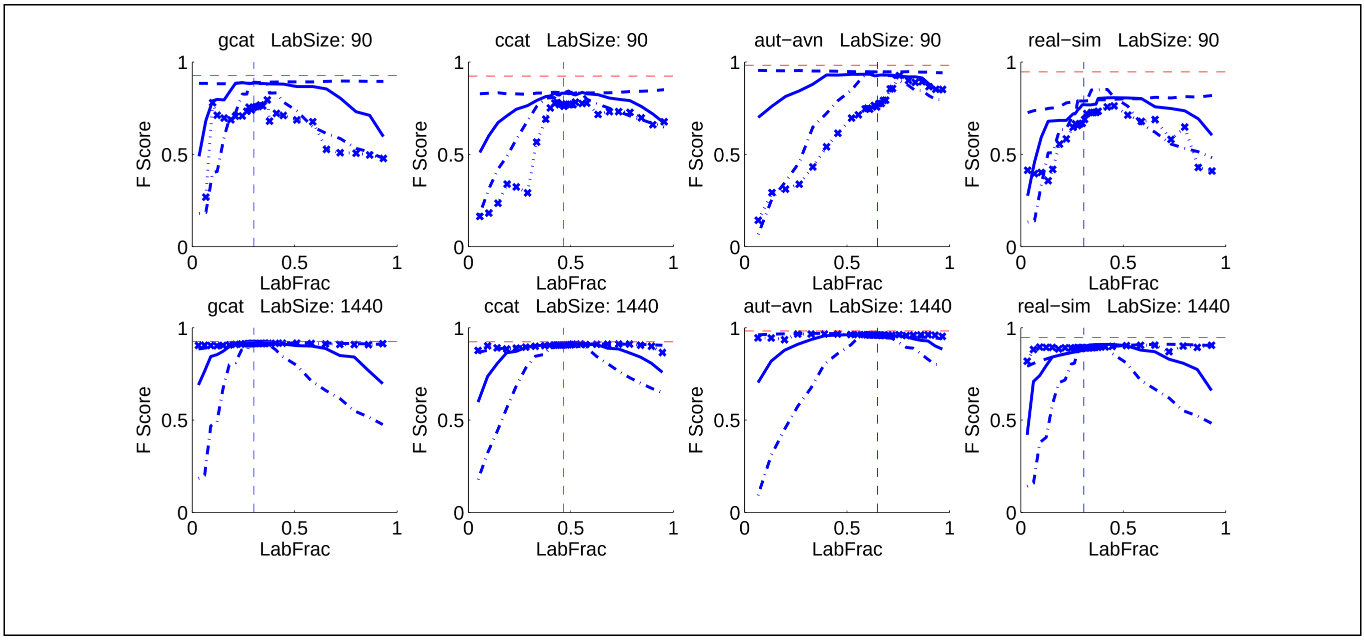

Figure 12: Comparison of f^{actual} estimation methods as applied to TSVM. Estimation Method 1: Dashdot. Estimation Method 2: Dotted with x. Estimation Method 3: Continuous. Dashed: Use $f = f^{actual}$ (Upper baseline)