Discriminative Sub-categorization

Minh Hoai **Andrew Zisserman**
University of Oxford, Oxford, UK
minhhoai@robots.ox.ac.uk, az@robots.ox.ac.uk

Abstract

The objective of this work is to learn sub-categories. Rather than casting this as a problem of unsupervised clustering, we investigate a weakly supervised approach using both positive and negative samples of the category.

We make the following contributions: (i) we introduce a new model for discriminative sub-categorization *which determines cluster membership for positive samples whilst simultaneously learning a max-margin classifier to separate each cluster from the negative samples; (ii) we show that this model does not suffer from the degenerate cluster problem that afflicts several competing methods (e.g., Latent SVM and Max-Margin Clustering); (iii) we show that the method is able to discover interpretable sub-categories in various datasets.*

The model is evaluated experimentally over various datasets, and its performance advantages over k-means and Latent SVM are demonstrated. We also stress test the model and show its resilience in discovering sub-categories as the parameters are varied.

1. Introduction

Many real-world categories are complex and require division into sub-categories for better modeling and classification. The sub-categories are often specified manually (e.g., frontal or profile faces [22, 23]), but can be determined automatically using unsupervised clustering [2, 6, 11, 14, 15, 17, 21, 29, 30].

Sub-categorization has been shown to improve performance in a wide variety of applications: analyzing animal behavior [10], understanding architecture [7], recognizing faces [33], classifying images [19], and detecting object categories [9]. Correspondingly a number of methods for sub-categorization have been developed and applied. The emphasis of the algorithm can be on grouping "similar" (in feature space) items (e.g., k-means, pLSA [14]) or on separating "dissimilar" items (e.g., Max-Margin Clustering (MMC) [29], DIFFRAC [2], universum clustering [32]). One particular problem of such methods is cluster degeneration, where clusters have few or no elements [13, 29].

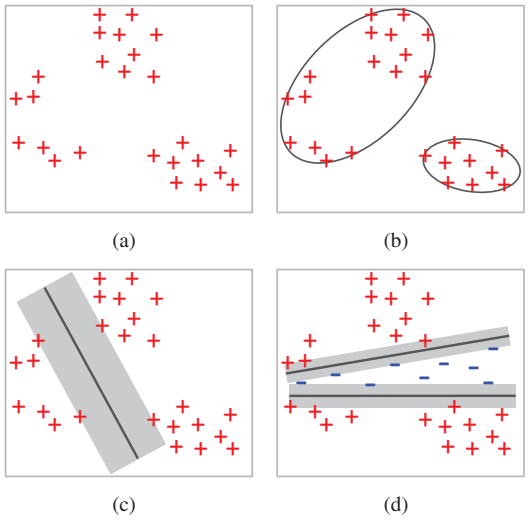


Figure 1. Different clustering criteria for sub-category discovery. (a): positive training examples. (b): methods such as k-means minimizes intra-cluster distances. (c) methods such as [6, 29] maximize separation among clusters. (d): our method maximizes the separation between clusters and negative data; it partitions positive examples (red plus) into clusters so that each cluster can be well separated from the negative examples (blue minus).

In this paper, we introduce a new model for determining sub-categories which also utilizes negative data, i.e., examples that do not belong to the category under consideration, as a means of defining similarity and dissimilarity. In essence, a sub-category is required to contains similar items *and* also be well separated from the negative examples. Given a set of positive and negative examples of a category, the model simultaneously determines the cluster label of each positive example, whilst learning an SVM for each cluster, discriminating it from the negative examples, as illustrated in Fig. 1(d). The requirement for negative examples is usually not a problem since they are readily available. As will be seen, the negative data is used at various stage of training, model selection, and parameter tuning.

Our formulation has several advantages. First, it jointly performs clustering and classifier training, unlike many existing techniques such as [16] where these two steps are

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.218

optimized independently leading to a suboptimal solution. Second, since it is based on linear SVMs, the model is simple to implement and efficient to run (both training and testing), and therefore suitable for high dimensional and large datasets. Third, it does not suffer from cluster degeneration. Experiments on datasets of varying complexity, from digits and letters to images, show that the model often discovers highly interpretable sub-categories.

It will be seen that our model bears some similarities to Multiple-Instance SVMs [1], Latent SVMs [9], Latent Structural SVMs [31], and to mixtures of linear SVMs [12]. Such methods improve classification performance using sub-categories, whereas here the emphasis is on obtaining the sub-categories. We discuss this further after specifying the model in the following section.

It is important to note that our method is distinct from the *Universum* approach [27] . We use negative data to sub-categorize a single class (discovering subcategories), while Universum [27] uses "non-examples" for learning a classifier to separate between *predefined* classes.

2. Subcategory discovery

We pose sub-categorization as a joint clustering and classification problem. In this section, we describe the formulation and contrast it with several alternatives.

2.1. Joint clustering and classification

For a particular category of interest, consider the task of discovering its sub-categories given a set of positive training examples ($\mathbf{x}_1^+, \cdots, \mathbf{x}_n^+ \in \Re^d$) and a set of negative training examples ($\mathbf{x}_1^-, \cdots, \mathbf{x}_m^- \in \Re^d$). We propose to find the sub-categories by grouping positive training examples into several clusters such that each cluster is well separated from the negative training examples. Let $y_i \in \{1, \cdots, k\}$ be the (latent) cluster label associated with the positive training example $\mathbf{x}_i^+$, the separation between cluster j and the negative examples can be measured using the SVM objective:

$$\begin{aligned} \min_{\mathbf{w}_j, b_j} \; & \frac{1}{2}||\mathbf{w}_j||^2 \\ \text{s.t. } & \mathbf{w}_j^T \mathbf{x}_i^+ + b_j \geq 1 \; \forall i : y_i = j, \\ & \mathbf{w}_j^T \mathbf{x}_i^- + b_j \leq -1 \; \forall i. \end{aligned} \tag{1}$$

The above only involves positive examples that belong to cluster j. To measure the total separability between sub-categories and the negative examples, we use the weighted sum of the above SVM objectives: $\sum_{j=1}^k \frac{n_j}{n}(\frac{1}{2}||\mathbf{w}_j||^2)$, where n_j is the cardinality of cluster j. This is equivalent to $\frac{1}{2n}\sum_{i=1}^n ||\mathbf{w}_{y_i}||^2$. We seek the cluster labels for positive examples and simultaneously train the SVMs that separate the resulting clusters from the negative examples:

$$\begin{aligned} \underset{\{\mathbf{w}_j, b_j, y_i\}}{\text{minimize}} \; & \frac{1}{2n}\sum_{i=1}^n ||\mathbf{w}_{y_i}||^2 \\ \text{s.t. } & \mathbf{w}_{y_i}^T \mathbf{x}_i^+ + b_{y_i} \geq 1 \; \forall i, \\ & \mathbf{w}_j^T \mathbf{x}_i^- + b_j \leq -1 \; \forall i \; \forall j. \end{aligned} \tag{2}$$

As in the case of SVMs, we allow the constraints to be violated but penalize for the total violation:

$$\underset{\{\mathbf{w}_j, b_j, y_i, \xi_i^+, \xi_i^-\}}{\text{minimize}} \; \frac{1}{2n}\sum_{i=1}^n ||\mathbf{w}_{y_i}||^2 + \frac{C}{n}\sum_{i=1}^n \xi_i^+ + \frac{C}{m}\sum_{i=1}^m \xi_i^- \tag{3}$$

$$\begin{aligned} \text{s.t. } & \mathbf{w}_{y_i}^T \mathbf{x}_i^+ + b_{y_i} \geq 1 - \xi_i^+ \; \forall i, \\ & \mathbf{w}_j^T \mathbf{x}_i^- + b_j \leq -1 + \xi_i^- \; \forall i \; \forall j, \\ & \xi_i^+ \geq 0, \xi_i^- \geq 0. \end{aligned}$$

where, as usual, $\{\xi_i^+\}, \{\xi_i^-\}$ are slack variables which allow for penalized constraint violation, and C is the parameter that controls the tradeoff between margins and constraint violation. Notably, in the above objective, a vector $\mathbf{w}_j$ is weighted by the cardinality of cluster j. This is different from the objective of Multi-class SVMs [4] or Latent SVMs [1, 9, 31] in which each vector $\mathbf{w}_j$ is weighted equally. Also in the above formulation, the cost (in the objective function) for assigning label j to point $\mathbf{x}_i^+$ is $\frac{1}{2n}||\mathbf{w}_j||^2 + \frac{C}{n}\max(0, 1 - \mathbf{w}_j^T \mathbf{x}_i^+ - b_j)$. Thus the cluster label y_i of $\mathbf{x}_i^+$ is:

$$y_i = \underset{j}{\text{argmin}} \left\{ \frac{1}{2}||\mathbf{w}_j||^2 + C\max(0, 1 - \mathbf{w}_j^T \mathbf{x}_i^+ - b_j) \right\}. \tag{4}$$

This is different from the class/cluster assignment in Multi-class SVMs and Latent SVMs, where y_i is given by $y_i = \arg\max_j \{\mathbf{w}_j^T \mathbf{x}_i^+ + b_j\}$.

There is a notable connection between (4) and k-means. Recall k-means seeks a set of centroids $\{\mathbf{w}_1, \cdots, \mathbf{w}_k\}$ and cluster labels $\{y_1, \cdots, \mathbf{y}_n\}$ to minimize $\frac{1}{2}\sum_{i=1}^n ||\mathbf{x}_i^+ - \mathbf{w}_{y_i}||^2$. The cluster label of a point $\mathbf{x}_i^+$ is given by $y_i = \arg\min_j \frac{1}{2}||\mathbf{x}_i^+ - \mathbf{w}_j||^2$, or equivalently, $y_i = \arg\min_j \{\frac{1}{2}||\mathbf{w}_j||^2 - \mathbf{w}_j^T \mathbf{x}_i^+\}$. Observe the similarity between this formula and (4). In our formulation, $\mathbf{w}_j^T \mathbf{x}_i^+$ is filtered through the Hinge loss.

2.2. Cluster degeneration

As noted in the introduction, several alternative formulations that are used for sub-category discovery suffer from the problem of cluster degeneration, i.e., the situation where a few clusters dominate and claim all the points, leading to many empty clusters. This is problematic for sub-category discovery because the number of sub-categories obtained is then smaller than the sought after number. Cluster degeneration has been pointed out to be an inherent problem of

discriminative clustering [13, 29]. Fortunately, our formulation does not suffer from this problem, and this section discusses the problem in detail.

Perhaps the formulation that is most similar to ours is:

$$\underset{\{\mathbf{w}_j, b_j, y_i, \xi_i^+, \xi_i^-\}}{\text{minimize}} \frac{1}{2k}\sum_{j=1}^{k}||\mathbf{w}_j||^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i^+ + \frac{C}{m}\sum_{i=1}^{m}\xi_i^- \tag{5}$$

$$\text{s.t. } \mathbf{w}_{y_i}^T\mathbf{x}_i^+ + b_{y_i} \geq 1 - \xi_i^+ \;\forall i,$$
$$\mathbf{w}_j^T\mathbf{x}_i^- + b_j \leq -1 + \xi_i^- \;\forall i\,\forall j,$$
$$\xi_i^+ \geq 0, \xi_i^- \geq 0.$$

This formulation is a particular realization of Latent SVM [9], Multiple-Instance SVM [1], and Latent Structural SVM [31], in which the latent variables are the cluster labels. For brevity, we will refer to this formulation as LSVM hereafter. As noted in Sec. 2.1, there are two key differences between LSVM and our formulation. First, LSVM minimizes the average of the squared L_2-norms of the weight vectors, while ours minimizes the weighted average. Second, the criteria for cluster assignment are different. These differences are the keys to address a major limitation of LSVM.

Our formulation has a natural mechanism for eliminating empty clusters without increasing the cost. First we show that if a cluster is empty, then it can be regenerated at no additional cost, then we show that the cost can be decreased. Suppose cluster j is empty. Consider the following steps: (i) pick a non-empty cluster l and split it into two arbitrary halves of size $\frac{n_l}{2}$; (ii) reassign one half to cluster j and copy the weight vector and bias term of cluster l to cluster j, i.e., $\mathbf{w}_j := \mathbf{w}_l, b_j := b_l$. These steps do not change the cost because $\frac{1}{2n}n_l||\mathbf{w}_l||^2 + \frac{1}{2n}0||\mathbf{w}_j||^2 = \frac{1}{2n}\frac{n_l}{2}||\mathbf{w}_l||^2 + \frac{1}{2n}\frac{n_l}{2}||\mathbf{w}_l||^2$. The total cost can possibly then be lowered in the subsequent optimization step that adjusts the weight vectors and the bias terms of the revived clusters. Moreover, if the cluster l contains some points that are not support vectors (i.e., points that are beyond the right side of the margin—corresponding to non-tight constraints) and these are reassigned to cluster j, then the margin corresponding to cluster j can be increased in subsequent optimization iterations. This means a reduction in the term $||\mathbf{w}_j||^2$ of the cost, because $||\mathbf{w}_j||$ is inversely proportional to the margin. In short, there exists a mechanism for eliminating empty clusters; this mechanism never increases the cost and it decreases the cost with high probability (unless every point is a support vector). The existence of this mechanism means that the objective (3) does not favor degenerate clusters.

Conversely, LSVM *does* suffer from cluster degeneration. This has been observed in practice (though not formally reported) and it is confirmed in our experiments (Sec. 3). A rigorous proof does not exist, but here are two possible reasons. First, the mechanism described above for regenerating an empty cluster at no cost does not apply. Eliminating an empty cluster j by reassigning some points from a non-empty cluster l and duplicating the weight vector ($\mathbf{w}_j := \mathbf{w}_l$) will *increase* the objective function, because $\frac{1}{2k}||\mathbf{w}_l||^2 + \frac{1}{2k}||\mathbf{w}_j||^2 < \frac{1}{2k}||\mathbf{w}_l||^2 + \frac{1}{2k}||\mathbf{w}_l||^2$ (recall cluster j is empty and $||\mathbf{w}_j|| = 0$). Second, LSVM encourages the domination of big clusters. This can be seen as follows: LSVM can be considered as a combination of multiple SVMs, equally weighted. Each of these SVMs has the same number of negative constraints, but the number of positive constraints depends on the cluster size. In general, the more positive constraints an SVM has, the the smaller the margin will be (assuming C is fixed; C is the parameter controlling the tradeoff for larger margin and less constraint violation). A smaller margin is equivalent to a larger magnitude of the weight vector (because they are inversely proportional). Thus, if cluster u is much larger than cluster v, the magnitude of weight vector $\mathbf{w}_u$ will be much larger than that of $\mathbf{w}_v$, i.e., $||\mathbf{w}_u|| \gg ||\mathbf{w}_v||$. Now since the clustering assignment of a data point is based on the dot product between itself and the weight vectors, cluster u will have an advantage over cluster v. It is likely that some points from cluster v will be reassigned to cluster u. Cluster u will grow larger while cluster v becomes smaller, increasing the size-gap between them.

Interestingly, cluster degeneration has been empirically observed for other types of classifiers. [28] noted that the boosting-tree classification-clustering framework of [25], which bases its splitting decision on classification confidence, also produces unbalanced clusters.

Cluster degeneration is also an inherent problem of MMC [29]. MMC requires every pair of clusters to be well separated by a margin. Thus every pair of clusters leads to a constraint on the maximum size of the margin. As a result, MMC favors a model with fewer number of clusters because less effort for separation is required. In the extreme, MMC can create a single cluster [13, 29].

2.3. Optimization

The learning formulation given in Eq. 3 is not convex, but it can be optimized with block coordinate descent in a similar manner to LSVM [1, 9, 31]. Block coordinate descent alternates between the following two procedures:

(A) Fix the cluster labels $\{y_i\}$, optimize the SVM parameters $\{\mathbf{w}_j, \mathbf{b}_j\}$ and $\{\xi_i^+, \xi_i^-\}$,

(B) Fix the SVM parameters $\{\mathbf{w}_j, \mathbf{b}_j\}$, optimize the cluster labels $\{y_i\}$.

Procedure (A) corresponds to a convex quadratic program, and can be optimized using stochastic gradient descent [3], where the weight vectors and the bias terms are

updated based on a single training example at each iteration. This procedure has been shown to be very effective and efficient for linear SVMs [3, 24].

Procedure (B) requires updating the cluster assignment for each positive training example. This is equivalent to finding the cluster label with minimum assignment cost, given in Eq. 4. This can be optimized efficiently with a few matrix-vector multiplications and min operators.

This block-coordinate descent algorithm is guaranteed to converge because both procedures (A) and (B) do not increase the energy of the objective function. We propose to initialize the algorithm as follows:

(i) Train a linear SVM to separate positive and negative classes, obtain the weight vector $\mathbf{w}$.

(ii) Project the positive examples on the weight vector $\mathbf{w}$, compute the residual vectors $\bar{\mathbf{x}}_i^+ := \mathbf{x}_i^+ - \frac{1}{||\mathbf{w}||}(\mathbf{w}^T\mathbf{x}_i^+)\mathbf{w}$. The residual is the component of $\mathbf{x}_i^+$ that is perpendicular to $\mathbf{w}$.

(iii) Perform k-means on the residual vectors $\{\bar{\mathbf{x}}_i^+\}$ to get the initial cluster labels.

3. Experiments

This section describes experiments on various datasets. Both quantitative and qualitative evaluations are provided, using purity measure [20, 26] and visual interpretability.

3.1. Clustering performance

We validated the clustering performance of our method on several publicly available datasets from the UCI repository[1] and the MNIST dataset [18]. The UCI repository contains many datasets, but most of them are irrelevant for our experiments. We exclude datasets that have fewer than six classes or fewer than 1500 data instances. We remove datasets that are only suitable for time series analysis or regression. We also exclude datasets that contain categorical or missing attributes. The final collection of datasets are: Gas Sensor (Gas Sensor Array Drift), Landsat (Stalog Landsat Satellite), Segmentation (Statlog Image Segmentation), Steel Plates (Steel Plates Faults), Wine Quality, Digits (Optical Recognition of Handwritten Digits), Semeion (Semeion Handwritten Digits), Letter (Letter Recognition), Isolet, and Amazon Reviews (Amazon Commerce Reviews). Additionally, we include MNIST, a dataset which is not in the UCI repository but frequently used for benchmarking machine learning algorithms. The number of classes, the number of dimensions, and the number of instances for these datasets are given in Tab. 1.

The experimental setting for each dataset is as follows. Each dataset has a predefined number of classes (ground truth labels). We randomly split the classes into two roughly equal halves, one is regarded as positive and the other as negative. Subsequently, each group is randomly divided into training and validation subsets. The training subsets (one positive and one negative) are used to learn the cluster models as in Eq. 3. The validation sets are used for parameter tuning. We set the number of clusters to the true number of classes of the positive group. We chose C among $10^2, 10^3, 10^4, 10^5$ based on the classification performance on the validation data. All datasets are L_2 normalized. Some datasets require an additional normalization step to bring the attribute values to the same scale (range $[0, 1]$) before applying L_2 normalization. Both LSVM and our method start from the same initialization, as explained in Sec. 2.3. Each experiment is repeated 50 times.

We compared the performance of our method against k-means and LSVM. To measure clustering performance, we followed the strategy used by [13, 29], where we first took a set of labeled data, removed the labels and ran the clustering algorithms. We then found the best one-to-one association between the resulting clusters and the ground truth classes (Hungarian algorithm). Finally, we calculated the percentage of correct assignment. This is referred to as *purity* in information theoretic measures [20, 26]. Notably, a purity measure requires no separate test set.

Tab. 1 displays the experimental results. *Init-label* performed similar to k-means on most of the datasets, despite the use of negative data. LSVM is initialized with Init-label, but the benefit of this approach is unclear: the clustering performance increases for three datasets, decreases for three others, and remains similar for the rest. On the other hand, also initialized with Init-label, our method improves the performance on seven out of eleven datasets. For the remaining four datasets, the performance of our method is similar to that of Init-label. This is perhaps due to the problem of local minima in optimization. For example, the Amazon Reviews dataset contains high dimensional sparse vectors. It is very likely that any subset of the positive class can be linearly separated from the negative class. Therefore, data points have little or no tendency to switch between clusters, and the optimization procedure terminates at a configuration that is similar to the one it starts with. Indeed, the performance of LSVM and our method are similar to the performance of their initialization on this dataset. Overall, comparing all approaches, our method performs the best or not significantly worse than the best on all datasets.

3.2. Discovering Head Orientations

This section describes experiments on discovering head orientations—the looking direction. Data for these experiments is extracted from the TV Human Interaction (TVHI) dataset [22]. This dataset contains 300 video clips compiled from 23 different TV shows. Every frame of these videos comes with the following annotation: the bounding boxes

[1] http://archive.ics.uci.edu/ml/

Table 1. Clustering purity measures (%) of k-means, LSVM, and our method on UCI datasets and MNIST. This table shows the mean and standard error of 50 runs on each dataset, and the datasets are ordered based on the number of classes they contain. For each dataset, results within one standard error of the maximum value are printed in bold. The second, third, and fourth columns list the numbers of classes, dimensions, and data points respectively. *Init-label* refers to the initial labels obtained by running k-means on the residual vectors as explained in Sec. 2.3. *Init-label* is the starting point of LSVM and our method. Our method performs the best or not significantly worse than the best on all datasets.

Dataset	#classes	#features	#points	k-means	Init-label	LSVM	ours
Gas Sensor	6	128	13910	46.38 ± 0.69	47.16 ± 0.65	56.74 ± 1.88	**60.82 ± 1.64**
Landsat	6	36	4435	**78.72 ± 2.08**	**77.45 ± 2.16**	69.37 ± 2.32	**76.73 ± 2.38**
Segmentation	7	19	2310	71.96 ± 1.75	71.47 ± 1.73	65.89 ± 2.36	**74.41 ± 1.85**
Steel Plates	7	27	1941	**53.29 ± 1.51**	**53.39 ± 1.89**	**52.64 ± 2.02**	**54.60 ± 1.98**
Wine quality	7	12	4898	43.43 ± 1.58	41.13 ± 1.14	**55.00 ± 2.35**	**54.21 ± 1.65**
Digits	10	64	5620	76.38 ± 1.72	**80.40 ± 1.23**	77.83 ± 1.57	**80.15 ± 1.18**
Semeion	10	256	1593	64.64 ± 1.20	65.05 ± 1.30	64.32 ± 1.58	**66.74 ± 1.43**
MNIST	10	784	60000	**65.38 ± 1.43**	63.84 ± 1.40	63.99 ± 1.36	**66.18 ± 1.34**
Letter	26	16	20000	33.35 ± 0.48	32.89 ± 0.52	40.27 ± 0.88	**44.38 ± 0.74**
Isolet	26	617	6238	62.15 ± 1.22	61.55 ± 1.20	61.95 ± 1.22	**64.08 ± 1.18**
Amazon Reviews	50	10000	1500	**24.93 ± 0.32**	**24.90 ± 0.36**	**24.89 ± 0.41**	**25.08 ± 0.38**

framing the upper bodies of the people present and their discrete head orientations. The label set for head orientations are Profile-Left, Frontal-Left, Frontal-Right, Profile-Right, and Backward. To obtain the head bounding box, regression is applied to the supplied bounding box of each upper body, as illustrated in Fig. 2. The head area is then normalized to a 64×64 pixel patch, and represented by a HOG descriptor [5]. The dimension of the descriptor is 1984 (the size of HOG cells is 8×8). Because the head areas of the same person in consecutive frames are often similar, it is unnecessary to consider all frames so we subsample them to obtain positive examples for this experiment. Data for training and validation are sampled from separate video subsets, based on the train/test split specified by the authors of the TVHI dataset [22]. This process yields 4040 and 4760 positive examples for training and validation, respectively. The negative examples are obtained from the negative images of the INRIA Person dataset[2] by applying the upper-body detector [8] on each image and retaining the top five detections. We also added five random patches per image. The numbers of negative examples for training and validation are 4872 and 4530 respectively. Finally, all feature vectors are normalized to have L_2 norms of approximately 1 (dividing them by the median of the L_2 norms of positive training examples). The training data is used to learn the cluster models as in Eq. 3, while the binary classification accuracy (positive versus negative) on the validation data is used for parameter tuning. The performance measure is cluster purity [20, 26], which requires no separate test set.

To test the ability to discover sub-categories, we set the number of clusters to five, the predefined number of head orientations. To circumvent the problem of local minima, all methods (k-means, Init-label, LSVM, and ours) are run

[2]http://pascal.inrialpes.fr/data/human/

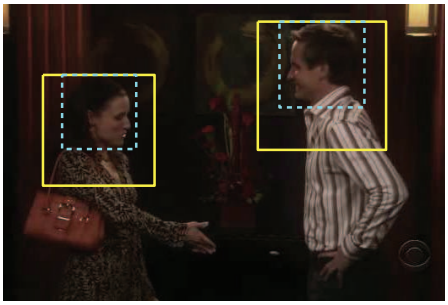
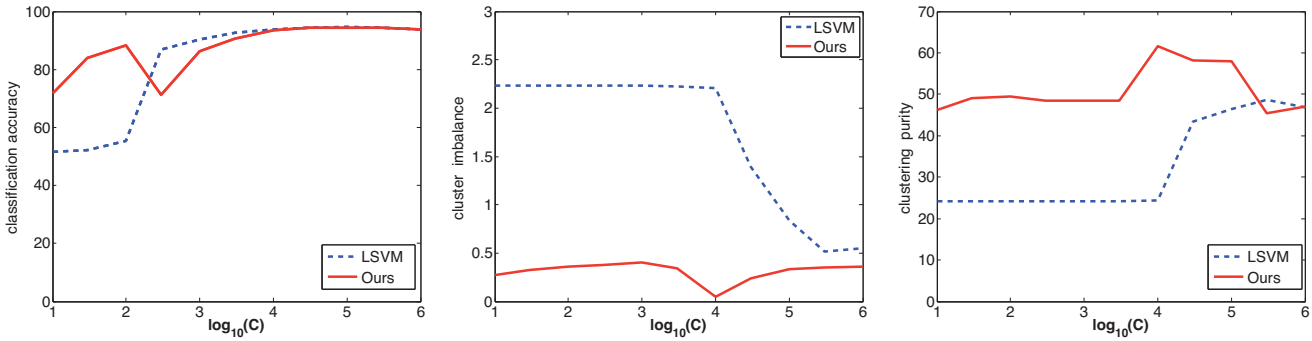


Figure 2. Extracting head regions from annotated upper bodies. The solid yellow squares are annotated upper bodies. The dash cyan squares are head regions used in our experiments.

50 times, and the run that yields minimum energy is selected. As in Sec. 3.1, we used purity measure to benchmark the clustering performance. The performance of k-means, Init-label, LSVM, and our method are 46.86, 46.14, 44.95, 58.07 respectively. These results were obtained with $C = 30000$, tuned based on the classification accuracy on validation data. We performed an additional experiment where the value of C was increased from 300 to 3000 to 30000, and the output of the previous step was used as the initialization for the next step. This multi-stage optimization procedure boosts the performance of our method to 62.23%, which is significantly better than the results of all other methods. Notably, this result is close to 69.05%, which is the state-of-the-art accuracy of five-way head classification using linear SVMs with HOG descriptors [22] (this is a comparison between the purity measure of an unsupervised method with the classification accuracy of a supervised method). We applied the same multi-stage optimization procedure to LSVM, but the performance degraded due to cluster degeneration at the early stage. Fig. 3 analyzes this problem.



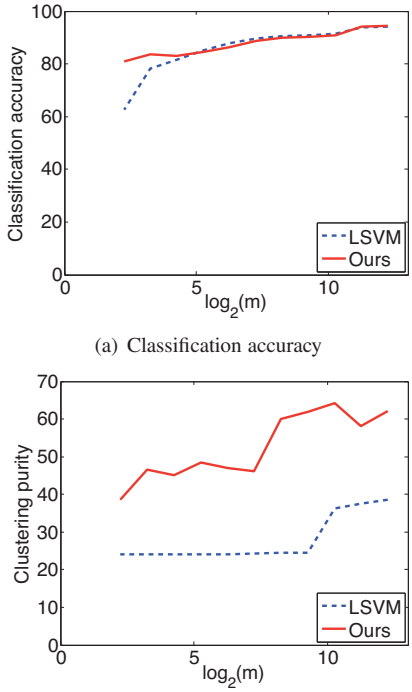
(a) Classification accuracy (b) Cluster imbalance (c) Purity measure

Figure 3. Several measures as a function of C. (a): classification accuracy on validation data; (b): imbalance index—the standard deviation of cluster sizes; (c): purity measure—the agreement between clusters and ground truth classes. Due to the problem of cluster degeneration, the clusters produced by LSVM can be highly unbalanced, even for the values of C that yield relatively high classification accuracy. Our method does not suffer from this problem, yielding low imbalance index and high purity measure on all values of C. For high values of C, LSVM does not suffer from the cluster degeneration problem, but the clustering performance is similar to the performance of the initialization. For a reference, the imbalance index of ground truth labels is 0.28. The upper limit of C is set to 10^6 because there are no practical benefits for going higher: (i) the classification performance decreases for $C > 10^5$; and (ii) the SVM optimization takes much longer to converge for large C.

We also study the classification accuracy (on validation data) and the clustering purity as the amount of negative data varies. Figs. 4(a) and 4(b) plot and discuss the results. Notably, for small amount of negative data, it is necessary to reduce the value of C to retain high classification accuracy. This is observed empirically. Theoretically, C controls the tradeoff between a large margin and low training loss. For small amount of training data, it is necessary to decrease the emphasis on training loss to avoid overfitting, and this is equivalent to using a smaller C. But for small values of C, the clusters of LSVM degenerate (as shown in Fig. 3). This explains why the clustering performance of LSVM is very poor in this experiment. In contrast, our method achieves good result with as few as 300 negative training examples.

Qualitative results are given in Figs. 5–8. Figs. 5 and 6 depict of the weight vectors obtained by our method and several representative images of highest and lowest ranks in each cluster. The learned weight vectors and the highest-rank images somewhat correspond to the five discrete ground truth head orientations. Here, the number of desired sub-categories is set to five, the number of discrete human labels. In practice, the number of sub-categories might be unknown, and furthermore, human annotation might not be optimal. We therefore performed experiments with different desired numbers of sub-categories. Figs. 7 and 8 visualize the models learned by our method and LSVM when the desired numbers of sub-categories are three and six, respectively. The models produced by our method have higher interpretability.

Regarding the classification performance, all methods perform similarly well. The classification accuracy of LSVM and our method are 94.13% and 94.39% respectively. The accuracy of five linear SVMs trained with ground truth head orientations is 94.78%. The accuracies of a single linear SVM and a single RBF-kernel SVM are 94.06% and 95.51% respectively. RBF-kernel SVM per-



(a) Classification accuracy

(b) Cluster purity

Figure 4. Classification accuracy and clustering purity as a function of m, the number of negative training examples. Though LSVM performs relatively well on the classification task, its clustering performance is much worse than ours. Our method obtains excellent clustering results, with as few as 300 negative examples.

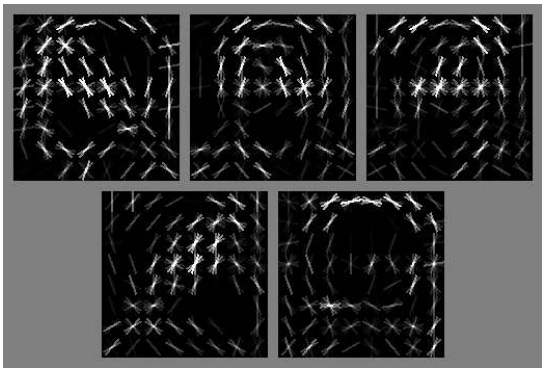
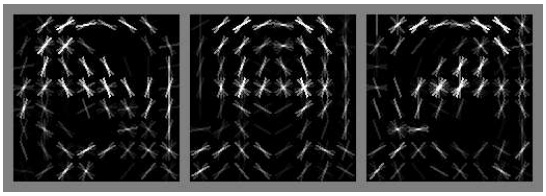
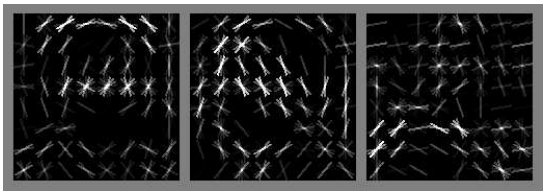


Figure 5. The positive components of the weight vectors that were learned by our method. Each subfigure shows 8×8 HOG cells, and each cell has 9 orientations. Dark values represent low weights and bright values represent high weights; a high weight for a particular direction at a particular cell means the model prefers to have a strong image edge of that direction at that cell. The learned weights somewhat correspond to the edge structures of the heads at different orientations.

Figure 6. High/low rank images – images with highest and lowest confidence scores in each cluster. High-rank images correspond to five discrete head orientations. Low-rank images are due to: i) the regression procedure fails to localize the head region; ii) subject exhibits a rare head pose; iii) the head is occluded; or iv) the image patch has low resolution, low contrast, or motion blur.

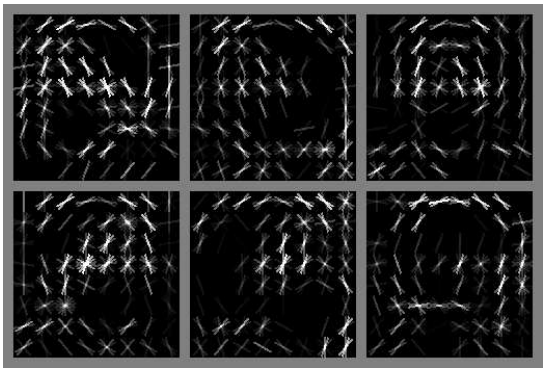


(a) 3-cluster model learned by our method

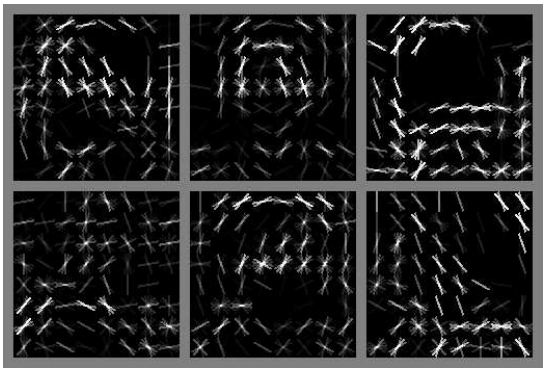


(b) 3-cluster model learned by LSVM

Figure 7. The learned clusters when the desired numbers of sub-categories are three. Our method produces models with higher interpretability. All clusters produced by our method are meaningful while the last cluster of LSVM is uninterpretable.



(a) 6-cluster model learned by our method



(b) 6-cluster model learned by LSVM

Figure 8. Clustering models produced by our method and LSVM when the desired numbers of sub-categories are six. Our method produces models with higher interpretability. All clusters produced by our method are meaningful while two clusters of LSVM are uninterpretable.

forms the best, but it does not produce sub-categories.

Our method is relatively efficient. It usually takes from 20 to 40 iterations of coordinate descent to converge. On this dataset (8912 training and 9290 testing examples of 1984 dimensions), our method (naive implementation without any speed optimization) took 260s for training and 0.7s for testing. For a comparison, LibSVM (both linear and RBF kernels) took around 210s for training and 200s for testing. This timing was performed on a Linux machine with Intel Xeon 4-core 2.5GHz.

4. Summary

We have introduced a new objective for learning sub-categories. The key novelty is incorporating negative samples into the learning framework. Furthermore, we show that assigning to clusters by a combination of Hinge loss and SVM margin avoids the degenerate configurations suffered by several popular methods that assign according to classifier score alone. The advantages of the method were empirically demonstrated on datasets of varying complexity, from MNIST and UCI to TVHI. In this paper, we developed the formulation for linear SVMs, but its extension to non-linear SVMs is straight forward. Armed with this model, methods such as [9] where sub-categories determine the initial templates, can now start from a stronger, and less fragile, basis.

Acknowledgement: This work was supported by EPSRC grant EP/I012001/1.

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, 2003.

[2] F. R. Bach and Z. Harchaoui. DIFFRAC: a discriminative and flexible framework for clustering. In *NIPS*. 2009.

[3] L. Bottou and Y. LeCun. Large scale online learning. In *NIPS*, 2004.

[4] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Machine Learning Research*, 2:265–292, 2001.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005.

[6] F. De la Torre and T. Kanade. Discriminative cluster analysis. In *Proc. ICML*, 2006.

[7] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes Paris look like Paris? *ACM Transactions on Graphics*, 2012.

[8] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari. 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. *IJCV*, 99:190–214, 2012.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE PAMI*, 32(9):1627–1645, 2010.

[10] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Nonparametric Bayesian learning of switching linear dynamical systems. In *NIPS*. 2009.

[11] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[12] Z. Fu and A. Robles-Kelly. Mixing linear SVMs for nonlinear classification. *IEEE Transactions on Neural Networks*, 21(12):1963–1975, 2010.

[13] M. Hoai and F. De la Torre. Maximum margin temporal clustering. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, 2012.

[14] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence*, 1999.

[15] A. Joulin, F. Bach, and J. Ponce. Efficient optimization for discriminative latent class models. In *NIPS*, 2010.

[16] C.-H. Kuo and R. Nevatia. Robust multi-view car detection using unsupervised sub-categorization. In *Proc. WACV*, 2009.

[17] T. Lange, V. Roth, M. L. Braun, and J. M. Buhmann. Stability-based validation of clustering solutions. *Neural Computation*, 16(6):1299–1323, 2004.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[19] A. Mansur and Y. Kuno. Improving recognition through object sub-categorization. In *Proceedings of International Symposium on Advances in Visual Computing*, 2008.

[20] M. Meila. Comparing clusterings – an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007.

[21] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2002.

[22] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman. Structured learning of human interactions in TV shows. *IEEE PAMI*, 2012.

[23] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *IJCV*, 56(3):151–177, 2004.

[24] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for SVM. In *Proc. ICML*, 2007.

[25] Z. Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *Proc. ICCV*, 2005.

[26] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. *IJCV*, 88(2):284–302, 2009.

[27] J. Weston, R. Collobert, F. Sinz, L. Bottou, and V. Vapnik. Inference with the universum. In *Proc. ICML*, 2006.

[28] B. Wu and R. Nevatia. Cluster boosted tree classifier for multi-view, multi-pose object detection. In *Proc. ICCV*, 2007.

[29] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *NIPS*. 2004.

[30] J. Ye, Z. Zhao, and M. Wu. Discriminative k-means for clustering. In *NIPS*, 2007.

[31] C.-N. J. Yu and T. Joachims. Learning structural SVMs with latent variables. In *Proc. ICML*, 2009.

[32] D. Zhang, J. Wang, and L. Si. Document clustering with universum. In *International Conference on Research and Development in Information Retrieval*, 2011.

[33] M. Zhu and A. Martinez. Subclass discriminant analysis. *IEEE PAMI*, 28(8):1274–1286, 2006.