Transfer Learning for Image Classification with Sparse Prototype Representations

Ariadna Quattoni[+], Michael Collins[+], and Trevor Darrell[+, *]
[+]MIT CSAIL
[*]UC Berkeley EECS & ICSI
{ariadna,mcollins}@csail.mit.edu, trevor@eecs.berkeley.edu

Abstract

To learn a new visual category from few examples, prior knowledge from unlabeled data as well as previous related categories may be useful. We develop a new method for transfer learning which exploits available unlabeled data and an arbitrary kernel function; we form a representation based on kernel distances to a large set of unlabeled data points. To transfer knowledge from previous related problems we observe that a category might be learnable using only a small subset of reference prototypes. Related problems may share a significant number of relevant prototypes; we find such a concise representation by performing a joint loss minimization over the training sets of related problems with a shared regularization penalty that minimizes the total number of prototypes involved in the approximation. This optimization problem can be formulated as a linear program that can be solved efficiently. We conduct experiments on a news-topic prediction task where the goal is to predict whether an image belongs to a particular news topic. Our results show that when only few examples are available for training a target topic, leveraging knowledge learnt from other topics can significantly improve performance.

1. Introduction

Learning visual category models from a small number of training examples is an important challenge in computer vision. It is well known that people can learn new categories from very few examples; to achieve similar performance with machines it is likely that visual category learning methods will need to leverage available prior knowledge derived from previously learned categories, as well as exploit unlabeled data to discover structure in the environment.

Semi-supervised learning methods exist which can find structure in available unlabeled data and use that structure to improve performance on a supervised task (e.g., [17]), but don't generally exploit knowledge learned from previous supervised tasks. A common goal of transfer learning methods is to discover representations from previous tasks that make learning a future related task possible with few examples. Existing methods for transfer learning often learn a prior model or linear manifold over classifier parameters [8, 2], discover a sparse set of common features [13, 3, 20, 6], or use a representation based on classifier outputs from related tasks [24], but do not generally take advantage of unlabeled data.

In this paper we develop a visual-category learning algorithm that can learn an efficient representation from a set of related tasks and which explicitly takes advantage of unlabeled data. Our method uses unlabeled data to define a prototype representation based on computing kernel distances to a potentially large set of unlabeled points. However, each visual category model may depend only on the distance to a small set of prototypes; if these prototypes were known, we might be able to learn with fewer examples by removing irrelevant prototypes from the feature space. In general we will not know this a priori, but related problems may share a significant number of such prototypes. Our transfer learning method identifies the set of prototypes which are jointly most relevant for a given set of tasks, and uses that reduced set of points as the feature space for future related tasks. Our experiments show that using the transferred representation significantly improves learning with small training sets when compared to the original feature space.

One of the advantages of our transfer learning method is that the prototype representation is defined using an arbitrary kernel function [4]. Recent progress has shown that visual category recognition can improve with the use of kernels that are optimized to particular tasks [22].

We discover an optimal subset of relevant prototypes with a jointly regularized optimization that minimizes the total number of reference prototypes involved in the approximation. Previous approaches in vision to joint feature learning have employed a greedy boosting approach [20]; our joint regularization exploits a norm derived from simultaneous sparse signal approximation methods [21], leading to an optimization problem that can be expressed as a linear

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

program.

We evaluate our method on a news-topic prediction task, where the goal is to predict whether an image belongs to a particular news-topic. Our results show that learning a representation from previous topics does improve future performance when supervised training data are limited.

In the following Section we review related work relevant to transfer learning of visual categories. We then describe our method for creating a prototype representation based on kernel distances to unlabeled datapoints, followed by our prototype selection technique based on a joint sparse optimization. Finally, we describe our experimental regime and discuss our results.

2. Previous work

Transfer learning has had a relatively long history in machine learning. Broadly speaking, the goal of transfer learning is to use training data from related tasks to aid learning on a future problem of interest.

Most transfer learning techniques can be broadly grouped into two categories: learning intermediate representations [19, 5, 16, 2], and learning small sets of relevant features that are shared across tasks [11, 13, 3, 1, 23]. The methods most closely related to our approach belong to the second category and are those of Obozinski *et al*. [13] , Argyriou *et al*. [3] and Amit *et al*. [1] on joint sparse approximation for multi-task learning.

Obozinki *et al*. proposed a joint regularization framework that extends l_1 regularization for a single task to a multi-task setting by penalizing the sum of l_2-norms of the block of coefficients associated with each feature across tasks. This can be expressed as a joint optimization that combines l_1 and l_2 norms on the coefficients matrix. They also suggest the use of the l_∞ norm in place of the l_2 norm. Argyriou *et al*. extended the formulation of Obozinki et al. by introducing an intermediate hidden representation. They developed an iterative algorithm to jointly optimize for the hidden representation and the classifiers' coefficients. Amit *et al*. [1] proposed an alternative joint regularization framework based on a trace-norm penalty on the coefficients matrix, where the trace-norm is defined as the sum of the matrix's singular values. For optimization they derived a method that performs gradient descent on a smooth approximation of the objective.

There are several differences between these feature sharing approaches and our prototype selection algorithm. One important difference is our choice of joint regularization norm, which allows us to express the optimization problem as a linear program. Additionally, while previous feature sharing approaches build a joint sparse classifier on the feature space [13], or a random [13] or hidden [3, 1] projection of that feature space, our method discovers a set of discriminative prototypes that can be transferred to solve a future problem. We utilize unlabeled data to compute a prototype representation and perform a joint sparse approximation on the prototype space.

Transfer learning for visual category recognition has received significant attention in recent years [8, 18, 12, 24, 9, 10, 20]. In the context of generative object models, Fei-Fei *et al*. proposed a Bayesian transfer learning approach for object recognition [8] where a common prior over visual classifier parameters is learnt; their results show a significant improvement when learning from a few labeled examples. Also in the context of constellation models, Zweig [24] has investigated transfer learning with a method based on combining object classifiers from different hierarchical levels into a single classifier.

In the context of discriminative (maximum margin) object models Fink [9] developed a method that learns distance metrics from related problems. Hertz *et al*. [10] reports a method based on training binary max margin classifiers on the product space of pairs of images, thus creating a distance function based on the output of those classifiers. Our method differers significantly from this approaches in that instead of learning a distance function we learn a sparse representation on a prototype space.

The work of Torralba *et al*. [20] on feature sharing for multi-class classification includes a joint boosting algorithm where the weak learners (step functions applied to individual features) are greedily selected so they can both separate well some bipartition of the set of classes and reduce the average empirical risk on all classifiers.

In the context of semi-supervised learning Raina *et al*. [15] described an approach that learns a sparse set of high-level features (i.e. linear combinations of the original features) from unlabeled data using a sparse coding technique.

Our approach builds on the work of Balcan *et al*. [4], who proposed the use of a representation based on kernel distances to unlabeled datapoints, and the work of Tropp [21] on simultaneous sparse approximation. The latter problem involves approximating a set of signals by a linear combination of elementary signals while at the same time minimizing the total number of elementary signals involved in the approximation. Tropp proposed a joint optimization with a shared regularization norm over the coefficients of each signal and gave theoretical guarantees for this approach. In our sparse transfer prototype algorithm we make use of the norm proposed by Tropp and Obozinski *et al*., but we use it to learn a sparse set of prototypes that are discriminative in a given domain.

3. Learning a sparse prototype representation from unlabeled data and related tasks

We now describe our sparse prototype learning algorithm for learning a representation from a set of unlabeled

Input 1: Unlabeled dataset
$U = \{x_1, x_2, \ldots, x_p\}$ for $x_i \in \mathcal{X}$ (e.g, $\mathcal{X} = \mathbb{R}^d$)
Input 2: Collection of related problems
$C = \{T_1, \ldots, T_m\}$ where
$T_k = \{(x_1^k, y_1^k), (x_2^k, y_2^k), \ldots, (x_{n_k}^k, y_{n_k}^k)\}$
for $x \in \mathcal{X}$ and $y \in \{+1, -1\}$
Input 3: Kernel function
$k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$
Input 4: Threshold θ
Input 5: Regularization constants λ_k, for $k = 1 : m$

Step 1: Compute the prototype representation

- Compute the kernel matrix for all unlabeled points :
  $K_{ij} = k(x_i, x_j)$ for $x_i \in U$, $x_j \in U$
- Compute eigenvectors of of K by performing SVD : Compute a projection matrix A of dimension $p \times p$ by taking the eigenvectors of K; where each column of A corresponds to an eigenvector.
- Project all points x_i^k in C to the prototype space:
  $z(x_i^k) = A^\top \varphi(x_i^k)$
  where
  $\varphi(x) = [k(x, x_1), \ldots, k(x, x_p)]^\top, \;\; x_i \in U$

Step 2: Discover relevant prototypes by joint sparse approximation
Let W be a $p \times m$ matrix where W_{jk} corresponds to the j-th coefficient of the k-th problem.

- Choose the optimal matrix W^* to be:
  $\min_{W,\boldsymbol{\varepsilon}} \; \sum_{k=1}^m \lambda_k \sum_{i=1}^{n_k} \varepsilon_i^k + \sum_{j=1}^p \max_k |W_{jk}|$
  s.t. for $k = 1 : m$ and $i = 1 : n_k$
  $y_i^k \mathbf{w}_k^\top z(x_i^k) \geq 1 - \varepsilon_i^k$
  $\varepsilon_i^k \geq 0$
  where $\mathbf{w}_k$ is the k-th column of W, corresponding to the parameters for problem k.

Step 3: Compute the relevant prototype representation

- Define the set of relevant prototypes to be:
  $R = \{r : \max_k |W_{rk}^*| > \theta\}$
- Create projection matrix B by taking all the columns of A corresponding to the indexes in R. B is then a $p \times h$ matrix, where $h = |R|$.
- Return the representation given by:
  $v(x) = B^\top \varphi(x)$

Output: The function $v(x) : \mathbb{R}^d \rightarrow \mathbb{R}^h$

Algorithm 1: The sparse prototype representation learning algorithm.

data points $U = \{x_1, x_2, \ldots, x_p\}$ and a collection of training sets of related problems $C = \{T_1, \ldots, T_m\}$, where $T_k = \{(x_1^k, y_1^k), \ldots, (x_{n_k}^k, y_{n_k}^k)\}$. In all cases, $x \in \mathcal{X}$ (for example, $\mathcal{X} = \mathbb{R}^d$) and $y \in \{+1, -1\}$.

In the following subsections we describe the three main steps of our algorithm. In the first step, we compute a prototype representation using the unlabeled dataset. In the second step, we use data from related problems to select a small subset of prototypes that is most discriminative for the related problems. Finally, in the third step we create a new representation based on kernel distances to the the selected prototypes. We will use the sparse prototype representation to train a classifier for a target problem. Algorithm 1 provides pseudo-code for the algorithm. The next three subsections describe the three steps of the algorithm in detail.

3.1. Computing the prototype representation

Given an unlabeled data set $U = \{x_1, x_2, \ldots, x_p\}$ and a kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, the first step of our algorithm computes a prototype representation based on kernel distances to the unlabeled data points in U.

We create the prototype representation by first computing the kernel matrix K of all points in U, i.e. $K_{ij} = k(x_i, x_j)$ for x_i and x_j in U. We then create a projection matrix A formed by taking all the eigenvectors of K corresponding to non-zero eigenvalues (the eigenvectors are obtained by performing SVD on K). The new representation is then given by:

$$z(x) = A^\top \varphi(x), \tag{1}$$

where $\varphi(x) = [k(x, x_1), \ldots, k(x, x_p)]^\top$, and $x_i \in U$. We will refer to the columns of A as prototypes.

This representation was first introduced by Balcan *et al.* [4], who proved it has important theoretical properties. In particular, given a target binary classification problem and a kernel function, they showed that if the classes can be separated with a large margin in the induced kernel space then they can be separated with a similar margin in the prototype space. In other words, the expressiveness of the prototype representation is similar to that of the induced kernel space. By means of this technique, our joint sparse optimization can take the advantage of a kernel function without having to be explicitly kernelized.

Another possible method for learning a representation from the unlabeled data would be to create a $p \times h$ projection matrix L by taking the top h eigenvectors of K and defining the new representation $g(x) = L^\top \varphi(x)$; we call this approach the low rank technique. The method we develop in this paper differs significantly from the low rank approach in that we use training data from related problems to select discriminative prototypes, as we describe in the next step. In the experimental Section we show the advantage of our approach compared to the low rank method.

3.2. Discovering relevant prototypes by joint sparse approximation

In the second step of our algorithm we use a collection of training sets from related problems, $C = \{T_1, \ldots, T_m\}$, where $T_k = \{(x_1^k, y_1^k), \ldots, (x_{n_k}^k, y_{n_k}^k)\}$, to find a subset of discriminative prototypes. Our method is based on the search for a sparse representation in the prototype space that is jointly optimal for all problems in C.

Consider first the case of learning a single sparse classifier on the prototype space of the form:

$$f(x) = \mathbf{w}^\top z(x), \tag{2}$$

where $z(x) = A^\top \varphi(x)$ is the representation described in step 1 of the algorithm. A sparse model will have a small number of prototypes with non-zero coefficients. Given a training set with n examples, a natural choice for parameter estimation in such a setting would be to take the optimal parameters $\mathbf{w}^*$ to be:

$$\min_{\mathbf{w}} \lambda \sum_{i=1}^{n} l(f(x_i), y_i) + \sum_{j=1}^{p} |w_j|. \tag{3}$$

The left term of Equation (3) measures the error that the classifier incurs on training examples, measured in terms of a loss function l. In this paper we will use the hinge loss, given by $l(f(x), y) = \max(0, (1 - yf(x)))$.

The right hand term of Equation (3) is an l_1 norm on the coefficient vector which promotes sparsity. In the context of regression Donoho [7] has proven that the solution with smallest l_1 norm is also the sparsest solution, i.e. the solution with the least number of non-zero coefficients. The constant λ dictates the trade off between sparsity and approximation error on the data.

For transfer learning, our goal is to find the most discriminative prototypes for the problems in C, i.e. find a subset of prototypes R such that each problem in C can be well approximated by a sparse function whose non-zero coefficients correspond to prototypes in R. Analogous to the single sparse approximation problem, we will learn jointly sparse classifiers on the prototype space using the training sets in C. The resulting classifiers will share a significant number of non-zero coefficients, i.e active prototypes. Let us define a $p \times m$ coefficient matrix W, where W_{jk} corresponds to the j-th coefficient of the k-th problem. In this matrix, the k-th column of W is the set of coefficients for problem k, which we will refer to as $\mathbf{w}_k$, while the j-th row corresponds to the coefficients of prototype j across the m problems. The m classifiers represented in this matrix correspond to:

$$f_k(x) = \mathbf{w}_k^\top z(x). \tag{4}$$

It is easy to see that the number of non-zero rows of W corresponds to the total number of prototypes used by any of the m classifiers. This suggests that a natural way of posing the joint sparse optimization problem would be to choose the optimal coefficient matrix W^* to be:

$$\min_{W} \sum_{k=1}^{m} \lambda_k \sum_{i=1}^{n_k} l(y_i^k, f_k(x_i^k)) + \|W\|_{r0} \tag{5}$$

where $\|W\|_{r0}$ is a pseudo-norm that counts the number of non-zero rows in W [1].

As in the single sparse approximation problem, the two terms in Equation (5) balance the approximation error against some notion of sparsity. Here, the left hand term minimizes a weighted sum of the losses incurred by each classifier on their corresponding training sets, where λ_k weights the loss for the k-th problem. The right hand term minimizes the number of prototypes that have a non-zero coefficient for some of the related problems. Due to the presence of the $r0$ pseudo-norm in the objective, solving (5) might result in a hard combinatorial problem. Instead of solving it directly we use a convex relaxation of the $r0$ pseudo-norm suggested in the literature of simultaneous sparse approximation [21], the (l_1, l_∞) norm, which takes the following form:

$$\sum_{j=1}^{p} \max_{k} |W_{jk}| \tag{6}$$

Using the (l_1, l_∞) norm we can rewrite Equation (5) as:

$$\min_{W} \sum_{k=1}^{m} \lambda_k \sum_{i=1}^{n_m} l(y_i^k, f_k(x_i^k)) + \sum_{j=1}^{p} \max_{k} |W_{jk}| \tag{7}$$

The right hand term on Equation (7) promotes joint sparsity by combining an l_1 norm and an l_∞ norm on the coefficient matrix. The l_1 norm operates on a vector formed by the maximum absolute values of the coefficients of each prototype across problems, encouraging most of these values to be 0. On the other hand, the l_∞ norm on each row promotes non-sparsity among the coefficients of a prototype. As long as the maximum absolute value of a row is not affected, no penalty is incurred for increasing the value of a row's coefficient. As a consequence only a few prototypes will be involved in the approximation but the prototypes involved will contribute in solving as many problems as possible.

When using the hinge loss the optimization problem in Equation (7) can be formulated as a linear program:

$$\min_{W, \boldsymbol{\varepsilon}, \mathbf{t}} \sum_{k=1}^{m} \lambda_k \sum_{i=1}^{n_k} \varepsilon_i^k + \sum_{j=1}^{p} t_j \tag{8}$$

[1] The number of non-zero rows is the number of rows for which at least one of its elements is different than 0.

such that for $j = 1 : p$ and $k = 1 : m$

$$-t_j \leq W_{jk} \leq t_j \tag{9}$$

and for $k = 1 : m$ and $i = 1 : n_k$

$$y_i^k \mathbf{w}_k^\top z(x_i^k) \geq 1 - \varepsilon_i^k \tag{10}$$

$$\varepsilon_i^k \geq 0 \tag{11}$$

The constraints in Equation (9) bound the coefficients for the j-th prototype across the m problems to lie in the range $[-t_j, t_j]$. The constraints in Equations (10) and (11) impose the standard slack variable constraints on the training samples of each problem.

3.3. Computing the relevant prototype representation

In the last step of our algorithm we take the optimal coefficient matrix W^* computed in Equation (8) of Step 2 and a threshold θ and create a new representation based on kernel distances to a subset of relevant prototypes. We define the set of relevant prototypes to be:

$$R = \{r : \max_k |W_{rk}^*| > \theta\}. \tag{12}$$

We construct a new projection matrix B by taking all the columns of A corresponding to indices in R, where A is the matrix computed in the first step of our algorithm. B is then a $p \times h$ matrix, where $h = |R|$. The new sparse prototype representation is given by:

$$v(x) = B^\top \varphi(x). \tag{13}$$

When given a new target problem we project every example in the training and test set using $v(x)$. We could potentially train any type of classifier on the new space; in our experiments we chose to train a linear SVM.

4. Experiments

We created a dataset consisting of 10,382 images collected from the Reuters news web-site[2] during a period of one week. Images on the Reuters website have associated story or topic labels, which correspond to different topics in the news. Images fell into 108 topics and 40 percent of the images belonged to one of the 10 most frequent topics, which we used as the basis for our experiments: SuperBowl, Golden Globes, Danish Cartoons, Grammys, Australian Open, Ariel Sharon, Trapped Coal Miners, Figure Skating, Academy Awards and Iraq. Figure 1 shows some example images from each of these categories.

[2]http://today.reuters.com/news/

The experiments involved the binary prediction of whether an image belonged to a particular news topic. The data was partitioned into unlabeled, training, validation and testing sets: we reserved 2,000 images as a source of unlabeled data, 1,000 images as potential source of validation data and 5,000 images as a source of supervised training data. The remaining 2,382 images where used for testing. For each of the 10 most frequent topics we created multiple training sets of different sizes $T_{y,n}$ for $n = 1, 5, 10, 15, \ldots, 50$ and $y = 1, 2, \ldots, 10$; where $T_{y,n}$ denotes a training set for topic y which has n positive examples from topic y and $4n$ negative examples. The positive and negative examples were drawn at random from the pool of supervised training data of size 5,000. The total number of positive images in this pool for the ten top topics were: 341, 321, 178, 209, 196, 167, 170, 146, 137 and 125 respectively.

We consider a transfer learning setting where we have access to unlabeled data U and a collection of training sets C from m related tasks. Our goal is to learn a representation from C and U and use it to train a classifier for the target task.

In our experimental setting we took the 10 most frequent topics and held out one of them to be the target task; the other nine topics were used to learn the sparse prototype representation. We did this for each of the 10 topics in turn. The training set for related topic j in the collection of related training sets C was created by sampling all n_j positive examples of topic j and $2n_j$ negative examples from the pool of supervised training data.

We test all models using the 2,382 held out test images but we remove images belonging to topics in C. We did this to ensure that the improvement in performance was not just the direct consequence of better recognition of the topics in C; in practice we observed that this did not make a significant difference to the experimental results.

Our notion of relatedness assumes that there is a small subset of relevant prototypes such that all related problems can be well approximated using elements of that subset; the size of the relevant subset defines the strength of the relationship. In this experiment we deal with problems that are intuitively only weakly related and yet we can observe a significant improvement in performance when only few examples are available for training. In the future we plan to investigate ways of selecting sets of more strongly related problems.

4.1. Baseline Representation

For all the experiments we used an image representation based on a bag of words representation that combined color, texture and raw local image information. For every image in the dataset we sampled image patches on a fixed grid and computed three types of features for each image patch:

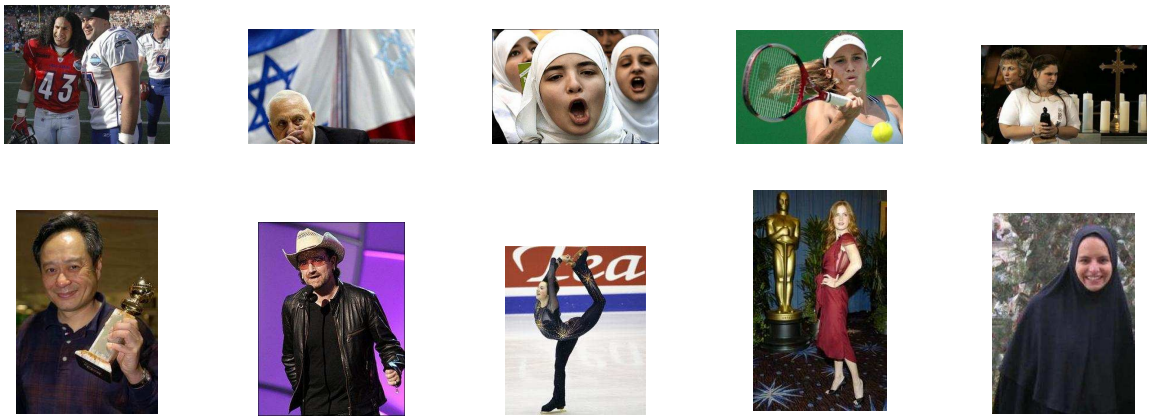

Figure 1. Example images. From top to bottom, left to right: SuperBowl, Sharon, Danish Cartoons, Australian Open, Trapped Coal Miners, Golden Globes, Grammys, Figure Skating, Academy Awards and Iraq.

color features based on HSV histograms, texture features consisting of mean responses of Gabor filters at different scales and orientations and 'raw' features made by normalized pixel values. For each feature type we created a visual dictionary by performing vector quantization on the sampled image patches; each dictionary contained 2,000 visual words.

To compute the representation for a new image x_i we sample patches on a fixed grid to obtain a set of patches $p_i = \{x_{i1}, x_{i2}, \ldots, x_{ih}\}$ and then match each patch to its closest entry in the corresponding dictionary. The final baseline representation for an image is given by the 6,000 dimensional vector: $[[cw_1, \ldots, cw_{2000}], [tw_1, \ldots, tw_{2000}], [rw_1, \ldots, rw_{2000}]]$ where cw_i is the number of times that an image patch in p_i was matched to the i-th color word, tw_i the number of times that an image patch in p_i was matched to the i-th texture word and rw_i the number of times that an image patch in p_i was matched to the i-th raw word.

4.2. Raw Feature Baseline Model

The raw feature baseline model (RFB) consists of training a linear SVM classifier over the bag of words representation by choosing the optimal parameters to be:

$$\mathbf{w}^* = \min_{\mathbf{w}} \lambda \sum_i l(f(x_i), y_i) + \frac{1}{2}\|\mathbf{w}\|^2 \qquad (14)$$

where $f(x) = \mathbf{w}^\top x$ and l is the hinge loss described in Section 3.2.

We conducted preliminary experiments using the validation data from topic 1 and found 0.01 to be the parameter λ resulting in best equal error rate for all training sizes (where we tried values: {0.01, 0.1, 1, 10, 100}); we also noticed that for the validation set the baseline model was not very sensitive to this parameter. We set the constant λ for this and all other models to 0.01.

A baseline model was trained on all training sets $T_{y,n}$ of the 10 most frequent topics and tested on the 2,382 test images. As explained in the previous Section, we removed from the testing set images that belonged to any of the other nine most frequent topics.

4.3. Low Rank Baseline Model

As a second baseline (LRB), we trained a linear SVM classifier but with the baseline feature vectors x in training and testing replaced by h-dimensional feature vectors: $g(x) = L^\top \varphi(x)$. L is the matrix described in Section 3.1 created by taking the top h eigenvectors of K, where K is the kernel matrix over unlabeled data points.

We present results for different values of $h = \{50, 100, 200\}$. For all experiments in this Section we used an RBF kernel over the bag of words representation: $k(x_i, x_j) = exp^{-\gamma||x_i - x_j||^2}$. In a preliminary stage, we tried a range of values $\gamma = \{0.003, 0.03, 0.3\}$ on the unlabeled data, 0.03 was the value that resulted in a non-degenerate kernel (i.e. neither a diagonal kernel nor a kernel made of ones). The value of γ was then fixed to 0.03 for all experiments.

4.4. The Sparse Prototype Transfer Model

We ran experiments using the sparse prototype transfer learning (SPT) approach described in Section 3.3 . For each of the ten topics we train a linear SVM on feature vectors $v(x)$ obtained by running the sparse prototype transfer learning algorithm on the training sets of the remaining nine topics.

For a target held out topic j we use the 2,000 unlabeled points in U and a collection of training sets from related

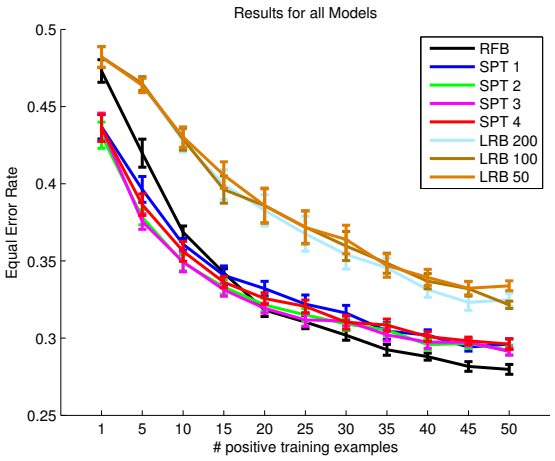

Figure 2. Mean Equal Error Rate over 10 topics for RFB, LRB (for $h = \{50, 100, 200\}$, see section 3.1 for details) and SPT (for $\theta = \{1, 2, 3, 4\}$, see section 3.3 for details).

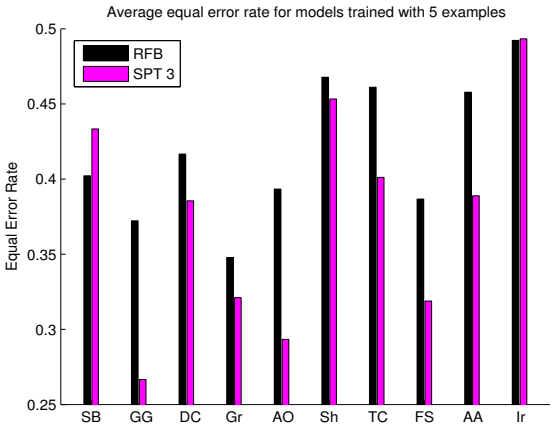

Figure 3. Mean Equal Error rate per topic for classifiers trained with five positive examples; for the RFB model and the SPT model for $\theta = 3$ (see Section 3.3 for details). SB: SuperBowl; GG: Golden Globes; DC: Danish Cartoons; Gr: Grammys; AO: Australian Open; Sh: Sharon; FS: Figure Skating; AA: Academy Awards; Ir: Iraq.

problems: $C = \{T_1, \ldots, T_{j-1}, T_{j+1}, \ldots\}$ to compute the sparse prototype representation $v(x)$ based on kernel distances to relevant prototypes. We report results for different values of the threshold θ; in practice θ could be validated using leave-one-out cross-validation.

4.5. Results

For all experiments we report the mean equal error rate and the standard error of the mean. Both measures were computed over nine runs of the experiments, where each run consisted of randomly selecting a training set from the pool of supervised training data.

Figure 2 shows the mean equal error rate averaged over the ten most frequent topics for RFB, LRB and SPT models; as we can observe from this figure the low rank approach fails to produce a useful representation for all choices of h. In contrast, our sparse transfer approach produces a representation that is useful when training classifiers with small number of training examples (i.e. less than 10 positive examples); the improvement is most significant for $\theta \geq 3$.

For larger training sets the RFB baseline gives on average better performance than the SPT model. We speculate that when larger training sets are available the sparse representation needs to be combined with the raw feature representation. In the future we plan to investigate methods for fusing both representations. However, we would like to emphasize that there exist important applications for which only very few examples might be available for training. For example when building a classifier for a user-defined category it would be unrealistic to expect the user to provide more than a handful of positive examples.

Figure 3 shows mean equal error rates for each topic when trained with 5 positive examples for the baseline model and the transfer learning model with $\theta = 3$. As we can see from these figures the sparse prototype transfer learning method significantly improves performance for 8 out of the 10 topics.

Figure 4 shows learning curves for the baseline and sparse transfer learning model for three different topics. The first topic, Golden Globes, is one of the topics that has the most improvement from transfer learning, exhibiting significantly better performance across all training sizes. The second topic, Academy Awards, shows a typical learning curve for the sparse prototype transfer learning algorithm; where we observe a significant improvement in performance when a few examples are available for training. Finally the third topic, Super Bowl, is the topic for which the sparse prototype transfer algorithm results in worst performance. We speculate that this topic might not be visually related to any of the other topics used for transfer. We have also noticed that this is one of the most visually heterogeneous topics since it contains images from the football field, a press conference and after-game celebrations.

5. Conclusion and Future Work

We have described a method for learning a sparse prototype image representation for transfer in visual category learning. Our approach leverages unlabeled data as well as data from related visual categories and can exploit any arbitrary kernel function. The method is based on performing a joint sparse approximation on the prototype space to find a subset of discriminative prototypes, we formulate this joint optimization as a linear program. Our experiments show that the sparse prototype representation improves the

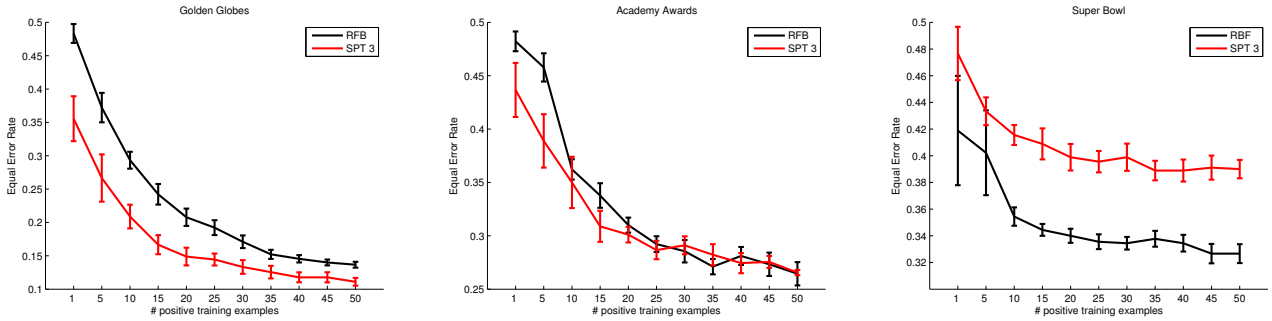

Figure 4. Learning curves for Golden Globes, Academy Awards and Super Bowl topics respectively for RFB and SPT model with $\theta = 3$, see Section 3.3 for details about parameter θ.

performance of news-topic image classifiers when learning with small training sets.

An alternative and complementary approach for transfer learning [2] is based on learning shared hidden representations (i.e. linear subspaces) using data from related tasks and can also be used for transfer learning on this data [14]. Future work will investigate ways of combining both approaches in a single optimization scheme.

References

[1] Y. Amit, M. Fink, N. Srebro, and S. Ullman. Uncovering shared structures in multiclass classification. In *Proceedings of ICML*, 2001.

[2] R. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

[3] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Proceedings of NIPS*, 2006.

[4] M. Balcan, A. Blum, and S. Vempala. Kernels as features: On kernels, margins, and low-dimensional mappings. In *Machine. Learning Journal, 65(1):79 94*, 2004.

[5] J. Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, 28, 1997.

[6] B.Epshtein and S.Ullmam. Identifying semantically equivalent object fragments. In *Proceedings of CVPR-2005*, 2005.

[7] D. Donoho. For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution. Technical report, Techical report, Statistics Dpt, Standford University, 2004.

[8] L. Fei-Fei, P. Perona, and R. Fergus. One-shot learning of object categories. *Pattern Analysis and Machine Intelligence*, 28(4), 2006.

[9] M. Fink. Object classification from a single example utilizing class relevance metrics. In *Proceedings of NIPS*, 2004.

[10] T. Hertz, A. Bar-Hillel, and D. Weinshall. Learning distance functions for image retrieval. In *Proceedings of CVPR*, 2004.

[11] T. Jebara. Multi-task feature and kernel selection for svms. In *Proceedings of ICML*, 2004.

[12] E. Miller, N. Matsakis, and P. Viola. Learning from one example through shared densities on transforms. In *Proceedings of CVPR*, 2001.

[13] G. Obozinski, B. Taskar, and M. Jordan. Multi-task feature selection. In *Technical Report*, 2006.

[14] A. Quattoni, M. Collins, and T. Darrell. Learning visual representations using images with captions. In *Proc. CVPR*, 2007.

[15] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of ICML*, 2007.

[16] R. Raina, A. Y. Ng, and D. Koller. Constructing informative priors using transfer learning. In *Proceedings of the 23rd International Conference on Machine learning*, pages 713–720, 2006.

[17] M. Seeger. Learning with labeled and unlabeled data. Technical report, Institute for Adaptive and Neural Computation, Univ. of Edinburgh, 2001.

[18] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Learning hierarchical models of scenes, objects, and parts. In *Proceedings of ICCV*, 2005.

[19] S. Thrun. Is learning the n-th thing any easier than learning the first? In *In Advances in Neural Information Processing Systems*, 1996.

[20] A. Torralba, K. Murphy, and W. Freeman. Sharing visual features for multiclass and multiview object detection. *Pattern Analysis and Machine Intelligence*, In press, 2006.

[21] J. Tropp. Algorithms for simultaneous sparse approximation, part ii: convex relaxation. In *Signal Process. 86 (3) 589-602*, 2006.

[22] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *Proceedings of ICCV*, 2007.

[23] X. Wang and E. George. A hierarchical bayes approach to variable selection for generalized linear models. In *Tech Report*, 2004.

[24] A. Zweig and D. Weinshall. Exploiting object hierarchy: Combining models from different category levels. In *Proceedings of ICCV*, 2007.