Uncovering Shared Structures in Multiclass Classification

Abstract

We suggest a method for multi-class learning with many classes by simultaneously learning shared characteristics common to the classes, and predictors for the classes in terms of these characteristics. We cast this as a convex optimization problem, using *trace-norm* regularization, study gradient-based optimization both for the linear case and the kernelized setting, and show how this approach can yield improved classification accuracy.

1. Introduction

In this paper we address the question of how to utilize hidden structure in order to improve multiclass classification accuracy. Our goal is to provide a mechanism for learning the underlying characteristics that are shared between the target classes, and to demonstrate the benefit of extracting common characteristics. We build upon the powerful notion of large margin linear classifiers, and specifically focus on the recent extensions to multiclass settings (Crammer & Singer, 2001).

The challenge of accurate classification of an instance into one of a large number of target classes surfaces in many domains, such as object recognition, face identification, textual topic classification, and phoneme recognition. In many of these domains it is natural to assume that even though there are a large number of classes (e.g. different people in a face recognition task), classes are related and build on some underlying common characteristics. For example, many different mammals share characteristics such as a striped texture or an elongated snout, and people's faces can be identified based on underlying characteristics such as gender, being Caucasian, or having red hair. Recovering the true underlying characteristics of a domain can significantly reduce the effective complexity of the multiclass problem and by that transfer knowledge between related classes.

The obvious question that arises is how to select the feature mapping appropriate for a given task. One method to resolve this need is by manually designing a domain specific kernel (e.g. (Shpigelman et al., 2002)). When the route of manual kernel design is not feasible one can attempt to learn a data specific feature mapping (Crammer et al., 2002). In practice, researchers often simply test several of the standard kernels in order to assess which attains better performance on a validation set. These approaches, however, fail to provide a clear mechanism for utilizing the existence of structures in selecting the appropriate feature mapping. We would therefore like to find an efficient way to learn feature mappings that capture the underlying structure of a given set of classes.

The observation that learning a hidden representation of some shared characteristics can facilitate learning has a long history in multiclass learning (e.g. Dekel et al. (2004)). This notion is often termed learning-to-learn or interclass transfer (Thrun, 1996). While some approaches assume some information on the shared characteristics is provided to the learner in advance (Fink & Levi, 2004; Fink et al., 2006), others rely on various learning heuristics in order to extract the shared features (Torralba et al., 2004).

Simultaneously learning the underlying structure between the classes and the class models is a challenging optimization task. Many of the heuristic approaches stated above aim at extracting powerful non-linear hidden characteristics. However, this goal often entails non-convex optimization tasks, prone to local minima problems. In contrast, we will focus on modeling the shared characteristics, as linear transformations of the input space. Thus, our model will postulate a linear mapping of shared features, followed by a multiclass linear classifier. We will show that such models can be efficiently learned in a convex optimization scheme and that they can significantly improve the accuracy of multiclass linear classifiers, despite the fact that they are restricted to simple linear mappings of the instance space.

The rest of this paper is organized as follows. We begin by introducing our learning setting, motivating our approach and formulating the suggested learning rule (Sec. 2). By studying the dual of the resulting optimization problem, we show, in Section 3, how to "kernalize" our learning rule. Then, in Section 4, we discuss the learning rule in the context of learning a latent feature representation. In Section 5 we derive an optimization scheme and in Section 7 demon-

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

strate our approach on picture classification and handwritten letter recognition tasks.

2. Formulation

The goal of multiclass classification is to learn a mapping $H : \mathcal{X} \to \mathcal{Y}$ from instances in $\mathcal{X}$ to labels in $\mathcal{Y} = \{1, ..., k\}$. We consider linear classifiers over $\mathcal{X} = \mathbb{R}^n$, parametrized by a weight vector $W_y \in \mathbb{R}^n$ for each class $y \in \mathcal{Y}$, and which take the form:

$$H_W(x) = \operatorname*{argmax}_{y \in \mathcal{Y}} W_y^t \cdot x \quad . \tag{1}$$

We wish to learn the weights from a set of m labeled training examples $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, which we summarize in a matrix $X \in \mathbb{R}^{n \times m}$ whose columns are given by $\mathbf{x}_i$. Inspired by the large margin approach for classification, Crammer and Singer (2001) suggest learning the weights by minimizing a trade-off between an average empirical *loss* (to be discussed shortly) and a regularizer of the form:

$$\sum_y \|W_y\|^2 = \|W\|_{\text{F}}^2 \tag{2}$$

where $\|W\|_{\text{F}}$ is the Frobenius norm of the matrix W whose columns are the vectors W_y. The loss function suggested by Crammer *et al*is the maximal hinge loss over all comparisons between the correct class and an incorrect class:

$$\ell\left(W; (\mathbf{x}, y)\right) = \max_{y' \neq y} \left[1 + W_{y'}^t \cdot \mathbf{x} - W_y^t \cdot \mathbf{x}\right]_+ \tag{3}$$

where $[z]_+ = \max(0, z)$. For a trade-off parameter C, the weights are then given by the following learning rule:

$$\min_W \frac{1}{2}\|W\|_{\text{F}}^2 + C \sum_{i=1}^m \ell\left(W; (\mathbf{x}_i, y_i)\right) \quad . \tag{4}$$

For a binary classification problem, $\mathcal{Y} = \{1, 2\}$, this formulation reduces to the familiar Support Vector Machine (SVM) formulation (with $W_1 = -W_2 = \frac{1}{2}\mathbf{w}_{\text{svm}}$ at the optimum, and C appropriately scaled). For larger number of classes, the formulation generalizes SVMs by requiring a margin between every pair of classes, and penalizing, for each training example, the amount by which the margin constraing it violated. Similarly to SVMs the optimization problem Eq. (4) is convex, and by introducing a "slack variable" for each example, it can be written as quadratic programming. Crammer *et al*discuss practical optimization approaches.

Recall that our goal is to learn W better by modeling characteristics shared among multiple classes. We restrict ourselves to modelling each common characteristics r as linear functions $F_r^t \mathbf{x}$ of the input vectors $\mathbf{x}$. The activation of each class y is then taken to be a linear function $G_y^t(F^t\mathbf{x})$ of the vector $F^t\mathbf{x}$ of common characteristics, instead of a linear function of the input vectors. Formally our model substitutes the weight matrix $W \in \mathbb{R}^{n \times k}$ with the product $W = FG$ of a weight matrix $F \in \mathbb{R}^{n \times t}$, whose columns define the t common characteristics, and $G \in \mathbb{R}^{t \times k}$, whose columns predict the classes based on the common characteristics:

$$H_{G,F}(x) = \operatorname*{argmax}_{y \in \mathcal{Y}} G_y^t \cdot (F^t x) = \operatorname*{argmax}_{y \in \mathcal{Y}} (FG)_y^t \cdot x \quad , \tag{5}$$

It should be emphasized that if F and G are not constrained in any way, the hypothesis space defined by Eq. (1) and by Eq. (5) is identical, since any linear transformations induced by applying F and then G can always be attained by a single linear transformation W. We aim to show that nevertheless, regularizing the decomposition FG, as we discuss shortly, instead of the Frobenius norm of the weight matrix W, can yield a significant generalization advantage.

When the common characteristics F are known, we can replace the input instances $\mathbf{x}_i$ with the vectors $F^t\mathbf{x}_i$ and revert back to our original formulation Eq. (4), with the matrix G taking the role of the weight matrix. Each characteristic r is now a *feature* $(F^t\mathbf{x}_i)_r$ in this transformed problem. The challenge we address in this paper is of simultaneously learning the common characteristics (or latent features) F *and* the class weights G.

In order for the regularizer $\|G\|_{\text{F}}$ to be meaningful, we must also control the magnitude of F, suggesting regularizing, in addition to $\|G\|_{\text{F}}$, also $\sum_r \|F\|^2 = \|F\|_{\text{F}}^2$, yielding the learning rule:

$$\min_{F,G} \frac{1}{2}\|F\|_{\text{F}}^2 + \frac{1}{2}\|G\|_{\text{F}}^2 + C \sum_{i=1}^m \ell\left(FG; (\mathbf{x}_i, y_i)\right) \quad . \tag{6}$$

The norm of each F_r determines how "easy" it is for class predictors to use this characteristic: increasing the norm $\|F_r\|$ allows smaller values of G_{yr} to yield the same prediction, making it "cheaper" to use the characteristic. It is thus beneficial for useful characteristics to have high norm. But generalization ability is ensured by limiting the overall norm of characteristics. It is important to note that, as we are accustomed to in large-margin methods, we do not have to also limit the number of characteristics t. We are relying here on the *norm* of F and G for regularization, rather than their *dimensionality*.

The optimization objective of Eq. (6) is non-convex, and involves matrices of unbounded dimensionality. However, instead of explicitly learning F, G, the optimization problem Eq. (6) can also be written directly as a convex learning rule for W. Following Srebro et al. (2005), we consider the trace-norm of a matrix W:

$$\|W\|_\Sigma = \min_{FG=W} \frac{1}{2}(\|F\|_{\text{F}}^2 + \|G\|_{\text{F}}^2) \tag{7}$$

The trace-norm is a convex function of W, and can be characterized as the sum of its singular values (Boyd & Vandenberghe, 2004).

$$\|W\|_\Sigma = \sum_i |\gamma_i| \ , \tag{8}$$

Using Eq. (7), we can rewrite Eq. (6) as:

$$\min_W \|W\|_\Sigma + C\sum_{i=1}^{m} \ell\left(W;(\mathbf{x}_i, y_i)\right) \ . \tag{9}$$

Furthermore, following Fazel et al. (2001) and Srebro et al. (2005), the optimization problem Eq. (9) can be formulated as a semi-definite program (SDP).

To summarize, we saw how learning to classify based on shared characteristics yields a learning rule in which the Frobenius-norm regularization is replaced with a trace-norm regularization.

3. Dualization and Kernelization

So far, we assumed we have direct access to the feature representation $\mathbf{x}$. However, much of the success of large-margin methods stems form the fact that one does not need access to the feature representation itself, but only to the inner product between feature vectors, specified by a *kernel function* $\mathbf{k}(\mathbf{x}, \mathbf{x}')$. In order to obtain a kernelized form of trace-norm regularized multi-class learning, we first briefly describe the dual of Eq. (9), and how the optimum W can be obtained from the dual optimum.

By applying standard Lagrange duality we deduce the dual of Eq. (9) is given by the following optimization problem, which may also be written as a semi-definite program:

$$\max \sum_i (-Q_{iy_i}) \quad \textbf{s.t.} \quad \begin{array}{c} \forall_{i,j\neq y_i} \ \ Q_{ij} \geq 0 \\ \forall_i \ \ (-Q_{iy_i}) = \sum_{j\neq y_i} Q_{ij} \leq c \\ \|XQ\|_2 \leq 1 \end{array}$$

where $Q \in \mathbb{R}^{n\times k}$ denotes the dual Lagrange variable and $\|XQ\|_2$ is the spectral norm of XQ (i.e. the maximal singular value of this matrix). The spectral norm constraint can be equivalently specified as $\|(XQ)^t(XQ)\|_2 = \|Q^t(X^tX)Q\|_2 \leq 1$. This form is particularly interesting, since it allows us to write the dual in terms of the Gram matrix $K = X^tX$ instead of the feature representation X explicitly:

$$\max \sum_i (-Q_{iy_i}) \quad \textbf{s.t.} \quad \begin{array}{c} \forall_{i,j\neq y_i} \ \ Q_{ij} \geq 0 \\ \forall_i \ \ (-Q_{iy_i}) = \sum_{j\neq y_i} Q_{ij} \leq c \\ \|Q^tKQ\|_2 \leq 1 \end{array} \tag{10}$$

Although Eq. (10) is not a semi-definite program, it is a convex problem on Q that involves a semi-definite constraint (the spectral-norm constraint) on a matrix whose size is independent of the size of the training set, and only depends on the number of classes k.

The following Representer Theorem describes the optimum weight matrix W in terms of the dual optimum Q, and allows the use of the kernel mechanism for prediction.

Theorem 1 *Let Q be the optimum of Eq. (10) and V be the matrix of eignevectors of $Q'KQ$, then for some diagonal $D \in \mathbb{R}^{k\times k}$, the matrix $W = X\left(QV^tDV\right)$ is an optimum of Eq. (9), with $\|W\|_\Sigma = \sum_r |D_{rr}|$.*

Proof Using complementary slackness and following arguments similar to those of Srebro et al. (2005), it can be shown that XQ and the optimum W of Eq. (9) share the same singular vectors. That is, if $XQ = USV$ is the singular value decomposition of XQ, then $W = UDV$ for some diagonal matrix D. Furthermore $D_{rr} = 0$ whenever $S_{rr} \neq 1$, i.e. $SD = D$. Note also that the right singular vectors V of $XQ = USV$ are precisely the eigenvectors of $(XQ)^t(XQ) = Q^tX^tXQ = Q^tKQ$. We can now express W as follows: First note that $W = UDV$. Since $D = SD$ we may express W as $USDV$. Since $VV^t = I$ we may further expand this expression to $USVV^tDV$. Finally, replacing USV with XQ we obtain $X\left(QV^tDV\right)$.

Corollary 1 *There exists $\alpha \in \mathbb{R}^{m\times k}$ s.t. $W = X\alpha$ is an optimum of Eq. (9)*

The situation is perhaps not as pleasing as for standard SVMs where the weight vector can be explicitly represented in terms of the dual optimum solution. Here, even after obtaining the dual optimum Q, we still need to recover the diagonal matrix D. However, substituting $W = XQV^tDV$ into Eq. (9), the first term becomes $\sum_r |D_{rr}|$, while the second is piecewise linear in KQV^tDV. We therefore obtain a linear program (LP) in the k unknown entries on the diagonal of D, which can be easily solved to recover D, and hence W. It is important to stress that the number of variables of this LP depends only on the number of classes, and not on the size of the data set, and that the entire procedure (solving Eq. (10), extracting V and recovering D) uses only the Gram matrix K and does not require direct access to the explicit feature vectors X.

Even if the dual is not directly tackled, the representation of the optimum W guaranteed by Thm. 1 can be used to solve the primal Eq. (9) using the Gram matrix K instead of the feature vectors X, as we discuss in Section 5.

4. Learning a Latent Feature Representation

As alluded to above, learning F can be thought of as learning a latent feature space F^tX, which is useful for prediction. Since F is learned jointly over all classes, it effectively transfers knowledge between the classes. Low-norm decompositions were previously discussed in these terms by Srebro et al. (2005). More recently, Argyriou et al. (2007) studied a formulation equivalent to using the trace-norm explicitly for transfer learning between multiple tasks: consider k binary classification tasks, and use W_j as a linear predictor for the jth task. Using an SVM to learn each class independently corresponds to the learning rule:

$$\min_W \sum_j (\frac{1}{2}\|W_i\|^2 + C\ell_j(W_j)) = \min_W \frac{1}{2}\|W\|_{\mathrm{F}}^2 + C\sum_j \ell_j(W_j)$$

where $\ell_j(W_j)$ is the total (hinge) loss of W_j on the training examples for task j. Replacing the Frobenius norm with the trace norm :

$$\min_W \|W\|_\Sigma + C\sum_j \ell_j(W_j) \qquad (11)$$

corresponds to learning a feature representation $\phi(\mathbf{x}) = F^t\mathbf{x}$ that allows good, low-norm prediction for all k task, where the linear predictor for task j, in this feature space, is given by V_j. After such a feature representation is learned, a new task can be learned directly using the feature vectors $F^t\mathbf{x}$ using standard SVM machinery, taking advantage of the transfered knowledge from the other, previously-learned, tasks.

In the multi-class setting, the predictors W_y are never independent, as even in the standard Frobenius norm formulation Eq. (4), the loss couples together the predictors for the different classes. However, the between-class transfer afforded by implicitly learning shared characteristics is much stronger. As will be demonstrated later, such transfer is particularly important if only a few number of examples are available from some class of interest.

Although this paper studies multi-class learning, the technical contributions, including the optimization approach, study of the dual problem, and kernelization, apply equally well also to the multi-task formulation Eq. (11).

It is interesting to note that we can learn a feature representation $\phi(\mathbf{x}) = F^t\mathbf{x}$ even when we are not given the feature representation X explicitly, but only a kernel $\mathbf{k}$ from which we can obtain the Gram matrix $K = X^tX$. In this situation we do not have access to X, nor can we obtain F explicitly. As discussed above, what we *can* obtain is a matrix α such that $W = X\alpha$ is an optimum of Eq. (9). Let $W = UDV$ be the singular value decomposition of W (which we cannot calculate, since we do not have access to X). We have that $F = U\sqrt{D}$ is an optimum of Eq. (6). What we *can* calculate is the singular value decomposition of $\alpha^tK\alpha = \alpha^tX^tX\alpha = W^tW = V^tD^2V$, and thus obtain D and V (but not U). Now, note that $D^{-1/2}V\alpha^tK = D^{-1/2}V(\alpha^tX^t)X = D^{-1/2}VW^tX = D^{-1/2}VV^tDU^tX = D^{1/2}U^tX = F^tX$, providing us with an explicit representation of the learned feature space that we can calculate from K and α alone.

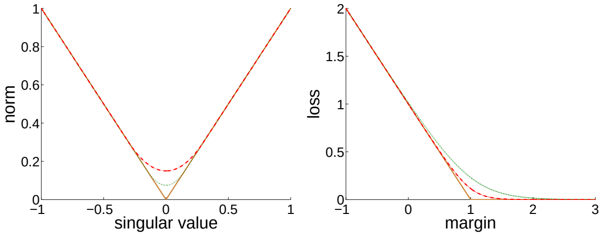

Figure 1. Left: The smoothed absolute value function g. Smaller values of r translate to a sharper function and a better estimate of the absolute values. Right: The binary version of the log-loss in comparison with the binary hinge-loss. Larger values of λ increase the accuracy of the log-loss approximation.

In either case, we should note the optimum of Eq. (6) is not unique, and so also the learned feature space is not unique: if F, G is an optimum of Eq. (6), then $(FR), (R^tG)$ is also an optimum, for any unitary matrix $RR^t = I$. Instead of learning the explicit feature representation $\phi(\mathbf{x}) = F^t\mathbf{x}$, we can therefore think of trace-norm regularization as learning the implied kernel $\mathbf{k}_\phi(\mathbf{x}', \mathbf{x}) = \langle F^t\mathbf{x}', F^t\mathbf{x}\rangle$. Even when F is rotated (and reflected) by R, the learned kernel $\mathbf{k}_\phi$ is unaffected.

5. Optimization

The optimization problem Eq. (9) can be formulated as a semi-definite program (SDP) and off-the-shelf SDP solvers can be used to recover the optimal W. However, such off-the-shelf solvers based on interior point methods scale very poorly with the size of the problem and typically cannot handle problems with more than several hundred dimensions, classes and training points. Moreover, the ability of interior point methods to obtain very accurate solutions to Eq. (9) is not particularly important in a machine learning application as the objective based on the training data is just a stochastic approximation of our true interest in generalization ability, and so obtaining a very precise solution to this approximation does not typically yield significant improvements in classification accuracy. Instead, we choose to optimize Eq. (9) using simple, but powerful gradient-based methods.

5.1. Gradient based optimization

The optimization problem Eq. (9) is non-differentiable and so not immediately amenable to gradient-based optimization. In order to perform the optimization, we consider a smoothed approximation to Eq. (9).

We begin by replacing the trace-norm with a smooth proxy. Eq. (8) characterizes the trace-norm as the sum of the singular values of W. Although the singular values are non-negative, the absolute value in Eq. (8) emphasizes the reason the trace-norm is non-differentiable when a singular value is zero and a singular vector abruptly changes direction. In order to obtain a smooth approximation to the trace-norm, we replace the non-smooth absolute value with a smooth function g defined as

$$g(\gamma) = \begin{cases} \frac{\gamma^2}{2r} + \frac{r}{2} & \gamma \leq r \\ |\gamma| & \text{otherwise} \end{cases} .$$

Where r is a some predefined cutoff point. Fig. 1 illustrates the function g and the effect of the parameter r. We can easily see that g is continuously differentiable, and that $\forall x : \big|g(x) - |x|\big| \leq \frac{r}{2}$. Our smoothed proxy for the trace norm is:

$$\|W\|_S = \sum_i g(\gamma_i) \tag{12}$$

where γ_i are the singular values of W. Its gradient can be calculated as:

$$\frac{\partial \|W\|_S}{\partial W} = U g'(D) V \tag{13}$$

where $W = UDV$ is the SVD of W and $g'(D)$ is an element-wise computation of the derivative g' of g on the diagonal of D.

We now turn our attention on the non-differentiable multiclass hinge-loss of Eq. (3). Since neither the hinge $[]_+$ nor the max operators are differentiable we employ an adaptation of the log-loss for the multiclass setting (Dekel et al., 2003), with a parameter γ controlling its sharpness (inspired by Zhang and Oles (2001)):

$$\ell_S\left(W; (\mathbf{x}_i y_i)\right) = \frac{1}{\lambda} \log\left(1 + \sum_{r \neq y_i} e^{\lambda \cdot (1 + W_r \cdot \mathbf{x}_i - W_{y_i} \cdot \mathbf{x}_i)}\right) .$$

This is a convex and continuously differentiable function of W which approaches the multiclass hinge-loss as $\lambda \to \infty$ (Fig. 1). In summary, instead of Eq. (9) we consider the following optimization problem:

$$\min_W \|W\|_S + C \sum_{i=1}^{m} \ell_S\left(W; (\mathbf{x}_i, y_i)\right) \tag{14}$$

which is a convex and continuously differentiable function.

Fig. 2 shows how optimization of the smoothed objective Eq. (14) approximately optimizes Eq. (9). We generated 160 training instances with 16 classes and 16-dimensional feature vectors using a weight matrix that is the product of two random 16×4 matrices. For each value of γ, and a fixed $r = 0.01$ we compared the weight matrix W recovered using conjugate gradient descent on Eq. (14) to the optimizer of Eq. (9) found using an interior point SDP solver . The figure plots the value of the original (non-smooth) objective of both solutions. For large values of γ, the smoothed optimization solves the original problem to within very good accuracy.

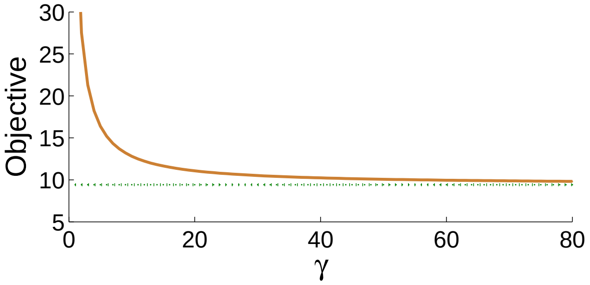

Figure 2. The values of the original (non-smooth) optimization objective Eq. (9) for minima of the smoothed objective Eq. (14) as a function of the smoothing parameter γ (solid) compared to the true optimum of Eq. (9) (dotted).

5.2. Kernelized gradient optimization

We now turn to devising a gradient-based optimization approach appropriate when only the Gram matrix $K = X^t X$ is available, but not the feature vectors X themselves. Corollary 1 assures us that the optimum of Eq. (9) is of the form $X\alpha$, and so we can substitute $W = X\alpha$ into Eq. (14) and minimize over α. To do so using gradient methods, we need to be able to compute both the smoothed objective and its derivative from K and α alone, without reference to X explicitly.

We first tackle the smoothed trace norm of $X\alpha$: Let $X\alpha = UDV$ denote the SVD of $X\alpha$ then the SVD of $\alpha^t K \alpha$ is given by $V^t D^2 V$. We can thus recover D from the SVD of $\alpha^t K \alpha$, and use Eq. (12) to calculate $\|X\alpha\|_S$.

In order to compute the gradient of $\|X\alpha\|_S$ with respect to α, we calculate:

$$\frac{\partial \|X\alpha\|_S}{\partial \alpha} = X^t \frac{\partial \|X\alpha\|_S}{\partial X\alpha} = X^t U g'(D) V$$

inserting $D(VV^t)D^{-1} = DID^{-1} = I$:

$$\begin{aligned} &= X^t U (D V V^t D^{-1}) g'(D) V \\ &= X^t (UDV) V^t D^{-1} g'(D) V \end{aligned}$$

and since $X\alpha = UDV$:

$$= X^t (X\alpha) V^t D^{-1} g'(D) V = K\alpha V^t D^{-1} g'(D) V \tag{15}$$

Recall that both V and D can be obtained from the SVD of $\alpha^t K \alpha$, and so Eq. (15) provides a calculation of the gradient in terms of K and α.

6. Spectral properties of the trace norm regularization

One way to appreciate the difference between the Frobenius norm and the trace norm of a matrix W is by observing that the squared Frobenius norm equals the sum of the squared singular values, $\sum_i \gamma_i^2$, while the trace norm is the sum of the singular values themselves, $\sum_i \gamma_i$. Thus, choosing to minimize $||U||^2_{Fro} + ||V||^2_{Fro}$ rather than $||W||^2_{\mathrm{F}}$, imposes a regularization preference for an L_1 norm on the spectrum of W (rather than an L_2 norm). When the various target classes share common characteristics we expect the spectrum of W to be non-uniform, since a large portion of the spectrum must be concentrated on few eigenvalues. In these cases the L_2 spectrum regularization imposed by the Frobenius norm will tend to attenuate the spectrum. In contrast, the L_1 spectrum regularization imposed by the trace norm does not share this tendency, and is thus better suited to preserve underlying structures of characteristics that are shared between the target classes.

In order to illustrate this effect we generated 100 classes over R^{120} and randomly sampled 4500 training instances from a 120-dimensional normal distribution. A 120×100 matrix W^* was then used to label the data, by choosing for each instance $\mathbf{x}$ the label $y = \underset{r \in \mathcal{Y}}{\operatorname{argmax}}\, W_r^* \cdot x$. The matrix W^* was selected to have a sigmoidal pattern of singular values, depicted in the dashed spectrum on Fig. 3. We then recovered two matrices W_{Fro} and W_Σ using the Frobenius norm optimization from Eq. (4) and the trace norm optimization from Eq. (9). The generalization error over 500 new test instances, was significantly higher for W_{Fro} (47%) than for W_Σ (31%). The spectrum of the two learned models is depicted in Fig. 3. It could be observed that Frobenius based regularization leads to the attenuated spectrum of W_{Fro}.

A question may arise whether it was possible to encourage the underlying common structure between the classes by applying a dimensionality reduction procedure to the weight matrix. In order to show this is not necessarily the case, we repeated the experiment described above, but W^* was selected to have the singular values form a harmonic series $(\frac{1}{1}, \frac{1}{2}, \ldots, \frac{1}{100})$. We similarly recovered two matrices W_{Fro} and W_Σ using the Frobenius norm optimization and the trace norm optimization . It was observed that the generalization error over 500 new test instances, was significantly higher for W_{Fro} (26%) than for W_Σ (17%).

Next, a singular value decomposition was performed on W_Σ and W_{Fro} followed by reconstructing these matrices using the p leading singular values and vectors ($p = 1, 2, \ldots, 100$). Performance of the reconstructed weight matrices was evaluated on the test set. It was observed that any SVD dimensionality reduction deteriorated the test performance. Moreover, the generalization error for the reduced W_{Fro} was consistently worse than the performance of the reduced W_Σ. It could therefore be concluded that post-hoc dimensionality reduction could not attenuate the importance of finding the underlying structure as an integral part of the learning procedure.

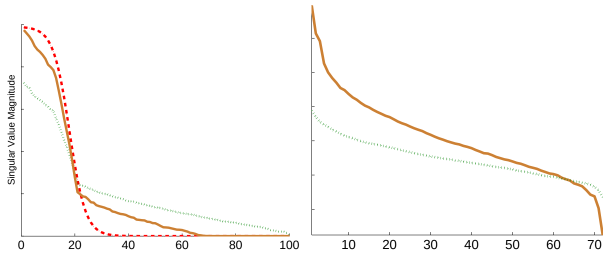

Figure 3. Spectra of learned matrices in the synthetic (left) and real (right) experiments. The weight matrix resulting from trace regularization (solid), and the weight matrix resulting from Frobenius regularization (dotted). The weight matrix that generated the data W^* (dashed) in the synthetic experiment only.

7. Experiments

7.1. Experiment I: Letter recognition

By analyzing over 100 writing systems, Changizi and Shimojo (2005) have demonstrated the fact that each writing system can be characterized by a set of underlying strokes. Therefore our first experiment focuses on recognition of the 26 characters made available in the UCI *letter* dataset. The data was composed of 2000 instances, roughly distributed over the 26 classes. The data was partitioned to three sets: 1000 were used as a training set, 500 were held out and used to select the optimal value of C and 500 were used as a test set. Data was represented using a Gaussian kernel with $\sigma = 0.07$.

We then recovered two matrices W_{Fro} (Frobenius norm regularization) from Eq. (4) and W_Σ (trace norm regularization) from Eq. (9). The trade-off parameter C was determined exhaustively by searching over 15 values between 2^{-10} and 2^5. The value was later fine tuned by searching within a smaller window within $C \cdot 2^{-1.5}$ and $C \cdot 2^{1.5}$. All values were tested on the fixed holdout set. Performance was evaluated over 500 new test instances, and the generalization error was significantly higher for W_{Fro} (10.1%) than for W_Σ (8.7%).

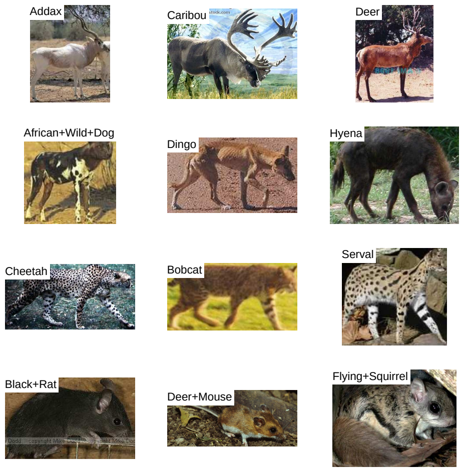

Figure 4. Representative images Deer (Addax, Caribou, Common Deer), Canines (African Wild Dog, Dingo, Hyena), Felines (Cheetah, Bobcat, Serval), and Rodents (Black Rat, Deer Mouse, Flying Squirrel).

7.2. Experiment II: Mammal recognition dataset

Our second experiment focused on the challenging task of classifying mammal images. We chose the 72 mammals that have at least 12 profile instances in the mammal benchmark made available by (Fink & Ullman, 2007). Of these, approximately 1,000 images were used for training and a similar number were used for testing. The test set was further partitioned, where half was held out and used to select C and the rest where used for testing. The number of instances of each class varied significantly from 6 to 30 training examples. It should be noted that the 72 target classes are expected to share many common characteristics due to genetic resemblance and evolutionary convergence. Four genetically related families (Deer, Canines, Felines and Rodents), are depicted in Fig. 4.

We build upon the comparison performed in (Zhang et al., 2006) in selecting an image representation suitable for the high degree of intraclass variability present in the mammal dataset. This representation is based on extracting a visual signature from the images. The visual signatures include 40 clusters of local descriptors, extracted from interest regions of the image. The resulting signatures are comparing using an Earth Moving Distance (EMD) Kernel. The EMD distance between signature-A and signature-B is found by solving the transportation problem, namely, by finding the minimal Euclidean distance necessary for converting the descriptors in signature-A to be identical to the descriptors of signature-B.

Using the above representation we learned the two matrices W_{Fro} (Frobenius norm regularization) and W_{Σ} (trace norm regularization). The trade-off parameter C was determined using the same procedure used in Experiment 7.1 The accuracy of the multiclass SVM based on trace norm regularization (33%) is observed to be higher than that attained using the Frobenius norm regularization (29%).

In the previous sections it was suggested that learning F can be thought of as learning a latent feature space $F^t X$, which is useful for prediction. Since F is learned jointly over all classes, it can be thought of as transferring knowledge between the classes. Under these conditions a new class can be acquired from fairly few training examples. We therefore predict that classes with few training examples will, on average gain more from applying trace norm regularization. This effect is depicted in Fig. 5. Specifically, it could be observed that of the few classes that gain from Frobenius regularization, four are of the top six most frequent mammals.

In order to verify this phenomenon we selected the most frequent class (Wombat), containing 30 training examples and repeatedly relearned W_{Fro} and W_{Σ} while gradually reducing the number of wombat examples to 24, 18, and 12. Under these conditions the accuracy of correct classification of wombats naturally deteriorated, but the effect was noticeably less severe for the trace norm regularization. While the Frobenius norm regularization performed better when all 30 instances where available during learning (by 2.2%), when 24 instances where available the gap had narrowed to 1.2%. When even fewer examples where available the leads where reversed and the trace norm outperformed the Frobenius norm by 1.4% for 18 instances and 3.7% for 12 instances. It should be noted that the false alarm rate over the remaining classes remained fairly constant. These results suggest that the learned common characteristics can indeed facilitate the acquisition of a novel class when only few examples are available for training.

Finally, the spectrum of the two learned models (Fig. 3), depicts the fact that Frobenius based regularization leads to the attenuated spectrum of W_{Fro}. It might be suggested that this effect manifests the advantage of trace norm regularization in preserving underlying structure between the mammal classes.

8. Discussion

We studied a learning rule for multi-class learning in which the magnitude of the factorization of the weight matrix is regularized, rather then the magnitude of the weights themselves. This is equivalent to regularizing the trace-norm of the weight matrix, instead of its Frobenius norm. We showed how this formulation can be kernelized, and solved efficiently either with direct access to the feature vectors or in a kernelized setting. We demonstrated the effectiveness of the formulation, particularly for classes with only a few

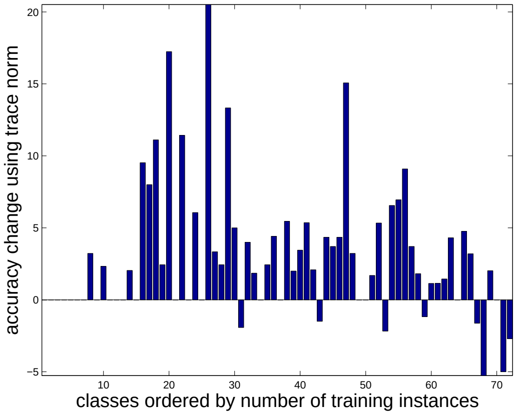

Figure 5. The x-axis contains the 72 mammal classes sorted by the number of instances available for training. The y-axis indicates the difference in performance entailed by choosing trace norm regularization over Frobenius norm regularization.

available training examples.

The multi-class formulation we study is a special case of a more general family of trace-norm regularized learning rules, where some general loss associated with the activation matrix W^tX replaces our multi-class loss:

$$\min_{W} \|W\|_{\Sigma} + C \cdot \mathit{loss}(W^tX). \tag{16}$$

Maximum Margin Matrix Factorization (Srebro et al., 2005) can be seen as a degenerate case of Eq. (16) where $X = I$ and the loss function decomposes over the entries of W. More recently, Argyriou et al. (2007) studied a learning rule which can be shown to be equivalent to Eq. (16), where the loss function again decomposes over the entries of W, corresponding to multiple independent tasks, but the feature vectors X are informative, and are explicitly available. Argyriou *et al*reach a different, but equivalent, formulation of the problem and suggest an optimization approach which requires iteratively solving multiple SVM problems. We believe that our formulation Eq. (16) is more direct and lends itself better to gradient-base optimization, which can be applied also for the multi-task setting. Additionally, our results on dualization, kernelization and representation of the learned latent feature space apply also to the multi-task setting studied by Argyriou *et al*, as well as to the general family of Eq. (16).

Another related learning rule using trace-norm regularization was studied by Abernethy et al. (2006). In their work, feature vectors are available for both “rows” and “columns” and so the prediction matrix is ZW^tX, rather than W^tX in Eq. (16), where Z is the matrix of “row” feature vectors. However, the trace-norm regularization is then applied to the prediction matrix ZW^tX, rather then to the weight matrix.

References

Abernethy, J., Bach, F., Evgeniou, T., & Vert, J.-P. (2006). *Low-rank matrix factorization with attributes*Technical report N24/06/MM). Ecole des Mines de Paris.

Argyriou, A., Evgeniou, T., & Pontil, M. (2007). Multi-task feature learning. In B. Schölkopf, J. Platt and T. Hoffman (Eds.), *Advances in neural information processing systems 19*. Cambridge, MA: MIT Press.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Changizi, M., & Shimojo, S. (2005). Character complexity and redundancy in writing systems over human history. *Proc Biol Sci. 2005 Feb 7;272(1560):267-75.*.

Crammer, K., Keshet, J., & Singer, Y. (2002). Kernel design using boosting.

Crammer, K., & Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*.

Dekel, O., Keshet, J., & Singer, Y. (2004). Large margin hierarchical classification. *Proceedings of the Twenty-First International Conference on Machine Learning*.

Dekel, O., Shalev-Shwartz, S., & Singer, Y. (2003). Smooth epsilon-insensitive regression by loss symmetrization. *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory*.

Fazel, M., Hindi, H., & Boyd, S. (2001). A rank minimization heuristic with application to minimum order system approximation. *Proceedings American Control Conference*.

Fink, M., & Levi, K. (2004). Encoding reusable perceptual features enables learning future categories from few examples. *Technical Report*.

Fink, M., Shalev-Schwartz, S., Singer, Y., & Ullman, S. (2006). Multiclass online learning by interclass hypotheseis sharing. *Proceedings of the International Conference on Machine Learning (ICML) 2006*.

Fink, M., & Ullman, S. (2007). From aardvark to zorro: A benchmark of mammal images.

Shpigelman, L., Singer, Y., Paz, R., & Vaadia, E. (2002). Spikernels: Embedding spiking neurons in inner-product spaces. *Advances in Neural Information Processing Systems 15*.

Srebro, N., Rennie, J., & Jaakkola, T. (2005). Maximum margin matrix factorization. *Advances in Neural Information Processing Systems (NIPS), 17*.

Thrun, S. (1996). *Learning to learn: Introduction*. Kluwer Academic Publishers.

Torralba, A., Murphy, K., & Freeman, W. (2004). Sharing visual features for multiclass and multiview object detection. *Proceedings of the Computer Vision and Pattern Recognition conference (CVPR) 2004*.

Zhang, J., Marszalek, M., Lazebnik, S., & Schmid, C. (2006). Local features and kernels for classification of texture and object categories: A comprehensive study. *CVPRW ’06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop* (p. 13). Washington, DC, USA: IEEE Computer Society.

Zhang, T., & Oles, F. J. (2001). Text categorization based on regularized linear classification methods. *Information Retrieval*, *4*, 5–31.