Classifier Learning with Hidden Information

Ziheng Wang Qiang Ji
ECSE, Rensselaer Polytechnic Institute, Troy, NY
wangz10@rpi.edu jiq@rpi.edu

Abstract

Traditional data-driven classifier learning approaches become limited when the training data is inadequate either in quantity or quality. To address this issue, in this paper we propose to combine hidden information and data to enhance classifier learning. Hidden information represents information that is only available during training but not available during testing. It often exists in many applications yet has not been thoroughly exploited, and existing methods to utilize hidden information are still limited. To this end, we propose two general approaches to exploit different types of hidden information to improve different classifiers. We also extend the proposed methods to deal with incomplete hidden information. Experimental results on different applications demonstrate the effectiveness of the proposed methods for exploiting hidden information and their superior performance to existing methods.

1. Introduction

A wide variety of computer vision problems can be formulated as a classification problem. Over the past decades, classifier learning methods have been mostly data-driven. The classifier is learned purely from a set of training instances $(x_1, y_1), \cdots, (x_n, y_n)$. Despite the substantial successes they have achieved for solving classification problems, data-driven approaches become very brittle and prone to overfitting when the training data is inadequate in either quantity or quality, which is unfortunately often the case in many real-world applications.

A natural solution to alleviate the limitations of data-driven approaches is incorporating additional prior information. In particular, there often exists a type of information which is available during training but not available during testing. It can be qualities, properties and context of the training instances, and can be found in a wide variety of applications. For example, in object recognition, besides the image features and object labels, during training, the learner may also have access to object attributes which describe high-level properties of the objects in each image. In human action recognition, besides the RGB video features and human action labels, the learner may also obtain depth information and human joint positions about each human action instance. Both object attributes and joint positions could be obtained offline for training data yet are very expensive to obtain for all the testing instances. The question is if we can effectively exploit such information that is available only during training to help improve the classification performance. In this paper we denote such information as *hidden information* and study how to combine hidden information and data to learn a better classifier. We call the new learning problem *learning with hidden information*.

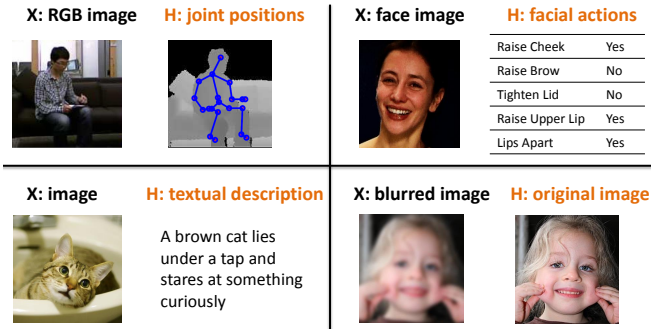

Figure 1: Examples of hidden information in different applications. X stands for primary measurement, H stands for hidden information.

With hidden information, we expect that a better classifier can be learned during training. The idea of learning with hidden information is also appealing because of its resemblance to human learning. A student may get various learning aids such as computer softwares in school. Yet they have to solve problems on their own without these learning aids latter on. These learning aids help improve students' learning.

However, learning with hidden information is challenging since hidden information is not available during testing and cannot be combined together with input features to predict the class label. Therefore hidden information has to be automatically and effectively encoded into the structure or parameters of the classifier during training. In this paper we focus on using hidden information to obtain better estimate of classifier parameters.

Learning with hidden information was originally proposed by Vapnik *et al*. [1]. Since then, it has been further explored by other researchers, who may refer hidden information as privileged information or side information. However, research in this area remains limited. First, existing approaches are all designed to exploit certain type of hidden information for certain type of classifiers. Second, they are generally based on strong and even unrealistic assumptions. Third, existing methods typically assume hidden information is complete for each training sample. However, for many real world applications, hidden information may only be available for a fraction of training data. Finally, existing methods typically treat each piece of hidden information independent of each other, ignoring their relationships.

To address these limitations, in this paper we systematically tackle the problem of learning with hidden information with two general approaches. First, besides hidden information represented as additional features, in this work we also study hidden information that is represented as additional targets or labels. Second, while most of the existing methods are specifically designed for SVM, the proposed methods are general and can be applied to different classifiers with different loss functions without strong assumptions. Finally, we also extend the proposed methods to deal with incomplete hidden information.

The remainder of this paper is organized as follows. A literature review is given in Section 2. After that we formally define the problem in Section 3 and introduce the proposed methods in Section 4. Experimental results are shown in Section 5 and the paper is concluded in Section 6.

2. Related Work

Hidden information, also referred to as privileged information [1] or side information [2], has been exploited to enhance different learning tasks such as classifier learning [1, 3, 2, 4, 5], feature learning [6], clustering [7], and metric learning [8]. Here we mainly review related work on classifier learning.

All existing work assume hidden information comes as additional features and is used to improve specific classifiers. The earliest approach is SVM+ proposed in [1]. It is based on very strong assumption that the slack variable (or upper bound of the loss function [9]) can be modeled as an unknown "correcting function" of hidden information. In other words, the upper bound of the loss for each training sample can be inferred from the corresponding hidden information through a function. The correcting function is learned simultaneously with the primary classifier and hence SVM+ is computationally more expensive than SVM. SVM+ has also been generalized to L1 regularized SVM+ [10], multi-class SVM+ [11], and multi-task multi-class SVM+ [12]. Besides, [13] showed that SVM+ can be formulated as a special case of instance weighted SVM. [14] studied the connection between SVM+ and multi-task learning. [9] investigated the generalization bound for a simplified version of SVM+ (i.e. SVM+ without any regularization terms). Although SVM+ and its variants have achieved success in some applications, the assumption that hidden information is functionally related to the slacks is too strong and hard to verify. Besides, the methods are all specifically developed for SVM and cannot generalize to other type of classifiers.

Besides SVM+, Sharmanska and Lampert proposed margin transfer SVM[1] and rank transfer SVM [3]. The basic idea is that the levels of difficulty for classifying the class label with the original feature and hidden information are the same. However, the assumption of equivalent classification margin may not hold.

Another method for learning with hidden information is AdaBoost+ proposed in [2]. Hidden information is treated as a set of side features and used to construct weak classifiers. Specifically, they first learn a set of weak classifiers from both the original features and side features. If a side feature is selected as the input to a weak classifier, then they learn a mapping function from the original feature to the side feature to predict that side feature during testing. However, learning this mapping function from original feature to the selected side feature could be more challenging than learning the original classifier. Moreover, the poorly predicted side features could adversely affect the subsequent classification.

In summary, existing methods are designed to exploits hidden information for specific type of classifiers and are generally based on strong assumptions that are hard to verify and may not even hold in practice. Besides, most of the existing approaches only focus on classifier learning with complete hidden information.

3. Problem Definition

We begin by presenting the mathematical definition of the standard learning problem and learning with hidden information. In this paper we assume binary supervised classification but the problem and proposed framework is not limited to binary case.

A **standard classification problem** is defined as follows: given n training instances $\{(x_i, y_i)|x_i \in \mathcal{X}, y_i \in \{-1, 1\}, i = 1, \cdots, n\}$ generated independently from an unknown distribution $P(x, y)$ where x is input feature, and y is output label, learn a classifier $f(x, w)$ to predict the label y from the input feature x. w is the parameter of the classifier.

A typical solution to the standard classification problem is empirical risk minimization shown in Equation 1 which learns a classifier by minimizing the empirical risk

[1] Without loss of generality, here we only discuss margin transfer.

$\sum_{i=1}^{n} \ell(y_i, x_i, w)$ and a regularization term over the parameters $\Omega(w)$ (e.g. ℓ_2 norm). λ is the regularization parameter and $\ell(y_i, x_i, w)$ is the loss function. In this paper, this is the baseline approach we want to beat by incorporating hidden information.

$$w^* = \arg\min \sum_{i=1}^{n} \ell(y_i, x_i, w) + \lambda\Omega(w) \tag{1}$$

Learning with hidden information (LHI) is formulated as follows: given n triplets $\{(x_i, h_i, y_i)|x_i \in \mathcal{X}, h_i \in \mathcal{H}, y_i \in \{-1, 1\}, i = 1, \cdots, n\}$ generated independently from an unknown distribution $P(x, h, y)$ where x is input feature, h is hidden information and y is output label, learn a classifier $f(x, w)$ to predict the label y from the input feature x. The goal is that by incorporating hidden information, we can get a better classifier, i.e., better parameter w. Throughout this paper we denote x as primary feature, y as primary target and $f(x)$ as primary classifier.

4. Approaches

Since hidden information is not available during testing, in order for it to influence the classification performance, hidden information has to be properly translated and encoded into the classifier parameters. In other words, the goal is to exploit hidden information to get a better estimate of the classifier parameter. Our basic idea is to encode hidden information as regularization terms to constrain and refine parameter estimation during training. Hidden information comes in different forms. Depending on its form, hidden information needs to be incorporated in different ways which we will discuss in the following two sections.

4.1. Hidden information as secondary features

First of all, hidden information h can be represented as an additional set of features for the target variable y in many applications. For example in computer vision, besides the images there are often corresponding text descriptions for the object. These text descriptions can be represented as a set of features and used as hidden information. We denote such hidden information as secondary features.

The primary and secondary features can be treated as features from two different views, and hence one would naturally think about using multi-view method to leverage hidden information in this case. Multi-view learning tackles the problem where features from multiple sources (or views) are available during both training and testing [15, 16, 17]. During training, it models each view with one classifier and jointly learns all classifiers by assuming all single-view classifiers produce similar outputs. This is generally achieved with a regularization term that penalizes the differences of different classifiers. During testing, it makes predictions by averaging the classification results from all different views. If we discard classifiers for other views and only keep the primary classifier for testing, it seems that multi-view learning method can be used for learning with secondary features.

However multi-view method may not be necessarily effective due to the following reasons. First, multi-view learning expects better performance by fusing different views. Hence it usually requires all views to be available during testing. However, learning with hidden information is asymmetric. Its goal is to improve the performance of the primary classifier with the help of the secondary features. Second, the assumption that different single view classifiers produce similar performances may not necessarily hold for the problem of learning with hidden information. In fact, hidden information is usually more discriminative than the primary features in many practical applications. For instance, text descriptions are usually better than the raw image pixels to classify the objects.

To address these issues we propose a regularization method for learning with secondary hidden features. Our method is motivated by the assumption that secondary feature is more informative for classification than the primary feature. While seemingly strong, this assumption holds for many applications (e.g attributes are more discriminative than image features). In addition, it can be generally satisfied if we combine x and h together as secondary features. Mathematically, the assumption means that if we have a primary classifier $f(x, w)$ to classify y from x, and a secondary classifier $f(h, \tilde{w})$ to classify y from h, then the loss for classifying y with x should be higher than the loss for classifying y with h. It can can be encoded as a set of ϵ-insensitive loss inequality constraints shown below. Here ϵ is used to account for uncertainties.

$$\ell(y_i, x_i, w) \geq \ell(y_i, h_i, \tilde{w}) - \epsilon_i, \;\; \epsilon_i \geq 0, \;\; \forall 1 \leq i \leq n$$

We propose a regularization method call **loss inequality regularization (LIR)** to leverage the ϵ-insensitive loss inequality constraints. The basic idea is to penalize the violation of these constraints, and the objective function is shown in Equation 2, where $[\cdot]_+ = \max(0, \cdot)$.

$$\sum_{i=1}^{n} \ell(y_i, x_i, w) + \eta \sum_{i=1}^{n} \ell(y_i, h_i, \tilde{w}) + \lambda\Omega(w) + \tilde{\lambda}\Omega(\tilde{w}) + \gamma \sum_{i=1}^{n} [\ell(y_i, h_i, \tilde{w}) - \ell(y_i, x_i, w)]_+ \tag{2}$$

The first two terms in the objective function are the empirical losses for the primary and secondary classifier. The next two are standard parameter based regularization terms such as ℓ_2 regularization. The last one is the proposed ϵ-insensitive loss inequality regularization term which penalizes the total violation of the loss inequality constraints for the training instances. The parameters λ, $\tilde{\lambda}$ and γ allow different degrees of trade off between empirical loss and reg-

ularization. The loss inequality regularization term serves as a bridge to relate the primary and secondary classifiers, and the parameter η determines which one will play a more important role to influence the other.

With the proposed objective function, a primary classifier with lower training loss than the secondary classifier will be penalized. Intuitively, the secondary classifier provides a reference for the learner to decide if the primary classifier overfits. With the ϵ-insensitive loss inequality regularization term, the primary hypothesis space is reduced and hence it is reasonable to expect better test performance.

Advantages of loss inequality regularization. First of all, loss inequality regularization (LIR) is more general. Margin transfer [3] relates the primary and secondary classifier through the **margin** and SVM+ relates the primary and secondary features through the slack variable. Therefore they are both limited to SVM classifiers. By contrast, LIR relates the two classifiers through the **loss function** and hence it is generally applied to different type of classifiers. Second, our assumption holds for many applications and can be generally satisfied if we combine primary and secondary features together as secondary features. This, unfortunately, is not the case for the existing methods. Margin transfer assumes that samples that are hard or easy (as called in their paper) to classify with secondary feature are also hard or easy to classify with the primary feature. This may not hold in practice. Consider the example of face recognition where low resolution image is the primary feature and high resolution image is the secondary feature. Even though it is easy to recognize face from high resolution image, it could still be very hard to recognize face from low resolution image. Besides, SVM+ assumes that hidden information is related to the margin through a correcting function, which is also strong and difficult to hold.

4.2. Hidden information as secondary targets

Besides secondary features, hidden information can also be represented as secondary targets or secondary labels. By this, we mean we can use the primary features to predict the hidden information just like using primary features to predict the original class label. For instance in image based object recognition, the object attributes can be treated as another set of labels. In this section we study how to exploit the secondary labels to improve classifier learning.

A straightforward method to exploit the secondary labels is to use them as the middle level representation. Specifically, instead of classifying the output y from x directly, one can first classify the secondary targets h from x and then use the predicted h (or together with x) to classify y. However, poor prediction of the secondary targets could confound the classification performance of the primary target. Besides, learning a large number of secondary target classifiers increase the computational cost.

Another straightforward method to exploit secondary targets is multi-task learning method. Instead of learning each task individually, multi-task learning utilizes the task relationships to learn all tasks simultaneously such that information can be shared across different tasks and hence improve their performances. The task relationships are generally characterized as model parameter relationships, which can be manually specified [18] or automatically learned from data [19, 20, 21]. If we treat each task as classifying one target (either primary target y or secondary target h^j) with x, we can directly apply multi-task learning approach to simultaneously learn all these tasks during training. Once all the classifiers are learned, we only keep the primary classifier during testing.

However, multi-task learning is not necessarily effective and efficient in our case, either. First of all, we are only interested in boosting the performance of the primary classification task instead of improving all the tasks. Second, multi-task learning significantly increases the computational cost since it requires learning more parameters of secondary classifiers. Finally, it is also very difficult to specify task parameter relationships.

To overcome these limitations we proposed another regularization method called **relationship preserving regularization**. We explicitly exploit relationships among the primary and secondary labels and our motivation is that we want predicted target labels by the learnt primary classifier to preserve their relationships with the secondary labels. Specifically, if y has a strong dependence with a secondary target h^j, then the predicted label of the primary classifier $\hat{y}$ should also have a strong dependence with h^j. On the contrary, if y has a very weak dependence with h^j, then $\hat{y}$ should also have a very weak dependence with h^j. For example, if an object is closely related to an attribute, then the predicted object to be closely related to this attribute (and vice-versa).

Mathematically, denote $R(y, [h^1, \cdots, h^k])$ as a certain mathematical measure of the relationships between the primary label y and secondary labels $h^1, \cdots, h^k$, we propose the following relationship preserving constraint, where $\hat{y}$ is the predicted label of y.

$$|R(y, [h^1, \cdots, h^k]) - R(\hat{y}, [h^1, \cdots, h^k])| < \epsilon \quad (3)$$

The relationship can be quantified with different mathematical measures. Here we measure it with pair-wise covariances $\sigma(y, h^j)$ between y and h^j, and we decompose the relationship preserving constraint into a set of pair-wise constraints as follows:

$$|\sigma(y, h^j) - \sigma(\hat{y}, h^j)| < \epsilon, 1 \leq j \leq k \quad (4)$$

The covariance between y and h^j is defined in Equation 5. Since the underlying distribution is unknown, we estimate the covariance with the training data with Equa-

tion 6, where $\bar{y}$ and $\bar{h}^j$ stand for the sample means. The covariance can be further written in the matrix form, where $Y = (y_1, \cdots, y_n)'$, and $H^j = (h_1^j, \cdots, h_n^j)'$. $M = I - \frac{1}{n}E$ is the matrix used to center the data. I is the identity matrix and E is a matrix of which all components are 1's.

$$\sigma(y, h^j) = E[(y - E(y))(h^j - E(h^j))] \quad (5)$$

$$\hat{\sigma}(y, h^j) = (MY)'(MH^j)/(n-1) \quad (6)$$

The proposed relationship preserving constraints are then encoded into a regularization term to regularize the learning of the primary classifier and the objective function as shown in Equation 7. The first term is the empirical training loss of the target classifier, the second term is a standard parameter based regularization such as $\|w\|_2^2$, and the last term is the proposed regularization term to penalize the violation of the relationship-preserving constraints. Specifically, $\hat{Y}$, Y, and H^j are three vectors consisting of the predicted target labels, the ground truth target labels, and the j^{th} ground truth secondary labels for the training instances. $(M\hat{Y})'MH^j$ measures the covariance between the predicted target label and the j^{th} ground truth secondary label. $(MY)'MH^j$ measures the covariance between the ground truth target label and the j^{th} ground truth secondary label. Since $\hat{y}$ is not a continuous function of the model parameter w, we approximate $\hat{y}$ with $f(x, w)$ during learning.

$$\sum_{i=1}^{n} \ell(y_i, x_i, w) + \lambda\Omega(w) + \frac{\gamma}{2}\sum_{j=1}^{k}\left[(M\hat{Y})'MH^j - (MY)'MH^j\right]^2 \quad (7)$$

Advantages of relationship preserving regularization. First, similar to loss inequality regularization, relationship preserving regularization (RPR) is also generally applied to different type of classifiers. Second, RPR exploits explicit label relationships instead of implicit parameter relationships in multi-task learning, which is difficult to obtain. Third, RPR does not increase complexity of the learning problem since it does not require learning any secondary classifier.

Depending on the role of hidden information, we can either apply loss inequality regularization (LIR), relationship preserving regularization, or combine both regularization terms to capture hidden information for classifier learning.

Learning with Incomplete Hidden Information The proposed methods can also be extended to deal with the case of incomplete hidden information. In other words, hidden information is only available for a subset of the training instances. This can be achieved by imposing either loss inequality regularization or relationship preserving regularization only on the training instances of which hidden information is available.

5. Experiments

We evaluate the proposed methods for different computer vision applications. The goal is to evaluate if the proposed methods can effectively exploit hidden information for classifier learning and compare their performances with related methods. We denote the proposed loss inequality regularization as **LIR** and the proposed relationship preserving regularization method as **RPR**. We compare them with the related methods **SVM+** [1], and Rank Transfer **RSVM+** [3]. Besides, we also compare the proposed methods with methods which can be straightforwardly applied to encode hidden information, including a general multi-view method **MV** [17], a multi-view method SVM2K [16] specifically designed for SVM, multi-task learning method **MTL** [22], and the method **MLR** that uses hidden information as middle level representation. For all methods, we perform five fold cross validation approach to tune the regularization parameters.

We use conjugate gradient descent for both LIR and RPR. The number of calculations to compute the objective value and gradient at each iteration is $O(n(d_1 + d_2))$ for LIR, and $O(nd_1)$ for RPR, where d_1 and d_2 are dimensions of original features and hidden information, and n is the number of training data. In practice it takes less than 0.1 second for the proposed methods to finish training in both our experiments with unoptimized matlab implementation.

5.1. Facial expression recognition with facial action units as hidden information

In this first experiment, we compare all the methods to use digital face images to recognize facial expression, including anger, contempt, disgust, fear, happiness, sadness and surprise. A typical approach is to collect appearance or geometric features from the face images and then train a classifier to classify the expressions. However, besides the raw image measurements, we may usually obtain the facial action units which represent the local facial muscle actions such as "raise eye brow", "raise cheek", and "pull lip corner". These facial muscle actions contain very discriminative information about the facial expressions. They can be annotated for the training images yet are usually expensive to obtain for testing images. In this experiment, we apply different methods to exploit facial action units as hidden information for facial expression recognition. An example of facial image and the corresponding facial action units are shown in Figure 1.

Data set and features. The experiment is performed on extended Cohn-Kanade dataset (CK+) dataset [23] which contains 327 video sequences performed by 210 subjects. We perform facial expression recognition on the peak frames of each video sequence. The primary features x we use in this case are the displacement of 68 facial landmarks of each face image, and the hidden information h is the hu-

Table 1: Results for facial expression recognition (* means primary and secondary features combined as secondary features). The values included in the parenthesis are the p-values of the one-tailed student-t test of the method against the baseline. Highlighted in gray background represents statistical significance against the baseline that do not use hidden information.

Method	# train subject: 168	# train subject: 105
Baseline without hidden information		
RLS	85.19±1.37	81.95±0.69
Hidden information as secondary targets		
MTL	86.61±1.19 (0.0700)	82.86±0.71 (0.0413)
MLR	85.33±0.77 (0.4630)	82.11±0.64 (0.4273)
RPR	87.57±1.27 (0.0023)	84.75±0.77 (0.1E-7)
Hidden information as secondary features		
MV	86.68±1.24 (0.0684)	82.04±0.83 (0.2778)
MV*	86.95 ±1.35 (0.0385)	82.34 ±0.67 (0.0576)
LIR	87.24±1.16 (0.0043)	84.32±0.73 (0.4E-5)
LIR*	87.46±1.32 (0.0075)	84.52±0.72 (0.2E-5)
Combining RPR and LIR together		
RPR+LIR	88.56±1.19 (0.2E-4)	84.81±0.73 (0.6E-6)
RPR+LIR*	**88.60±1.26 (0.7E-4)**	**84.83±0.72 (0.9E-7)**
Related methods		
SVM	86.67±1.46	82.85±0.69
SVM2K[16]	86.65±1.24 (0.4243)	82.91±0.66 (0.3822)
SVM+	86.95±1.13 (0.3504)	83.40±0.73 (0.2854)

man annotated 17 binary facial action units.

Implementation details. In order to classify 7 expressions, we train a total of 21 binary classifiers for each pair of expressions. The final prediction is made by comparing the predictions from all the binary classifiers. The performance is measured by the average recognition accuracy of all the 7 expressions. We test the performance of the methods using different training data size, and we repeat the procedure of train/test split for 20 times to get statistics of the performance. Specifically, each time we randomly select images from either 105 or 168 subjects as training data, and images of the remaining subjects as testing data. We use linear classifier and square loss for all the methods in this experiment.

Results. The detailed results are shown in Table 1. RLS is the baseline empirical risk minimization method using square loss. Facial action units can be treated as either features or labels. Therefore we can apply both proposed regularization methods to exploit them. Below we analyze and compare different methods depending on how we utilize facial action units. Specifically, we analyze both the empirical significance of each method, and the statistical significance by performing one tailed student-t test using a significance level of 0.05.

Secondary targets. We first compare three methods discussed in Section 4.2 that encode hidden information as secondary targets. They are MTL (multi-task method), MLR (hidden information as middle level representation), and RPR (relationship preserving regularization). First of all, we can see that all methods improve the baseline by utilizing facial action units as hidden information. In particular, RPR outperforms the baseline by 2.4% (2.8%) when the training subject number is 168 (105). Student t-test shows that RPR significantly outperforms the baseline in both cases. However, Student t-test shows that both MTL and MLR do not significantly better than the baseline. All these results demonstrate that RPR can successfully exploits facial action units as hidden information to improve facial expression recognition.

Secondary features. We also compare the two methods discussed in Section 4.1 which treats hidden information as secondary features. They are MV(multi-view method) and LIR (loss-inequality regularization). We can directly use facial action units as secondary features. We can also combine the primary feature and facial action units together as secondary features. We use $*$ to indicate methods that use combined features as hidden information. We test the performances of both methods in both cases. We can see that in both cases, facial action units, used as hidden information, can successfully boost the performance of the baseline. In particular, the proposed method LIR outperforms the baseline by more than 2% in all cases. We also perform student t-test to evaluate the statistical significance of the methods against the baseline. Results show that with hidden information, the performance improvement by LIR and LIR* over the baseline method is statistically significant, while MV and MV* are not.

Combined regularization. Facial action units can be used as either features or targets. Hence we can combine the two regularization terms together to exploit them. We can see that by combining both regularization terms together, we can further marginally improve the performance.

Comparison with SVM+. We also compare proposed methods with the related approach SVM+ [1]. We can see that SVM+ outperforms its counter part SVM in both cases yet the improvement is not statistically significant. On the contrary, all the proposed methods achieve statistically significant results by incorporating hidden information. Student t test was also performed to compare RPR+LIR* and SVM+. The results show that when training data size is small (105 training subjects), the improvement over SVM+ is statistically significant (p-value is 0.0492). When training data size gets larger (168 training subjects), the improvement over SVM+ gets smaller (p-value is 0.0978).

Performance with different training data size. We tested the performance of different methods with different training data size. We can see that when the training data size is smaller, the proposed methods tend to achieve more improvements with hidden information. This indicates that hidden information is more useful when the training data size is smaller.

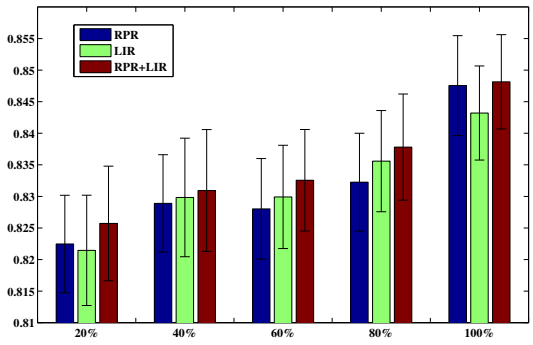

Figure 2: Results of different methods with incomplete hidden information for facial expression recognition. Y axis: average accuracy. X axis: percentage of hidden information for training data.

Incomplete hidden information. The proposed methods can also be extended for the case where hidden information is only available for a subset of the training data. Without loss of generality, we test LIR, RPR and LIR+RPR with incomplete data. Specifically, we randomly select data from 105 subjects as training data and the rest as testing data. For each selected training data, we sample different number of facial action units as hidden information to train the proposed methods. This procedure is repeated for 20 times. The average results and standard errors are shown in Figure 2. We can see that the performance of methods increase as we increase the percentage of available hidden information.

5.2. Object classification with attribute scores as hidden information

The human annotated attributes describe the high-level semantic properties about the objects in the image. In our second experiment we test the methods for object classification where the object attributes are used as hidden information.

Following the same experimental setting in [3], we perform 45 pair-wise classifications among a total of 10 classes, namely, *chimpanzee, giant panda, leopard, persian cat, pig, hippopotamus, humpback whale, raccoon, rat and seal* from the Animals with Attributes (AwA) dataset [24]. For each binary classification problem, we use 50 images for training and another 200 images for testing. This procedure is repeated for 20 times. The average precision (AP), which corresponds to the area under the precision-recall curve, is used as the measure of performance. Results of SVM, SVM+, RSVM+ are directly collected from [3]. Regularization parameters of the proposed method are tuned using five fold cross-validation.

The L_1 normalized 2000 dimensional SURF descriptors [25] extracted from the raw images are used as the primary feature x. The estimated scores of a total of 85 binary attributes for each image are used as the hidden information h. Since we are only given the estimated scores of attributes instead of ground truth attributes, we only use LIR in this experiment. For fair comparison with the results reported [3], we use linear classifier and hinge loss for LIR.

Results. The detailed results are shown in Table 2, where the results of SVM, SVM+ and RSVM+ are directly collected from [3]. First of all, the best result of each task is highlighted in boldface, which in total is 4 for SVM, 2 for SVM+, 13 for RSVM+ and 27 for LIR with hinge loss. We can see that the proposed method achieves the best performance in most number of tasks. Second, we also perform student t-test to analyze the statistical significance of the methods that utilize hidden information against SVM. Results show that LIR achieves statistically significant improvement in 19 cases, RSVM+ achieves statistically significant improvement in 5 cases, and SVM+ achieves statistically significant improvement in 0 case. All these results demonstrate that the proposed method improves the related methods RSVM+ and SVM+ in encoding hidden information for classifier learning. A graphical comparison of different methods over the baseline SVM is shown in Figure 3.

6. Conclusion

In this paper we proposed two regularization methods to incorporate hidden information, which is only available during training but not available during testing, to learn a better classifier. Specifically, loss inequality regularization approach is used to exploit hidden information as secondary features, and relationship preserving regularization approach is used to exploit hidden information as secondary targets. They can be used individually or simultaneously to capture hidden information. Compared to the existing methods, the proposed approaches are general (applicable to different types of hidden information and classifiers), without strong assumptions. The proposed methods are evaluated on different applications. Experimental results demonstrate the effectiveness of the proposed method for exploiting hidden information, as well as its superior performance to the related methods.

Acknowledgement

The work described in this paper is supported in part by the grant IIS 1145152 from the National Science Foundation.

References

[1] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22:544–557, July 2009. 2, 5, 6, 8

[2] Jixu Chen, Xiaoming Liu, and Siwei Lyu. Boosting with side information. In *ACCV*, 2012. 2

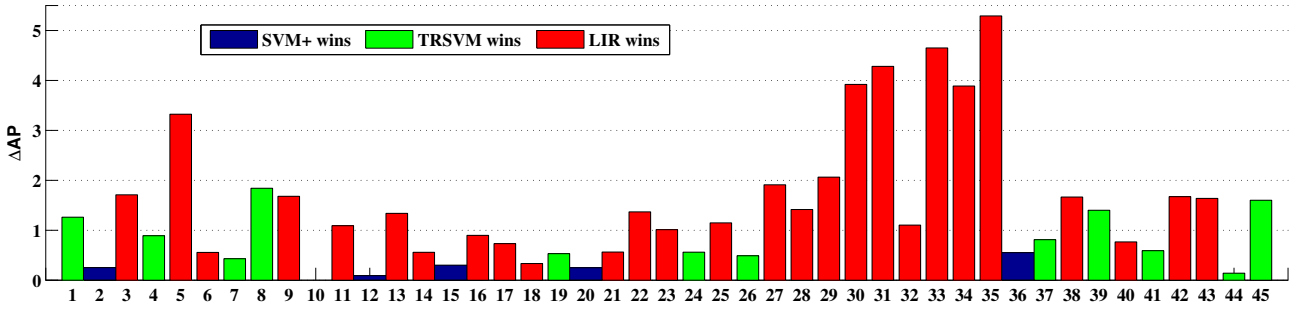

Figure 3: Comparison of different methods for object recognition. The 45 bars correspond to 45 binary classification tasks. The y axis represents the AP improvement of the best method over SVM. Each distinct color represents one method.

Table 2: Results of object recognition with attribute scores as hidden information. The numbers are mean and standard error of the AP performance over 20 runs with N = 50 training samples per class.

		SVM	SVM+ [1]	RSVM+ [3]	LIR(hinge loss)
1	Chimpanzee versus Giant panda	88.88 ± 0.51	88.07 ± 0.57	**89.33 ± 0.50**	88.28 ±0.47
2	Chimpanzee versus Leopard	**93.74 ± 0.26**	93.49 ± 0.29	93.70 ± 0.23	93.36 ±0.15
3	Chimpanzee versus Persian cat	90.14 ± 0.40	89.88 ± 0.42	91.00 ± 0.39	**91.59 ±0.40**
4	Chimpanzee versus Pig	85.64 ± 0.57	85.19 ± 0.53	**86.08 ± 0.43**	83.74 ±0.35
5	Chimpanzee versus Hippopotamus	86.40 ± 0.55	86.31 ± 0.59	86.92 ± 0.45	**89.63 ±0.31**
6	Chimpanzee versus Humpback whale	98.03 ± 0.18	97.74 ± 0.22	98.08 ± 0.18	**98.30 ±0.16**
7	Chimpanzee versus Raccoon	87.01 ± 0.46	86.64 ± 0.47	**87.07 ± 0.48**	85.90 ±0.63
8	Chimpanzee versus Rat	85.42 ± 0.53	84.83 ± 0.68	**86.67 ± 0.56**	85.43 ±0.48
9	Chimpanzee versus Seal	91.74 ± 0.39	91.10 ± 0.59	91.54 ± 0.43	**92.78 ±0.42**
10	Giant panda versus Leopard	93.71 ± 0.38	**94.03 ± 0.28**	93.76 ± 0.29	92.81 ±0.48
11	Giant panda versus Persian cat	92.55 ± 0.41	92.66 ± 0.32	92.57 ± 0.43	**93.75 ±0.29**
12	Giant panda versus Pig	86.64 ± 0.45	**86.55 ± 0.40**	86.00 ± 0.52	84.19 ±0.69
13	Giant panda versus Hippopotamus	90.04 ± 0.56	89.93 ± 0.56	90.89 ± 0.36	**91.27 ±0.35**
14	Giant panda versus Humpback whale	98.38 ± 0.17	98.11 ± 0.19	98.53 ± 0.15	**98.67 ±0.11**
15	Giant panda versus Raccoon	**89.36 ± 0.44**	89.06 ± 0.49	88.66 ± 0.60	86.90 ±0.74
16	Giant panda versus Rat	88.49 ± 0.49	87.86 ± 0.48	87.53 ± 0.51	**88.76 ±0.37**
17	Giant panda versus Seal	92.81 ± 0.32	92.59 ± 0.38	92.40 ± 0.40	**93.32 ±0.31**
18	Leopard versus Persian cat	95.08 ± 0.25	94.93 ± 0.24	**95.26 ± 0.25**	**95.26 ±0.22**
19	Leopard versus Pig	88.55 ± 0.28	88.37 ± 0.36	**88.90 ± 0.28**	85.34 ±0.50
20	Leopard versus Hippopotamus	**92.98 ± 0.29**	92.73 ± 0.31	92.86 ± 0.26	92.54 ±0.28
21	Leopard versus Humpback whale	98.49 ± 0.30	98.27 ± 0.33	98.63 ± 0.23	**98.83 ±0.11**
22	Leopard versus Raccoon	80.31 ± 0.75	79.94 ± 0.73	79.84 ± 0.59	**81.31 ±0.67**
23	Leopard versus Rat	88.74 ± 0.35	88.92 ± 0.35	89.27 ± 0.28	**89.93 ±0.28**
24	Leopard versus Seal	93.87 ± 0.36	93.74 ± 0.37	**94.30 ± 0.36**	94.12 ±0.21
25	Persian cat versus Pig	81.55 ± 0.59	81.45 ± 0.57	81.68 ± 0.46	**82.60 ±0.58**
26	Persian cat versus Hippopotamus	92.42 ± 0.34	92.33 ± 0.33	**92.82 ± 0.30**	92.00 ±0.49
27	Persian cat versus Humpback whale	95.92 ± 0.29	95.45 ± 0.38	95.84 ± 0.30	**97.36 ±0.15**
28	Persian cat versus Raccoon	90.19 ± 0.40	90.31 ± 0.41	90.38 ± 0.39	**91.72 ±0.34**
29	Persian cat versus Rat	67.19 ± 0.60	67.56 ± 0.63	69.07 ± 0.48	**69.62 ±0.84**
30	Persian cat versus Seal	84.79 ± 0.60	84.46 ± 0.54	85.66 ± 0.49	**88.38 ±0.44**
31	Pig versus Hippopotamus	74.42 ± 0.48	73.47 ± 0.55	75.57 ± 0.58	**77.75 ±0.51**
32	Pig versus Humpback whale	96.01 ± 0.33	95.75 ± 0.30	95.93 ± 0.37	**96.85 ±0.18**
33	Pig versus Raccoon	77.73 ± 0.80	76.96 ± 0.85	79.13 ± 0.63	**81.61 ±0.71**
34	Pig versus Rat	68.66 ± 0.76	68.58 ± 0.41	70.77 ± 0.73	**72.47 ±0.55**
35	Pig versus Seal	77.91 ± 0.71	77.32 ± 0.73	79.26 ± 0.77	**82.61 ±0.55**
36	Hippopotamus versus Humpback whale	**92.19 ± 0.44**	91.64 ± 0.60	92.17 ± 0.44	91.08 ±0.63
37	Hippopotamus versus Raccoon	85.54 ± 0.60	85.03 ± 0.60	**85.84 ± 0.70**	85.72 ±0.63
38	Hippopotamus versus Rat	84.49 ± 0.39	84.25 ± 0.37	85.62 ± 0.48	**85.91 ±0.48**
39	Hippopotamus versus Seal	69.79 ± 0.83	69.43 ± 0.84	**70.83 ± 0.79**	69.79 ±0.70
40	Humpback whale versus Raccoon	96.67 ± 0.28	96.57 ± 0.31	96.90 ± 0.29	**97.34 ±0.20**
41	Humpback whale versus Rat	94.52 ± 0.19	93.97 ± 0.24	**94.56 ± 0.22**	92.95 ±0.68
42	Humpback whale versus Seal	84.60 ± 0.49	84.24 ± 0.49	84.81 ± 0.38	**85.91 ±0.57**
43	Raccoon versus Rat	77.65 ± 0.64	78.36 ± 0.54	78.61 ± 0.72	**80.00 ±0.57**
44	Raccoon versus Seal	91.43 ± 0.36	91.37 ± 0.38	**91.51 ± 0.40**	89.21 ±0.43
45	Rat versus Seal	78.45 ± 0.65	78.28 ± 0.75	**79.88 ± 0.69**	79.02 ±0.50
Average		87.28	87.53	87.92	88.13

[3] Viktoriia Sharmanska, Novi Quadrianto, and Christoph H Lampert. Learning to rank using privileged information. In *ICCV*, 2013. 2, 4, 5, 7, 8

[4] Ziheng Wang, Xiaoyang Wang, and Qiang Ji. Learning with hidden information. In *ICPR*, pages 238–243. IEEE, 2014. 2

[5] Ziheng Wang, Tian Gao, and Qiang Ji. Learning with hidden information using a max-margin latent variable model. In *ICPR*, pages 1389–1394. IEEE, 2014. 2

[6] Zuoguan Wang, Gerwin Schalk, and Qiang Ji. Anatomically constrained decoding of finger flexion from electrocorticographic signals. In *Advances in Neural Information Process-*

ing Systems, pages 2070–2078, 2011. 2

[7] Jan Feyereisl and Uwe Aickelin. Privileged information for data clustering. *Information Sciences*, 194:4–23, 2012. 2

[8] Shereen Fouad, Peter Tino, Somak Raychaudhury, and Petra Schneider. Learning using privileged information in prototype based models. In *ICANN*. 2012. 2

[9] Dmitry Pechyony and Vladimir Vapnik. On the Theory of Learning with Privileged Information. In *NIPS*, 2010. 2

[10] Lingfeng Niu, Yong Shi, and Jianmin Wu. Learning using privileged information with l-1 support vector machine. In *WI-IAT*, 2012. 2

[11] Jun Liua, Wenxin Zhua, and Ping Zhonga. A new multi-class support vector algorithm based on privileged information. *Journal of Information and Computational Science*, 2013. 2

[12] You Ji, Shiliang Sun, and Yue Lu. Multitask multiclass privileged information support vector machines. In *ICPR*, 2012. 2

[13] Maksim Lapin, Matthias Hein, and Bernt Schiele. Learning using privileged information: Svm+ and weighted svm. *arXiv preprint arXiv:1306.3161*, 2013. 2

[14] Feng Cai and Vladimir Cherkassky. Generalized smo algorithm for svm-based multitask learning. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(6):997–1003, 2012. 2

[15] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *ICML Workshop*, 2005. 3

[16] Jason Farquhar, David Hardoon, Hongying Meng, John S Shawe-taylor, and Sandor Szedmak. Two view learning: Svm-2k, theory and practice. In *NIPS*, 2005. 3, 5, 6

[17] David S Rosenberg and Peter L Bartlett. The rademacher complexity of co-regularized kernel classes. In *International Conference on Artificial Intelligence and Statistics*, 2007. 3, 5

[18] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117. ACM, 2004. 4

[19] Tsuyoshi Kato, Hisashi Kashima, Masashi Sugiyama, and Kiyoshi Asai. Multi-task learning via conic programming. In *NIPS*, 2007. 4

[20] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *The Journal of Machine Learning Research*, 8:35–63, 2007. 4

[21] Laurent Jacob, Francis Bach, Jean-Philippe Vert, et al. Clustered multi-task learning: A convex formulation. In *NIPS*, volume 21, pages 745–752, 2008. 4

[22] Yu Zhang and Dit-Yan Yeung. A convex formulation for learning task relationships in multi-task learning. In *26th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 733–742, 2010. 5

[23] P. Lucey, J.F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *CVPR Workshop*, pages 94–101, june 2010. 5

[24] C. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot learning of object categories. In *PAMI*, 2013. 7

[25] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*. 2006. 7