Incorporating Privileged Information Through Metric Learning

Shereen Fouad, Peter Tino, Somak Raychaudhury, and Petra Schneider

***Abstract*—In some pattern analysis problems, there exists expert knowledge, in addition to the original data involved in the classification process. Vast majority of existing approaches simply ignore such auxiliary (privileged) knowledge. Recently a new paradigm - Learning Using Privileged Information was introduced in the framework of SVM+. This approach is formulated for binary classification and, as typical for many kernel based methods, can scale unfavorably with the number of training examples. While speeding-up training methods and extensions of SVM+ to multi-class problems are possible, In this contribution we present a more direct novel methodology for incorporation of valuable privileged knowledge in the model construction phase, primarily formulated In the framework of Generalized Matrix Learning Vector Quantization. This is done by changing the global metric in the input space, based on distance relations revealed by the privileged information. Hence, unlike in SVM+, any convenient classifier can be used after such metric modification, bringing more flexibility to the problem of incorporating privileged information during the training. Experiments demonstrate that manipulation of input space metric based on privileged data improves classification accuracy. Moreover, our methods can achieve competitive performance against the SVM+ formulations.**

***Index Terms*—Learning Using Privileged Information (LUPI), Generalized Matrix Learning Vector Quantization (GMLVQ), Distance Metric Learning (DML), Information Theoretic Metric Learning (ITML).**

I. Introduction

TRADITIONALLY in classification learning problems the learner is given a labeled training set T of examples $x_i \in X$ from a data space X and aims to find a decision function f (preferably with a small generalization error) over the domain X. Although the main data set plays an important role when designing a classifier, additional privileged knowledge (represented through 'privileged space' X^*) may contain substantial information that might be used when constructing f. Designing classifiers that incorporate privileged knowledge along with the original data set is an important and challenging research issue.

Shereen Fouad and Peter Tino are with the School of Computer Science, The University of Birmingham, Birmingham B15 2TT, United Kingdom, (e-mail: saf942, P.Tino@cs.bham.ac.uk.)

Somak Raychaudhury is with the School of Physics and Astronomy, The University of Birmingham, Birmingham B15 2TT, United Kingdom, (e-mail:somak@star.sr.bham.ac.uk).

Petra Schneider is with the School for Clinical and Experimental Medicine, The University of Birmingham, Birmingham B15 2TT, United Kingdom, (e-mail:p.schneider@bham.ac.uk).

Recently, [1], [2] integrated privileged knowledge in Support Vector Machine (SVM) classifier via a new learning paradigm called *Learning Using Privileged Information* (LUPI). In the training stage, along with training input $x_i \in X$, a classifier may be given some additional information $x_i^* \in X^*$ about x_i. Such additional (privileged) information, however, will not be available in the test phase, where labels must be estimated using the trained model for previously unseen inputs $x \in X$ only (without x^*). In the SVM context, the additional information is used to estimate a slack variable model in SVM+. However,

1) SVM classifiers use decision hyperplane[1] and are inherently constructed to deal with binary classification problems. Even though there have been developments in extending SVM to multi-class scenarios (e.g. [3]), such formulations do not naturally represent the multi-class nature of the data in a single model.
2) It may be difficult to interpret how exactly the additional information influences the resulting classifier through the slack model in SVM+.
3) SVM+ training can be computationally expensive (even impractical for large-scale data sets).

This paper proposes a completely different approach to learning with privileged information through metric learning in prototype based models, particularly in the Learning Vector Quantization (LVQ) frameworks. LVQ models lend themselves naturally to multi-class problems, are more amenable to interpretations and can be constructed at a smaller computational cost.

In this paper we extend a recently proposed modification of LVQ, termed Generalized Matrix LVQ (GMLVQ) [4], [5], to the case of additional (privileged) information available only during the training phase. In GMLVQ the prototype positions, as well as the (global) metric in the data space X can be modified.

The main idea behind our approach is the modification of the metric in the original data space X based in data proximity 'hints' obtained from the privileged information space X^*. We present two approaches for metric manipulation in X based on X^*. We also introduce two methods for incorporating the new metric in X in the context of prototype based classification.

One of the main advantages of our approach is that, unlike in the SVM+ formulation [1], [2], the privileged information is used to manipulate the metric in the input

[1]in the original, or feature spaces

space and thus any convenient classifier can be subsequently used, bringing more flexibility to the problem of incorporating privileged information during the training.

We experimentally study the performance of our general methodology and compare it with the SVM+ model [1]. In addition, we illustrate its advantages in galaxy morphology classification using a large scale astronomical data set (on which application of the standard SVM based methodology would be computationally costly[2]).

This paper has the following organization: Section II gives the background and briefly describes previous methods related to this study. Sections III and IV introduce novel approaches for incorporation of privileged knowledge in prototype based classification. Experimental results presented in section VI are discussed in section VII. Section VIII concludes the study by summarizing the key contributions.

II. Background and Related Work

A. Learning Using Privileged Information (LUPI)

Learning Using Privileged Information (LUPI) framework [1], [2] aims to improve learning in the presence of an additional (privileged) information $x^* \in X^*$ about training examples $x \in X$, where the privileged information will not be available at the test stage, For example, when classifying proteins based on their amino-acid sequences, protein 3D-structures can be used as privileged information In [1]. Another example is time series prediction, where future events (present in the training set, but not available in the test phase) form privileged information. The incorporation of the privileged information into training was formulated within the Support Vector Machine (SVM) framework, in particular, [1], [2] present a new learning scheme for SVM based on SVM+. In addition, several approaches have been introduced for incorporation of privileged information in the *unsupervised* learning context, e.g. [7].

The basic process of the original supervised Support Vector Machine (SVM) model starts with mapping the training data from the original input space into a higher dimensional feature space, by using kernels, so that a linearly non separable problem is transformed into a linearly separable one. Within the feature space, the hyperplane with maximum margin is constructed to separate two classes in case of binary classification. In order to find the hyperplane, SVM model presents an objective function in a dual form and employs quadratic programming to solve the optimization problem. If the training set is not linearly separable, the standard SVM model allows the decision margin to make a few "mistakes" represented by slack variables (ξ_i).

In the standard SVM classification [8] we are given a set of (input,label) pairs, $\{(x_1, y_1), ..., (x_n, y_n)\}, x_i \in X, y_i \in \{-1, 1\}, i = 1, ..., n$, generated according to a fixed (but unknown) probability measure $P(x, y)$. The data is used to estimate a decision function $h(z_i) = \langle w, z_i\rangle + b$, where $\langle \cdot, \cdot \rangle$ represents the dot product and w, b are solutions of:

$$\min_{w,b,\xi_i} \frac{1}{2}||w||_2^2 + B\sum_{i=1}^{n}\xi_i \quad \text{under the constraints,}$$

$$\forall 1 \le i \le n, \quad y_i(\langle w, z_i\rangle + b) \ge 1 - \xi_i, \quad \xi_i \ge 0,$$

where $B \ge 0$ is a hyper-parameter. Training inputs x_i are (implicitly) transformed to their feature space images z_i through the use of 'kernel trick': Given a kernel $\mathcal{K}$, $\mathcal{K}(x_i, x_j)$ represents a dot product $\langle z_i, z_j\rangle$ in the corresponding Hilbert space.

In the LUPI framework additional information $x_i^* \in X^*$ may be given about a training example $x_i \in X$ during the training stage. However, such information will not be available (i.e. hidden) at the test stage. In the SVM+ model we are given a set of training triplets,

$$\{(x_1, x_1^*, y_1), ..., (x_n, x_n^*, y_n)\}\, x_i \in X, x_i^* \in X^*,$$

$$y_i \in \{-1, 1\}, \, i = 1, ..., n,$$

generated according to a fixed (unknown) probability measure $P(x, x^*, y)$. The training triplets are used to estimate two linear functions concurrently:

1) The decision function $h(z_i) = \langle w, z_i\rangle + b$
2) A correcting function (i.e. slack function) $\xi_i = \langle w^*, z_i^*\rangle + b^*$, where w^* , w, b and $b*$ are the solutions of

$$\min_{w,b,w^*,b^*} \frac{1}{2}||w||_2^2 + \frac{\rho}{2}||w^*||_2^2 + B\sum_{i=1}^{n}(\langle w^*, z_i^*\rangle + b^*)$$

$$\text{under the constraints,} \quad \forall 1 \le i \le n,$$

$$y_i(\langle w, z_i\rangle + b) \ge 1 - (\langle w^*, z_i^*\rangle + b^*), (\langle w^*, z_i^*\rangle + b^*) \ge 0$$

In SVM+ model correcting functions control the slack variables based on the privileged information. The objective function of SVM+ contains two hyper-parameters $B, \rho > 0$. Training triplets $(x_1, x_1^*, y_1), ..., (x_n, x_n^*, y_n)$ are transformed into the triplets $(z_1, z_1^*, y_1), ..., (z_n, z_n^*, y_n)$ by mapping vectors $x \in X$ into $z \in Z$ and $x^* \in X^*$ into $z^* \in Z^*$, where Z and Z^* are the corresponding feature spaces endowed with inner products $\langle z_i, z_j\rangle = \mathcal{K}(x_i, x_j)$, $\langle z_i^*, z_j^*\rangle = \mathcal{K}^*(x_i^*, x_j^*)$ defined by kernels $\mathcal{K}$ and $\mathcal{K}^*$.

In [1] another related approach, dSVM+, is introduced. In dSVM+ the space of admissible non-negative correcting functions is constrained to a 1-dimensional space (d-space). Privileged information x_i^* is transformed into so-called deviation (scalar) values d_i and the SVM+ method is applied to training triplets (x_i, d_i, y_i). For more details see [1].

It has been experimentally verified that classifiers trained with both privileged information $x_i^* \in X^*$ and original data $x_i \in X$ can improve over classifiers fitted on $x_i \in X$ only [1].

[2]There have been developments in the SVM literature aiming to handle large data sets (e.g.[6]). However, direct transformation of the LUPI framework to such formulations would be non-trivial

B. Prototype Based Learning Algorithms

Learning Vector Quantization (LVQ) [9] constitutes a family of supervised learning algorithms which are widely used for the classification of potentially high dimensional data. The classifiers are parametrized by a set of prototypical-vectors which represent the classes in the input space. In the working phase, an unknown sample is assigned to the class represented by the closest prototype. Kohonen introduced the original LVQ1 scheme in 1986 [10] which uses Hebbian online learning to adapt the prototypes to the training data. Meanwhile, researchers proposed numerous modifications of the basic learning scheme. Recent variations can be derived from an explicit cost function [11] or allow for the incorporation of adaptive distance measures [12], [4], [5].

Assume training data $(x_i, y_i) \in \mathbb{R}^m \times \{1, ..., K\}$, $i = 1, 2, ..., n$ is given, m denoting the data dimensionality and K is number of different classes. A typical LVQ network consists of L prototypes $w_q \in \mathbb{R}^m$, $q = 1, 2, 3, ..., L$, characterized by their location in the input space and their class label $c(w_q) \in \{1, ..., K\}$. Obviously, at least one prototype per class needs to be included in the model. The overall number of prototypes is a model hyper-parameter optimized e.g. in a data driven manner through a validation process. Given the (squared) Euclidean distance $d(x, w) = (x - w)^T(x - w)$ in $\mathbb{R}^m$ between input vectors and prototypes, classification is based on a winner-takes-all scheme: a data point $x_i \in \mathbb{R}^m$ is assigned to the label $c(w_j)$ of prototype w_j with $d(x, w_j) < d(x, w_q), \forall j \neq q$. Each prototype w_j with class label $c(w_j)$ will represent a receptive field in the input space[3]. Points in the receptive field of prototype w_j will be assigned class $c(w_j)$ by the LVQ model. The goal of learning is to adapt prototypes automatically such that the distances between data points of class $c \in \{1, ..., K\}$ and the corresponding prototypes with label c (to which the data belong) is minimized. In the training phase for each data point x_i with class label $c(x_i)$, the closest prototype with the same label is rewarded by pushing it closer to the training input; the closest prototype with different label is penalized by moving it away of the pattern x_i. In Generalized LVQ (GLVQ) algorithm [11], which is an expansion of the basic LVQ, prototypes adaptation is derived by minimizing of an explicit cost function with a stochastic gradient descent procedure. However, GLVQ suffers from the problem that the classification is based on a predefined Euclidean metric. The squared Euclidean distance can only be useful, if the data displays a Euclidean characteristic. This is particularly problematic in case of high-dimensional, heterogeneous data sets, where noise accumulates the data or different scaling and correlations of dimensions can be observed.

Recently, special attention was paid to schemes for manipulating the input space metric used to quantify 'similarity' between prototypes and feature vectors [12], [4]. Generalized Relevance LVQ (GRLVQ) [12] uses an adaptive diagonal matrix acting as a metric tensor defining the distance in the input space. The distance is a weighted squared Euclidean metric $d_\pi(x, w) = \sum_i \pi_i (x_i - w_i)^2$ with $\pi \in \mathbb{R}^m$, $\pi_i \geq 0$, $\sum_i \pi_i = 1$. During classification the parameters π_i weight the input dimensions according to their relevance, which helps to prune out 'irrelevant' dimensions (with respect to the classification task). An empirical and theoretical comparison of GRLVQ with SVM [13] has shown that the two model classes share several crucial advantages, such as convergence to global optimum[4], interpretation as large margin optimizers for which dimensionality independent generalization bounds exist and formulation of learning in a feature space defined by non-linear kernels.

The diagonal metric tensor of GRLVQ was further extended in [4], [5] to a fully adaptive metric tensor accounting for relevance factors as well as rotations of coordinate axis.

C. Generalized Matrix LVQ (GMLVQ)

Generalized Matrix LVQ (GMLVQ, see [4], [5]) is a new heuristic extension of the GRLVQ [12] with a full (e.g. not only diagonal elements) matrix tensor based distance measure. Matrix learning in the GMLVQ allows to account for different scalings and pairwise correlations between different features. Given an $(m \times m)$ positive definite matrix $\mathbf{\Lambda} \succ \mathbf{0}$[5], the algorithm uses a generalized form of the squared Euclidean distance

$$d_{\mathbf{\Lambda}}(x_i, w) = (x_i - w)^T \mathbf{\Lambda} (x_i - w). \tag{1}$$

Positive definiteness of $\mathbf{\Lambda}$ can be achieved by substituting $\mathbf{\Lambda} = \mathbf{\Omega}^T \mathbf{\Omega}$, where $\mathbf{\Omega} \in \mathbb{R}^{m \times m}$ is a full-rank matrix[6]. Furthermore, $\mathbf{\Lambda}$ needs to be normalized after each learning step to prevent the algorithm from degeneration. Here, we set $\sum_i \mathbf{\Lambda}_{ii} = 1$ to fix the sum of diagonal elements (eigenvalues) to be constant.

The model is trained in an on-line-learning manner, minimizing the cost function

$$f_{GMLVQ} = \sum_{i=1}^{n} \phi(\mu_{\mathbf{\Lambda}}(x_i)) \quad \text{where}$$
$$\mu_{\mathbf{\Lambda}}(x_i) = \frac{d_{\mathbf{\Lambda}}(x_i, w^+) - d_{\mathbf{\Lambda}}(x_i, w^-)}{d_{\mathbf{\Lambda}}(x_i, w^+) + d_{\mathbf{\Lambda}}(x_i, w^-)} \tag{2}$$

based on the steepest descent method. Φ is a monotonic function, e.g. the logistic function or the identity $\phi(\ell) = \ell$, $d_{\mathbf{\Lambda}}(x_i, w^+)$ is the distance of data point x_i from the closest prototype with the same class label $c(w^+) = c(x_i) = y_i$, and $d_{\mathbf{\Lambda}}(x_i, w^-)$ is the distance to x_i from the closest prototype w^- with a different class label than y_i. Note that the numerator is smaller than 0 if the classification of the data point is correct. The smaller the numerator, the

[3]The receptive field of prototype w is defined as the set of points in the input space which pick this prototype as their winner.

[4]if GRLVQ is combined with the Neural Gas model

[5] We use the notation $\mathbf{A} \succ \mathbf{0}$ and $\mathbf{A} \succeq \mathbf{0}$ to signify that $\mathbf{A}$ is positive definite and positive semi-definite, respectively.

[6] In some cases parts of the data relevant for classification can lie in a linear subspace of $\mathbb{R}^m$, In such situations $\mathbf{\Omega}$ (and thus $\mathbf{\Lambda}$) can be lower rank.

greater the 'security'[7] of classification, i.e. the difference of the distance from a correct and wrong prototype. The denominator scales the argument of ϕ such that it falls in the interval $[-1, 1]$.

Hebbian-like on-line updates are implemented for prototypes w^+, w^- along with the metric parameter $\mathbf{\Omega}$: w^+ is pushed towards the training instance x_i and w^- is pushed away from it. The derivatives of f_{GMLVQ} with respect to the prototypes w^+, w^- and the metric parameter $\mathbf{\Omega}$ yield the following adaptation rules [4], [5].

$$\begin{aligned}
&\Delta w^+ = +\epsilon_w \cdot \phi\prime(\mu_{\mathbf{\Lambda}}(x_i)) \cdot \gamma^+ \cdot \mathbf{\Lambda} \cdot (x_i - w^+),\\
&\Delta w^- = -\epsilon_w \cdot \phi\prime(\mu_{\mathbf{\Lambda}}(x_i)) \cdot \gamma^- \cdot \mathbf{\Lambda} \cdot (x_i - w^-),\\
&\Delta \mathbf{\Omega} = -\epsilon_{\mathbf{\Omega}} \phi\prime(\mu_{\mathbf{\Lambda}}(x_i))\\
&\left[\gamma^+ \cdot (\mathbf{\Omega}(x_i - w^+)(x_i - w^+)^T)\right.\\
&\left.-\gamma^- \cdot (\mathbf{\Omega}(x_i - w^-)(x_i - w^-)^T)\right],
\end{aligned}$$

where

$$\gamma^+ = \frac{4 d_{\mathbf{\Lambda}}(x_i, w^-)}{(d_{\mathbf{\Lambda}}(x_i, w^+) + d_{\mathbf{\Lambda}}(x_i, w^-))^2},$$
$$\gamma^- = \frac{4 d_{\mathbf{\Lambda}}(x_i, w^+)}{(d_{\mathbf{\Lambda}}(x_i, w^+) + d_{\mathbf{\Lambda}}(x_i, w^-))^2},$$

$\phi\prime$ is the derivative of ϕ and ϵ_w, $\epsilon_{\mathbf{\Omega}}$ are positive learning rates for prototypes and metric, respectively. For more details, please consult [4], [5].

D. Distance Metric Learning (DML)

Over the last few years, there has been considerable research on Distance Metric Learning (DML) algorithms which aim to optimize a target distance for a given set of data points under various types of constraints (given in the form of side information) [14], [15], [16], [17], [18], [19], [20], [21].

In the context of supervised metric learning, the distance metric is learned from training data associated with explicit class labels and pairwise similarity constraints. Such constraints indicate that points in the same class should have smaller distances to each other than points in different classes (e.g. Neighbourhood Components Analysis [14], Large Margin Nearest Neighbor [15]). In [22] generalization error of a regularized supervised DML formulation has been investigated - under appropriate constraints the generalization error is independent from the data dimensionality. In a different research stream [23], the metric is estimated within the Empirical Risk Minimization (ERM) framework. The learnt metric is consistent in the asymptotic regime of training set size approaching infinity. This work was further extended in [24] by proposing a constrained ERM DML framework. Generalization bound proved in [24] demonstrates the importance of the employed constraints.

[7]Note that, the 'security' of classification characterizes the hypothesis margin of the classifier. The larger this margin, the more robust is the classification of a data pattern with respect to noise in the input or function parameters [4], [12]

Supervised subspace selection approaches can be viewed as 'appropriately' changing the input features and metric in order to enhance the classification performance, e.g. Fisher's Linear Discriminant Analysis (FLDA) [25]. In multi-class classification, multi-class FLDA may merge classes which are close in the original data space. This problem has been addressed in [26]. Assuming (as in FLDA) that the classes are Gaussian-distributed with the same covariance matrix, the algorithm maximizes the geometric mean (rather than the arithmetic mean implicitly used in FLDA) of the (normalized) Kullback-Leibler (KL) divergences between the projected class distributions. The requirement of the same covariance matrix shared by all classes has been relaxed in the kernelized version of Max-Min Distance Analysis (MMDA) approach [27]. The method separates all class pairs by maximizing the minimum distance between the projected class pairs.

In semi-supervised metric learning, the distance metric is learnt from a weaker supervisory information, such as pairwise similarity constraints and partially available or completely absent class labels. The similarity constraints describe pairs of points that should, or should not be grouped together (e.g. Relevance Component Analysis [17], Discriminant Component Analysis [18])).

In the context of supervised clustering, the algorithm presented in [19] learns a metric using a semi-definite programming through minimizing the sum of squared distances between similarly labeled examples, while imposing a lower bound on the distances between examples with different labels. However, the algorithm suffers from high computational cost especially in the case of high-dimensional data.

In this research we will utilize an exiting supervised DML method, namely, Information Theoretic Metric Learning (ITML) [16] to learn a Mahalanobis distance metric for the original space X using a supervisory information (pairwise similarity constraints and class labels) extracted from the privileged space X^*.

In ITML [16] given a set of n points $\{x_1, ..., x_n\}$, $x_i \in \mathbb{R}^m$, one learns a positive definite matrix $\boldsymbol{A} \succ \mathbf{0}$ defining the the (squared) Mahalanobis distance $d_{\boldsymbol{A}}(x_i, x_j) = (x_i - x_j)^T \boldsymbol{A} (x_i - x_j)$, subject to categorical pairwise similarity information on the data points that should be preserved. In semi-supervised multi-class settings, the constraints are taken directly from the provided labels: points in the same class are constrained to be 'similar', and points in different classes are constrained to be 'dis-similar'. Consider distance bounds $l < u$. Then the new distance on the data space should be as close to the squared Euclidean metric as possible, subject to

$$\begin{aligned}
&d_{\boldsymbol{A}}(x_i, x_j) \le l, \quad \text{if } x_i, x_j \quad \text{are 'similar'}, \quad \text{and}\\
&d_{\boldsymbol{A}}(x_i, x_j) \ge u, \quad \text{if } x_i, x_j \quad \text{are 'dis-similar'}.
\end{aligned}$$

The closeness relation between the original Euclidean metric and the new one is measured through K-L divergence between the multivariate zero-mean Gaussians having $\boldsymbol{I}$ and $\boldsymbol{A}$ as precision matrices.

In this research we adopt ITML [16] as a supervised DML method because it can naturally incorporate prior distances and can be solved through efficient optimization avoiding costly computations (e.g. semi-definite programming as in [19])

III. LUPI in the Prototype Based Model GMLVQ

This section presents two metric learning approaches of incorporating Privileged information in the GMLVQ's learning phase. In the following algorithms data metric $\boldsymbol{U}$ is learnt in the original space informed by inter point distances in the privileged space.

A. Metric Fusion (MF) Approach

We propose a method that incorporates the distance structure in the privileged space X^* into the metric in the original space X. Assume that we are given a global metric tensor $\boldsymbol{M}$ on space X which parametrizes the (squared) Mahalanobis distance

$$d_{\boldsymbol{M}}(x_i, x_j) = (x_i - x_j)^T \boldsymbol{M}(x_i - x_j), \quad (x_i, x_j) \in X. \quad (3)$$

We assume that the data set is ordered such that the first $p \leq n$ data items have privileged information. The sum of pairwise squared distances of the training points with privileged information is then equal to

$$D = \sum_{i<j}^{p} d_{\boldsymbol{M}}(x_i, x_j). \quad (4)$$

Assume further that we are given a global metric tensor $\boldsymbol{M}^*$ on space X^* which parametrizes the (squared) Mahalanobis distance

$$d_{\boldsymbol{M}^*}(x_i^*, x_j^*) = (x_i^* - x_j^*)^T \boldsymbol{M}^*(x_i^* - x_j^*), \quad (x_i^*, x_j^*) \in X^*. \quad (5)$$

The sum of pairwise squared distances of the training points in X^* is then equal to

$$D^* = \sum_{i<j}^{p} d_{\boldsymbol{M}^*}(x_i^*, x_j^*). \quad (6)$$

In order to be able to directly compare the distances in X and X^*, we need to rescale the distances in X^* by a scaling factor α that levels out the difference in scales of D and D^*:

$$\alpha = \arg\min_{a>0} \left[D - aD^*\right]^2, \quad \text{leading to} \quad \alpha = \tfrac{D}{D^*}.$$

The proposed distance metric learning is formulated as the following optimization problem: Find a full-rank matrix $\boldsymbol{U}$ of size $m \times m$, parameterizing a positive-definite matrix $\boldsymbol{C} = \boldsymbol{U}^T\boldsymbol{U}$, that minimizes the cost function

$$I(\boldsymbol{C}) = \frac{2\gamma}{p(p-1)} \sum_{i<j}^{p} \left(d_{\boldsymbol{C}}\left(x_i, x_j\right) - \alpha\ d_{\boldsymbol{M}^*}\left(x_i^*, x_j^*\right)\right)^2 + \frac{2(1-\gamma)}{n(n-1)} \sum_{i<j}^{n} \left(d_{\boldsymbol{C}}\left(x_i, x_j\right) - d_{\boldsymbol{M}}\left(x_i, x_j\right)\right)^2. \quad (7)$$

where $\gamma \in [0, 1]$ is constant that determines the 'importance' of the auxiliary metric. There are two forces at play in the above expression: One pulls the new metric $d_{\boldsymbol{C}}$ in the direction of the metric $d_{\boldsymbol{M}^*}$ in the privileged space X^*, the other one prevents $d_{\boldsymbol{C}}$ from deviating too far from the distance $d_{\boldsymbol{M}}$ in the original space X. Note that the normalization terms $2/(p(p-1))$ and $2/(n(n-1))$ appear since not all training items have an associated privileged information (only $p \leq n$ out of n training points).

The cost function $I(\boldsymbol{U}^T\boldsymbol{U})$ is quartic (degree 4) in $\boldsymbol{U}$, which means that a gradient based optimization of I can get stuck in a local optimum. However, for unconstrained $\boldsymbol{C}$, $I(\boldsymbol{C})$ is quadratic in $\boldsymbol{C}$. We will initialize gradient descent optimization of $I(\boldsymbol{U}^T\boldsymbol{U})$ by first finding the unconstrained minimizer of $I(\boldsymbol{C})$ analytically, and then projecting it to the space of positive definite matrices parametrized by $\boldsymbol{U}^T\boldsymbol{U}$. In order to find $\boldsymbol{C}$ minimizing $I(\boldsymbol{C})$ we first differentiate

$$\begin{aligned}\frac{\mathrm{d}I}{\mathrm{d}\boldsymbol{C}} = \frac{4\gamma}{p(p-1)} \sum_{i<j}^{p} \big[&(x_i - x_j)^T\boldsymbol{C}(x_i - x_j) \\ &-\alpha(x_i^* - x_j^*)^T\boldsymbol{M}^*(x_i^* - x_j^*)\big] \cdot (x_i - x_j)(x_i - x_j)^T \\ &+\frac{4(1-\gamma)}{n(n-1)} \sum_{i<j}^{n} \big[(x_i - x_j)^T\boldsymbol{C}(x_i - x_j) - \\ &(x_i - x_j)^T\boldsymbol{M}(x_i - x_j)\big] \cdot (x_i - x_j)(x_i - x_j)^T. \quad (8)\end{aligned}$$

Denoting the rank-1 matrix $(x_i - x_j)(x_i - x_j)^T$ by $\boldsymbol{J}^{(i,j)}$, the optimal $\boldsymbol{C}$ is the solution of

$$\begin{aligned}&\frac{4\gamma}{p(p-1)} \sum_{i<j}^{p} (x_i - x_j)^T\boldsymbol{C}(x_i - x_j)\boldsymbol{J}^{(i,j)} \\ &+\frac{4(1-\gamma)}{n(n-1)} \sum_{i<j}^{n} (x_i - x_j)^T\boldsymbol{C}(x_i - x_j)\boldsymbol{J}^{(i,j)} \\ &= \frac{4\gamma}{p(p-1)} \sum_{i<j}^{p} \alpha(x_i^* - x_j^*)^T\boldsymbol{M}^*(x_i^* - x_j^*)\ \boldsymbol{J}^{(i,j)} \\ &+\frac{4(1-\gamma)}{n(n-1)} \sum_{i<j}^{n} (x_i - x_j)^T\boldsymbol{M}(x_i - x_j)\ \boldsymbol{J}^{(i,j)}. \quad (9)\end{aligned}$$

Note that

$$\begin{aligned}&(x_i - x_j)^T\boldsymbol{C}(x_i - x_j)\boldsymbol{J}^{(i,j)} \\ &= [(x_i - x_j)^T\boldsymbol{C}(x_i - x_j)](x_i - x_j)(x_i - x_j)^T \\ &= (x_i - x_j)(x_i - x_j)^T\boldsymbol{C}(x_i - x_j)(x_i - x_j)^T \\ &= \boldsymbol{J}^{(i,j)}\boldsymbol{C}\boldsymbol{J}^{(i,j)}.\end{aligned}$$

Therefore, denoting the RHS of (9) by $\boldsymbol{H}$, and introducing further notation

$$\boldsymbol{P}^{(i,j)} = 2\sqrt{\frac{\gamma}{p(p-1)}}\boldsymbol{J}^{(i,j)}, \quad \boldsymbol{N}^{(i,j)} = 2\sqrt{\frac{1-\gamma}{n(n-1)}}\boldsymbol{J}^{(i,j)},$$

we have

$$\sum_{i<j}^{p} \boldsymbol{P}^{(i,j)}\boldsymbol{C}\boldsymbol{P}^{(i,j)} + \sum_{i<j}^{n} \boldsymbol{N}^{(i,j)}\boldsymbol{C}\boldsymbol{N}^{(i,j)} = \boldsymbol{H}. \quad (10)$$

The solution $\boldsymbol{C}$ of the encapsulating sum system (10) can be written as

$$\mathrm{Vec}(\boldsymbol{C}) = \left[\sum_{i<j}^{p} \boldsymbol{P}^{(i,j)^T} \bigotimes \boldsymbol{P}^{(i,j)} + \sum_{i<j}^{n} \boldsymbol{N}^{(i,j)^T} \bigotimes \boldsymbol{N}^{(i,j)}\right]^{-1} \cdot \mathrm{Vec}(\boldsymbol{H}).$$

where $\bigotimes$ denotes the Kronecker product and Vec is the vectorization operator on matrices.

We found that the unconstrained solution $\boldsymbol{C}$ was typically already 'close' to being symmetric positive-definite. The L_2 projection of $\boldsymbol{C}$ onto the space of matrices parametrized by $\boldsymbol{U}^T\boldsymbol{U}$ can be found by minimizing

$$\boldsymbol{U}_0 = \arg\min_{\boldsymbol{U}} \ ||\boldsymbol{U}^T\boldsymbol{U} - \boldsymbol{C}||_2,$$

which is achieved e.g. by first finding a 2-norm positive approximant $\boldsymbol{G}$ of $\boldsymbol{C}$ [28] and then decomposing the positive definite matrix $\boldsymbol{G} \succ \boldsymbol{0}$ into the product $\boldsymbol{U}_0^T\boldsymbol{U}_0$ (Cholesky decomposition).

The projection $\boldsymbol{U}_0$ then initializes a gradient descent algorithm

$$\boldsymbol{U}_{t+1} = \boldsymbol{U}_t - \eta \cdot \frac{\mathrm{d}I(\boldsymbol{U}_t^T\boldsymbol{U}_t)}{\mathrm{d}\boldsymbol{U}_t}. \tag{11}$$

where $0 \leq \eta \leq 1$ is a positive step size parameter[8] and

$$\frac{\mathrm{d}I(\boldsymbol{U}^T\boldsymbol{U})}{\mathrm{d}\boldsymbol{U}} = \frac{8\gamma}{p(p-1)} \sum_{i<j}^{p} \Big[(x_i - x_j)^T\boldsymbol{U}^T\boldsymbol{U}(x_i - x_j) - \alpha(x_i^* - x_j^*)^T\boldsymbol{M}^*(x_i^* - x_j^*)\Big] \cdot \boldsymbol{U}(x_i - x_j)(x_i - x_j)^T + \frac{8(1-\gamma)}{n(n-1)} \sum_{i<j}^{n} \Big[(x_i - x_j)^T\boldsymbol{U}^T\boldsymbol{U}(x_i - x_j) - (x_i - x_j)^T\boldsymbol{M}(x_i - x_j)\Big] \cdot \boldsymbol{U}(x_i - x_j)(x_i - x_j)^T.$$

Unconstrained analytically obtained minimizer $\boldsymbol{C}$ of the cost function I (eq. (7)) is projected (with respect to the L_2-norm) onto the manifold $\mathcal{M}$ of symmetric positive definite matrices. The projection $\boldsymbol{U}_0^T\boldsymbol{U}_0$ is not necessarily the constrained minimizer of I (constrained to the manifold $\mathcal{M}$). We therefore run a gradient descent on I constrained to $\mathcal{M}$ to find the minimizer of I parametrized as $\boldsymbol{U}^T\boldsymbol{U}$.

B. Information Theoretic (IT) Approach

In the previous approach, the resulting squared metric $d_{\boldsymbol{C}}$ formed a 'compromise' between the squared metric $d_{\boldsymbol{M}}$ in the original space X and the scaled squared metric $\alpha \cdot d_{\boldsymbol{M}^*}$ in the privileged space X^*. The actual pairwise distances played a crucial role. In this section we suggest another approach where the privileged information is used to describe closeness relations between some of the points in a categorical manner only - e.g. the points are 'close' or 'far apart'. This categorical information is then imposed on the original space through the framework of Information Theoretic Metric Learning (ITML) [16] (see section II-D). Our aim is to learn a new metric in the original space which imposes small distances on points within the same class and with 'similar' associated privileged data, and large distances between points across different classes and with 'dis-similar' associated privileged information.

Consider training data (x_i, y_i), $i = 1, 2, ..., n$, as in section II-B. As before, additional information $x_i^* \in X^*$ is given about training examples $x_i \in X$, $i = 1, 2, ..., p \leq n$. Assume that we are given a global metric tensor $\boldsymbol{M}$ on space X defining the squared Mahalanobis distance $d_{\boldsymbol{M}}$ (3) We would like to modify $d_{\boldsymbol{M}}$ so that the distances under the new metric $d_{\boldsymbol{C}}$ on X are enlarged and shrunk for pairs of points that have 'dis-similar' and 'similar' privileged information, respectively.

In the ITML approach, two sets of pairs of data points from X are formed corresponding to the 'similar and dis-similar' data items:

- $S_+ = \{(x_i, x_j) | x_i \text{ and } x_j \text{ are judged to be similar}\}$
- $S_- = \{(x_i, x_j) | x_i \text{ and } x_j \text{ are judged to be dis-similar}\}$

We construct these two sets based in proximity information in the privileged space X^*. In particular, assume we are given a global metric tensor $\boldsymbol{M}^*$ on X^* giving the squared Mahalanobis distance $d_{\boldsymbol{M}^*}$ (5). We calculate all pairwise squared distances $d_{\boldsymbol{M}^*}(x_i^*, x_j^*)$, $1 \leq i < j \leq p$. These distances are then sorted in ascending order and, given a lower percentile parameter $a^* > 0$, a distance threshold l^* is found such that a^* percent of the lowest pairwise squared distances $d_{\boldsymbol{M}^*}(x_i^*, x_j^*)$ are smaller than l^*. Analogously, given an upper percentile parameter $b^* > a^*$, a distance threshold $u^* > l^*$ is found such that $(1 - b^*)$ percent of the largest pairwise squared distances $d_{\boldsymbol{M}^*}(x_i^*, x_j^*)$ are greater than u^*.

The sets S_+ and S_- are constructed using privileged information as follows:

- If $d_{\boldsymbol{M}^*}\left(x_i^*, x_j^*\right) \leq u^*$ and $c\left(x_i\right) = c\left(x_j\right) = y_i$ (same class label) then $(x_i, x_j) \in S_+$.
- If $d_{\boldsymbol{M}^*}\left(x_i^*, x_j^*\right) \geq l^*$ and $c\left(x_i\right) \neq c\left(x_j\right) \neq y_i$ (different class labels), then $(x_i, x_j) \in S_-$.

Note that it is not necessary for all training points in X to be involved pairs of points in S_+ or S_-.

In IT approach the 'similarity' between two metrics $d_{\boldsymbol{C}}$ and $d_{\boldsymbol{M}}$ on $X \subset \mathbb{R}^m$, given by metric tensors $\boldsymbol{C}$ and $\boldsymbol{M}$, respectively, is measured through the Bregman divergence (Burg). The divergence is defined over the cone of positive definite matrices as [16]:

$$D_{Burg}\left(\boldsymbol{C}, \boldsymbol{M}\right) = tr\left(\boldsymbol{C}\boldsymbol{M}\right)^{-1} - \log\det\left(\boldsymbol{C}\boldsymbol{M}\right) - m,$$

where tr denotes the trace operator and m is the data dimensionality. Given distance thresholds $0 < l < u$ on X, the Bregman divergence is minimized while enforcing the desired constraints:

$$\min_{\boldsymbol{C} \succ \boldsymbol{0}} D_{Burg}\left(\boldsymbol{C}, \boldsymbol{M}\right), \quad \text{subject to} \tag{12}$$

$$d_{\boldsymbol{C}}\left(x_i, x_j\right) \leq l, \text{ if } \left(x_i, x_j\right) \in S_+, \quad \text{and}$$
$$d_{\boldsymbol{C}}\left(x_i, x_j\right) \geq u, \text{ if } \left(x_i, x_j\right) \in S_-.$$

[8] We employed a line search algorithm to identify the 'optimal' value of η.

As in the original ITML formulation [16], in order to guarantee the existence of a feasible solution for $\boldsymbol{C}$, a slack variable ν is introduced: Let $s(i,j)$ denote the index of the (i,j)-th constraint, and let ξ be a vector of slack variables, initialized to ξ_0, with components equal l for similarity constraints and u for dissimilarity constraints. Then the optimization problem can be reformulated as [16]:

$$\min_{\boldsymbol{C}\succ\mathbf{0},\xi} D_{Burg}\left(\boldsymbol{C},\boldsymbol{M}\right)+\nu\cdot D_{Burg}\left(diag(\xi),diag(\xi_0)\right) \quad (13)$$

subject to

$$\begin{aligned} &d_{\boldsymbol{C}}\left(x_i,x_j\right)\leq\xi_{s(i,j)},\ \text{if}\ \ (x_i,x_j)\in S_+, \qquad \text{and}\\ &d_{\boldsymbol{C}}\left(x_i,x_j\right)\geq\xi_{s(i,j)},\ \text{if}\ \ (x_i,x_j)\in S_-. \end{aligned}$$

In IT approach the trade-off between the minimization problem and satisfying the constraints is controlled by the parameter ν, set through cross-validation. As in [16], optimizing (13) involves repeatedly projecting (Bregman projections) the current solution onto a single constraint, via the update:

$$\boldsymbol{C_{t+1}}=\boldsymbol{C}_t+\beta_t\boldsymbol{C}_t(x_{i_t}-x_{j_t})(x_{i_t}-x_{j_t})^T\boldsymbol{C}_t, \quad (14)$$

where x_{i_t} and x_{j_t} are data points associated with one of the (dis)similarity constraints from $S_\pm$ at time t and β_t is a projection parameter computed by the algorithm. The algorithm is initialized with $\boldsymbol{C}$ equal to the Mahalanobis matrix of the data distribution in the original space.

IV. Incorporating Privileged Information in Classifiers

We propose two approaches for incorporation of the learnt metric $d_{\boldsymbol{C}}$ into a classifier operating on X. The first approach linearly transforms data in the original space X so that the distance information from the privileged space X^* is 'preserved'. The classifier is then trained on the transformed points. In the second approach,specially designed for the GMLVQ classification, the new metric $d_{\boldsymbol{C}}$ is used for only retraining the prototype positions in X, given that the metric tensor on X has changed. This is achieved by running GMLVQ with $d_{\boldsymbol{C}}$ fixed.

A. Transformed Basis (TB)

Recall that $d_{\boldsymbol{C}}$ is found in the parametrized form $\boldsymbol{C}=\boldsymbol{U}^T\boldsymbol{U}$. Then for any $x\in X$, we have

$$\|x\|^2_{\boldsymbol{C}}=x^T\boldsymbol{C}x=x^T\boldsymbol{U}^T\boldsymbol{U}x=\tilde{x}^T\tilde{x}=\|\tilde{x}\|_2^2,$$

where $\tilde{x}=\boldsymbol{U}x$ is the image of x under the basis transformation $\boldsymbol{U}$. The layout of the transformed points $\tilde{x}_i=\boldsymbol{U}x_i$ now reflects the 'similarity/dis-similarity' information from X^*. Data points with 'similar' privileged data representation will now in general be closer than in the original data layout. Likewise, data points with more distant privileged representations will tend to move further apart. The classification algorithm (e.g. GMLVQ in its original form) is now applied to the transformed data $\{(\tilde{x}_1,y_1),....,(\tilde{x}_n,y_n)\}$. We stress that the TB approach is flexible and, unlike SVM+, allows for application of *any* suitable metric-based classifier , e.g. k-nearest neighbors (k-NN).

B. Extended Model (Ext)

Unlike the TB approach, this methodology is specially designed to incorporate the privileged-information-induced learned metric $\boldsymbol{C}$ in the GMLVQ algorithm. First, GMLVQ is run on the training set $(x_i,y_i)\in\mathbb{R}^m\times\{1,...,c\}$, $i=1,2,...,n$, yielding a global metric $d_{\boldsymbol{M}}$ (given by metric tensor $\boldsymbol{M}$) and a set of prototypes $w_j\in\mathbb{R}^m$, $j=1,2,...,L$. Then, one of the two techniques of section III is used to find metric $d_{\boldsymbol{C}}$ on X that will replace the metric $d_{\boldsymbol{M}}$ originally found by GMLVQ. Hence, the Ext in GMLVQ squared metric will have the form

$$d_{\boldsymbol{C}}(w,x)=(x-w)^T\boldsymbol{C}(x-w).$$

The metric $d_{\boldsymbol{C}}$ incorporates the privileged information. Finally, GMLVQ is run once more with metric tensor $\boldsymbol{C}$ fixed to modify the prototype positions[9].

V. Computational Complexity Analysis

Our methodology incorporates three main steps:

1) metric learning in the original space X via Metric Fusion (MF) or Information Theoretic approach (IT),
2) incorporation of the learned metric in the underlying classifier - Transformed Basis (TB) or Extended Model (Ext),
3) forming the resulting classifier.

We study the computational complexity of each by each phase separately.

1) Analytical computation of the unconstrained matrix $\boldsymbol{C}$ in MF by solving the quadratic problem $I(\boldsymbol{C})$ (Eq. (7)) costs $O(n^2+m^2)$, where n is the number of training examples and m is the data dimensionality. This is also the cost of each iteration of gradient descent in Eq. (11). Learning matrix $\boldsymbol{C}$ in IT costs $O(m^2)$ per projection (Eq. 14). Each iteration of IT costs $O(s\cdot m^2)$, where s is the number of pairwise constrains ($s=|S_+\cup S_-|$) [29].
2) TB linearly transforms each data point (cost $O(n)$). The complexity of the closest correct and incorrect prototypes' adaptation in each step of Ext costs $O(m^2\cdot N_w)$, where N_w is the number of updated prototypes [4].
3) In the TB case, the complexity depends on the classifier used. For example, The original GMLVQ costs $O(m^2)$ for matrix adaptation in each adaptation step together with $O(m^2\cdot N_w)$ for the closest correct and incorrect prototypes adaptation in each adaptation step [4]. In the case of Ext, the cost per adaptation steps is $O(m^2\cdot N_w)$.

VI. Experiments and Evaluations

The effectiveness of the proposed methodology, integrating privileged information in learning, was evaluated in the context of classification accuracy obtained against

[9] The prototype positions will in general change, since the metric has been changed from $d_{\boldsymbol{M}}$ to $d_{\boldsymbol{C}}$.

the state of art algorithms GMLVQ, used in the original space. In addition, since the privileged information is used to manipulate metric in the original input space X, we also employed simple k-Nearest Neighbor (k-NN) metric based classifier operating in the modified metric[10]. The two proposed metric learning methodologies, metric fusion (MF, section III-A) and information theoretic approach (IT, section III-B) were assessed in three experiments.

In all experiments, (hyper-)parameters of metric learning and classification algorithms were tunned via 5-fold cross-validation on the training set. In MF approach, parameter γ was tuned over the values $0.2, 0.3, ..., 1$. In both classification scenarios (GMLVQ and k-NN), the metric tensor $\boldsymbol{M}^*$ in X^* was set to the precision matrix[11] of the privileged training points $x_1^*, x_2^*, ..., x_r^*$ (Mahalanobis distance in X^*). The same applies to the initial metric tensor $\boldsymbol{M}$ in the original space X. In IT[12] approach, lower and upper bounds for the privileged and original spaces were chosen over the values of $\{2, 3, 5, 7, 10\}$ for (a, a^*) and of $\{80, 85, 90, 95\}$ for (b, b^*). Furthermore, the slack parameter ν was tuned over the values $\{0.01, 0.1, 1\}$.

For GMLVQ, the number of prototypes per class was tunned over the set $\{1, 2, 3, 4, 5\}$. The class prototypes were initialized as means of random subsets of training samples selected from the corresponding class. Relevance matrices were normalized after each training step to $\sum_i \boldsymbol{\Lambda}_{ii} = 1$ (see section II-C). Initial learning rates for prototypes ϵ_w and relevance metric $\epsilon_{\boldsymbol{\Omega}}$ were chosen through cross-validation[13]. They decrease monotonically with training epoch index e [30]:

$$\epsilon_g(e) = \frac{\epsilon_g}{1 + \tau(e-1)}, \quad g \in \{\boldsymbol{\Omega}, w\},$$

with $\tau > 0$ set to 10^{-5}. For the k-NN classification algorithm, k was cross validated over the range 1...8[14].

The 'optimal' metric tensor $\boldsymbol{U}$ in X, resulting from the above metric learning algorithms, is then incorporated in the GMLVQ classification process via one of the two scenarios: transformed basis (TB, section IV-A) and extended model (Ext, section IV-B). Note that when using k-NN only TB approach is applicable. We summarize the models constructed within our framework in Table I. The models are build along two degrees of freedom, namely metric learning and incorporation of the learnt metric.

TABLE I
SUMMARY OF MODELS CONSTRUCTED WITHIN OUR FRAMEWORK.

Metric Modification	**Metric incorporation**	
	Transformed Basis (TB)	Extended Model (Ext)
Metric Fusion (MF)	MF-TB	MF-Ext
Information Thoeretic (IT)	IT-TB	IT-Ext

TABLE II
CROSS-VALIDATED VALUES OF (HYPER-)PARAMETERS FOR THE IRIS, PIMA, AND ABALONE DATA SETS OBTAINED FOR GMLVQ AND K-NN CLASSIFICATIONS.

Algorithm	**Hyper-parameter**	**Iris**	**Pima**	**Abalone**
GMLVQ	Prototypes per class	1	3	1
	(a^*, b^*, a, b)	(10,90,5,95)	(5,90,5,90)	(2,85,5,90)
	ν	1	0.01	1
	γ	0.7	0.2	0.2
k-NN	k	3	4	4
	(a^*, b^*, a, b)	(10,90,5,95)	(5,90,5,90)	(5,90,5,90)
	ν	1	0.01	1
	γ	0.7	0.2	0.2

A. Initial Controlled Experiments

In this section we report on experiments performed using three classification datasets from the UCI database [31], namely Iris, Pima, and Abalone sets. Here we have a control over what features constitute the 'original' and 'privileged' spaces X and X^*, respectively. In order to demonstrate the potential of methods able to incorporate the privileged information, we used the least informative features (from the point of view of classification) as the original features, the rest as the privileged ones. We also studied the effect of downsizing the amount of privileged information in the training set.

1) Data Sets: The Iris data set contains 150 items, has four input features and three classes. The 8-dimensional Pima data set contains 768 data items classified into two classes. Finally, the 8-dimensional Abalone data set has 4177 data items classified into three classes.

As mentioned above, in order to create the experimental testbed, input features were first categorized into 'privileged' and 'original'. This categorization is driven by feature relevance for the underlying classification. Diagonal elements in the GMLVQ relevance matrix effectively order the input features with respect to their relevance for classification (higher value means higher relevance). For each data set, we first ran the GMLVQ algorithm on the training set[15] and then took the lower half of input features as the 'original' ones, the second half as the privileged features.

2) Experimental Settings and Results: Cross-validated values of (hyper-)parameters of the studied methods are presented Table II. We randomly selected 75% of data items of each class for training and use the remaining data for testing. Mean misclassification rates (± Std. dev) are reported across 10 runs (10 random re-samplings of the training/test sets). Table III presents results for the case where each training point has both original and privileged information. Our findings confirm that all our

[10]We are thankful to the anonymous reviewer for this suggestion.

[11]The inverse of the covariance matrix.

[12]We modified the ITML Matlab code available from http://www.cs.utexas.edu/users/pjain/itml/. The parameters were tuned via cross-validation.

[13]We imposed $\epsilon_w > \epsilon_{\boldsymbol{\Omega}}$, implying slower rate of changes to the metric, when compared with prototype modification.

[14]larger values of k did not bring performance improvements

[15]random selection of 75% points from the original data set

TABLE III
MEAN MISCLASSIFICATION RATES FOR GMLVQ AND K-NN CLASSIFICATIONS, ALONG WITH STANDARD DEVIATIONS (±) ACROSS 10 TRAINING/TEST RE-SAMPLING, OBTAINED ON IRIS, PIMA, AND ABALONE DATA SETS. EACH TRAINING POINT HAS BOTH THE ORIGINAL AND PRIVILEGED INFORMATION. THE BEST RESULTS ARE MARKED WITH BOLD FONT.

Algorithm	**Metric learning**	**Iris**	**Pima**	**Abalone**
GMLVQ	N/A	0.22 ±(0.05)	0.35 ±(0.01)	0.45 ±(0.009)
	IT-TB	**0.16 ±(0.03)**	**0.30 ±(0.007)**	**0.42 ±(0.01)**
	IT-Ext	0.17 ±(0.03)	**0.30 ±(0.006)**	0.43 ±(0.01)
	MF-TB	0.18 ±(0.02)	0.33 ±(0.01)	0.43 ±(0.05)
	MF-Ext	0.18 ±(0.1)	0.31 ±(0.008)	0.44 ±(0.01)
k-NN	N/A	0.45 ±(0.02)	0.37 ±(0.05)	0.50 ±(0.02)
	IT-TB	0.39 ±(0.03)	0.35 ±(0.04)	0.48 ±(0.01)
	MF-TB	0.41 ±(0.01)	0.35 ±(0.02)	0.47 ±(0.02)

metric learning methods are able to successfully incorporate privileged information during the classifier building stage, even though in the test phase (reported results) the privileged information is not available. For the GMLVQ classification, the IT approach achieves the best overall performance for both metric incorporation methods (TB and Ext). On average, it outperforms (relatively) the baseline GMLVQ (trained on X only) by 25%, 14%, and 5% on Iris, Pima, and Abalone data sets, respectively. For the k-NN classification, on average (across the three data sets) the IT-TB and MF-TB outperformed (relatively) the baseline k-NN (trained on X only) with 7% and 6% , respectively. Compared with k-NN, GMLVQ is more successful because it not only incorporates the privileged information in terms of learnt metric on X, but also repositions the class prototypes 'optimally' with respect to the modified metric.

3) Studying the Effect of Downsizing Privileged Information in Space X^*: Obtaining privileged data may be costly. Therefore it is quite natural to expect that in real applications the number of data items in X^* will be relatively small, compared to the number of available data in X. Thus, in the next experiment (conducted using the GMLVQ in Transformed Basis scenario only (best performing)) we removed privileged information for randomly chosen 40% of the training points. Results are reported in Table IV. Naturally, the performance levels of GMLVQ algorithm decrease - the performance of IT-TB and MF-TB relatively decreased by 10% and 6% (in the three data sets), respectively. The IT-TB still retains the best performance. We found (not reported here) that GMLVQ based methods were more robust to reducing the privileged information than the k-NN ones, with k-NN performance deteriorating rapidly as the amount of privileged information was reduced.

TABLE IV
MEAN MISCLASSIFICATION RATES FOR GMLVQ CLASSIFICATION (USING THE TRANSFORMED BASIS SCENARIO ONLY), ALONG WITH STANDARD DEVIATIONS (±) ACROSS 10 TRAINING/TEST RE-SAMPLING, OBTAINED ON IRIS, PIMA, AND ABALONE DATA SETS. ONLY 60% OF TRAINING POINTS HAVE PRIVILEGED INFORMATION. THE BEST RESULTS ARE MARKED WITH BOLD FONT.

Algorithm	**Iris**	**Pima**	**Abalone**
GMLVQ	0.22±(0.05)	0.35±(0.02)	0.45±(0.009)
IT-TB + GMLVQ	**0.201±(0.03)**	**0.34±(0.01)**	**0.43±(0.01)**
MF-TB + GMLVQ	0.204±(0.2)	0.35±(0.01)	0.45±(0.03)

B. Comparison with SVM and SVM+

In this section we compare the approaches developed here with the recently introduced SVM-based technique for incorporation of privileged information [1] (see section II-A). We use one of the three scenarios of incorporating privileged information addressed in [1], namely, the privileged information as a holistic description. Images of digits (original space) are enhanced with poetic image description (represented as privileged information). We followed the same experimental settings used by [1].

1) Data Set: This experiment uses the MNIST hand writing database[16]. It consists of 60,000 training examples and 10,000 test samples, each of which is a 28×28 pixel gray scale image. As in [1], we used the subset of the MNIST data set corresponding to digits '5' and '8' with the images rescaled to 10×10 pixels. Training inputs (in space X) consist of the first 50 samples of digits '5' and '8' from the MNIST training data (making 100 training points). Testing data has 1,866 samples of digits '5' and '8' from the MNIST test data. Poetic descriptions describing images, with the help of language experts, were designed and used by [1] as privileged information. Poetic descriptions were translated by experts into 21-dimensional feature vectors [17] and considered as the privileged data (in space X^*). As in [1], we used training sets of increasing size 40, 50, ..., 90 (each training set containing the same number of digits '5' and '8'). We selected 12 different random samples from each training data set and we reported the average of test errors.

2) Experimental Settings and Results : Cross-validated values of (hyper-)parameters of the studied methods are presented Table V. Results are shown in Figure 1. As in the previous experiment, GMLVQ with incorporated privileged information outperforms the standard GMLVQ. Analogously for the k-NN classifier, even though the k-NN results are again inferior to the GMLVQ ones. The best performing algorithm (IT-TB in GMLVQ) was compared against the existing SVM+ based models (see Figure 2).

[16]The MNIST dataset can be downloaded from http://yann.lecun.com/exdb/mnist/

[17] The reader is referred to http://www.nec-labs.com/research/machine/ml_we a detailed description of the dataset exists.

TABLE V
CROSS-VALIDATED VALUES OF (HYPER-)PARAMETERS FOR THE MNIST DATA SET (IMAGES '5' AND '8') OBTAINED FOR GMLVQ AND K-NN CLASSIFICATIONS.

GMLVQ	Prototypes per class	(a^*, b^*, a, b)	ν	γ
	1	(5,80,5,95)	0.01	0.5
k-NN	k	(a^*, b^*, a, b)	ν	γ
	4	(5,80,5,95)	0.01	0.2

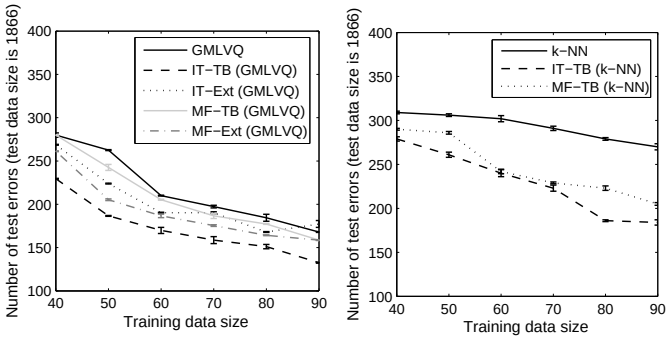

Fig. 1. Number of misclassified points obtained by GMLVQ (left figure) and k-NN (right figure) classifications (error bars report standard deviation across 12 training re-sampling) conducted on the MNIST data set (images '5' and '8').

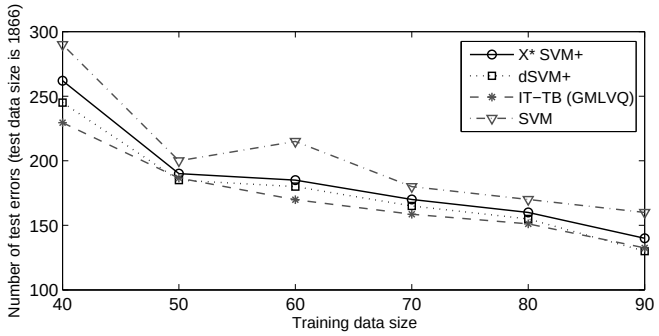

Fig. 2. Number of misclassified points obtained by the IT-TB in GMLVQ and the previously introduced SVM+ based models for LUPI conducted on the MNIST data set (images '5' and '8').

In particular, IT-TB in GMLVQ achieves relative performance improvement of 14%, 6%, and 2% over the SVM, X^*SVM+, and dSVM+, respectively.

C. Galaxy Morphological Classification using Full Spectra as Privileged Information

Morphological galaxy classification aims to classify galaxies based on their structure and appearance. It is the first step towards a greater understanding of the origin and formation process of galaxies, as well as the evolution processes of the Universe [32], [33]. The most common classification scheme is galaxy separation into three classes - *Elliptical*, *Spiral*, and *Irregular*. There have been several approaches to Galaxy morphology classification, e.g. [34], [35], [36]. Most of these approaches rely heavily on the galaxy photometric data, ignoring spectroscopic information. Huge amount of information about the physical properties of galaxies comes from their electromagnetic spectrum [37]. It is therefore of paramount importance to be able to consider detailed spectral data when training galaxy classifiers. However, obtaining a full spectrum is much more costly than measuring coarse spectral features and basic morphological characteristics. Nevertheless, for many galaxies full spectra have been measured and should not be discounted, even though for a new galaxy to be classified we may not have the privilege to have such an information. This is exactly the arena of learning with privileged information - construct a classifier using both basic and advanced (more costly) spectral information, while in the 'test' phase the classifier will take as inputs only the basic ('original') features.

1) Data Set: Sample of galaxy identifications numbers (IDs) was extracted from Galaxy Zoo project catalogs [38], [39]. The Galaxy Zoo project launched in 2007 has provided visual morphological classifications for around one million galaxies, extracted from the Sloan Digital Sky Survey (SDSS) (data release 7) [40]. Astronomers and general public experts were invited to visually inspect and classify these galaxies via the main analysis page from the Galaxy Zoo website[18]. The project had obtained a huge number of classifications made by 100,000 participants. From the Galaxy Zoo catalog we extracted galaxy objects that had more than 50 votes with 95% agreement among the votes. The galaxy IDs were then used to extract features characteraizing the galaxies in the original (bulk measurement) space X, as well as, if available, in the privileged space X^* of full spectra.

Basic Imaging Features (X): It was shown by [41] that imaging parameters associated with colors, profile-fitting, adaptive shape, concentration and texture, are useful in separating the galaxy objects into the basic three morphological classes. For each galaxy, we extracted 9 essential imaging parameters defined in [41] from the SDSS DR7 data catalogues[19]. After detailed discussions with astronomers, we added 4 additional basic features (e.g. coarse spectral measures).

Detailed Spectral Features (X^*): Input spectra parameters for the extracted galaxy objects were obtained from the MPA-JHU DR7 release of spectrum measurements[20]. Originally, there were 138 spectral features. Based on consultations with astronomers, we downsized the amount of features to 40. Out of these we selected only the most relevant ones (for the purposes of classification) using diagonal elements in the relevance matrix provided by GMLVQ. There were 8 spectral features showing high significance for galaxy classification that were confirmed as highly important by astronomers.

Overall, our dataset contained 20,000 galaxies characterized by 13 'original' features (in X) and 8 'privileged ' spectral features (in X^*). On the set of this size, we found it infeasible to run extensive sets of experiments using the SVM+ based approaches.

[18]http://data.galaxyzoo.org/
[19]http://cas.sdss.org/astro/en/tools/crossid/upload.asp
[20]http://www.mpa-garching.mpg.de/SDSS/DR7/

TABLE VI
CROSS-VALIDATED VALUES OF (HYPER-)PARAMETERS FOR THE GALAXY DATA SET OBTAINED FOR GMLVQ AND K-NN CLASSIFICATIONS.

GMLVQ	Prototypes per class	(a^*, b^*, a, b)	ν	γ
	(20,10,5)	(3,90,5,90)	0.1	1
k-NN	k	(a^*, b^*, a, b)	ν	γ
	6	(3,90,5,90)	0.1	0.8

TABLE VII
MEAN MISCLASSIFICATION RATES, ALONG WITH STANDARD DEVIATIONS (±) ACROSS 10 TRAINING/TEST RE-SAMPLING, FOR THE GALAXY MORPHOLOGICAL CLASSIFICATION. THE BEST RESULTS ARE MARKED WITH BOLD FONT.

Algorithm	Metric learning	Misclassification
GMLVQ	N/A	0.023±(0.001)
	IT-TB	**0.019±(0.001)**
	IT-Ext	0.020±(0.002)
	MF-TB	0.020±(0.001)
	MF-Ext	0.020±(0.003)
k-NN	N/A	0.025±(0.004)
	IT-TB	0.022±(0.003)
	MF-TB	0.023±(0.004)

2) Experimental Setting and Results: On the set of 20,000 galaxies, we conducted 10 experimental runs, in each run the galaxy set was randomly split into training set (75%) and test set (25%). Mean misclassification rates (± Std. dev) are reported across 10 runs (10 random re-samplings of the training/test sets).

Cross-validated values of (hyper-)parameters[21] of the studied methods are presented Table VI. In general, using the spectral privileged information in the model building phase enhances the classification accuracy, even though in the test phase the models are fed with the original 'coarse' features only. For the GMLVQ classification, the average relative improvement (in both metric incorporation scenarios (TB and Ext)) in the classification accuracy over the GMLVQ baseline is 15% and 13% for IT and MF, respectively. It is interesting that in this case even the k-NN base classifier works well. As expected, the inclusion of full spectral information improves its accuracy (e.g. IT-TB in k-NN). However, the best (and most stable) results are obtained by the IT-TB method in GMLVQ.

3) Studying the Effect of Downsizing Privileged Information in Space X^*: Extracting galaxy spectral parameters is complex and expensive task. SDSS has photometric data for around fifty million galaxies [40]. However, the spectroscopic features are available for only relatively few galaxy objects. We quantified deterioration of the classification accuracy with decreasing number of galaxies having privileged spectral information. The above experiment (conducted for the GMLVQ formulations in IT-TB and MF-TB scenario only) was repeated with 5000, 10,000 and 15,000 galaxy objects (randomly selected over 10 runs) having the privileged information. The results are shown in Figure 3. As in the case of UCI datasets (section VI-A2) the IT model is more robust to limited amounts of privileged information in the training data.

[21]Due to large data set size and imbalanced nature of the 3 classes, we allowed for larger and different number of prototypes in each class.

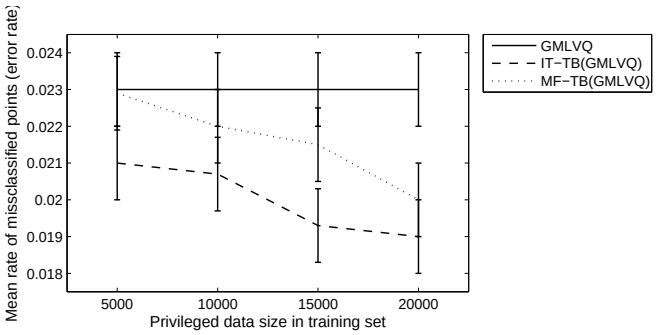

Fig. 3. Mean misclassification rates (error bars report standard deviation across 10 training/test re-sampling) obtained using varying amounts of privileged information.

VII. DISCUSSION

The principal difference between the IT and MF approaches is in the way the distance information in the privileged space X^* is treated. While the MF approach emphasize the exact values of the distances, the IT approach works on a qualitative level only (similar/dissimilar representations in X^*). This makes the IT framework more robust to deficiencies in the privileged information. Treating distance information in X^* as qualitative only (similar/dis-similar) instead of paying full attention to precise distances can be beneficial when the link between the original features and the privileged information is loose, e.g. poetic descriptions of images of digits (section VI-B). Figure 1 clearly demonstrates superiority of IT-TB over MF-TB. Note that If the privileged information is less credible (e.g. contaminated with noise, or of subjective character as in the digits experiment), the model can reduce its influence in the model building phase via the regularization parameters γ and ν in the (MF and IT) formulations, respectively.

In the GMLVQ classifications, the overall performance of the two metric incorporation scenarios considered in this study - Transformed Basis (TB) and Extended Model (Ext) - is comparable, with TB being slightly better most of the time. In the Ext approach, the prototypes get retrained one more time using GMLVQ, given the modified metric tensor in X. If we continued updating both prototypes and metric tensor on X further (as in GMLVQ), all information from the privileged space X^* would get eventually lost. On the hand, in the TB scenario the privileged information is 'permanently' coded in X by changing the distribution of points in X on the basis of distance relations in X^*. The subsequent runs of GMLVQ operate on this new layout of training points in X with the privileged information contribution not lost during further training.

Finally, we remark that we also tried to impose on X^* the metric obtained by running GMLVQ on the privileged data only, but this did not (at least for the data sets used here) improve (compared to using precision matrix (Mahalanobis distance) on X^*) the classification performance. The same applies to initialization of metric tensor in the original input space X. Using metric tensor obtained from GMLVQ on X was not preferable to simple initialization of the metric tensor with the precision matrix in X.

VIII. Conclusion

We have introduced a novel framework for learning with privileged information through metric learning. The framework can be naturally cast in prototype based classification with metric adaptation (GMLVQ). The privileged information is incorporated into the model operating on the original space X by changing the global metric in X, based on distance relations revealed by the privileged information in X^*. Unlike in the existing SVM-based approaches for learning with privileged information, the privileged information is used to manipulate the input space or its metric and thus any classifier (e.g. simple k-NN) can be subsequently used. This provides more flexibility for the task of incorporating privileged information during the training. Moreover, prototype based approaches have the additional advantages of providing more interpretable models and natural formulation of multi-class classifiers.

We verified our framework in three experimental settings: **(1)** controlled experiments using three data sets from UCI repository, **(2)** handwritten digit recognition using poetic descriptions as privileged information [1] and **(3)** a real world application of great practical and theoretical importance in astronomy - galaxy morphological classification. Here, the privileged information takes the form of costly-to-obtain full galaxy spectra.

Acknowledgments

This work was supported by a grant from the Biotechnology and Biological Sciences Research Council [H012508/1].

References

[1] V. Vapnik and A. Vashist, "A new learning paradigm: Learning using privileged information," *Neural Networks*, vol. 22, pp. 544–557, 2009.

[2] V. Vapnik and S. Kotz, *Estimation of Dependences Based on Empirical Data: Empirical Inference Science (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[3] J. Cervantes, X. Li, and W. Yu, "Multi-class svm for large data sets considering models of classes distribution," in *International Conference on Data Mining*, 2008, pp. 30–35.

[4] P. Schneider, M. Biehl, and B. Hammer, "Adaptive relevance matrices in learning vector quantization," *Neural Computation*, vol. 21, no. 12, pp. 3532–3561, 2009.

[5] P. Schneider, "Advanced methods for prototype-based classification," PhD Dissertation, University of Groningen, 2010.

[6] J. Cervantes, X. Li, W. Yu, and K. Li, "Support vector machine classification for large data sets via minimum enclosing ball clustering," *Neural Networks: Algorithms and Applications, 4th International Symposium on Neural Networks*, vol. 71, no. 4-6, pp. 611–619, 2008.

[7] J. Feyereisl and U. Aickelin, "Privileged information for data clustering," *Information Sciences*, vol. 194, pp. 4–23, Jul. 2012.

[8] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning (ML)*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[9] T. Kohonen, *The Handbook of Brain Theory and Neural Networks, 2nd Edition*. MIT Press, 2003.

[10] ——, "Learning vector quantization for pattern recognition," Laboratory of Computer and Information Science, Department of Technical Physics, Helsinki University of Technology, Espoo, Finland, Technical Report TKK-F-A601, 1986.

[11] A. Sato and K. Yamada, "Generalized learning vector quantization," *Advances in Neural Information Processing Systems*, vol. 7, pp. 423–429, 1995.

[12] B. Hammer and T. Villmann, "Generalized relevance learning vector quantization," *Neural Networks*, vol. 15, no. 8-9, pp. 1059–1068, Oct. 2002.

[13] B. Hammer, M. Strickert, and T. Villmann, "Relevance lvq versus svm," in *Artificial Intelligence and Soft Computing, Volume 3070 Of Springer Lecture Notes in Artificial Intelligence*. Springer, 2004, pp. 592–597.

[14] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information Processing Systems 17*. MIT Press, 2004, pp. 513–520.

[15] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *The Journal of Machine Learning Research*, vol. 10, pp. 207–244, Jun. 2009.

[16] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proceedings of the 24th international conference on Machine learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 209–216.

[17] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *In Proceedings of the Twentieth International Conference on Machine Learning*, 2003, pp. 11–18.

[18] S. C. Hoi, W. Liu, M. R. Lyu, and W. Ma, "Learning distance metrics with contextual constraints for image retrieval," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 2072–2078.

[19] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell, "Distance metric learning with application to clustering with side-information," in *Neural Information Processing Systems*, vol. 15. MIT Press, 2002, pp. 505–512.

[20] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning a mahalanobis metric from equivalence constraints," *Journal of Machine Learning Research*, vol. 6, pp. 937–965, December 2005.

[21] L. Yang and A. R. Jin, "Distance metric learning: A comprehensive survey," Michigan State University, Technical Report, 2006.

[22] R. Jin, S. Wang, and Y. Zhou, "Regularized distance metric learning: Theory and algorithm." in *NIPS*. Curran Associates, Inc., 2009, pp. 862–870.

[23] W. Bian and D. Tao, "Learning a distance metric by empirical loss minimization," in *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Two*, ser. IJCAI'11. AAAI Press, 2011, pp. 1186–1191.

[24] ——, "Constrained empirical risk minimization framework for distance metric learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1194–1205, 2012.

[25] G. J. McLachlan, *Discriminant analysis and statistical pattern recognition*, ser. Wiley series in probability and mathematical statistics. J. Wiley and sons, 1992.

[26] D. Tao, X. Li, X. Wu, and S. J. Maybank, "Geometric mean for subspace selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 260–274, 2009.

[27] W. Bian and D. Tao, "Max-min distance analysis by using sequential sdp relaxation for dimension reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 1037–1050, 2011.

[28] N. J. Higham, "Matrix nearness problems and applications," in *Applications of Matrix Theory*, M. J. C. Gover and S. Barnett, Eds. University of Manchester, University Press, 1989, pp. 1–27.

[29] B. Kulis, P. Jain, and K. Grauman, "Fast similarity search for learned metrics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2143–2157, 2009.

[30] C. Darken, J. Chang, J. C. Z, and J. Moody, "Learning rate schedules for faster stochastic gradient search," in *Neural Networks for Signal Processing 2 - Proceedings of the 1992 IEEE Workshop*. IEEE Press, 1992.

[31] A. Frank and A. Asuncion, "UCI machine learning repository," 2010.

[32] C. Elting, C. A. L. Bailer-Jones, and K. W. Smith, "Photometric classification of stars, galaxies and quasars in the sloan digital sky survey dr6 using support vector machines," *Proceedings of the International Conference:Classification and Discovery in Large Astronomical Surveys. AIP Conference Proceedings*, vol. 1082, pp. 9–14, 2008.

[33] N. M. Ball and R. J. Brunner, "Data mining and machine learning in astronomy," *Instrumentation and Methods for Astrophysics in the International Journal of Modern Physics*, vol. 91, pp. 1049–1106, 2010.

[34] D. B. Wijesinghe, A. M. Hopkins, B. C. Kelly, N. Welikala, and A. J. Connolly, "Morphological classification of galaxies and its relation to physical properties," *Monthly Notices of the Royal Astronomical Society*, vol. 404, no. 4, pp. 2077–2086, 2010.

[35] J. de la Calleja and O. Fuentes, "Machine learning and image analysis for morphological galaxy classification," *Monthly Notices of the Royal Astronomical Society*, vol. 349, pp. 87–93, 2004.

[36] S. Kasivajhula, N. Raghavan, and H. Shah, "Morphological galaxy classification using machine learning," *Monthly Notices of the Royal Astronomical Society*, vol. 8, pp. 1–8, 2007.

[37] C. Reichardt, R. Jimenez, and A. Heavens, "Recovering physical parameters from galaxy spectra using moped," *Monthly Notices of the Royal Astronomical Society*, vol. 327, no. 3, pp. 849–867, 2001.

[38] C. J. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, M. J. Raddick, R. C. Nichol, A. Szalay, D. Andreescu, P. Murray, and J. Berg, "Galaxy zoo: morphologies derived from visual inspection of galaxies from the sloan digital sky survey," *Monthly Notices of the Royal Astronomical Society*, vol. 389, 2008.

[39] C. J. Lintott, K. Schawinski, S. Bamford, A. Slosar, K. Land, D. Thomas, E. Edmondson, K. Masters, R. Nichol, J. Raddick, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg, "Galaxy zoo 1 : Data release of morphological classifications for nearly 900,000 galaxies," *Monthly Notices of the Royal Astronomical Society*, pp. 1–14, 2010.

[40] K. Abazajian, "The seventh data release of the sloan digital sky survey," *The Seventh Data Release of the Salon Digital Sky Survey Journal reference The Astrophysical Journal Supplement*, vol. 82, 2009.

[41] M. Banerji, O. Lahav, C. J. Lintott, F. B. Abdalla, K. Schawinski, S. P. Bamford, D. Andreescu, P. Murray, M. J. Raddick, A. Slosar, A. Szalay, D. Thomas, and J. Vandenberg, "Galaxy zoo: reproducing galaxy morphologies via machine learning," *Monthly Notices of the Royal Astronomical Society*, vol. 406, no. 1, pp. 342–353, 2010.