SVM Classifier Estimation from Group Probabilities

Stefan Rueping STEFAN.RUEPING@IAIS.FRAUNHOFER.DE
Fraunhofer IAIS, Schloss Birlinghoven, 53754 St. Augustin, Germany

Abstract

A learning problem that has only recently gained attention in the machine learning community is that of learning a classifier from group probabilities. It is a learning task that lies somewhere between the well-known tasks of supervised and unsupervised learning, in the sense that for a set of observations we do not know the labels, but for some groups of observations, the frequency distribution of the label is known. This learning problem has important practical applications, for example in privacy-preserving data mining. This paper presents an approach to learn a classifier from group probabilities based on support vector regression and the idea of inverting a classifier calibration process. A detailed analysis will show that this new approach outperforms existing approaches.

1. Introduction

A learning problem that has only recently gained attention in the machine learning community is that of learning a classifier from group probabilities (Kueck & de Freitas, 2005; Quadrianto et al., 2008; 2009). It is a learning task that lies somewhere between the well-known tasks of supervised and unsupervised learning, in the sense that for a set of observations we do not know the labels, but for some groups of observations, the frequency distribution of the label in the groups is known (see Figure 1). The goal is, from this information alone, to estimate a classifier that works well on the labeled data.

As noted in (Quadrianto et al., 2009), this learning problem has received surprisingly little attention so far, even though it has many interesting applications. One of the most natural applications comes in analyzing the outcomes of political elections, where the population of all voters in an electoral district is known, but only the total number of votes per party in each district is revealed. However, from an analysis of this data, e.g. the dependence of votes on variables such as income or household types, can show up interesting connections, and may be used to uncover election fraud when outliers from this model are uncovered.

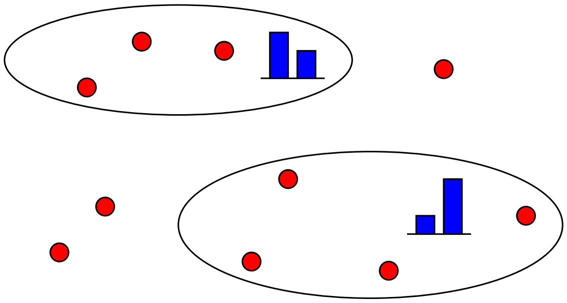

Figure 1. Classifier estimation from group probabilities, cf. (Quadrianto et al., 2009).

Another interesting application comes from privacy-preserving data mining. In some settings, revealing the label of the observations may imply serious privacy concerns. For example, in medical research following the outbreak patterns of a new type of influenza virus is an important task, but revealing which patient actually got infected with the new virus may be viewed as information that is confidential between him and his treating physician. However, outbreak frequencies in certain risk groups are usually anonymous data, such that they are not sensitive information.

To give another example, in fraud detection it is common practice to apply machine learning to fraud / non-fraud data. While this seems to be straight-forward, in practice labeling some person as a fraudster has serious legal implications, in particular when this data is given to a third person for analysis. Even if it is clear that a person has not paid for some merchandise or service he received, there may be perfectly legal rea-

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010. Copyright 2010 by the author(s)/owner(s).

sons not to do so. Hence, it may only be legally safe to label someone as a fraudster if he was convicted by a court of law.

In the end, storing only risk probabilities over small groups of people may be the legally advisable way in these cases. To put it more plainly, the difference between fraud labels for observations and group probabilities in this case translate to the difference between the statements *this person is a fraudster* and the much less aggressive *in this group of 5 people the risk probability is 20%.*

In this paper, we will present an algorithm for learning a classifier from group probabilities, which is based on ideas from support vector regression and classifier calibration.

The remainder of this paper is structured as follows: in the following section related work is discussed, before Section 3 introduces the new algorithm, which will be called Inverse Calibration. Section 4 empirically compares the new algorithm to existing approaches. Finally, Section 5 concludes.

2. Related Work

In this section, we will first present related work for learning a classifier from group probabilities. We will also present existing work on the related task of estimating conditional probabilities from a given classifier, which will be relevant later on.

2.1. Estimation of a Classifier from Group Probabilities

The task of estimating a classifier from set probabilities describes the setting, where groups of unlabeled observations are given and the only information about the distribution of the labels comes from the frequencies of the labels in each group.

A method for estimating a classifier from group probabilities, the Mean Map method, has been proposed in (Quadrianto et al., 2009). The method is based on modeling the conditional class probability $p(y|x, \theta)$ using conditional exponential models:

$$p(y|x, \theta) = exp((\Phi(x, y)\theta) - g(\theta|x))$$

with a normalizing function g. The parameter θ of the model is estimated by taking the known observation means of the groups and inferring from them and the known class frequencies per group the example means given classes.

(Quadrianto et al., 2009) defines the learning problem with a transductive component as well, where the distribution of the labels in the test set is known. However, in this paper we do not assume that this information is known.

Algorithmically, the Mean Map method boils down to solving a convex optimization problem. While the method is defined for joint kernels on $X \times Y$, a special case exists for the case of binary classification where $k((x, y), (x', y')) = yy'k'(x, x')$. Since in this paper we are only interested in binary classification, this variant is used in the experiments.

The paper (Quadrianto et al., 2009) also gives a detailed overview of other related techniques, such as methods based on kernel density estimation, discriminative sorting, or generative models and MCMC (Kueck & de Freitas, 2005). However, it was found that none of these methods can outperform their Mean Map method, and hence they are not investigated in detail in this paper.

2.2. Estimating Conditional Probabilities

Given a binary classification task described by an unknown probability distribution $P(X, Y)$ on an input space X and a set of labels $Y = \{-1, 1\}$, a probabilistic classifier is a function $f_{prob} : X \to [0, 1]$ that returns an estimate of the conditional class probability, i.e.

$$f_{prob}(x) \approx P(Y = 1|x).$$

A standard approach to probabilistic classification is to calibrate a numerical classifier. That is, for a numerical classification function

$$cl(x) = sign(f_{num}(x))$$

the task is to find an appropriate scaling function $\sigma : \mathcal{R} \to [0, 1]$ such that

$$\sigma(f_{num}(x)) \approx P(Y = 1|x)$$

holds.

A comparative study by (Niculescu-Mizil & Caruana, 2005) revealed that Platt Calibration (Platt, 1999) and Isotonic Regression (Zadrozny & Elkan, 2002) are the most effective probabilistic calibration techniques for a wide range of classifiers.

Platt Calibration (Platt, 1999) was originally introduced for scaling Support Vector Machine (SVM) outputs, but has been shown to be efficient for many other numerical decision functions as well (Niculescu-Mizil & Caruana, 2005). Based on an empirical analysis of the distribution of SVM decision function values, Platt suggests to use a scaling function of the form

$$\sigma_{Platt}(f(x)) = \frac{1}{1 + exp(-Af(x) + B)}.$$

The parameters A and B are optimized using gradient descent to minimize the cross-entropy error

Isotonic Regression (Zadrozny & Elkan, 2002) assumes a monotonic dependency between the decision function and the conditional class probabilities and finds a piecewise constant, monotonic scaling function that minimizes the quadratic loss by making use of the pair-adjacent violators algorithm (Ayer et al., 1955).

Other probabilistic calibration techniques have also been proposed in the literature, for example Softmax Scaling, Binning, or calibration by Gaussian modeling of the decision function.

A finding that holds particular significance for our approach is that often even very trivial calibration techniques without an elaborate parameter estimation procedure can produce reasonably good probability estimates (Rüping, 2004).

3. The Algorithm

Problem formulation: Let $P(X, Y)$ be a fixed, but unknown probability distribution and let $(x_1, y_1), \ldots, (x_n, y_n) \subset X \times \{-1, 1\}$ be drawn i.i.d. from P. Assume we are given m subsets of $(x_1, \ldots, x_n)$, where we identify the k-th subset by the set of its indices $S_k = \{i_{k,1}, \ldots, i_{k,|S_k|}\}$. Let $p_k = |\{i \in S_k : y_i = 1\}|/|S_k|$ be the estimate of the conditional class probability $P(Y = 1|S_k)$. The goal is to find a classifier $f : X \to \{-1, 1\}$ with minimal error according to P, given only the $x_1, \ldots, x_n$, the $S_1, \ldots, S_m$ and the $p_1, \ldots, p_m$ are known.

Inversion of Class Probability Estimation: Our approach is to invert the process of estimating conditional class probabilities from Section 2.2. In conditional class probability estimation a classifier f is trained first, and then a scaling function σ is fitted such that $\sigma(f(x))$ is a good estimate of $P(Y = 1|x)$.

We instead start with given probability estimates p, fix a scaling function σ, apply this inverse scaling function and train an SVM to predict the values $\sigma^{-1}(p)$. This approach is partly motivated by (Rüping, 2004), which shows that even very trivial probabilistic scaling functions without an elaborate parameter fitting procedure can give reasonable estimates of p.

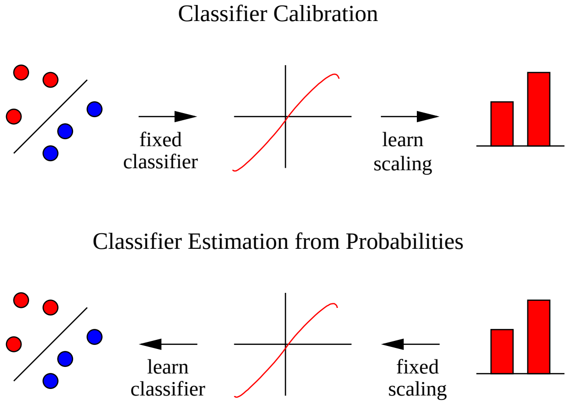

Figure 2. Classifier estimation from group probabilities by inverting the classifier calibration process.

In our algorithm, we use the scaling function

$$p = \sigma(y) = \frac{1}{1 + exp(-y)}$$

which can be seen as a special case of Platt scaling (Platt, 1999) with $A = 1$ and $B = 0$. In particular, we will make use of it's inverse

$$y = \sigma^{-1}(p) = -\log(\frac{1}{p} - 1).$$

To simplify notation, in the following, we use p and y, or p_i and y_i, interchangeably and always imply $p = \sigma(y)$. In order to avoid undefined values of y, we clip p to the interval $[\varepsilon, 1-\varepsilon]$, where ε is a parameter defining the minimum required precision of the estimate. A reasonable choice is to take $\varepsilon = 1/\#$examples.

Our goal is to estimate a linear classification function $f(x) = wx + b$. In order to classify well, we require it to predict y well, which in turn implies that $\sigma \circ f$ is a good estimate of p. However, in our problem we are not given estimates of p for every observation x, but only for sets S of observations. In particular, depending on the construction of S, the optimal class probability estimates of the single observations in S may very much vary around their average p. To circumvent this problem, we only require that f predicts y well on average:

$$\forall i : \frac{1}{|S_i|} \sum_{j \in S_i} (wx_j + b) \approx y_i.$$

We can now formally define the learning task formally in the spirit of Support Vector Regression (Vapnik, 1998), which will in particular allow a kernelization later.

Primal Problem:

$$\begin{aligned}
\frac{1}{2}||w||^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) &\quad\rightarrow\quad \min \\
&s.t. \\
\forall_{i=1}^{m} : \xi_i, \xi_i^* &\quad\geq\quad 0 \\
\forall_{i=1}^{m} : \frac{1}{|S_i|}\sum_{j\in S_i}(wx_j + b) &\quad\geq\quad y_i - \varepsilon_i - \xi_i \\
\forall_{i=1}^{m} : \frac{1}{|S_i|}\sum_{j\in S_i}(wx_j + b) &\quad\leq\quad y_i + \varepsilon_i + \xi_i^*
\end{aligned}$$

The formulation requires to both minimize the complexity of the model and to try to keep the class probability estimate of S_i close to p_i, where the maximum tolerable error is defined by ε_i. Note that to keep the optimization problem in the form of a quadratic problem, the error is defined with respect to y_i and not p_i, which is the true target. We will fix this inconsistency in the following.

Usually in Support Vector Regression one would use a constant value for the required precision, i.e. $\forall i : \varepsilon_i = \varepsilon$. However, in this case the goal is not to estimate y_i but p_i, such that instead of setting a precision limit on y we actually require

$$\begin{aligned}
& p_i - \varepsilon \leq \sigma(y_i) \leq p_i + \varepsilon \\
\Leftrightarrow\quad & p_i - \varepsilon \leq \frac{1}{1 + exp(-y_i)} \leq p_i + \varepsilon \\
\Leftrightarrow\quad & -\log(\frac{1}{p_i - \varepsilon} - 1) \leq y_i \leq -\log(\frac{1}{p_i + \varepsilon} - 1)
\end{aligned}$$

A Taylor expansion of order 1 of the function $p \mapsto -\log(\frac{1}{p+\varepsilon} - 1)$ around the point $p = p_i$ yields

$$\begin{aligned}
-\log(\frac{1}{p_i + \varepsilon} - 1) &\approx -\log(\frac{1}{p_i} - 1) + \frac{\varepsilon}{p_i(1-p_i)} \\
&= y_i + \frac{\varepsilon}{p_i(1-p_i)}.
\end{aligned}$$

and hence we set

$$\varepsilon_i = \frac{\varepsilon}{p_i(1-p_i)}.$$

Dual Problem: It is straightforward to prove that the primal problem can be efficiently solved in its dual form, which is

$$\begin{gathered}
\tfrac{1}{2}\sum_{i,j=1}^{m} \tfrac{(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)}{|S_i||S_j|} \sum_{i'\in S_i, j'\in S_j} K(x_{i'}, x_{j'}) \\
+ \sum_{i=1}^{m} \left(\alpha_i(\varepsilon_i - y_i) + \alpha_i^*(\varepsilon_i + y_i)\right) \rightarrow \min
\end{gathered}$$

s.t.

$$\begin{gathered}
\sum_{i=1}^{m}(\alpha_i - \alpha_i^*) = 0 \\
\forall_{i=1}^{m} : 0 \leq \alpha_i, \alpha_i^* \leq C.
\end{gathered}$$

Table 1. Datasets used in the Experiments.

Data set	Size	Dimension
heart-c	303	23
primary-tumor	339	24
ionosphere	351	34
colic	368	61
vote	435	17
soybean	683	84
credit-a	690	43
breast-w	699	10
diabetes	768	9
vehicle	846	19
anneal	898	64
credit-g	1000	60

where K is a kernel function. The minimization can be carried out by a standard solver for quadratic optimization problems.

In the following, this approach will be called Inverse Calibration.

4. Experiments

We compared the new approach, called Inverse Calibration, to learning classifiers from set probabilities empirically on 12 data sets from the UCI machine learning repository (Asuncion & Newman, 2007). Table 1 lists the data sets that were used. To construct probability examples, we picked different set sizes k and randomly partitioned the original data sets into sets of size k (plus one set of size $< k$, if necessary). Values of $k = 2, 4, 8, 16, 32$ and 64 were chosen in the experiments. We performed tests with linear kernels, radial basis kernels with parameters $\gamma = 0.01, 0.1$ and 1 and polynomial kernels with degrees 2 and 3. Hence, in total $12 * 6 * 6 = 432$ experiments were performed. A 10-fold cross-validation was executed in each experiment. For tuning the parameters of the methods, an internal cross-validation loop was applied in each training phase.

As performance measure, we want to use the accuracy of predicting the labels in the test set. That is, we assume that while set probabilities are given in the training examples, the ultimate goal is to induce a classifier that accurately predicts the labels. In order for the analysis to be independent of the default error rate in each data sets $\mathcal{D}$, we use the accuracy of a method $\mathcal{M}$ relative to the accuracy that can be achieved by a standard classification SVM that has full access to the labels on the training set as our performance measure of choice:

$$accuracy_{rel}(\mathcal{M}, \mathcal{D}) := \frac{accuracy(\mathcal{M}, \mathcal{D})}{accuracy(\text{full SVM}, \mathcal{D})}$$

We compared the Inverse Calibration algorithm with the Mean Map method which has been proven to perform superior to all competing approaches in (Quadrianto et al., 2009). In order to find out how much of the performance of the Inverse Calibration method is due to the general properties of SVMs, we compare our method against simpler approaches of applying SVMs to the problem of learning from probabilities. The following trivial variants are included in our tests:

`Reg`: directly predicting the transformed probabilities of each example using a regression SVM. The same label y was used for each element of a set.

`RegS`: directly predicting the transformed probabilities using a regression SVM, but using only the mean of the vectors in each set as an example (i.e., one example per set).

`Class`: directly predicting the label of each example, using the label 1 for every example in a set S iff the probability of S is higher than the default probability in the complete data set.

`ClassS`: same as `Class`, but taking only the mean of the vectors in each set as an example (i.e., one example per set).

Table 2 list the number of wins of each method in the columns against each method in the row, and against all other methods in total, over all trials. It can be seen that Inverse Calibration and the Mean Map method are clearly superior to the trivial methods: in a direct comparison, the trivial methods lose against the more advanced ones in at least 85% of all trials. In only 6% of all cases, a trivial method outperforms the other methods. Hence, in the following, only those two methods will be compared in more detail.

4.1. Dependency of the Performance on k

Figure 3 shows the relative accuracies of the Inverse Calibration versus the Mean Map method over all 432 tests. Both approaches generally achieve high relative accuracies, it can be seen that in most of the trials at least 70% of the accuracy of a classification SVM with full information was achieved.

To show up an interesting structure in the results, trials with low values of k, namely $k = 2, 4, 8$, were plotted as blue triangles, while trials with high values of k, namely $k = 16, 32, 64$, are plotted as red circles. For low k, i.e. for many sets with few observations per set, both approaches seem to perform roughly similar (see below for detailed statistical tests), and also often not much worse than the standard SVM. On the other hand, for high k, i.e. in a situation with less information, it can be seen that Inverse Calibration frequently outperforms the Mean Map method.

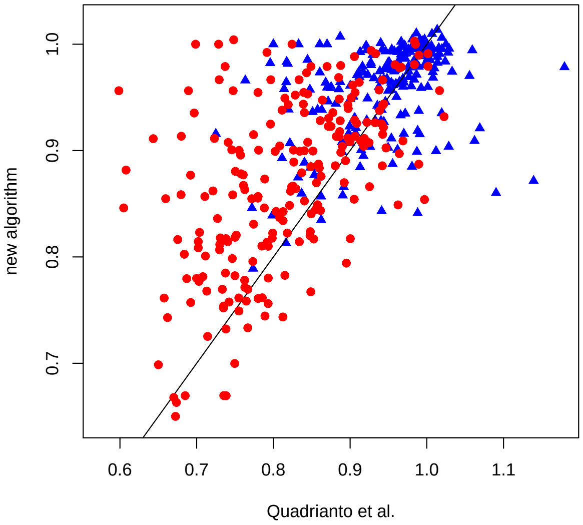

Figure 3. Relative Accuracy of Inverse Calibration (vertical axis) vs. Relative Accuracy of Mean Map (horizontal axis) over all tests. Blue triangles depict test with low values of k, while red circle depict trials with high k.

To investigate the dependency on k in detail, Figure 4 shows boxplots of the relative accuracy of Inverse Calibration minus the relative accuracy of the Mean Map method over all trials for each k. It can be seen that while generally Inverse Calibration performs better for all k (mean values are positive), the difference become especially pronounced the larger the k.

The same effect can be seen in Figure 5, which plots the actual relative accuracies of the two methods over all k. In addition, the relative accuracy of the best trivial method, `classS`, is also plotted. Again, while the relative accuracies decrease with increasing k, the gap between the different methods widens.

4.2. Dependency of the Performance on the Kernel

Finally, we are interested in the effect of the kernel function. Figure 6 shows boxplots of the relative ac-

Table 2. Comparison of methods over 432 trials. Table lists the number of wins of the method in a column against the method in a row. Last row compares the method in column against all other methods.

Method	Inv. Calibration	Mean Map	Reg.	Class.	Reg. Sets	Class. Sets
Inverse Calibration	-	139	8	13	10	33
Mean Map	292	-	13	52	15	62
Regression	423	419	-	331	150	369
Classification	416	380	101	-	111	277
Regression over Sets	422	417	36	321	-	360
Classification over Sets	395	370	58	149	66	-
All methods	268	132	0	6	1	19

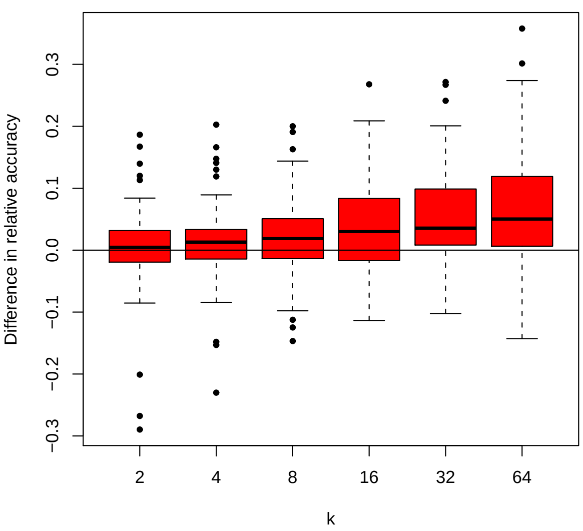

Figure 4. Boxplot of Difference in Relative Accuracy of Inverse Calibration vs. Mean Map over all k. Higher values imply better performance of Inverse Calibration.

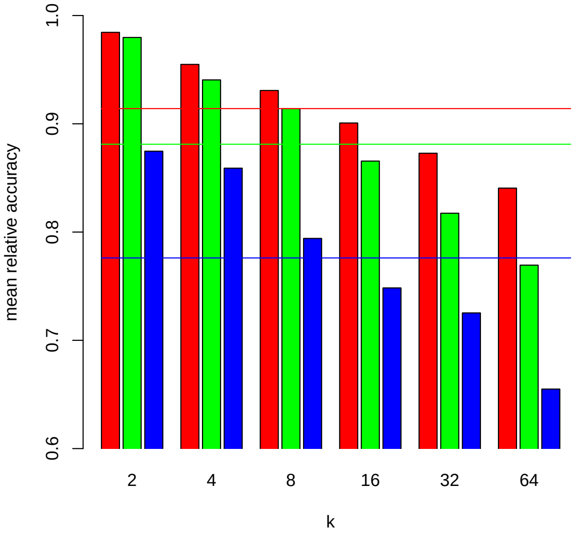

Figure 5. Relative Accuracy of Inverse Calibration (left bar, red) vs. Mean Map (middle bar, green) vs. Classification on Sets (right bar, blue) over all k. Horizontal lines show averages over all k.

curacy of the Inverse Calibration method minus the relative accuracy of the Mean Map method over all trials for each kernel. It can be seen that the results are quite stable, with a slightly better performance for the linear kernel. However, the RBF kernel with parameter $\gamma = 1$ shows a high variance.

The explanation of the erratic performance of this kernel can be found in Figure 7, which shows the actual accuracies of the methods. It can be seen that this kernel on the average shows a worse performance than the other ones, which is possibly due to overfitting the training data, As a consequence, random variations have a much higher influence on the performance of the learners in this case.

4.3. Overall Results

Table 3 shows a detailed comparison of Inverse Calibration with the Mean Map method over all kernels and all values of k. In total, the Inverse Calibration method outperforms Mean Map in 28 of the experiments, performs equally well on 4 and is worse on another 4 experiments (note that each experiment is a test over 12 data sets). A Wilcoxon signed rank test, as suggested by (Demsar, 2006), confirms the statistical significance of the results. Over all trials, a p-value of $6.91e-07$ is achieved, which confirms that the new Inverse Calibration method outperforms the Mean Map method. Further, the Inverse Calibration method per-

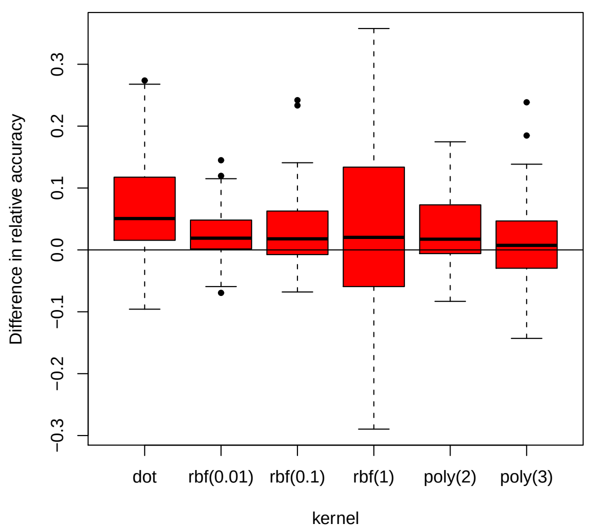

Figure 6. Difference in Relative Accuracy of Inverse Calibration vs. Mean Map over all kernels. Higher values imply better performance of Inverse Calibration.

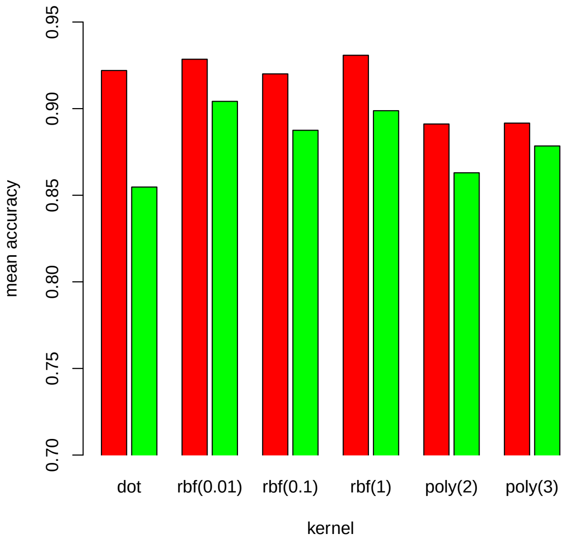

Figure 7. Comparison of the accuracy of Inverse Calibration (left bar, red) vs. Mean Map (right bar, green) over all kernels. Higher values imply better performance of Inverse Calibration.

forms particularly well for the linear kernel and RBF kernels with low γ.

Vice versa, it can be seen that the new approach is only significantly outperformed for the RBF kernel with parameter $\gamma = 1$ and $k = 2, 4$. However, as has been already discussed in Section 4.2, this kernel function exhibits a low performance on the average and hence is not of a particularly high importance compared to the other kernels.

5. Conclusions and Future Work

Estimating classifiers from group probabilities is an important learning task with many practical applications. However, it has only recently begun to receive attention in the research community.

In this paper, we have presented a new algorithm for estimating a classifier from group probabilities based on support vector regression and inverse classifier calibration. A detailed comparison of the new Inverse Calibration algorithm with the best previously known approach of (Quadrianto et al., 2009) has revealed that the new algorithm performs significantly better, in particular in the case of linear kernels and high compression factors, i.e. high number k of observations per group. In all other cases, both approaches have been shown to exhibit comparable performance. Algorithmically, the Inverse Calibration method is a quadratic optimization problem, for which efficient solvers exists, while the Mean Map method has to be optimized with more general solvers.

While the new method works only for binary classification, Quadrianto's approach is also defined for an arbitrary number of classes. It would be interesting to see if it is possible to extend the new approach to multiple classes, for example by making use of ideas from algorithms for multiclass SVMs (Duan & Keerthi, 2005).

A practically very interesting direction for future work lies in taking the construction process of groups into account. In this paper, we have taken an i.i.d. assumption, which is reasonable when one does not know otherwise. However, in situations like privacy-preserving data mining, where full information is available to one party, but not the second party that is analyzing the data, both parties could still agree on a process that tries to set up the groups in a way to both guarantees data privacy and allow for an effective classifier estimation under these constraints. Such a process could, for example, be built up upon ideas from active learning (Tong & Koller, 2000).

Table 3. Comparison of Inverse Calibration vs. Mean Map for each k and kernel. Table list the number of wins/ties/losses and the p-value of a Wilcoxon signed rank test of the hypothesis that the Inverse Calibration method is better than Mean Map. Results that are significant at the 10% level are printed in bold.

K	DOT	RBF(0.01)	RBF(0.1)	RBF(1)	POLY(2)	POLY(3)	ALL
2	**11/0/1**	7/2/3	5/0/7	3/0/9	8/0/4	6/0/6	40/2/30
	p = 0.005	*p = 0.130*	*p = 0.782*	*p = 0.989*	*p = 0.118*	*p = 0.535*	*p = 0.351*
4	**10/0/2**	9/0/3	8/0/4	4/0/8	6/0/6	7/0/5	**44/0/28**
	p = 0.015	*p = 0.156*	*p = 0.143*	*p = 0.893*	*p = 0.333*	*p = 0.465*	***p = 0.083***
8	**10/0/2**	9/0/3	9/0/3	7/0/5	9/0/3	5/0/7	**49/0/23**
	p = 0.034	*p = 0.130*	*p = 0.143*	*p = 0.602*	*p = 0.130*	*p = 0.688*	***p = 0.072***
16	**9/0/3**	**10/0/2**	**9/0/3**	**7/0/5**	6/1/5	6/0/6	**47/1/24**
	p = 0.056	***p = 0.056***	***p = 0.070***	***p = 0.050***	*p = 0.302*	*p = 0.512*	***p = 0.002***
32	**11/0/1**	**11/0/1**	**9/0/3**	**10/0/2**	**9/1/2**	8/0/4	**58/1/13**
	p = 0.018	***p = 0.027***	***p = 0.079***	***p = 0.004***	***p = 0.092***	*p = 0.235*	***p < 1e-3***
64	**10/0/2**	9/0/3	**9/0/3**	**10/0/2**	**8/0/4**	**9/0/3**	**55/0/17**
	p = 0.015	*p = 0.143*	***p = 0.070***	***p = 0.003***	***p = 0.087***	***p = 0.044***	***p < 1e-3***
	61/0/11	**55/2/15**	**49/0/23**	**41/0/31**	**46/2/24**	41/0/31	**293/4/135**
	p < 1e-3	***p = 0.027***	***p = 0.020***	***p = 0.044***	***p = 0.037***	*p = 0.275*	***p < 1e-3***

References

Asuncion, A. and Newman, D.J. UCI machine learning repository. http://www.ics.uci.edu/~mlearn/, institution = University of California, Irvine, School of Information and Computer Sciences, 2007.

Ayer, M., Brunk, H., Ewing, G., Reid, W., and Silverman, E. An empirical distribution function for sampling with incomplete information. *Annals of Mathematical Statistics*, 5:641–647, 1955.

Demsar, Janez. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

Duan, Kai-Bo and Keerthi, S. Sathiya. Which is the best multiclass svm method? an empirical study. In Oza, Nikunj C., Polikar, Robi, Kittler, Josef, and Roli, Fabio (eds.), *Proc. Multiple Classifier Systems (MCS 2005)*, volume 3541 of *LNCS*, pp. 278–285. Springer, 2005.

Kueck, H. and de Freitas, N. Learning about individuals from group statistics. In *Uncertainty in Artificial Intelligence (UAI)*, pp. 332339, Arlington, Virginia, 2005. AUAI Press.

Niculescu-Mizil, Alexandru and Caruana, Rich. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 625–632, 2005.

Platt, John. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Smola, A., Bartlett, P., Schölkopf, B., and Schuurmans, D. (eds.), *Advances in Large Margin Classifiers*. MIT Press, 1999.

Quadrianto, Novi, Smola, Alex J., Caetano, Tibrio S., and Le, Quoc V. Estimating labels from label proportions. In Cohen, W., McCallum, A., , and Roweis, S. (eds.), *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, pp. 776783. Omnipress, 2008.

Quadrianto, Novi, Smola, Alex J., Caetano, Tibrio S., and Le, Quoc V. Estimating labels from label proportions. *Journal of Machine Learning Research*, 10: 2349–2374, Oct 2009.

Rüping, Stefan. A simple method for estimating conditional probabilities in SVMs. In Abecker, A., Bickel, S., Brefeld, U., Drost, I., Henze, N., Herden, O., Minor, M., Scheffer, T., Stojanovic, L., and Weibelza hl, S. (eds.), *LWA 2004 - Lernen - Wissensentdeckung - Adaptivität*. Humboldt-Universität Berlin, 2004.

Tong, S. and Koller, D. Restricted bayes optimal classifiers. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*, 2000.

Vapnik, V. *Statistical Learning Theory*. Wiley, Chichester, GB, 1998.

Zadrozny, Bianca and Elkan, Charles. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 694–699, 2002.