Learning with Dataset Bias in Latent Subcategory Models

Dimitris Stamos[1]
d.stamos@cs.ucl.ac.uk

Samuele Martelli[2]
samuele.martelli@iit.it

Moin Nabi[2]
moin.nabi@iit.it

Andrew McDonald[1]
a.mcdonals@cs.ucl.ac.uk

Vittorio Murino[2,3]
vittorio.murino@iit.it

Massimiliano Pontil[1]
m.pontil@cs.ucl.ac.uk

[1] Department of Computer Science, University College London, UK
[2] Pattern Analysis & Computer Vision, Istituto Italiano di Tecnologia, Italy
[3] Dipartimento di Informatica, Università di Verona, Italy

Abstract

Latent subcategory models (LSMs) offer significant improvements over training linear support vector machines (SVMs). Training LSMs is a challenging task due to the potentially large number of local optima in the objective function and the increased model complexity which requires large training set sizes. Often, larger datasets are available as a collection of heterogeneous datasets. However, previous work has highlighted the possible danger of simply training a model from the combined datasets, due to the presence of bias. In this paper, we present a model which jointly learns an LSM for each dataset as well as a compound LSM. The method provides a means to borrow statistical strength from the datasets while reducing their inherent bias. In experiments we demonstrate that the compound LSM, when tested on PASCAL, LabelMe, Caltech101 and SUN09 in a leave-one-dataset-out fashion, achieves an average improvement of over 6.5% *over a previous SVM-based undoing bias approach and an average improvement of over* 8.5% *over a standard LSM trained on the concatenation of the datasets.*

1. Introduction

The problem of object recognition has received much attention in Computer Vision. One of the most successful object recognition systems is based on Deformable Part-based Models (DPM), see [5, 9, 10, 30] and references therein. A special case of Latent SVMs are latent subcategory models (LSM) [5, 11, 30]. This approach has proved useful when the object we wish to classify/detect consists of multiple components, each of which captures different characteristics of the object class. For example, components may be associated with different viewpoints, light conditions, etc. Under these circumstances, training a single global classifier on the full dataset may result in a low complexity model which underfits the data. To address this, latent subcategory models train multiple subclassifiers simultaneously, each of which is associated with a specific linear classifier capturing specific characteristics of the object class.

Training these models is a challenging task due to the presence of many local optima in the objective function and the increased model complexity which requires large training set sizes. An obvious way to have larger training set sizes is to merge datasets from different sources. However, it has been observed by [21, 26] that training from combined datasets needs to be done with care. Although we would expect training a classifier from all available data to be beneficial, it may in fact result in decreased performance because standard machine learning methods do not take into account the bias inherent in each dataset. To address this problem several approaches have been considered, some of which we review in the next section.

The principal contribution of this paper is to extend LSMs to deal with multiple biased datasets. We address this problem from a multitask learning perspective [7], combining ideas from Computer Vision which have been put forward in [15]. This methodology leverages the availability of multiple biased datasets to tackle a common classification task (e.g. car classification) in a principled way. Specifically, we simultaneously learn a set of biased LSMs as well as a compound LSM (visual world model) which is constrained to perform well on a concatenation of all datasets. Although we focus on LSMs, the framework we present in this paper extends in a natural manner to training multiple latent part-based models. We describe a training procedure for our method and provide experimental analysis, which indicates that the method offers a significant improvement over both simply training a latent subcategory model from the concatenation of all datasets as well as the undoing bias

978-1-4673-6964-0/15/$31.00 ©2015 IEEE

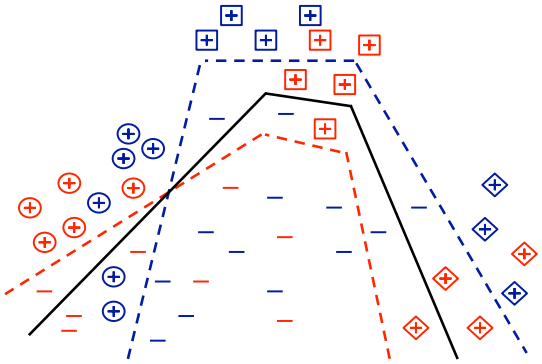

Figure 1. Parameter sharing across datasets can help to train a better subcategory model of the visual world. Here we have two datasets (red and blue) of a class (e.g. "dog"), each of which is divided into three subcategories (e.g. viewpoints). The red and blue classifiers are trained on their respective datasets. Our method, in black, both learns the subcategories and undoes the bias inherent in each dataset.

method of [15]. Hence, our approach achieves the best of both worlds, see Figure 1 for an illustration of the method.

As we noted earlier, training DPMs requires solving a difficult nonconvex optimization problem, which is prone to many local minima. Therefore, a good initialization heuristic is important in order to reach a good solution. As a second contribution of this paper, we observe that if the positive examples admit a good K-means clustering, and the regularization parameter associated used in the LSM is small relative to the cluster separation, then a good suboptimal solution for the LSM can be obtained by simply clustering the positive class and then training independent SVMs to separate each cluster from the negative examples. This result supports a commonly used heuristic for training subcategory models [5]. Furthermore, we describe how clustering initialization is performed in the multitask setting.

The paper is organized in the following manner. In Section 2 we review previous related work. Section 3 gives a short account of LSMs and Section 4 provides a justification for K-means based initialization schemes. Next, Section 5 presents our approach for training multiple LSMs from a collection of biased datasets. Section 6 reports our experimental results with this new method. Finally, Section 7 summarizes our findings and outlines future directions of research.

2. Related Work

Latent subcategory models (sometimes also called mixture of template models) [5, 11, 30] are a special case of DPMs [9, 10] and structured output learning [27]. Closely related methods have also been considered in machine learning under the name of multiprototype models or multiprototype support vector machines [1], as well as in optimization [17]. An important issue in training these models is the initialization of the subclassifier weight vectors. This issue has been addressed in [5, 11], where clustering algorithms such as K-means are used to cluster the positive class and subsequently independent SVMs are trained to initialize the weight vectors of the subclassifiers. In Section 4, we observe that K-means clustering can be justified as a good initialization heuristic when the positive class admits a set of compact clusters. Furthermore we discuss how this initialization can be adapted to our undoing bias setting.

We note that other initialization heuristics are discussed in [10]. Furthermore other interesting latent subcategory formulations are presented in [12] and [29]. While we do not consider their framework here, our method could also be extended to those settings, leading to interesting future research directions.

Most related to our paper is the work by Khosla *et al.* [15], which considers jointly training multiple linear max-margin classifiers from corresponding biased datasets. The classifiers pertain to the same classification (or detection) task (e.g. "car classification") but each is trained to perform well on a specific "biased" dataset. Their method is similar to the regularized multitask learning framework of Evgeniou and Pontil [7] with the addition that the common weight vector ("visual world" classifier) is constrained to fit the union of all the training datasets well. A key novelty of our approach is that we enhance these methods by allowing the common vector and bias vectors to be LSMs. We show experimentally that our method improves significantly over both [15] and a standard LSM trained on the concatenation of all datasets.

At last, we note that our model is different from the supervised domain adaptation methodology in [23, 16], which focuses on learning transformations between a source and a target domain. A key difference compared to these methods is that they require labels in the target domain, whereas our setting can be tested on unseen datasets, see also [15] for a discussion. Other related works include [13, 19, 20, 24].

3. Background on Latent Subcategory Models

In this section, we review latent subcategory models (LSMs). We let K be the number of linear subclassifiers and let $(\mathbf{w}_1, b_1), \ldots, (\mathbf{w}_K, b_K)$ be the corresponding parameters. A point $\mathbf{x}$ belongs to the subclass k if $\langle \mathbf{w}_k, \mathbf{x} \rangle + b_k > 0$, where $\langle \cdot, \cdot \rangle$ denotes the standard scalar product between two vectors. For simplicity, throughout the paper we drop the threshold b_k since it can be incorporated in the weight vector using the input representation $(\mathbf{x}, 1)$. A point $\mathbf{x}$ is classified as positive provided at least one subclassifier gives a positive output[1], that is, $\max_k \langle \mathbf{w}_k, \mathbf{x} \rangle > 0$. The geometric

[1]The model does not exclude that a positive point belongs to more than one subclass. For example, this would be the case if the subclassifiers are

interpretation of this classification rule is that the negative class is the intersection of K half-spaces. Alternatively, the positive class is the union of K half-spaces.

A standard way to learn the parameters is to minimize the objective function [5, 10, 30]

$$E_{K,\lambda}(\mathbf{w}) = \sum_{i=1}^{m} L(y_i \max_k \langle \mathbf{w}_k, \mathbf{x}_i \rangle) + \lambda \Omega(\mathbf{w}) \quad (1)$$

where $(\mathbf{x}_i, y_i)_{i=1}^m$ is a training sequence, L is the loss function and, with some abuse of notation, $\mathbf{w}$ denotes the concatenation of all the weight vectors. In this paper, we restrict our attention to the hinge loss function, which is defined as $L(z) = \max(0, 1 - z)$. However, our observations extend to any convex loss function which is monotonic nonincreasing, such as the logistic loss.

We denote by P and N the index sets for the positive and negative examples, respectively, and decompose the error term as

$$\sum_{i \in P} L(\max_k \langle \mathbf{w}_k, \mathbf{x}_i \rangle) + \sum_{i \in N} L(-\max_k \langle \mathbf{w}_k, \mathbf{x}_i \rangle). \quad (2)$$

Unless $K = 1$ or $K = |P|$, problem (1) is typically nonconvex[2] because the loss terms on the positive examples are nonconvex. To see this, note that $L(\max_k \langle \mathbf{w}_k, \mathbf{x} \rangle) = \min_k L(\langle \mathbf{w}_k, \mathbf{x} \rangle)$, which is neither convex nor concave[3]. On the other hand the negative terms are convex since $L(-\max_k \langle \mathbf{w}_k, \mathbf{x} \rangle) = \max_k L(-\langle \mathbf{w}_k, \mathbf{x} \rangle)$, and the maximum of convex functions remains convex.

The most popular instance of (1) is based on the regularizer $\Omega(\mathbf{w}) = \sum_k \|\mathbf{w}_k\|^2$ [5, 11, 30] and is a special case of standard DPMs [9]. Note that the case $K = 1$ corresponds essentially to a standard SVM, whereas the case $K = |P|$ reduces to training $|P|$ linear SVMs, each of which separates one positive point from the negatives. The latter case is also known as exemplar SVMs [18].

It has been noted that standard DPMs suffer from the "evaporating effect", see e.g. [10] for a discussion. This means that some of the subclassifiers are redundant, because they never achieve the maximum output among all subclassifiers. To overcome this problem, the regularizer has been modified to [10] $\Omega(\mathbf{w}) = \max_k \|\mathbf{w}_k\|^2$. This regularizer encourages weight vectors which have the same size at the optimum (that is, the same margin is sought for each component), thereby mitigating the evaporating effect. The corresponding optimization problem is slightly more involved since the above regularizer is not differentiable. However, similar techniques to those described below can be applied to solve the problem.

associated to different nearby viewpoints.

[2] If λ is large enough the objective function in (1) is convex, but this choice yields a low complexity model which may perform poorly.

[3] The function $L(\langle \mathbf{w}_k, \mathbf{x} \rangle)$ is convex in $\mathbf{w}_k$ but the minimum of convex functions is neither convex or concave in general, see e.g. [3].

A common training procedure to solve (1) is based on alternating minimization. We fix some starting value for $\mathbf{w}_k$ and compute the subclasses $P_k = \{i \in P : k = \operatorname{argmax}_\ell \langle \mathbf{w}_\ell, \mathbf{x}_i \rangle\}$. We then update the weights $\mathbf{w}_k$ by minimizing the convex objective function

$$F_{K,\lambda}(\mathbf{w}) = \sum_{k=1}^{K} \sum_{i \in P_k} L(\langle \mathbf{w}_k, \mathbf{x} \rangle) + \sum_{i \in N} L(-\max_k \langle \mathbf{w}_k, \mathbf{x}_i \rangle) + \lambda \Omega(\mathbf{w}_1, \ldots, \mathbf{w}_K). \quad (3)$$

This process is then iterated a number of times until some convergence criterion is satisfied. The objective function decreases at each step and in the limit the process converges to a local optimum.

A variation to (1) is to replace the error term associated with the negative examples by

$$\sum_k \sum_{i \in N} L(-\langle \mathbf{w}_k, \mathbf{x}_i \rangle) \quad (4)$$

see for example [11, 29]. This results in a simpler training procedure, in that the updating step reduces to solving K independent SVMs, each of which separates one of the clusters from all the negatives. Each step can then be solved with standard SVM toolboxes. Often in practice problem (1) is solved by stochastic subgradient methods, which avoid computations that require all the training data at once and are especially convenient for distributed optimization. Since the objective function is nonconvex, stochastic gradient descent (SGD) is applied to a convex upper bound to the objective, which uses a DC decomposition (difference of convex functions, see e.g. [2, 14] and references therein): the objective function is first decomposed into a sum of a convex and a concave function and then the concave term is linearly approximated around the current solution. This way we obtain an upper bound to the objective which we then seek to mininimize in the next step. We refer to [10] for more information.

Finally we note that LSMs are a special case of DPMs without parts. Specifically, a DPM classifies an image $\mathbf{x}$ into one of two classes according to the sign of the function $\max_{k,h} \mathbf{w}_k^\top \boldsymbol{\phi}_k(\mathbf{x}, h)$. Here $k \in \{1, \ldots, K\}$ is the latent component and h is an additional latent variable which specifies the position and scale of prescribed parts in the object, represented by the feature vector $\boldsymbol{\phi}_k(\mathbf{x}, h)$. LSMs do not consider parts and hence they choose $\boldsymbol{\phi}_k(\mathbf{x}, h) = \mathbf{x}$ and discard the maximum over h. Our methodology, presented below, extends to DPMs in a natural manner, however for simplicity in this paper we focus on LSMs.

4. Effect of Clustering Initialization

As we noted above, the objective function of an LSM (1) is nonconvex. In this section, we argue that if the posi-

tive points admit a good K-means clustering, then the minimizer of the function (3) provides a good suboptimal solution to the problem of minimizing (1). Our observations justify a standard initialization heuristic which was advocated in [5, 28].

Specifically, we assume that we have found a good K-means clustering of the positive data, meaning that the average distortion error

$$\sum_{i \in P} \min_k \|\mu_k - \mathbf{x}_i\|_2^2 \tag{5}$$

is small relative to the total variance of the data. In the above formula $\mu_1, \ldots, \mu_K$ denote the K means. We also let k_i be cluster index of point $\mathbf{x}_i$, that is $k_i = \operatorname{argmin}_k \|\mathbf{x}_i - \mu_k\|$, we let $\delta_i = \mathbf{x}_i - \mu_{k_i}$ and $\epsilon = \sum_{i \in P} \|\delta_i\|$. Then we can show that

$$\min_{\mathbf{w}} F_{K,\lambda'}(\mathbf{w}) \leq \min_{\mathbf{w}} E_{K,\lambda}(\mathbf{w}) \leq \min_{\mathbf{w}} F_{K,\lambda}(\mathbf{w}) \tag{6}$$

where $\lambda' = \lambda - 2\epsilon$. In other words, if ϵ is much smaller than λ then the gap between the upper and lower bounds is also small. In this case, the initialization induced by K-means clustering provides a good approximation to the solution of problem (1).

The right inequality in (6) holds since the objective function in problem (3) specifies the assignment of each positive point to a subclassifier and hence this objective is greater or equal to that in problem (1). The proof of the left inequality uses the fact that the hinge loss function is Lipschitz with constant 1, namely $|L(\xi) - L(\xi')| \leq |\xi - \xi'|$. In particular this allows us to give a good approximation of the loss $\min_k(1 - \langle \mathbf{w}_k, \mathbf{x}_i \rangle)$ in terms of the loss of the corresponding mean, that is, $\min_k(1 - \langle \mathbf{w}_k, \mu_{k_i} \rangle)$. A detailed derivation is presented in the supplementary material.

The bound (6) has a number of implications. First, as K increases, the gap between the upper and lower bound shrinks, hence the quality of the suboptimal solution improves. As K decreases down to $K = 2$ the initialization induced by K-means provides a more coarse approximation of problem (1), see also [30] for related considerations. Second, the bound suggests that a better initialization can be obtained by replacing K-means by K-medians, because the latter algorithm directly optimizes the quantity ϵ appearing in the bound.

We notice that a similar reasoning to the one presented in this section applies when the negative error term in (1) is replaced by (4). In this case, clustering the positive points, and subsequently training K independent SVMs which separate each cluster from the set of all negative points yields a good suboptimal solution of the corresponding nonconvex problem, provided the distortion parameter ϵ is smaller than the regularization parameter λ.

5. Learning from Multiple Biased Datasets

In this section, we extend LSMs described in Section 2 to a multitask learning setting [7]. Following [15] we assume that we have several datasets pertaining to the same object classification or detection task. Each dataset is collected under specific conditions and so it provides a biased view of the object class (and possibly the negative class as well). For example, if the task is people classification one dataset may be obtained by labelling indoor images as people / not people, whereas another dataset may be compiled outdoors, and other datasets may be generated by crawling images from the internet, etc. Although the classification task is the same across all datasets, the input data distribution changes significantly from one dataset to another. Therefore a classifier which performs well on one dataset may perform poorly on another dataset. Indeed, [15] empirically observed that training on the concatenation of all the datasets and testing on a particular dataset is outperformed by simply training and testing on the same dataset, despite the smaller training set size.

In the sequel, we let T be the number of datasets and for $t \in \{1, \ldots, T\}$, we let m_t be the sample size in training dataset t and let $\mathcal{D}_t = \{(x_{t1}, y_{t1}), \ldots, (x_{tm_t}, y_{tm_t})\} \subset \mathbb{R}^d \times \{-1, 1\}$ be the corresponding data examples. We assume that the images in all the datasets have the same representation so the weight vectors can be compared by simply looking at their pairwise Euclidean distance.

5.1. Undoing Bias SVM

In [15], the authors proposed a modified version of the multitask learning framework in [7] in which the error term includes an additional term measuring the performance of a compound model (visual world classifier) on the concatenation of all the datasets. This term is especially useful when testing the compound model on an "unseen" dataset, a problem we return upon in the sequel. Specifically, in [15] the authors learn a set of linear max-margin classifiers, represented by weight vectors $\mathbf{w}_t \in \mathbb{R}^d$ for each dataset, under the assumption that the weights are related by the equation $\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t$, where $\mathbf{w}_0$ is a compound weight vector (which is denoted as the *visual world* weight in [15]) and the vector $\mathbf{v}_t$ captures the bias of the t-th dataset. The weights $\mathbf{w}_0$ and $\mathbf{v}_1, \ldots, \mathbf{v}_T$ are then learned by minimizing a regularized objective function which leverages the error of the biased vectors on the corresponding dataset, the error of the visual world vector on the concatenation of the datasets and a regularization term which encourages small norms of all the weight vectors.

5.2. Undoing Bias LSM

We now extend the above framework to the latent subcategory setting. We let $\mathbf{w}_t^1, \ldots, \mathbf{w}_t^K \in \mathbb{R}^d$ be the weight

Object	Bird	Car	Person
K-means	33.8 $\pm$0.4	65.8 $\pm$ 0.4	67.5 $\pm$ 0.2
Random	29.4 $\pm$0.6	61.3 $\pm$ 0.5	64.7 $\pm$ 0.5

Table 1. AP (over 30 runs) of our method with and without K-means initialization for three object classification tasks.

vectors for the t-th dataset, for $t = 1, \ldots, T$. For simplicity, we assume that the number of subclassifiers is the same across the datasets, but the general case can be handled similarly. Following [7, 15], we assume that the weight vectors representative of the k-th subcategory across the different datasets are related by the equation

$$\mathbf{w}_t^k = \mathbf{w}_0^k + \mathbf{v}_t^k \tag{7}$$

for $k = 1, \ldots, K$ and $t = 1, \ldots, T$. The weights $\mathbf{w}_0^k$ are shared across the datasets and the weights $\mathbf{v}_t^k$ capture the bias of the k-th weight vector in the t-th dataset. We learn all these weights by minimizing the objective function

$$C_1 \sum_{t=1}^{T} \sum_{i=1}^{m_t} L(y_{ti} \max_k \langle \mathbf{w}_0^k + \mathbf{v}_t^k, \mathbf{x}_{ti} \rangle) \tag{8}$$

$$+ \quad C_2 \sum_{t=1}^{T} \sum_{i=1}^{m_t} L(y_{ti} \max_k \langle \mathbf{w}_0^k, \mathbf{x}_{ti} \rangle) \tag{9}$$

$$+ \quad \sum_{k=1}^{K} \left(\|\mathbf{w}_0^k\|^2 + \rho \sum_{t=1}^{T} \|\mathbf{v}_t^k\|^2 \right). \tag{10}$$

In addition to the number of subclassifiers K, the method depends on three other nonnegative hyperparameters, namely C_1, C_2 and ρ, which can be tuned on a validation set. Note that the method reduces to that in [15] if $K = 1$ and to the one in [7] if $K = 1$ and $C_2 = 0$. Furthermore our method reduces to training a single LSM on the concatenation of all datasets if $C_1 = 0$.

The parameter ρ plays an especially important role: it controls the extent to which the datasets are similar, or in other words the degree of bias of the datasets. Taking the limit $\rho \to \infty$ (or in practice setting $\rho \gg 1$) eliminates the bias vectors $\mathbf{v}_t^k$, so we simply learn a single LSM on the concatenation of all the datasets, ignoring any possible bias present in the individual datasets. Conversely, setting $\rho = 0$ we learn the bias vectors and visual world model independently. The expectation is that a good model lies at an intermediate value of the parameter ρ, which encourages some sharing between the datasets.

5.3. Implementation

A common method used to optimize latent SVMs as well as LSMs is stochastic gradient descent (SGD), see for example [25]. At each iteration we randomly select a dataset t and a point $\mathbf{x}_{ti}$ from that dataset and update the bias weight vector $\mathbf{v}_t^k$ and $\mathbf{w}_0^k$ by subgradient descent. We either train the SGD method with a fixed number of epochs or set a convergence criterion that tests the maximum change of the weight vectors. Furthermore, we use the adapting cache trick: if a point is correctly classified by at least two base and bias pairs $(\mathbf{w}_0^k, \mathbf{w}_t^k)$ then we give it a long cooldown. This means that the next 5 or 10 times the point is selected, we instead skip it. A similar process is used in [9, 15] and we verified empirically that this results in improved training times, without any significant loss in accuracy.

5.4. Initialization

It is worth discussing how the weight vectors are initialized. First, we group all the positive points across the different datasets and run K-means clustering. Let P_k be the set of points in cluster k, let $P_{t,k}$ be the subset of such points from dataset t and let N_t be the set of negative examples in the dataset t. For each subcategory $k \in \{1, \ldots, K\}$ we initialize the corresponding weight vectors $\mathbf{w}_0^k$ and $\mathbf{v}_1^k, \ldots, \mathbf{v}_T^k$ as the solution obtained by running the undoing datasets' bias method from [15], with training sets $\mathcal{D}_t = \{(\mathbf{x}_{ti}, y_{ti}) : i \in P_{t,k} \cup N_t\}$. We then iterate the process using SGD for a number of epochs (we use 100 in our experiments below).

Our observations in Section 4 extend in a natural way to the undoing bias LSMs setting. The general idea is the same: if the data admit a good K-means clustering then the initialization induced by K-means provides a good suboptimal solution of the problem. We have experimentally verified that the improvement offered by this initialization over a random choice is large. Table 1 reports the performance of the our method after 100 epochs of SGD starting with and without the K-means initialization. Average performance and standard deviation are reported over 30 trials. As it can be seen K-means initialization offers a substantial improvement[4].

6. Experiments

In this section, we present an empirical study of the proposed method. The goal of the experiments is twofold. On the one hand, we investigate the advantage offered by our method over standard LSMs trained on the union (concatenation) of all the datasets. Intuitively, we expect our method to learn a better set of visual world subclassifiers since it filters out dataset bias. On the other hand, we compare our method to the "undoing bias" method in [15], where each dataset is modelled as a linear SVM classifier (so no subclassifiers are learned in this case). As we already noted, both methods are special cases of ours for a certain choice of the hyperparameters.

[4] Preliminary experiments further indicate that K-medians improves by 0.4% over K-means, in agreement with our theoretical observations in Section 4, however but a detailed analysis is deferred to future work.

Test	$\mathbf{w}_{\text{PASCAL}}$	$\mathbf{w}_{\text{LabelMe}}$	$\mathbf{w}_{\text{Caltech101}}$	$\mathbf{w}_{\text{SUN09}}$	$\mathbf{w}_{\text{vw}}$	Aggregate	Independent
PASCAL	66.8 (64.8)	55.6 (50.5)	56.3 (54.2)	65.9 (51.2)	66.5 (57.0)	63.7 (57.4)	**67.1** (65.9)
LabelMe	73.1 (68.8)	**75.2** (72.9)	75.0 (71.2)	71.6 (73.3)	75.1 (72.4)	72.9 (72.9)	72.4 (71.7)
Caltech101	96.5 (94.8)	97.5 (94.6)	98.2 (**99.7**)	97.6 (95.6)	98.0 (98.9)	98.9 (97.0)	98.8 (99.4)
SUN09	57.2 (40.1)	57.6 (46.5)	57.7 (50.2)	58.0 (**59.6**)	57.8 (54.1)	53.9 (54.0)	58.9 (55.3)
Average	73.4 (67.1)	71.2 (66.2)	71.8 (68.8)	73.3 (69.9)	**74.5** (70.6)	72.4 (70.3)	74.3 (73.0)

Table 2. Average precision (AP) of "car classification" on seen datasets for our method ($K = 2$) and, within brackets, AP for the undoing bias method in [15].

In the experiments, we focus on object classification tasks as this allows us to directly compare with the results in [15] using the publicly available features provided by the authors[5]. However, the method can also be employed for detection experiments. Following the setting in [15] we employ four datasets: Caltech101 [8], LabelMe [22], PASCAL2007 [6] and SUN09 [4]. We use the bag-of-words representation provided by [15]. It is obtained by extracting SIFT descriptors at multiple patches, followed by local linear coding and a 3-level spatial pyramid with linear kernel. Performance of the methods is evaluated by average precision (AP).

We use the same training and test splits provided in [15]. Furthermore, to tune the model parameter C_1, C_2 and ρ, we use 75% of training data of each dataset for actual training and the remaining 25% for validation. We use the following parameter range for validation: $\rho = 10^r$, for r ranging from -9 to 4 with a step of 1 and $C_1, C_2 = 10^r$, for r ranging from -9 to 4 with a step of 0.5.

In our experiments, the number of subclassifiers K is regarded as a free hyperparameter chosen from the test set and we try values from 1 to 10. Although related work by [5] recommends using values of K up to 50, they consider detection tasks. However, as we show below, smaller values of K provide the best results for classification tasks, since the features employed in this case are extracted from larger images which are often dominated by the background rather than the object itself. This makes it more difficult to learn finer subcategories.

We test the methods in two different scenarios, following the "seen dataset" and "unseen dataset" settings outlined in [15]. In the first scenario we test on the same datasets used for training. The aim of this experiment is to demonstrate that the visual world model works better than a single model trained on the concatenation of the datasets, and it is competitive with a specific model trained only on the same domain. Furthermore, we show the advantage over setting $K = 1$. In the second scenario, we test the model on a new dataset, which does not contribute any training points. Here our aim is to show that the visual world model improves over just training a model on the concatenation of the training datasets as well as the visual world model from [15]. We discuss the results in turn.

6.1. Testing on Seen Datasets

In the first set of experiments, we test our method on "car classification" datasets. Results are reported in Table 2. The main numbers indicate the performance of our method, while within brackets we report performance for $K = 1$, which corresponds to the undoing bias method in [15]. In columns 2-5 we test the $\mathbf{w}_t^k$ on all datasets, for $t \in \{\text{PASCAL, LabelME, Calthech101, SUN09}\}$. In column 6 we test the visual world vectors $\mathbf{w}_0^k$ (denoted by $\mathbf{w}_{\text{vw}}$ in the tables). As noted above, in this set of experiments we test the method on the same datasets used during training (by this we mean that the training and test sets are selected within the same domain). For this reason and since the datasets are fairly large, we do not expect much improvement over training on each dataset independently (last column in the table). However, the key point of the table is that training a single LSM model on the union of all datasets (we call this the aggregate model in the tables) yields a classifier which neither performs well on any specific dataset nor does it perform well on average. In particular, the performance on the "visual world" classifier is much better than that of the aggregated model. This finding, due to dataset bias, is in line with results in [15], as are our results for $K = 1$. Our results indicate that, on average, using a LSM as the core classifier provides a significant advantage over using single max-margin linear classifier. The first case corresponds to a variable number of subclassifiers, the second case corresponds to $K = 1$. This is particularly evident by comparing the performance of the two visual world classifiers in the two cases.

6.2. Testing on Unseen Datasets

In the next set of experiments, we train our method on three out of four datasets, retain the visual world classifier and test it on the dataset left out during training[6]. Results are reported in Figure 2, where we show the relative improvement over training a single LSM on all datasets (ag-

[5] See the link *http://undoingbias.csail.mit.edu/*.

[6] That is, we predict as $\text{sign}(\max_k \langle \mathbf{w}_0^k, \mathbf{x} \rangle)$, where $\mathbf{w}_0^k$ are the compound subcategory models in equation (7).

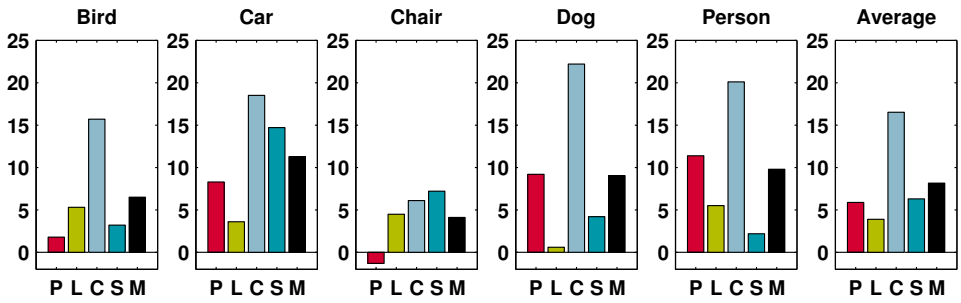

Figure 2. Relative improvement of undoing dataset bias LSM vs. the baseline LSM trained on all datasets at once (aggregated LSM). On all datasets at once (P: PASCAL, L: LabelMe, C: Caltech101, S: SUN09, M: mean).

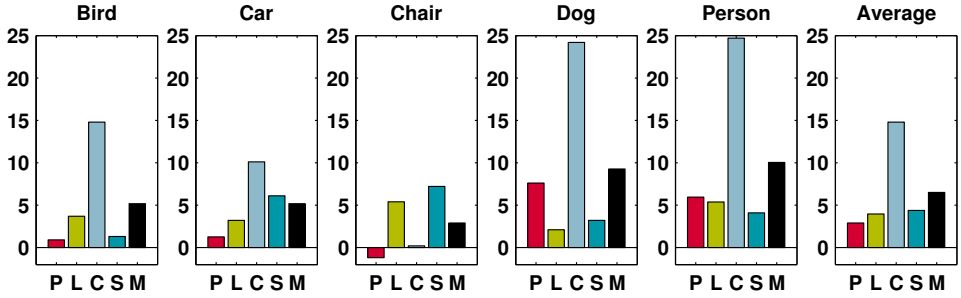

Figure 3. Relative improvement of undoing dataset bias LSM vs. undoing bias SVM [15]. (Legend as in Figure 2)

gregated LSM), and Figure 3, where we show the relative improvement of our method over the method in [15]. Overall our method gives an average improvement of more than 8.5% over the aggregated LSMs and an average improvement of more than 6.5% over [15]. On some datasets and objects the improvement is much more pronounced than others, although overall the method improves in all cases (with the exception of "chair classification" on the PASCAL dataset, where our method is slightly worse than the two baselines). Although our method tends to improve more over aggregated LSMs than undoing bias SVMs, it is interesting to note that on the Caltech101 "person" or "dog" datasets, the trend reverses. Indeed, these object classes contain only one subcategory (for "person" a centered face image, for "dogs" only Dalmatian dogs) hence when a single subcategory model is trained on the remaining three datasets a more confused classifier is learned.

To further illustrate the advantage offered by the new method, we display in Figure 4 the car images which achieved a top score on each of the four datasets for our method and the visual world classifier from [15]. In our case we use $K = 2$ subclassifiers because this gives the best performance on this object class. Note that among the two visual world subclassifiers, $\mathbf{w}_0^1$ and $\mathbf{w}_0^2$, the former tends to capture images containing a few cars of small size with a large portion of background (in order to verify this property please zoom in the figure), while the latter concentrates on images which depict a single car occupying a larger portion of the image. This effect is especially evident on the PASCAL and LabelMe datasets. On Caltech101, the first subclassifier is empty, which is not surprising as this dataset contains only images with well centered objects, so no cars belong to the first discovered subcategory. Finally the SUN09 dataset has fewer images of large cars and contributes less to the second subcategory. Note, however, that we still find images of single cars although of smaller size. The right portion of Figure 4 reports similar scores for the visual world classifier trained in [15] ($K = 1$). In this case we see that images of the two different types are present among the top scores, which indicates that the model is too simple and underfits the data.

7. Discussion and Conclusion

We presented a method for learning latent subcategories in presence of multiple biased datasets. Our approach is a natural extension of previous work on multitask learning to the setting of latent subcategory models (LSMs). In addition to the number of subclassifiers, the model depends upon two futher hyperparameters, which control the fit of the visual world LSM to all the datasets and the fit of each biased LSM to the corresponding dataset. In experiments, we demonstrated that our method provides significant improvement over both standard LSMs and previous undoing bias methods based on SVMs. Both methods are included in our framework for a particular parameter choice and our empirical analysis indicates our model achieves the best of

Figure 4. Left and center, the top scoring images for visual world subclassifiers $\mathbf{w}_0^1$ and $\mathbf{w}_0^2$ using our method. Right, the top scoring image for single category classifier $\mathbf{w}_0$ from [15].

both worlds: it mitigates the negative effect of dataset bias and still reaps the benefits of learning object subcategories. In future work, it would be valuable to extend ideas presented here to the setting of DPMs, in which the subclassifiers are part-based models. Furthermore, our observations on K-means initialization may be extended to other clustering schemes and other LSMs such as those described in [12] and [29]. Finally, learning LSMs across both biased datasets and different object classes provides an important direction of research.

Acknowledgements. We thank the reviewers for their helpful comments. MP acknowledges support from GENES and by the French National Research Agency grant Labex-ECODEC.

References

[1] F. Aiolli and A. Sperduti. Multiclass classification with multi-prototype support vector machines. *Journal of Machine Learning Research*, pages 817–850, 2005. 2

[2] A. Argyriou, R. Hauser, C. A. Micchelli, and M. Pontil. A DC-programming algorithm for kernel selection. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, 2006. 3

[3] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2009. 3

[4] M. Choi, J. J. Lim, A. Torralba, and A. Willsky. Exploiting hierarchical context on a large database of object categories. In *CVPR*, pages 129–136, 2010. 6

[5] S. K. Divvala, A. A. Efros, and M. Hebert. How important are deformable parts in the deformable parts model? In *ECCV Workshops and Demonstrations*, pages 31–40, 2012. 1, 2, 3, 4, 6

[6] M. Everingham, L. Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, (88):303–338, 2010. 6

[7] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 109–117, 2004. 1, 2, 4, 5

[8] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007. 6

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 1, 2, 3, 5

[10] R. Girshick and J. Malik. Training deformable part models with decorrelated features. In *ICCV*, pages 3016–3023, 2013. 1, 2, 3

[11] C. Gu and X. Ren. Discriminative mixture-of-templates for viewpoint classification. In *ECCV*, pages 408–421. 2010. 1, 2, 3

[12] M. Hoai and A. Zisserman. Discriminative sub-categorization. In *CVPR*, 2013. 2, 8

[13] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko. Discovering latent domains for multisource domain adaptation. In *ECCV*, pages 702–715. 2012. 2

[14] R. Horst and N. V. Thoai. DC programming: overview. *Journal of Optimization Theory and Applications*, 103:1–41, 1999. 3

[15] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV*, pages 158–171. 2012. 1, 2, 4, 5, 6, 7, 8

[16] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, pages 1785–1792, 2011. 2

[17] A. Magnani and S. Boyd. Convex piecewise-linear fitting. *Optimization and Engineering*, 10(1):1–17, 2009. 2

[18] T. Malisiewicz, A. Gupta, and A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, pages 89–96, 2011. 3

[19] A. Maurer, M. Pontil, and B. Romera-Paredes. Sparse coding for multitask and transfer learning. In *Proceedings of the 30th International Conference on Machine Learning*, pages 343–351, 2013. 2

[20] F. Mirrashed and M. Rastegari. Domain adaptive classification. In *ICCV*. 2013. 2

[21] J. Ponce, T. Berg, M. Everingham, D. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, B. C. Schmid, C.and Russell, A. Torralba, et al. Dataset issues in object recognition. In *Toward category-level object recognition*, pages 29–48. Springer, 2006. 1

[22] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1):157–173, 2008. 6

[23] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226. 2010. 2

[24] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, pages 1481–1488, 2011. 2

[25] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical Programming*, 127(1):3–30, 2011. 5

[26] A. Torralba and A. Efros. Unbiased look at dataset bias. In *CVPR*, pages 1521–1528, 2011. 1

[27] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, pages 1453–1484, 2005. 2

[28] Q. Ye, Z. Han, J. Jiao, and J. Liu. Human detection in images via piecewise linear support vector machines. *IEEE Transactions on Image Processing*, 22(2):778–789, 2013. 4

[29] X. Zhu, D. Anguelov, and D. Ramanan. Capturing long-tail distributions of object subcategories. In *CVPR*, 2014. 2, 3, 8

[30] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes. Do we need more training data or better models for object detection? In *BMVC*, pages 80.1–80.11, 2012. 1, 2, 3, 4

Supplementary Material

A. Derivation of Bound (6)

The right inequality readily follows by noting that the objective function in problem (3) considers an *a-priori* assignment of each positive point to a subclassifier, hence the objective is greater or equal to that in (1).

We now prove the left inequality. We consider a more general setting, in which the loss function L is convex and Lipschitz. The latter property means that there exists a constant ϕ such that for every $\xi, \xi' \in \mathbb{R}$, $|L(\xi) - L(\xi')| \leq \phi|\xi - \xi'|$. For example the hinge loss is Lipschitz with constant $\phi = 1$.

Choosing $\xi = \langle \mathbf{w}_k, \mathbf{x}_i \rangle$, $\xi' = \langle \mathbf{w}_k, \mu_{k_i} \rangle$ and letting $\delta_i = \mathbf{x}_i - \mu_{k_i}$, we obtain

$$|L(\langle \mathbf{w}_k, \mu_{k_i} \rangle) - L(\langle \mathbf{w}_k, \mathbf{x}_i \rangle)| \leq \phi |\langle \mathbf{w}_k, \delta_i \rangle| \leq \phi \|\mathbf{w}_k\| \|\delta_i\|$$

where the last step follows by Cauchy-Schwarz inequality. Furthermore, using the property that, for every choice of functions $f_1, \ldots, f_K$, it holds $|\min_k f_k(x) - \min_k f_k(x')| \leq \max_k |f_k(x) - f_k(x')|$, we have

$$|\min_k L(\langle \mathbf{w}_k, \mu_{k_i} \rangle) - \min_k L(\langle \mathbf{w}_k, \mathbf{x}_i \rangle)| \leq \max_k |L(\langle \mathbf{w}_k, \mu_{k_i} \rangle) - \min_k L(\langle \mathbf{w}_k, \mathbf{x}_i \rangle)| \leq \phi \max_k \|\mathbf{w}_k\| \|\delta_i\|.$$

Letting $\epsilon = \phi \sum_{i \in P} \|\delta_i\|$, we conclude, for every choice of the weight vectors $\mathbf{w}_1, \ldots, \mathbf{w}_K$, that

$$\sum_{k=1}^{K} p_k L(\langle \mathbf{w}_k, \mu_k \rangle) - \epsilon \max_k \|\mathbf{w}_k\| \leq \sum_{i \in P} \min_k L(\langle \mathbf{w}_k, \mathbf{x}_i \rangle) \leq \sum_{k=1}^{K} p_k \min_\ell L(\langle \mathbf{w}_\ell, \mu_k \rangle) + \epsilon \max_k \|\mathbf{w}_k\| \tag{11}$$

where P is the set of positive points, $P_k = \{i \in P : k_i = k\}$ and $p_k = |P_k|$, that is the number of positive points in cluster k.

Now, we define the surrogate convex function

$$S_{K,\lambda}(\mathbf{w}) = \sum_{k=1}^{K} p_k L(\langle \mathbf{w}_k, \mu_k \rangle) + \sum_{i \in N} L(-\max_k \langle \mathbf{w}_k, \mathbf{x}_i \rangle) + \lambda \max_k \|\mathbf{w}_k\|, \tag{12}$$

where $\mathbf{w}$ is a shorthand for the concatenation of the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_K$. Using equation (**??**) we obtain that

$$S_{K,\lambda-\epsilon}(\mathbf{w}) \leq E_{K,\lambda}(\mathbf{w}) \leq S_{K,\lambda+\epsilon}(\mathbf{w}). \tag{13}$$

Now using the fact that

$$L(\langle \mathbf{w}_k, \mu_{k_i} \rangle) \geq L(\langle \mathbf{w}_k, \mathbf{x}_i \rangle) - \|\mathbf{w}_k\| \|\delta_i\|$$

and recalling equation (3), we conclude that

$$F_{K,\lambda-2\epsilon}(\mathbf{w}) = \sum_{k=1}^{K} \sum_{i \in P_k} L(\langle \mathbf{w}_k, \mathbf{x}_i \rangle) + \sum_{i \in N} L(-\max_k \langle \mathbf{w}_k, \mathbf{x}_i \rangle) + (\lambda - 2\epsilon) \max_k \|\mathbf{w}_k\| \leq S_{K,\lambda-\epsilon}(\mathbf{w}).$$

The result follows by combining the left inequality in (**??**) with the above inequality and minimizing over the weight vectors $\mathbf{w}_1, \ldots, \mathbf{w}_K$.

B. Effect of Initialization in Undoing Bias LSM

As we noted in the Section 5.4, the initialization induced by K-means clustering can be extended in a natural way to the undoing bias LSM setting. We first run K-means on the aggregate set of positive points from all datasets. We let $P_{t,k}$ be the subset of positive points in dataset t which belong to cluster k and let N_t be the set of negative points in the same dataset. For each subcategory $k \in \{1, \ldots, K\}$, we initialize the corresponding weight vectors $\mathbf{w}_0^k$ and $\mathbf{v}_1^k, \ldots, \mathbf{v}_T^k$ as the solution obtained by running the undoing bias method in [15], with training sets $\mathcal{D}_t = \{(\mathbf{x}_{ti}, y_{ti}) : i \in P_{t,k} \cup N_t\}$. Specifically, for each k, we solve the problem

$$\sum_{t=1}^{T} \sum_{i \in P_{t,k} \cup N_t} \Big[C_1 L(y_{ti} \langle \mathbf{w}_0 + \mathbf{v}_t, \mathbf{x}_{ti} \rangle) + C_2 L(y_{ti} \langle \mathbf{w}_0, \mathbf{x}_{ti} \rangle) \Big] + \|\mathbf{w}_0\|^2 + \rho \sum_{t=1}^{T} \|\mathbf{v}_t\|^2.$$

Test	bird	car	chair	dog	person
random	3.9 (0.1)	17.9 (0.1)	7.3 (0.1)	3.6 (0.3)	21.5 (0.1)
random followed by optimization	29.4 (0.6)	61.3 (0.5)	34.5 (0.1)	27.7 (0.8)	64.7 (0.5)
K-means	18.3 (0.3)	51.2 (0.6)	30.2 (0.2)	24.3 (0.3)	61.2 (0.3)
K-means followed by optimization	33.8 (0.4)	65.8 (0.4)	35.2 (0.2)	31.4 (0.3)	67.5 (0.2)

Table 3. Effect of initialization on AP on different image classification problems. Top to bottom: random initialization, random initialization and optimization (100 epochs of SGD), K-means initialization, K-means initialization and optimization (100 epochs of SGD).

We then attempt to minimize the objective function formed by equations (8)–(10) with SGD for a number of epochs. The computation of a subgradient for this objective function is outlined in Algorithm **??** below.

Using arguments similar to those outlined above we can show that this initialization gives a good approximation to the minimum of the non-convex objective (8)–(10), provided the average distortion error $\sum_t \sum_{i \in P} \|\delta_{ti}\|$ is small, where we let $\delta_{ti} = \min_k \|\mathbf{x}_{ti} - \mu_k\|$[7].

Table **??** illustrates the importance of this initialization process, using a fixed parameter setting over 30 runs, in a seen dataset setting. The first row shows the performance (average precision) of a random choice of $\mathbf{w}_0$ and $\mathbf{v}_1, \ldots, \mathbf{v}_T$. The second row shows the performance of our method starting from this random initialization. The third row shows the performance of the K-means induced initilization reviewed above. Finally, the fourth row is our method. As we see, K-means based initialization on its own already provides a fair solution. In particular, for "chair", "dog" and "person" there is a moderate gap between the performance of K-means based initialization and K-means followed by optimization. Furthermore, in all cases K-means followed by optimization provides a better solution than random initialization followed by optimization.

for $k \leq K$ **do**
  if $k = \arg\max_j \langle \mathbf{w}_0^j, \mathbf{x}_{ti} \rangle$ **and** $k = \arg\max_j \langle \mathbf{w}_0^j + \mathbf{v}_t^j, \mathbf{x}_{ti} \rangle$ **and** $y_{ti} \langle \mathbf{w}_0^k, \mathbf{x}_{ti} \rangle \leq 1$ **and** $y_{ti} \langle \mathbf{w}_0^k + \mathbf{v}_t^k, \mathbf{x}_{ti} \rangle \leq 1$ **then**
    $\partial_{\mathbf{w}_0^k} J = -C_1 y_{ti} \mathbf{x}_{ti} - C_2 y_{ti} \mathbf{x}_{ti} + \mathbf{w}_0^k$
  else if $k = \arg\max_j \langle \mathbf{w}_0^j, \mathbf{x}_{ti} \rangle$ **and** $y_{ti} \langle \mathbf{w}_0^k, \mathbf{x}_{ti} \rangle \leq 1$ **then**
    $\partial_{\mathbf{w}_0^k} J = -C_2 y_{ti} \mathbf{x}_{ti} + \mathbf{w}_0^k$
  else if $k = \arg\max_j \langle \mathbf{w}_0^j + \mathbf{v}_t^j, \mathbf{x}_{ti} \rangle$ **and** $y_{ti} \langle \mathbf{w}_0^k + \mathbf{v}_t^k, \mathbf{x}_{ti} \rangle \leq 1$ **then**
    $\partial_{\mathbf{w}_0^k} J = -C_1 y_{ti} \mathbf{x}_{ti} + \mathbf{w}_0^k$
  else
    $\partial_{\mathbf{w}_0^k} J = \mathbf{w}_0^k$
end

for $k \leq K$ **do**
  if $k = \arg\max_j \langle \mathbf{w}_0^j + \mathbf{v}_t^j, \mathbf{x}_{ti} \rangle$ **and** $y_{ti} \langle \mathbf{w}_0^k + \mathbf{v}_t^k, \mathbf{x}_{ti} \rangle \leq 1$ **then**
    $\partial_{\mathbf{v}_t^k} = -C_1 y_{ti} \mathbf{x}_{ti} + \rho \mathbf{v}_t^k$
  else
    $\partial_{\mathbf{v}_t^k} = \rho \mathbf{v}_t^k$
end

Algorithm 1: Computation of subgradient for the objective function (8)–(10).

[7]More precisely, our analysis of K-means initialization can be extended to regularizer $S(\mathbf{w}) = \sum_k ||\mathbf{w}_k||^2$ as well as the formulation in equations (8)-(10). To this end, we need an additional step using the inequality $\lambda S(\mathbf{w}) - \epsilon \max_k \|\mathbf{w}_k\| > (\lambda - \epsilon) S(\mathbf{w}) - \epsilon/4$. We postpone the full details to a future occasion.