Support Vector Machines for Polycategorical Classification

Ioannis Tsochantaridis and Thomas Hofmann

Department of Computer Science, Brown University
Box 1910, Providence, RI 02912, USA
{it,th}@cs.brown.edu

Abstract. Polycategorical classification deals with the task of solving multiple interdependent classification problems. The key challenge is to systematically exploit possible dependencies among the labels to improve on the standard approach of solving each classification problem independently. Our method operates in two stages: the first stage uses the observed set of labels to learn a joint label model that can be used to predict unobserved pattern labels purely based on inter-label dependencies. The second stage uses the observed labels as well as inferred label predictions as input to a generalized transductive support vector machine. The resulting mixed integer program is heuristically solved with a continuation method. We report experimental results on a collaborative filtering task that provide empirical support for our approach.

1 Introduction

The standard supervised classification setting of inferring a single discriminant function based on a finite sample of labeled patterns has been investigated for decades. More recently, the question of how to make use of additional unlabeled examples has received a lot of attention. Such methods include the Fisher kernel [12] and maximum entropy discrimination method [13], maximum likelihood estimation via EM in text categorization [15], co-training [3], transductive inference [14], and kernel expansion methods [21]. The general hope in this line of research is that unlabeled data provide useful information about the pattern distribution that can be exploited to improve the classification performance, either by inducing an improved pattern representation or by enabling a more robust estimation of the discriminant function. In most cases, certain assumptions have to be made to guarantee that unlabeled data help to improve the performance.

In this paper, we investigate a more general setting, called *polycategorical classification*. Assume that we have multiple (binary) concepts represented by labeling processes $\mathcal{P}^j$, $1 \leq j \leq k$, i.e. each $\mathcal{P}^j$ denotes a joint probability distribution over labeled patterns $(\mathbf{x}, y) \in \Re^d \times \{-1, 1\}$. For each concept a sample set $\mathcal{S}^j$ is available, where samples in $\mathcal{S}^j$ have been generated i.i.d. according to $\mathcal{P}^j$. The goal is to simultaneously learn all k binary classification tasks. Of course, if these tasks were unrelated then one would apply a standard classification method to each sample set $\mathcal{S}^j$ independently. However, we assume that there are non-trivial dependencies between the labeling processes.

T. Elomaa et al. (Eds.): ECML, LNAI 2430, pp. 456–467, 2002.

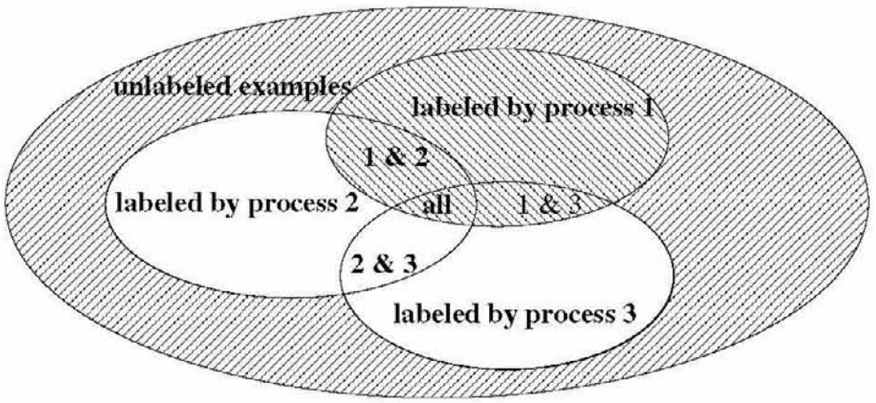

Fig. 1. Illustration of the set relationships between patterns labeled by different labeling processes

Notice how this setting can be viewed as a generalization of supervised learning with unlabeled data. For each classification problem $\mathcal{P}^j$ we have the corresponding sample set $\mathcal{S}^j$, but in addition we have patterns that occur in one or more of the other sample sets $\mathcal{S}^l$, $l \neq j$ and are thus annotated by labels from other labeling processes. This induces a rich set structure between the two extrema of examples that are correctly labeled according to the specific concept to be learned and examples that have not been labeled by any process (cf. Fig. 1). Intuitively, a pattern labeled by some other process $\mathcal{P}^l$ will on average be more useful w.r.t. $\mathcal{P}^j$ than an unlabeled pattern, in particular if there is some dependency between the two labeling processes. For example, if one knew *a priori* that two concepts are identical, $\mathcal{P}^i = \mathcal{P}^l$, then one could simply use the union $\mathcal{S}^i \cup \mathcal{S}^l$ for training of both concepts, which would drastically increase the number of available training examples. The goal in polycategorical classification is to exploit such dependencies to effectively augment the available training data in order to learn more accurate classification functions.

As a motivating example of why these types of problems are actually of relevance in practice, consider the scenario of information filtering in multi-user or multi-agent systems: Each user may define personalized categories for items such as text documents, movies or CDs. In particular, users may annotate items by whether an item is relevant (label $+1$) or irrelevant (label -1). These preferences or categories will be specific to a particular person, yet there might be similarities between user interests that induce dependencies among the category labels. For example, a document $\mathbf{x}_i$ labeled with $y_i^j \in \{-1, 1\}$ by some user or agent u_j might provide evidence about how another user or agent u_l might label this example, in particular if both users have shown similar responses on items

in $\mathcal{S}^j \cap \mathcal{S}^l$. There are thus two sources of evidence that are important in predicting y_i^j given $\mathbf{x}_i$: the input space representation (which is ordinarily exploited in classification) and dependencies between the labeling processes. The latter is closely related to a technique known as collaborative filtering [7, 17, 20] which makes predictions or recommendations purely based on inter-label dependencies. The success of these techniques in (commercial) recommendation systems shows that a substantial amount of cross-information is contained in user profiles. In polycategorical classification, one aims at combining these two sources of evidence, the item's feature vector representation and the dependencies between labels provided by different users. This problem has been discussed in the context of recommender systems as the problem of combining content-based and collaborative filtering, cf. for example [2, 16]. Yet, none of the methods proposed so far has shown how to generalize state-of-the-art discriminative methods to incorporate "collaborative" information.

The approach we propose can be decomposed into two almost independent stages. The first stage, deals with the problem of learning a probabilistic model of inter-label dependencies. In other words, the goal of the first stage is to estimate the joint label probability $P(\mathbf{y_i} = (y_i^1, \ldots, y_i^j, \ldots, y_i^k)|y(\mathbf{x}_i))$ for each pattern $\mathbf{x}_i$ that occurs in one of the sample sets. Here $y(\mathbf{x}_i)$ denotes the set of known labels for pattern $\mathbf{x}_i$. By marginalization we will then obtain prior probabilities $P(y_i^j|y(\mathbf{x}_i))$. Notice that these probabilities do not depend on the actual feature representation $\mathbf{x}_i$, but just on its observed (partial) label vector $y(\mathbf{x}_i)$. Since this estimate does not depend on the observation $\mathbf{x}_i$ we will also refer to the latter as the *prior label probability*. In the second stage the sample sets $\mathcal{S}^j$ are augmented by probabilistically labeled examples. The latter is then used as the input to a generalized transductive Support Vector Machine (SVM) to produce the desired classification functions. The challenge at this stage is how to combine the prior label estimates with the actual feature representations.

The rest of the paper is organized as follows: Section 2 describes a statistical model and a corresponding learning algorithm to compute predictions for unobserved labels based on observed labels. Section 3 deals with the generalization of the transductive SVM, while section 4 presents an experimental evaluation on a real-world data set.

2 Modeling Inter-label Dependencies

In this section, we will completely ignore the pattern representation and solely focus on modeling inter-label dependencies. If we denote by m the total number of distinct patterns $\mathbf{x}_i$, $m \equiv |\bigcup_j S^j|$, then all labels can be arranged in a $m \times k$ matrix $\mathbf{Y}$ with entries $y_i^j \in \{-1, ?, 1\}$, referring to the label the j-th labeling process assigns to the i-th pattern. Here we suggestively use the special symbol '?' to denote missing entries. In most cases, this matrix will be sparse in the sense that, $N = \sum_j |\mathcal{S}^j| \ll m \cdot k$, i.e. only a very small fraction of the entries will actually be observed. The goal is to estimate a matrix $\hat{\mathbf{Y}} \in [-1; 1]^{m \times k}$ with coefficients $\hat{y}_i^j$ corresponding to the expected value of the label Y_i^j under

the model, where Y_i^j denotes the random variable associated with the label of the i–th pattern with respect to the j–th labeling process.

2.1 Log-Likelihood Function

As an objective function between the probabilistic estimates $\hat{\mathbf{Y}}$ and the observed matrix $\mathbf{Y}$ it is natural to consider the log-likelihood,

$$l(\hat{\mathbf{Y}}; \mathbf{Y}) = \sum_{i,j:y_i^j=1} \log \frac{1+\hat{y}_i^j}{2} + \sum_{i,j:y_i^j=-1} \log \frac{1-\hat{y}_i^j}{2}\,, \tag{1}$$

which we want to maximize. Notice that $P(Y_i^j = \pm 1) = (1 \pm E[Y_i^j])/2$ and $\hat{y}_i^j = E[Y_i^j]$ by definition, so this just measures the average log-probability of the true label under the model leading to the approximation $\hat{Y}$.

2.2 Probabilistic Latent Semantic Analysis Model

There are many possibilities to define a joint label model. In this paper, we investigate the use of the probabilistic latent semantic analysis (pLSA) approach presented in [8]. We have previously applied this model in the context of collaborative filtering [10, 9], so it seems to be a good starting point for polycategorical classification. The pLSA model can be written in the following form:

$$\hat{y}_i^j = \sum_{r=1}^{R} \phi_i^r \psi_r^j, \quad \text{with} \quad \phi_i^r \in [-1;1],\ \psi_r^j \in [0;1] \quad \text{and} \quad \sum_{r=1}^{R} \psi_r^j = 1, \tag{2}$$

here R denotes the rank of the approximation, which we assume to be given for now. Notice that the total number of free parameters in the model is $R \cdot m + (R-1) \cdot k$, which can be far less than $m \cdot k$, if $R \ll \min\{m, k\}$. Intuitively, we can think of $(\phi_i^r)_r$ for each r as a prototype vector with probabilistic labels for each pattern $\mathbf{x}_i$ and of the coefficients $(\psi_r^j)_j$ as defining a convex combination of these vectors for the j-th classification problem. The pLSA model clearly bears a resemblance with soft-clustering models; concepts are probabilistically clustered into R groups, where each group corresponds to a super-concept that is characterized by a vector of probabilistic labels over patterns.

2.3 Expectation Maximization Algorithm

In fitting the above model, we would like to maximize the likelihood in Eq. (1) with respect to the parameters (ϕ, ψ). Explicitly inserting the model into the log-likelihood function and ignoring additive constants results in

$$l(\phi, \psi; \mathbf{Y}) = \sum_{i,j:y_i^j=1} \log \sum_r (1+\phi_i^r)\psi_r^j + \sum_{i,j:y_i^j=-1} \log \sum_r (1-\phi_i^r)\psi_r^j \tag{3}$$

Since the logarithm of a sum of terms is hard to optimize, we follow the standard Expectation-Maximization (EM) approach [5] of iteratively improving Eq. (3) until a local maximum is reached. We denote by $\phi(t)$, $\psi(t)$ the parameter estimates at time step t of the EM procedure. The goal in step $t+1$ is to improve on the estimate obtained in step t, which can be quantified in terms of the differential log-likelihood

$$\begin{aligned}\triangle l^{t+1} = &\sum_{i,j:y_i^j=1} \log \frac{\sum_r (1+\phi_i^r(t+1))\psi_r^j(t+1)}{\sum_r (1+\phi_i^r(t))\psi_r^j(t)} \\ &+ \sum_{i,j:y_i^j=-1} \log \frac{\sum_r (1-\phi_i^r(t+1))\psi_r^j(t+1)}{\sum_r (1-\phi_i^r(t))\psi_r^j(t)} \,. \end{aligned} \tag{4}$$

Using a concavity argument (Jensen's inequality) the differential log-likelihood can be lower bounded as follows

$$\begin{aligned}\triangle l^{t+1} \geq &\sum_{i,j:y_i^j=1} \sum_r h_{ij}^r(t) \log \frac{(1+\phi_i^r(t+1))\psi_r^j(t+1)}{(1+\phi_i^r(t))\psi_r^j(t)} \\ &+ \sum_{i,j:y_i^j=-1} \sum_r h_{ij}^r(t) \log \frac{(1-\phi_i^r(t+1))\psi_r^j(t+1)}{(1-\phi_i^r(t))\psi_r^j(t)} \,, \end{aligned} \tag{5}$$

where

$$h_{ij}^r(t) \equiv \begin{cases} \frac{(1+\phi_i^r(t))\psi_r^j(t)}{\sum_s (1+\phi_i^s(t))\psi_s^j(t)} \,, \text{ if } y_i^j = 1 \\ \frac{(1-\phi_i^r(t))\psi_r^j(t)}{\sum_s (1-\phi_i^s(t))\psi_s^j(t)} \,, \text{ if } y_i^j = -1. \end{cases} \tag{6}$$

After augmenting the lower bound in Eq. (5) by appropriate Lagrange multipliers to enforce the constraints on $\psi(t+1)$ one can set the gradient with respect to the new parameters estimates $\phi(t+1)$ and $\psi(t+1)$ to zero. This yields explicit solution of the following form,

$$\phi_i^r(t+1) = \frac{\sum_{j:y_i^j=1} h_{ij}^r(t) - \sum_{j:y_i^j=-1} h_{ij}^r(t)}{\sum_{j:y_i^j=1} h_{ij}^r(t) + \sum_{j:y_i^j=-1} h_{ij}^r(t)} \tag{7}$$

$$\psi_r^j(t+1) = \frac{\sum_{i:y_i^j=\pm 1} h_{ij}^r(t)}{\sum_s \sum_{i:y_i^j=\pm 1} h_{ij}^s(t)} \tag{8}$$

Eq. (6) corresponds to the E-step (expectation step), while Eqs. (7,8) form the M-step (maximization step). As can be seen, the previous parameter values only enter the M-step equations through the h_{ij}^r variables. Hence one can maximize the log-likelihood by alternating E-steps and M-steps until convergence is reached. The fact that the EM algorithm converges follows from the fact that the log-likelihood is increased in every step, while being bounded from above.

2.4 Comments

We have voted for the pLSA approach to model inter-label dependencies in this paper. However, we would like to point out that due to the modularity of our approach there are other options that could be employed and combined with the generalization of transductive inference SVMs presented in the subsequent section. For example, as an alternative to pLSA one could use graphical models such as Bayesian networks, where the label vector $\mathbf{y}^j$ for each labeling process can be treated as an instance and the Bayesian network consists of n nodes, one for every pattern $\mathbf{x}_i$. This approach to collaborative filtering has been pursued in [11]. We plan to investigate this research direction in future work.

3 Transductive SVM with Probabilistic Labels

3.1 Support Vector Machines

The Support Vector Machine (SVM) [22] is a popular classification method that is based on the principle of margin maximization. SVMs generalize the linear discrimination method known as the maximum margin hyperplane. Assume we parameterize linear classifiers by a weight vector $\mathbf{w}$ and a bias term b, $f(\mathbf{x}) = \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b)$. For linearly separable data $\mathcal{S}$, there are in general many hyperplanes that separate the training data perfectly. These hyperplanes form the so-called version space. The maximum margin principle suggests to choose $\mathbf{w}^*$ and b^* among the parameters in the version space so that they maximize the minimal distance (the *margin*) between the hyperplane and any of the training points.

SVMs generalize this idea in two ways. First of all in order to be able to deal with non-separable data sets one introduces slack variables ξ_i, one for every data point, and augments the objective function by an additional penalty term. The penalty term is usually proportional to the sum of the slack variables (L_1-norm), other choices include a squared error. With L_1-norm penalties one arrives at the following standard quadratic program for soft-margin SVMs:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i \geq 0, \quad i = 1, \ldots, n \end{aligned} \tag{9}$$

Here n denotes the number of available training patterns. After introducing Lagrange parameters α_i for the inequality margin constraints one can explicitly solve for $\mathbf{w}$, b and ξ_i to obtain the dual formulation (Wolfe dual, cf. [22])

$$\begin{aligned} \text{maximize} \quad & \theta(\alpha) = -\frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \sum_i \alpha_i \\ \text{subject to} \quad & C \geq \alpha_i \geq 0, \quad i = 1, \ldots, n \\ & \sum_i y_i \alpha_i = 0 \end{aligned} \tag{10}$$

Since the Gram matrix with coefficients $k_{ij} = \langle x_i, x_j \rangle$ is symmetric and positive semi-definite, the resulting problem is a convex quadratic minimization problem.

Furthermore, in SVM learning, one can take advantage of the fact that the dual function only depends on the gram matrix and replace the inner products between patterns in the input representation by an inner product computed via kernel functions K and simply define a new Gram matrix by $k_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. One then effectively gets a non-linear classification function in the original input representation. Details on kernel methods can be found in [19].

3.2 Transductive SVMs

In transductive SVMs (TSVMs), one aims at incorporating additional unlabeled data to get more reliable estimates for the optimal discriminant. The key observation is that a discriminant function which results in small margins for an unlabeled data point will not achieve a good separation, no matter what the true label of the unlabeled data point is. This idea is formalized in TSVMs by introducing additional integer variables $\bar{y}_i \in \{-1, 1\}$ to model the *unknown* labels and to optimize a joint objective over the integer variables and the parameters $\mathbf{w}, b$ or - equivalently - the dual parameters α. In the following, we use the primal formulation, mainly because it is more comprehensible for the purpose of this presentation. We assume for simplicity that the labeled patterns are numbered from $1, \ldots, n$ and the unlabeled examples are numbered from $n+1, \ldots, m$.

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i + \bar{C}\sum_{i=n+1}^{m}\xi_i, \quad \text{over} \quad \mathbf{w}, \xi \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \qquad i = 1, \ldots, n \\ & \bar{y}_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \qquad i = n+1, \ldots, m \\ & \xi_i \geq 0, \qquad i = 1, \ldots, m \\ & \bar{y}_i \in \{-1, 1\}, \qquad i = n+1, \ldots, m \end{aligned} \tag{11}$$

Alternative formulations of the TSVM problem that avoid the use of integer variables and result in non-convex optimization have been investigated in [4].

For large problems, there is (currently) no hope to find the exact solution to the above mixed integer quadratic program. Instead, one has to resort to optimization heuristics to compute an approximate solution. The heuristic proposed in [14] optimizes the integer variables in an outer loop and then solves the standard SVM-QP in the inner loop. Since [14] proposes to keep the proportion of positive and negative labels constant, labels $\bar{y}_i$ and $\bar{y}_j$ with $\bar{y}_i \neq \bar{y}_j$ are swapped between pairs of unlabeled examples $\mathbf{x}_i$, $\mathbf{x}_j$, if this reduces the overall objective function. Finally, there is yet another outer loop which employs a continuation method to reduce the sensitivity of the optimization heuristic with respect to local minima. Starting from a small value for the penalty $\bar{C}$, $\bar{C}$ is iteratively increased until it reaches a given final value $\bar{C}^* \leq C$. Notice that for small values of $\bar{C}$, the labeled data dominate the objective function, so that the TSVM solution will be close to the SVM solution which can be computed exactly. As $\bar{C}$

is increased, the penalty for having unlabeled data points close to the decision boundary increases and more attention will be paid to the configuration of the unlabeled data points and the imputed labels $\bar{y}_i$.

3.3 SVM with Probabilistic Labels

In order to use the prior label estimates derived from inter-label dependencies, we propose to generalize TSVMs in a way that they can handle "uncertain" labels, where we think of labeled and unlabeled patterns as extreme cases of uncertain labels. Hence let us assume label probabilities $\hat{y}_i^j$ for $i = 1, \ldots, n$ are given, where $\hat{y}_i^j = y_i^j$ for observed labels. We will drop the superscript j to refer to a generic labeling process. Let us introduce binary integer variables $\bar{y}_i \in \{-1, 1\}$ as in TSVMs and define the following optimization problem (using the same numbering convention as before)

$$\text{minimize} \qquad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i + \bar{C}\left[\sum_{i=n+1}^{m}\xi_i + DH(\mathbf{y}, \hat{\mathbf{y}})\right] \tag{12}$$

$$\text{where} \qquad H(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=n+1}^{m}\left[\frac{1+\bar{y}_i}{2}\log\frac{1+\hat{y}_i}{2} + \frac{1-\bar{y}_i}{2}\log\frac{1-\hat{y}_i}{2}\right] \tag{13}$$

$$\begin{aligned}\text{subject to} \qquad & y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i, \qquad i = 1, \ldots, n \\ & \bar{y}_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i, \qquad i = n+1, \ldots, m \\ & \xi_i \geq 0, \qquad i = 1, \ldots, m \\ & \bar{y}_i \in \{-1, 1\}, \qquad i = n+1, \ldots, m \end{aligned} \tag{14}$$

The function H measures the cross entropy between the (deterministically) imputed labels $\bar{y}_i$ and the predictions derived from the inter-label model, $\hat{y}_i$. It acts as a soft penalty that penalizes labels that deviate from the prior predictions. The relative weight $D \in \Re^+$ controls the influence of this penalty relative to the margin penalty encoded in the slack variables ξ_i, thereby trading off the inter-label information encoded in $\hat{y}_i$ with the information encoded in the feature representation $\mathbf{x}_i$. In practice, one can use a cross-validation scheme to determine the optimal value for D. Notice that in the special case of $\hat{y}_i = 0$, i.e. in the case of a maximally entropic prior with a label uncertainty of one bit, our formulation reduces to TSVM, since the corresponding log-ratio term in H will reduce to a constant.

For a given hyperplane, the update step for the labels $\bar{y}_i$ is simple. First notice that the slack variable ξ_i will depend on $\bar{y}_i$, because the associated constraint involves $\bar{y}_i$. Since ξ_i is non-negative and large values are penalized, the optimal choice is given by

$$\xi_i = \max\{0, 1 - \bar{y}_i\gamma_i\} = 1 - \min\{1, \bar{y}_i\gamma_i\}, \quad \text{where} \quad \gamma_i \equiv \langle \mathbf{w}, \mathbf{x}_i\rangle + b\,. \tag{15}$$

Notice that for data points that are strictly inside the margin tube, this value will be positive for both, $\bar{y}_i = 1$ and $\bar{y}_i = -1$. It is now straightforward to

Initialize $\bar{C}$ to a small value ϵ
Initialize integer variables $\bar{y}_i = \text{sign}(\hat{y}_i)$
Repeat until $\bar{C} = \bar{C}^*$
  Repeat until convergence, i.e. no integer variable needs to be changed
    Compute the optimal hyperplane $\mathbf{w}, b$, given the integer variables $\{\bar{y}_i\}$
    Re-compute the integer variables $\{\bar{y}_i\}$ for given parameters $\mathbf{w}, b$
  end
  $\bar{C} = 2 * \bar{C}$
end

Fig. 2. Generalized SVM algorithm for polycategorical classification

compute the optimal value for $\bar{y}_i$ by comparing the cost induced by the two possible choices,

$$h_i^+ = \max\{0, 1 - \gamma_i\} - D \log(1 + \hat{y}_i)/2 \tag{16}$$

$$h_i^- = \max\{0, 1 + \gamma_i\} - D \log(1 - \hat{y}_i)/2 \tag{17}$$

$$h_i^- - h_i^+ = \min\{1, \gamma_i\} - \min\{1, -\gamma_i\} + D \log \frac{1 + \hat{y}_i}{1 - \hat{y}_i} \tag{18}$$

$$= \begin{cases} (\gamma_i + 1) + D \log \frac{1+\hat{y}_i}{1-\hat{y}_i}, & \text{for } \gamma_i \geq 1 \\ (\gamma_i - 1) + D \log \frac{1+\hat{y}_i}{1-\hat{y}_i}, & \text{for } \gamma_i \leq -1 \\ 2\gamma + D \log \frac{1+\hat{y}_i}{1-\hat{y}_i}, & \text{otherwise} \end{cases}$$

$$\bar{y}_i^* = \text{sign}(h_i^- - h_i^+) \tag{19}$$

Notice that if $\hat{y}_i \gamma_i \geq 0$, both contributions are in agreement, i.e. the data point is on the +1/-1 side of the hyperplane and the prior probability for a +1/-1 label is higher. However, if $\hat{y}_i \gamma_i < 0$, these two contributions are in conflict, in which case the weighting factor D determines how to compare the log ratio with the margin difference and which one to favor, the prior belief or the location of the feature vector relative to the current decision boundary. The complete algorithm is described in pseudo-code in Fig. 2

4 Experiments and Results

4.1 Data Generation and Preprocessing

In order to experimentally verify the proposed method for polycategorical classification, we have used the well–known EachMovie [6] data set which contains about 1600 movies and more than 60,000 user profiles with a total number of approximately 2.8 million labels/votes. We have augmented this data set with movie synopses based on descriptions provided at [1]. The movie pages have been automatically crawled, parsed and indexed. Movies have then been represented as vectors $\mathbf{x}_i$ in the standard term frequency representation used in the vector space model [18] for information retrieval. We have been able to obtain

Table 1. Classification accuracy results on the augmented EachMovie data set. The first row denotes accuracies obtained by ignoring the feature representation, the second row summarizes the results by using the SVM. The first columns refers to the case of no model for inter-label dependencies, the second column to the popularity model and the third column to the pLSA model

	independent classification	popularity baseline	pLSA model
no features		66.0%	73.0%
SVM	63.0%	68.6%	**74.3**%

descriptions for 1217 movies which constitutes the set of pattern used in the experiments. For computational reasons, we have subsampled the database and randomly selected a subset of 1000 user profiles among the profiles with at least 100 votes. The actual votes have been converted into binary labels by thresholding the ratings: 4-5 stars have been mapped to a +1 and 0-3 stars to a -1 label. For each user, the available labels have been randomly split into a training set (90%) and a test set (10%).

4.2 Experiments

We have performed the following experimental comparison. For each of the 1000 users, we have trained a SVM just based on the feature representation. In all the experiments we have restricted our attention to linear kernels. This provides a benchmark that is purely based on the extracted content information. Moreover, we have trained a pLSA model to predict unobserved labels based on the observed label matrix $\mathbf{Y}$. We have chosen a model with $R = 200$, by coarse optimization based on the predictive log-likelihood. This provides a benchmark that is purely based on label dependencies. In addition, we have investigated the use of a simple popularity baseline model which estimates the expected label by uniformly averaging over the population of users. Finally, the pLSA predictions as well as the popularity predictions have been used as prior predictions for the polycategorical SVM algorithm.

Table 1 summarizes the results in terms of classification accuracy. First of all, notice that the use of inter-label dependencies leads to a significant absolute improvement of more than 11% in terms of classification accuracy compared to the SVM learning. This clearly demonstrates that a lot can be gained by the polycategorical treatment compared to the straightforward approach of independently solving each classification problem. It also shows that in this particular example, the content features are relatively weak for discrimination between movies, at least given the available training sample size. It seems that individual words occurring in short movie summaries are rather weakly correlated with most users' preferences. Secondly, notice that using the features representation

yields a small yet consistent improvement in the performance of both, the simple popularity model as well as the pLSA model for inter-label dependencies. Despite the fact that the "collaborative" information between labels seems to be more precise than the information encoded in the content descriptions, there is still extra information that can be gained from the feature representation. One also sees that the difference is larger in the case of the popularity baseline - 2.6% vs. 1.1% gain in accuracy. One also has to consider that previous experiments [9] have shown that pLSA is a highly competitive collaborative filtering technique, so improving upon it is not trivial. We speculate that the improvement will be larger in cases, where both the feature representation and the inter-label dependencies yield predictions of comparable accuracy. Along these lines, we are currently investigating ways to extract stronger features like genre information for movies.

5 Conclusion

We have presented a novel approach to jointly solving large scale classification problems with many interdependent concepts. The proposed method uses state-of-the-art classification methods, namely SVMs, to learn from feature representations. In order to incorporate inter-label dependencies the transductive SVM framework has been extended to deal with weak label information. An efficient optimization heuristic has been proposed to compute approximate solutions of the resulting mixed integer program. On a real-world data set, the proposed method outperforms both, methods that are purely based on a feature representation and methods that are only taking into account inter-label dependencies.

Acknowledgments

We would like to thank the Compaq Equipment Corporation for making the EachMovie data set available. This work has in part be sponsored by an NSF-ITR grant, award number IIS-0085836.

References

[1] http://www.allmovie.com. 464

[2] C. Basu, H. Hirsh, and W. W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the AAAI/IAAI*, pages 714–720, 1998. 458

[3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. 11th Annual Conf. Computational Learning Theory*, pages 92–100, 1998. 456

[4] A. Demiriz and C. Bennett. Optimization approaches to semi-supervised learning. In M. C. Ferris, O. L. Mangasarian, and J. S. Pang, editors, *Applications and Algorithms of Complementarity*. Kluwer Academic Publishers, Boston, 2000. 462

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B*, 39:1–38, 1977. 460

[6] http://research.compaq.com/SRC/eachmovie. 464

[7] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992. 458

[8] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning Journal*, 42(1):177–196, 2001. 459

[9] T. Hofmann. What people (don't) want. In *European Conference on Machine Learning (ECML)*, 2001. 459, 466

[10] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *Proceedings of the International Joint Conference in Artificial Intelligence*, 1999. 459

[11] C. Kardie J. Breese, D. Heckerman. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 1998. 461

[12] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems 11*. MIT Press, 1998. 456

[13] T. Jaakkola, M. Meila, and T. Jebara. Maximum entropy discrimination. In *Neural Information Processing Systems 12*. MIT Press, 1999. 456

[14] T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. 16th International Conf. on Machine Learning*, pages 200–209. Morgan Kaufmann, San Francisco, CA, 1999. 456, 462

[15] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2-3):103–134, 2000. 456

[16] A. Popescul, L. H. Ungar, D. M. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *17th Conference on Uncertainty in Artificial Intelligence*, 2001. 458

[17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, 1994. 458

[18] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983. 464

[19] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002. 462

[20] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating 'word of mouth'. In *Proceedings of the Computer Human Interaction Conference (CHI95)*, 1995. 458

[21] M. Szummer and T. Jaakkola. Kernel expansions with unlabeled examples. In *Advances in Neural Information Processing Systems 13*. MIT Press, 2000. 456

[22] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, Berlin, 1995. 461