Estimating Labels from Label Proportions

Novi Quadrianto NOVI.QUAD@GMAIL.COM
Alex J. Smola ALEX.SMOLA@GMAIL.COM
Tiberio S. Caetano TIBERIO.CAETANO@GMAIL.COM
Statistical Machine Learning, NICTA and RSISE, Australian National University
Quoc V. Le QUOC.LE@CS.STANFORD.EDU
Computer Science Department, Stanford University

Abstract

Consider the following problem: given sets of unlabeled observations, each set with known label proportions, predict the labels of another set of observations, also with known label proportions. This problem appears in areas like e-commerce, spam filtering and improper content detection. We present consistent estimators which can reconstruct the correct labels with high probability in a uniform convergence sense. Experiments show that our method works well in practice.

1 Introduction

Assume that a web services company wants to increase its profit in sales. Obviously sending out discount coupons will increase sales, but sending coupons to customers who would have purchased the goods anyway decreases the margins. Alternatively, failing to send coupons to customers who would only buy in case of a discount reduces overall sales. We would like to identify the class of would-be customers who are most likely to change their purchase decision when receiving a coupon. The problem is that there is no direct access to a sample of would-be customers. Typically only a sample of people who buy regardless of coupons (those who bought when there was no discount) and a mixed sample (those who bought when there was discount) are available. The mixing proportions can be reliably estimated using random assignment to control and treatment groups. How can we use this information to determine the would-be customers?

Likewise, consider the problem of spam filtering. Datasets of spam are likely to contain almost pure spam (this is achieved e.g. by listing e-mails as spam bait), while user's inboxes typically contain a mix of spam and non-spam. We would like to use the inbox data to improve estimation of spam. In many cases it is possible to estimate the *proportions* of spam and non-spam in a user's inbox much more cheaply than the actual labels. We would like to use this information to categorize e-mails into spam and non-spam.

Similarly, consider the problem of filtering images with "improper content". Datasets of such images are readily accessible thanks to user feedback, and it is reasonable to assume that this labeling is highly reliable. However the rest of images on the web (those not labeled) is a far larger dataset, albeit without labels (after all, this is what we would like to estimate the labels for). That said, it is considerably cheapter to obtain a good estimate of the *proportions* of proper and improper content in addition to having one dataset of images being of likely improper content. We would like to obtain a classifier based on this information.

In this paper we present a method to estimate labels *directly* in such situations, assuming that only label proportions be known. In the above examples, this would be helpful in identifying potential customers, spam e-mails and improper images. We prove bounds indicating that the estimates obtained are close to those from a fully labeled scenario. The formal setting though is more general than the above examples might suggest: we do not require *any* label to be known, only their proportions within each of the involved datasets. Also we are not restricted to the binary case but instead can deal with large numbers of classes.

Problem Formulation Assume that we have n sets of observations $X_i = \left\{x_1^i, \ldots, x_{m_i}^i\right\}$ of respective sample sizes m_i (our calibration set) as well as a set $X = \{x_1, \ldots, x_m\}$ (our test set). Moreover, assume that we know the fractions π_{iy} of patterns of labels $y \in \mathcal{Y}$ ($|\mathcal{Y}| \leq n$) contained in each set X_i and assume that we also know the marginal probability $p(y)$ of the

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).

Table 1. Notation Conventions

X_i	i^{th} set of observations: $X_i = \{x_1^i, \ldots, x_{m_i}^i\}$
m_i	number of observations in X_i
X	test set of observations: $X = \{x_1, \ldots, x_m\}$
Y	test set of labels: $Y = \{y_1, \ldots, y_m\}$
m	number of observations in the test set X
π_{iy}	proportion of label y in set i
$\phi(x, y)$	map from (x, y) to a Hilbert Space

Table 2. Major quantities of interest in the paper

Numbers on the left represent the order in which the corresponding quantity is computed in the algorithm (letters denote the variant of the algorithm: 'a' for general feature map $\phi(x, y)$ and 'b' for factorizing feature map $\phi(x, y) = \psi(x) \otimes \varphi(y)$). Lowercase subscripts refer to model expectations, uppercase subscripts are sample averages.

Expectations with respect to the model:		
	μ_{xy}	$:= \mathbf{E}_{(x,y)\sim p(x,y)}[\phi(x, y)]$
	$\mu_x^{\text{class}}[y, y']$	$:= \mathbf{E}_{(x)\sim p(x\|y)}[\phi(x, y')]$
	$\mu_x^{\text{set}}[i, y']$	$:= \mathbf{E}_{(x)\sim p(x\|i)}[\phi(x, y')]$
	$\mu_x^{\text{class}}[y]$	$:= \mathbf{E}_{(x)\sim p(x\|y)}[\psi(x)]$
	$\mu_x^{\text{set}}[i]$	$:= \mathbf{E}_{(x)\sim p(x\|i)}[\psi(x)]$
Expectations with respect to data:		
	μ_{XY}	$:= \frac{1}{m}\sum_{i=1}^m \phi(x_i, y_i)$
(1a)	$\mu_X^{\text{set}}[i, y']$	$:= \frac{1}{m_i}\sum_{x\in X_i} \phi(x, y')$ (**known**)
(1b)	$\mu_X^{\text{set}}[i]$	$:= \frac{1}{m_i}\sum_{x\in X_i} \psi(x)$ (**known**)
Estimates:		
(2)	$\hat{\mu}_x^{\text{class}}$	$= (\pi^\top \pi)^{-1} \pi^\top \mu_X^{\text{set}}$
(3a)	$\hat{\mu}_{XY}$	$= \sum_{y\in\mathcal{Y}} p(y) \hat{\mu}_x^{\text{class}}[y, y]$
(3b)	$\hat{\mu}_{XY}$	$= \sum_{y\in\mathcal{Y}} p(y)\varphi(y) \otimes \hat{\mu}_x^{\text{class}}[y]$
(4)	$\hat{\theta}^*$	solution of (5) for $\mu_{XY} = \hat{\mu}_{XY}$.

test set X.[1] It is our goal to design algorithms which are able to obtain conditional class probability estimates $p(y|x)$ solely based on this information. As an illustration, take the spam filtering example. We have X_1 = "mail in spam box" (only spam) and X_2 = "mail in inbox" (spam mixed with non-spam). The test set X then may be X_2 itself, for example. The goal is to find $p(\text{spam}|\text{mail})$ in X_2. Note that (for general π_{iy}) this is more difficult than transduction, where we have at least one dataset with actual labels plus an unlabeled test set where we might have an estimate as to what the relative fractions of class labels might be.

2 Mean Operators

Our idea relies on uniform convergence properties of the expectation operator and of corresponding risk functionals (Altun & Smola, 2006). In doing so, we are able to design estimators with the same performance guarantees in terms of uniform convergence as those with full access to the label information.

2.1 Exponential Families

Denote by $\mathcal{X}$ the space of observations and let $\mathcal{Y}$ be the space of labels. Moreover, let $\phi(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathcal{H}$ be a feature map into a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ with kernel $k((x, y), (x', y'))$. In this case we may state conditional exponential models via

$$p(y|x, \theta) = \exp\left(\langle \phi(x, y), \theta\rangle - g(\theta|x)\right) \text{ with} \tag{1}$$

$$g(\theta|x) = \log \sum_{y\in\mathcal{Y}} \exp \langle \phi(x, y), \theta\rangle, \tag{2}$$

where the normalization g is called the log-partition function. For $\{(x_i, y_i)\}$ drawn iid from a distribution $p(x, y)$ on $\mathcal{X} \times \mathcal{Y}$ the conditional log-likelihood is

$$\log p(Y|X, \theta) = \sum_{i=1}^m \left[\langle \phi(x_i, y_i), \theta\rangle - g(\theta|x_i)\right] \tag{3}$$

$$= m \langle \mu_{XY}, \theta\rangle - \sum_{i=1}^m g(\theta|x_i)$$

where μ_{XY} is defined as in Table 2. In order to avoid overfitting one commonly maximizes the log-likelihood penalized by a prior $p(\theta)$. This means that we need to solve the following optimization problem

$$\theta^* := \underset{\theta}{\operatorname{argmin}} \left[-\log p(Y|X, \theta)p(\theta)\right]. \tag{4}$$

For instance, for a Gaussian prior on θ, i.e. for $-\log p(\theta) = \lambda \|\theta\|^2 + \text{const.}$ we have

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \left[\sum_{i=1}^m g(\theta|x_i) - m \langle \mu_{XY}, \theta\rangle + \lambda \|\theta\|^2\right]. \tag{5}$$

The problem is that in our setting we do not know the labels y_i, so the sufficient statistics μ_{XY} cannot be computed exactly. The only place where the labels enter the estimation process is via the mean μ_{XY}. Our strategy is to exploit the fact that this quantity, however, is statistically well behaved and converges under relatively mild technical conditions at rate $O(m^{-\frac{1}{2}})$ to its expected value (see Theorem 2)

$$\mu_{xy} := \mathbf{E}_{(x,y)\sim p(x,y)}[\phi(x, y)]. \tag{6}$$

Our goal therefore will be to estimate μ_{xy} and use it as a proxy for μ_{XY}, and only then solve (5) with the estimated $\hat{\mu}_{XY}$ instead of μ_{XY}. We will discuss explicit convergence guarantees in Section 3 after describing how to compute the mean operator in detail.

2.2 Estimating the Mean Operator

In order to obtain θ^* we would need μ_{XY}, which is impossible to compute exactly, since we do not have

[1] Label dictionaries $\mathcal{Y}_i$ do not need to be the same across all sets i: define $\mathcal{Y} := \cup_i \mathcal{Y}_i$ and allow for $\pi_{iy} = 0$ as needed.

Y. However, we know that μ_{XY} and μ_{xy} are close. Hence, if we are able to approximate μ_{xy} this, in turn, will be a good estimate for μ_{XY}.

Our quest is therefore as follows: express μ_{xy} as a linear combination over expectations with respect to the distributions on the datasets $X_1, \ldots, X_n$ (where $n \geq |\mathcal{Y}|$). Secondly, show that the expectations of the distributions having generated the sets X_i ($\mu_x^{\text{set}}[i, y']$, see Table 2) can be approximated by empirical means ($\mu_X^{\text{set}}[i, y']$, also see Table 2). Finally, we need to combine both steps to provide guarantees for μ_{XY}.

It will turn out that in certain cases some of the algebra can be sidestepped, in particular whenever we may be able to identify several sets with each other (e.g. the test set X is one of the training datasets X_i) or whenever $\phi(x, y)$ factorizes into $\psi(x) \otimes \varphi(y)$.

Mean Operator: Since μ_{xy} is a linear operator mapping $p(x, y)$ into a Hilbert Space we may expand μ_{xy}

$$\mu_{xy} = \sum_{y \in \mathcal{Y}} p(y) \mathbf{E}_{x \sim p(x|y)}[\phi(x, y)] = \sum_{y \in \mathcal{Y}} p(y) \mu_x^{\text{class}}[y, y]$$

where the shorthand $\mu_x^{\text{class}}[y, y]$ is defined in Table 2. This means that if we were able to compute $\mu_x^{\text{class}}[y, y]$ we would be able to "reassemble" μ_{xy} from its individual components. We now show that $\mu_x^{\text{class}}[y, y]$ can be estimated directly.

Key to our assumptions is that $p(x|y, i) = p(x|y)$. In other words, we assume that the *conditional* distribution of x is independent of the index i, as long as we know the label y. This yields the following:

$$p(x|i) = \sum_y p(x|y) \pi_{iy}. \tag{7}$$

This allows us define the following means

$$\mu_x^{\text{set}}[i, y'] := \mathbf{E}_{x \sim p(x|i)}[\phi(x, y')] \overset{(7)}{=} \sum_y \pi_{iy} \mu_x^{\text{class}}[y, y'].$$

Note that in order to compute $\mu_x^{\text{set}}[i, y']$ we do *not* need any label information with respect to $p(x|i)$. However, since we have at least $|\mathcal{Y}|$ of those equations and we assumed that π has full rank, they allow us to solve a linear system of equations and compute $\mu_x^{\text{class}}[y, y]$ from $\mu_x^{\text{set}}[i, y']$ for all i. That is, we may use

$$\mu_x^{\text{set}} = \pi \mu_x^{\text{class}} \text{ and hence } \mu_x^{\text{class}} = (\pi^\top \pi)^{-1} \pi^\top \mu_x^{\text{set}} \tag{8}$$

to compute $\mu_x^{\text{class}}[y, y]$ for all $y \in \mathcal{Y}$. Whenever $\pi \in \mathbb{R}^{n \times n}$ is invertible (8) reduces to $\mu_x^{\text{class}} = \pi^{-1} \mu_x^{\text{set}}$. With some slight abuse of notation we have μ_x^{class} and μ_x^{set} represent the *matrices* of terms $\mu_x^{\text{class}}[y, y']$ and $\mu_x^{\text{set}}[i, y']$ respectively.

Algorithm 1

```
Input datasets $X$, $\{X_i\}$, probabilities $\pi_{iy}$ and $p(y)$
for $i = 1$ to $n$ and $y' \in \mathcal{Y}$ do
    Compute empirical means $\mu_X^{\text{set}}[i, y']$
end for
Compute $\hat{\mu}_x^{\text{class}} = (\pi^\top \pi)^{-1} \pi^\top \mu_X^{\text{set}}$
Compute $\hat{\mu}_{XY} = \sum_{y \in \mathcal{Y}} p(y) \hat{\mu}_x^{\text{class}}[y, y]$
Solve the minimization problem
    $\hat{\theta}^* = \operatorname*{argmin}_\theta \left[ \sum_{i=1}^m g(\theta|x_i) - m \langle \hat{\mu}_{XY}, \theta \rangle + \lambda \|\theta\|^2 \right]$
Return $\hat{\theta}^*$.
```

Obviously we cannot compute $\mu_x^{\text{set}}[i, y']$ explicitly, since we only have *samples* from $p(x|i)$. However the same convergence results governing the convergence of μ_{XY} to μ_{xy} also hold for the convergence of $\mu_X^{\text{set}}[i, y']$ to $\mu_x^{\text{set}}[i, y']$. Hence we may use the empirical average $\mu_X^{\text{set}}[i, y']$ as the estimate for $\mu_x^{\text{set}}[i, y']$ and from that find an estimate for μ_{XY} (see Algorithm 1).

2.3 Special Cases

In some cases the calculations described in Algorithm 1 can be carried out more efficiently.

Minimal number of sets, i.e. $|\mathcal{Y}| = n$: Provided that π has full rank, $(\pi^\top \pi)^{-1} \pi^\top = \pi^{-1}$. This means that the inverse can be computed more directly.

Testing on one of the calibration sets, i.e. $X = X_i$: This means that X is one of the training sets. We only need one less set of observations. This is particularly useful for factorizing feature maps.

Special feature map $\phi(x, y) = \psi(x) \otimes \varphi(y)$: In this case the calculations of $\hat{\mu}_x^{\text{class}}[y, y']$ and $\mu_X^{\text{set}}[i, y']$ are greatly simplified, since we may pull the dependency on y out of the expectations. Defining $\mu_x^{\text{class}}[y], \mu_x^{\text{set}}[i]$, and $\mu_X^{\text{set}}[i]$ as in Table 2 allows us to simplify

$$\hat{\mu}_{XY} = \sum_{y \in \mathcal{Y}} p(y) \varphi(y) \otimes \hat{\mu}_x^{\text{class}}[y] \tag{9}$$

$$\text{where } \hat{\mu}_x^{\text{class}} = (\pi^\top \pi)^{-1} \pi^\top \mu_X^{\text{set}}. \tag{10}$$

A significant advantage of (10) is that we only need to perform $O(n)$ averaging operations rather than $O(n \cdot |\mathcal{Y}|)$. Obviously the cost of computing $(\pi^\top \pi)^{-1} \pi^\top$ remains unchanged but the latter is negligible in comparison to the operations in Hilbert Space.

Binary classification: One may show that the feature map $\phi(x, y)$ takes on a particularly appealing form of $\phi(x, y) = y\psi(x)$ where $y \in \{\pm 1\}$. This follows since we can always re-calibrate $\langle \phi(x, y), \theta \rangle$ by an offset independent of y such that $\phi(x,1)+\phi(x,-1)=0$.

If we moreover assume that X_1 only contains class 1 and $X_2 = X$ contains a mixture of classes with labels 1 and -1 with proportions $p(1) =: \rho$ and $p(-1) = 1-\rho$ respectively, we obtain the mixing matrix

$$\pi = \begin{bmatrix} 1 & 0 \\ \rho & 1-\rho \end{bmatrix} \Rightarrow \pi^{-1} = \begin{bmatrix} 1 & 0 \\ \frac{-\rho}{1-\rho} & \frac{1}{1-\rho} \end{bmatrix}$$

Plugging this into (10) and the result in (9) yields

$$\begin{aligned}\hat{\mu}_{XY} &= \rho\mu_X^{\text{set}}[1] - (1-\rho)\left[\tfrac{-\rho}{1-\rho}\mu_X^{\text{set}}[1] + \tfrac{1}{1-\rho}\mu_X^{\text{set}}[2]\right] \\ &= 2\rho\mu_X^{\text{set}}[1] - \mu_X^{\text{set}}[2]. \qquad (11)\end{aligned}$$

Consequently taking a simple weighted difference between the averages on two sets, e.g. one set containing spam whereas the other one containing an unlabeled mix of spam and non-spam allows one to obtain the sufficient statistics needed for estimation.

3 Convergence Bounds

The obvious question is how well $\hat{\mu}_{XY}$ manages to approximate μ_{XY} and secondly, how badly any error in estimating μ_{XY} would affect the overall quality of the solution. We approach this problem as follows: first we state the uniform convergence properties of μ_{XY} and similar empirical operators relative to μ_{xy}. Secondly, we apply those bounds to the cases discussed above, and thirdly, we show that the approximate minimizer of the log-posterior has a bounded deviation from what we would have obtained by knowing μ_{XY} exactly.

3.1 Uniform Convergence for Mean Operators

In order to introduce the key result we need to introduce Rademacher averages:

Definition 1 (Rademacher Averages) *Let $\mathcal{X}$ be a domain and p a distribution on $\mathcal{X}$ and assume that $X := \{x_1, \ldots, x_m\}$ is drawn iid from p. Moreover, let $\mathcal{F}$ be a class of functions $\mathcal{X} \to \mathbb{R}$. Furthermore denote by σ_i Rademacher random variables, i.e. $\{\pm 1\}$ valued with zero mean. The Rademacher average is*

$$R_m(\mathcal{F}, p) := \mathbf{E}_X \mathbf{E}_\sigma \left[\sup_{f \in \mathcal{F}} \left| \frac{1}{m} \sum_{i=1}^m \sigma_i f(x_i) \right| \right]. \qquad (12)$$

This quantity measures the flexibility of the function class $\mathcal{F}$ — in our case linear functions in $\phi(x,y)$.

Theorem 2 (Convergence of Empirical Means) *Denote by $\phi : \mathcal{X} \to \mathcal{B}$ a map into a Banach space $\mathcal{B}$, denote by $\mathcal{B}^*$ its dual space and let $\mathcal{F}$ the class of linear functions on $\mathcal{B}$ with bounded $\mathcal{B}^*$ norm by 1. Let $R > 0$ such that for all $f \in \mathcal{F}$ we have $|f(x)| \leq R$. Moreover, assume that X is an m-sample drawn from p on $\mathcal{X}$. For $\bar{\epsilon} > 0$ we have that with probability at least $1 - \exp(-\bar{\epsilon}^2 m / 2R^2)$ the following holds:*

$$\|\mu_X - \mu_x\|_{\mathcal{B}} \leq 2R_m(\mathcal{F}, p) + \bar{\epsilon} \qquad (13)$$

For $k \geq 0$ we only have a failure probability of $1 - \exp(-\bar{\epsilon}^2 m / R^2)$.

Theorem 3 (Bartlett & Mendelson (2002)) *Whenever $\mathcal{B}$ is a Reproducing Kernel Hilbert Space with kernel $k(x,x')$ the Rademacher average can be bounded from above by $R_m(\mathcal{F}) \leq m^{-\frac{1}{2}} \left[\mathbf{E}_x[k(x,x)]\right]^{\frac{1}{2}}$*

Our approximation error can be bounded as follows. From the triangle inequality we have:

$$\|\hat{\mu}_{XY} - \mu_{XY}\| \leq \|\hat{\mu}_{XY} - \mu_{xy}\| + \|\mu_{xy} - \mu_{XY}\|.$$

For the second term we may employ Theorem 2 directly. To bound the first term note that by linearity

$$\epsilon := \hat{\mu}_{XY} - \mu_{xy} = \sum_y p(y) \left[(\pi^\top \pi)^{-1} \pi^\top \hat{\epsilon}\right]_{y,y} \qquad (14)$$

where we define the matrix of coefficients

$$\hat{\epsilon}\,[i, y'] := \mu_x^{\text{set}}[i, y'] - \mu_X^{\text{set}}[i, y']. \qquad (15)$$

Now note that all $\hat{\epsilon}\,[i, y']$ also satisfy the conditions of Theorem 2 since the sets X_i are drawn iid from the distributions $p(x|i)$ respectively. We may bound each term individually in this fashion and subsequently apply the union bound to ensure that all $n \cdot |\mathcal{Y}|$ components satisfy the constraints. Hence each of the terms needs to satisfy the constraint with probability $1 - \delta/(n|\mathcal{Y}|)$ to obtain an overall bound with probability $1 - \delta$. To obtain bounds we would need to bound the linear operator mapping $\hat{\epsilon}$ into ϵ.

3.2 Special Cases

A closed form solution in the general case is not particularly useful. However, we give an explicit analysis for two special cases: firstly the situation where $\phi(x,y) = \psi(x) \otimes \varphi(y)$ and secondly, the binary classification setting where $\phi(x,y) = y\psi(x)$ and $X_i = X$, where much tighter bounds are available.

Special feature map We only need to deal with n rather than with $n \times |\mathcal{Y}|$ empirical estimates, i.e. $\mu_X^{\text{set}}[i]$ vs. $\mu_X^{\text{set}}[i, y']$. Hence (14) and (15) specialize to

$$\epsilon = \sum_y p(y) \sum_{i=1}^n \varphi(y) \otimes \left[(\pi^\top \pi)^{-1} \pi^\top\right]_{yi} \hat{\epsilon}[i] \qquad (16)$$

$$\hat{\epsilon}\,[i] := \mu_x^{\text{set}}[i] - \mu_X^{\text{set}}[i]. \qquad (17)$$

Assume that with high probability each $\hat{\epsilon}[i]$ satisfies $\|\hat{\epsilon}[i]\| \leq c_i$ (we will deal with the explicit constants c_i later). Moreover, assume for simplicity that $|\mathcal{Y}| = n$ and that π has full rank (otherwise we need to follow through on our expansion using $(\pi^\top \pi)^{-1}\pi^\top$ instead of π^{-1}). This implies that

$$\begin{aligned}\|\epsilon\|^2 &= \sum_{i,j} \langle \hat{\epsilon}[i], \hat{\epsilon}[j]\rangle \times \\ & \quad \sum_{y,y'} p(y)p(y')k(y,y') \left[\pi^{-1}\right]_{yi} \left[\pi^{-1}\right]_{y'j} \\ &\leq \sum_{i,j} c_i c_j \left|\left[\pi^{-1}\right]^\top K^{y,p} \pi^{-1}\right|_{ij} \qquad (18)\end{aligned}$$

where $K^{y,p}_{y,y'} = k(y,y')p(y)p(y')$. Combining several bounds we have the following theorem:

Theorem 4 *Assume that we have n sets of observations X_i of size m_i, each of which drawn from distributions with probabilities π_{iy} of observing data with label y. Moreover, assume that $k((x,y),(x',y')) = k(x,x')k(y,y') \geq 0$ where $k(x,x) \leq 1$ and $k(y,y) \leq 1$. Finally, assume that $m = |X|$. In this case the mean operator μ_{XY} can be estimated by $\hat{\mu}_{XY}$ with probability at least $1-\delta$ with precision*

$$\|\mu_{XY} - \hat{\mu}_{XY}\| \leq \left[2 + \sqrt{\log((n+1)/\delta)}\right] \times$$
$$\left[m^{-\frac{1}{2}} + \left[\sum_{i,j} m_i^{-\frac{1}{2}} m_j^{-\frac{1}{2}} \left|\left[\pi^{-1}\right]^\top K^{y,p}\pi^{-1}\right|_{ij}\right]^{\frac{1}{2}}\right]$$

Proof We begin our argument by noting that both for $\phi(x,y)$ and for $\psi(x)$ the corresponding Rademacher averages R_m for functions of RKHS norm bounded by 1 is bounded by $m^{-\frac{1}{2}}$. This is a consequence of all kernels being bounded by 1 in Theorem 3 and $k \geq 0$.

Next note that in Theorem 2 we may set $R = 1$, since for $\|f\| \leq 1$ and $k((x,y),(x,y)) \leq 1$ and $k(x,x) \leq 1$ it follows from the Cauchy Schwartz inequality that $|f(x)| \leq 1$. Solving $\delta \leq \exp -m\epsilon^2$ for ϵ yields $\epsilon \leq m^{-\frac{1}{2}}\left[2 + \sqrt{\log(1/\delta)}\right]$.

Finally, note that we have $n+1$ deviations which we need to bound: one between μ_{XY} and μ_{xy}, and n for each of the $\epsilon[i]$ respectively. Dividing the failure probability δ into $n+1$ cases yields bounds of the form $m^{-\frac{1}{2}}\left[2 + \sqrt{\log((n+1)/\delta)}\right]$ and $m_i^{-\frac{1}{2}}\left[2 + \sqrt{\log((n+1)/\delta)}\right]$ respectively. Plugging all error terms into (18) and summing over terms yields the claim and substituting this back into the triangle inequality proves the claim. ■

Binary Classification Next we consider the special case of binary classification where $X_2 = X$. Using (11) we see that the corresponding estimator is given by

$$\hat{\mu}_{XY} = 2\rho\mu_X^{\text{set}}[1] - \mu_X^{\text{set}}[2]. \qquad (19)$$

Since $\hat{\mu}_{XY}$ shares a significant fraction of terms with μ_{XY} we are able to obtain tighter bounds as follows:

Theorem 5 *With probability $1-\delta$ (for $1 > \delta > 0$) the following bound holds:*

$$\|\hat{\mu}_{XY} - \mu_{XY}\| \leq 2\rho\left[2 + \sqrt{\log(2/\delta)}\right]\left[m_1^{-\frac{1}{2}} + m_+^{-\frac{1}{2}}\right]$$

m_+ is the number of observations with $y = 1$ in X_2.

Proof Denote by $\mu[X_+]$ and $\mu[X_-]$ the averages over the subsets of X_2 with positive and negative labels respectively. By construction we have that

$$\begin{aligned}\mu_{XY} &= \rho\mu[X_+] - (1-\rho)\mu[X_-] \\ \hat{\mu}_{XY} &= 2\rho\mu_X^{\text{set}}[1] - \rho\mu[X_+] - (1-\rho)\mu[X_-]\end{aligned}$$

Taking the difference yields $2\rho\left[\mu_X^{\text{set}}[1] - \mu[X_+]\right]$. To prove the claim note that we may use Theorem 2 both for $\left\|\mu_X^{\text{set}}[1] - \mathbf{E}_{x\sim p(x|y=1)}[\psi(x)]\right\|$ and for $\left\|\mu[X_+] - \mathbf{E}_{x\sim p(x|y=1)}[\psi(x)]\right\|$. Taking the union bound and summing over terms proves the claim. ■

The bounds we provided show that $\hat{\mu}_{XY}$ converges at the same rate to μ_{xy} as μ_{XY} does, assuming that the sizes of the sets X_i increase at the same rate as X.

3.3 Stability Bounds

To complete our reasoning we need to show that those bounds translate in guarantees in terms of the minimizer of the log-posterior. In other words, estimates using the correct mean μ_{XY} vs. its estimate $\hat{\mu}_{XY}$ do not differ by a significant amount. For this purpose we make use of (Altun & Smola, 2006, Lemma 17).

Lemma 6 *Denote by f a convex function on $\mathcal{H}$ and let $\mu, \hat{\mu} \in \mathcal{H}$. Moreover let $\lambda > 0$. Finally denote by $\theta^*, \in \mathcal{H}$ the minimizer of*

$$L(\theta, \mu) := f(\theta) - \langle \mu, \theta\rangle + \lambda\|\theta\|^2 \qquad (20)$$

with respect to θ and $\hat{\theta}^*$ the minimizer of $L(\hat{\theta}, \hat{\mu})$ respectively. In this case the following inequality holds:

$$\left\|\theta^* - \hat{\theta}^*\right\| \leq \lambda^{-1}\|\mu - \hat{\mu}\|. \qquad (21)$$

This means that a good estimate for μ immediately translates into a good estimate for the minimizer of the approximate log-posterior. This leads to the following bound on the risk minimizer.

Corollary 7 *The deviation between θ^*, as defined in (4) and $\hat{\theta}^*$, the minimizer of the approximate log-posterior using $\hat{\mu}_{XY}$ rather than μ_{XY}, is bounded by $O(m^{-\frac{1}{2}} + \sum_i m_i^{-\frac{1}{2}})$.*

Finally, we may use (Altun & Smola, 2006, Theorem 16) to obtain bounds on the quality of $\hat{\theta}^*$ when considering how well it minimizes the *true* negative log-posterior. Using the bound

$$L(\hat{\theta}^*, \mu) - L(\theta^*, \mu) \leq \left\|\hat{\theta}^* - \theta^*\right\| \|\hat{\mu} - \mu\| \qquad (22)$$

yields the following bound for the log-posterior:

Corollary 8 *The minimizer $\hat{\theta}^*$ of the approximate log-posterior using $\hat{\mu}_{XY}$ rather than μ_{XY} incurs a penalty of at most $\lambda^{-1} \|\hat{\mu}_{XY} - \mu_{XY}\|^2$.*

4 Extensions

Note that our analysis so far focused on a specific setting, namely maximum-a-posteriori analysis in exponential families. While this is a common and popular setting, the derivations are by no means restricted to this. We have the entire class of (conditional) models described by Altun & Smola (2006); Dudík & Schapire (2006) at our disposition. They are characterized via

$$\underset{p}{\text{minimize}} -H(p) \text{ subject to } \|\mathbf{E}_{z\sim p}[\phi(z)] - \mu\| \leq \epsilon$$

Here p is a distribution, H is an entropy-like quantity defined on the space of distributions, and $\phi(z)$ is some evaluation map into a Banach space. This means that the optimization problem can be viewed as an approximate maximum entropy estimation problem, where we do not enforce exact moment matching of μ but rather allow ϵ slack. In both Altun & Smola (2006) and Dudík & Schapire (2006) the emphasis lay on *unconditional* density models: the dual of the above optimization problem. In particular, it follows that for H being the Shannon-Boltzmann entropy, the dual optimization problem is the maximum a posteriori estimation problem, which is what we are solving here.

In the conditional case, p denotes the collection of probabilities $p(y|x_i)$ and the operator $\mathbf{E}_{z\sim p}[\phi(z)] = \frac{1}{m}\sum_{i=1}^m \mathbf{E}_{y|p(y|x_i)}[\phi(x_i, y)]$ is the conditional expectation operator on the set of observations. Finally, $\mu = \frac{1}{m}\sum_{i=1}^m \phi(x_i, y_i)$, that is, it describes the empirical observations. We have two design parameters:

Function Space: Depending on which Banach Space norm we may choose to measure the deviation between μ and its expectation with respect to p in terms of e.g. the ℓ_2 norm, the ℓ_1 norm or the ℓ_∞ norm. The latter would lead to sparse coding and convex combinations.

Entropy and Regularity: Depending on the choice of entropy and divergence functionals we obtain a range of diverse estimators. For instance, if we were to choose the *unnormalized* entropy instead of the entropy, we would obtain AdaBoost style problems. We may also use Csiszar and Bregmann divergences.

The key point is that our reasoning of estimating μ_{XY} based on an aggregate of samples with unknown labels but known label proportions is still applicable. This means that it should be possible to design boosting algorithms and sparse coding methods which could operate on similarly restricted sets of observations.

5 Related Work and Alternatives

Transduction Gärtner et al. (2006) and Mann & McCallum (2007) performed transduction by enforcing a proportionality constraint on the unlabeled data via a Gaussian Process model. At first glance these methods might seem applicable for our problem but as stated in Section 1, they do require that we have at least some labeled instances of *all classes* at our disposition which need to be drawn in an unbiased fashion. This is clearly not the case in our setting.

Self consistent proportions Kück & de Freitas (2005) introduced a more informative variant of the binary multiple-instance learning, in which groups of instances are given along with estimates of the fraction of positively-labeled instances per group. This is then used to design a hierarchical probabilistic model which will generate consistent fractions. The optimization is solved via a MCMC sampler. While only described for a binary problem it could be easily extended to multiclass settings. Chen et al. (2006) and Musicant et al. (2007) use a similar approach with similar drawbacks, since we typically only have as many sets as classes.

Conditional Probabilities A seemingly valid alternative approach is to try building a classifier for $p(i|x)$ and subsequently recalibrating the probabilities to obtain $p(y|x)$. At first sight this may appear promising since this method is easily applicable in conjunction with most discriminative methods. The idea would be to reconstruct $p(y|x)$ by

$$p(y|x) = \sum_i \pi_{iy} p(i|x). \qquad (23)$$

However, this is not a useful estimator in our setting for a simple reason: it assumes the conditional independence $y \perp\!\!\!\perp x \mid i$, which obviously does not hold.

A simple example illustrates the problem. Assume that $\mathcal{X}, \mathcal{Y} = \{1, 2\}$ and that $p(y = 1|x = 1) = p(y = 2|x = 2) = 1$. In other words, the estimation problem is solvable since the classes are well separated. Moreover, assume that π is given by

$$\pi = \begin{bmatrix} 0.5-\epsilon & 0.5+\epsilon \\ 0.5 & 0.5 \end{bmatrix} \text{ for } 0 < \epsilon \ll 1.$$

Here, $p(i|x)$ is useless for estimating $p(y|x)$, since we will only exceed random guessing by at most ϵ.

Reduction to Binary For binary classification and real-valued classification scores we may resort to yet another fairly straightforward method: build a classifier which is able to distinguish between the sets X_1 and X_2 and subsequently threshold labels such that the appropriate fraction of observations in X_1 and X_2 respectively has its proper labels. Unfortunately, multi-class variants of this reduction are nontrivial and experiments show that even for the binary case this method is inferior to our approach.

Density Estimation Finally, one way of obtaining the probabilities $p(x, y|i)$ is to perform density estimation for each set of observations X_i. Subsequently we may use

$$p(x|y) = \sum_i \left[\pi^{-1}\right]_{yi} p(x, y|i) \tag{24}$$

to re-calibrate the probability estimates. Bayes' theorem is finally invoked to compute posterior probabilities. This tends to fail for high-dimensional data due to the curse of dimensionality in density estimation.

6 Experiments

Datasets: We used binary and three-class classification datasets from the UCI repository[2] and the LibSVM site.[3] Whenever separate training, validation and testing sets were available, we merged all of them prior to performing nested 10-fold cross-validation. We limited ourselves to three classes, since we need to generate as many splits as we have classes.

For the binary datasets we used half of the data for X_1 and the remainder for X_2. Moreover, we removed all instances of class 2 from X_1. That is, the conditional class probabilities in X_2 match those from the repository, whereas in X_1 their counterparts were deleted.

For the three-class datasets (protein, dna, senseit) we investigated two different partitions. In scenario A we used class 1 exclusively in X_1, class 2 exclusively in X_2, and a mix of all three classes weighted by $(0.5 \cdot p(1), 0.7 \cdot p(2), 0.8 \cdot p(3))$ to generate X_3. In scenario B we used the following splits

$$\begin{bmatrix} c_1 \cdot 0.4 \cdot p(1) & c_1 \cdot 0.15 \cdot p(2) & c_1 \cdot 0.14 \cdot p(3) \\ c_2 \cdot 0.1 \cdot p(1) & c_2 \cdot 0.15 \cdot p(2) & c_2 \cdot 0.06 \cdot p(3) \\ c_3 \cdot 0.5 \cdot p(1) & c_3 \cdot 0.7 \cdot p(2) & c_3 \cdot 0.8 \cdot p(3) \end{bmatrix}$$

Here the constants c_1, c_2 and c_3 are chosen such that the probabilities are properly normalized. As before, X_3 contained half of the data.

Model Selection: As stated, we perform nested 10-fold cross-validation. That is, we perform 10-fold cross-validation to assess the performance of the estimators. Within each fold, we perform 10-fold cross-validation to find a suitable value for the parameters.

For supervised classification, i.e. for discriminative sorting, such a procedure is quite straightforward, since we may directly optimize for classification error. For kernel density estimation, we used the log-likelihood as our criterion.

Due to the high number of hyper-parameters (at least 8) in MCMC it would be very difficult to perform *nested* 10-fold crossvalidation. Instead, we choose the *best* parameters from a simple 10-fold crossvalidation run. In other words, we are giving the MCMC method an unfair advantage over our approach by effectively reporting the best performance during the model selection procedure.

Finally, for the re-calibrated sufficient statistics $\hat{\mu}_{XY}$ we use the estimate of the log-likelihood on the validation set as the criterion for cross-validation, since no other quantity, such as classification errors is readily available for estimation.

Algorithms: For discriminative sorting we used an SVM with a Gaussian kernel. Its width was set to the median between pairs of observations, as suggested in (Schölkopf, 1997). Moreover, we chose the regularization parameter by cross-validation. For our re-calibrated sufficient statistics approach we adopted the same strategy. For kernel density estimation we used Gaussian kernels with diagonal densities. Cross-validation was performed over the kernel width. For MCMC 10 000 samples were generated after a burn in period of 10 000 steps (Kück & de Freitas (2005)).

Optimization: Bundle methods (Smola et al., 2007) are used for solving the regularized risk minimization problem in Algorithm 1. It performs successive Taylor approximations of the conditional log-likelihood and optimizes over these lower bounds. For our regularized log-likelihood problem, the solver converges to ϵ precision in $O(\log \epsilon)$ steps.

Results The results of binary and multiclass classification are summarized in Table 3. Our method clearly

[2] `http://archive.ics.uci.edu/ml/`

[3] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/`

beats kernel density estimation and performs favorably in comparison to discriminative sorting. Note that in terms of computation, our approach is somewhat cheaper to implement, since it effectively only needs to deal with a smaller sample size (only X rather than the union of all datasets X_i). The training time for our method is less than two minutes for all of the datasets, whereas for MCMC on average it takes 15 minutes and this may be even much longer whenever the number of active kernels and/or observations are high. However, for large number of partitions n, the approach described in the MCMC procedure might potentially have an edge over our method since at present we do not take full advantage of this setting. We believe, however, that this could be achieved easily by optimizing the condition number of the pseudoinverse of the then redundant linear system of equations. Note that kernel density estimation fails on two datasets due to numerical problems related to too high dimensionality.

On multiclass datasets we observe the same property. As described in Section 3.2, the quality of our minimizer of the log-posterior will depend on the mixing matrix and this is noticeable in the reduction of performance for the dense mixing matrix (scenario B) in comparison to the better conditioned sparse mixing matrix (scenario A). In other words, for ill conditioned π even our method has its limits, due simply to numerical considerations of effective sample size.

7 Conclusion

In this paper we showed a rather surprising result, namely that it is possible to consistently reconstruct the labels of a dataset if we only have, in addition to the observations, information about the proportions of occurrence of each class (in at least as many data aggregates as there are classes). In particular, we prove that up to constants, our algorithm enjoys the same rates of convergence afforded to methods which have full access to all label information.

This has beneficial applications in a range of applications like e-commerce, spam filtering and improper content detection. It may also suggest that techniques used to anonymize observations, in particular for demographic data, may not be really safe. (At least when it is possible to estimate the missing information in principle, provided enough data.) Experiments show that our approach is eminently feasible and performs favorably when compared to related methods.

Acknowledgments We thank Hendrik Kück for his MCMC codes and Choon Hui Teo for his bundle methods codes and the useful discussions with them. NICTA is funded by the Australian Government's Backing Australia's Ability and the Centre of Excellence programs. We received funding of the FP7 Network of Excellence by the European Union.

References

Altun, Y., & Smola, A. (2006). Unifying divergence minimization and statistical inference via convex duality. *COLT'06*, LNCS, 139–153. Springer.

Bartlett, P. L., & Mendelson, S. (2002). Rademacher and Gaussian complexities: Risk bounds and structural results. *JMLR*, *3*.

Chen, B., Chen, L., Ramakrishnan, R., & Musicant, D. (2006). Learning from aggregate views. *ICDE'06*, 3–12.

Dudík, M., & Schapire, R. E. (2006). Maximum entropy distribution estimation with generalized regularization. *COLT'06*, Springer.

Gärtner, T., Le, Q., Burton, S., Smola, A., & Vishwanathan, S. (2006). Large-scale multiclass transduction. *NIPS'06*

Kück, H., & de Freitas, N. (2005). Learning about individuals from group statistics. In *UAI'05*, 332–339.

Mann, G., & McCallum, A. (2007). Simple, robust, scalable semi-supervised learning via expectation regularization. In *ICML'07*.

Musicant, D., Christensen, J., & Olson, J. (2007). Supervised learning by training on aggregate outputs. In *IEEE ICDM*.

Schölkopf, B. (1997). *Support Vector Learning*. Oldenbourg Verlag.

Smola, A., Vishwanathan, S. V. N., & Le, Q. (2007). Bundle methods for machine learning. In *NIPS'07*.

Table 3. Classification error on the UCI/LibSVM database

Errors are reported in % with standard error. (%) ± SE. The best result and those results not significantly worse than it, are highlighted in boldface. We used a one-sided paired Welch t-test with 95% confidence level as reference.

`MM`: Mean Map, i.e. our method of re-estimating the sufficient statistics; `KDE`: Kernel Density Estimation; `DS`: Discriminative Sorting. This is only applicable for binary classification. `MCMC`: A sampling based hierarchical probabilistic model. `BA`: Baseline, i.e. the baseline obtained by predicting the majority class. †: The program did not complete, since the dimensionality was too high (only `KDE`). −: not applicable (only `DS` for multicategory problems). ‡: Algorithm as implemented does not scale to large datasets (only `MCMC`).

Data	MM	KDE	DS	MCMC	BA
iono	18.4±3.2	**17.5±3.2**	**12.2±2.6**	18.0±2.1	35.8
iris	**10.0±3.6**	**16.8±3.4**	**15.4±1.1**	**21.1±3.6**	29.9
optd	1.8±0.5	**0.7±0.4**	9.8±1.2	2.0±0.4	49.1
page	**3.8±2.3**	7.1±2.8	18.5±5.6	**5.4±2.8**	43.9
pima	**27.5±3.0**	34.8±0.6	34.4±1.7	**23.8±1.8**	34.8
tic	31.0±1.5	34.6±0.5	**26.1±1.5**	31.3±2.5	34.6
yeast	**9.3±1.5**	**6.5±1.3**	25.6±3.6	10.4±1.9	39.9
wine	**7.4±3.0**	**12.1±4.4**	18.8±6.4	**8.7±2.9**	40.3
wdbc	**7.8±1.3**	**5.9±1.2**	10.1±2.1	15.5±1.3	37.2
sonar	**24.2±3.5**	35.2±3.5	31.4±4.0	39.8±2.8	44.5
heart	**30.0±4.0**	38.1±3.8	**28.4±2.8**	**33.7±4.7**	44.9
brea	**5.3±0.8**	14.2±1.6	**3.5±1.3**	**4.8±2.0**	34.5
aust	**17.0±1.7**	33.8±2.5	**15.8±2.9**	30.8±1.8	44.4
svm3	**20.4±0.9**	27.2±1.3	25.5±1.5	24.2±0.8	23.7
adult	**18.9±1.2**	24.5±1.3	22.1±1.4	**18.7±1.2**	24.6
cleve	**19.1±3.6**	35.9±4.5	**23.4±2.9**	**24.3±3.1**	22.7
derm	**4.9±1.4**	27.4±2.6	**4.7±1.9**	14.2±2.8	30.5
musk	**25.1±2.3**	28.7±2.6	**22.2±1.8**	**19.6±2.8**	43.5
ger	**32.4±1.8**	41.6±2.9	37.6±1.9	**32.0±0.6**	32.0
cove	37.1±2.5	41.9±1.7	**32.4±1.8**	41.1±2.2	45.9
spli	**25.2±2.0**	35.5±1.5	**26.6±1.7**	28.8±1.6	48.4
giss	**10.3±0.9**	†	**12.2±0.8**	50.0±0.0	50.0
made	**44.1±1.5**	†	**46.0±2.0**	49.6±0.2	50.0
cmc	**37.5±1.4**	43.8±0.7	45.1±2.3	46.9±2.6	49.9
bupa	**48.5±2.9**	50.8±5.1	**40.3±4.9**	50.4±0.8	49.7
protA	**44.6±0.3**	60.2±0.1	-	65.3±1.9	61.2
protB	**45.7±0.6**	61.2±0.0	-	67.7±1.8	61.2
dnaA	**16.6±1.0**	30.7±0.8	-	37.7±0.8	40.5
dnaB	**29.1±1.0**	33.0±0.7	-	40.5±0.0	40.5
sensA	**19.8±0.1**	43.1±0.0	-	‡	43.2
sensB	**21.0±0.1**	43.1±0.0	-	‡	43.2