Maximum Mutual Information Regularized Classification

Jim Jing-Yan Wang\textsuperscript{a}, Yi Wang\textsuperscript{b}, Shiguang Zhao\textsuperscript{c}, Xin Gao\textsuperscript{a,}\textsuperscript{*}

\textsuperscript{a}Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia
\textsuperscript{b}Department of Computer Science and Engineering, The Ohio State University, Columbus OH 43210, USA
\textsuperscript{c}Department of Neurosurgery, The First Affiliated Hospital of Harbin Medical University, Harbin, Heilongjiang 150001, P.R. China

Abstract

In this paper, a novel pattern classification approach is proposed by regularizing the classifier learning to maximize mutual information between the classification response and the true class label. We argue that, with the learned classifier, the uncertainty of the true class label of a data sample should be reduced by knowing its classification response as much as possible. The reduced uncertainty is measured by the mutual information between the classification response and the true class label. To this end, when learning a linear classifier, we propose to maximize the mutual information between classification responses and true class labels of training samples, besides minimizing the classification error and reducing the classifier complexity. An objective function is constructed by modeling mutual information with entropy estimation, and it is optimized by a gradient descend method in an iterative algorithm. Experiments on two real world pattern classification problems show the significant improvements achieved by maximum mutual information regularization.

Keywords: Pattern Classification, Maximum Mutual Information, Entropy, Gradient Descend

1. Introduction

The pattern classification problem is a problem of assigning a discrete class label to a given data sample represented by its feature vector \([1, 2, 3]\). It has many applications in various fields, including bioinformatics \([4, 5, 6, 7, 8]\), biometrics verification \([9, 10, 11]\), computer networks \([12, 13, 14]\), and computer...
vision [15, 16, 17]. For example, in the face recognition problem, given a face image, the target of pattern classification is to assign it to a person who has been registered in a database [18, 19, 20]. This problem is usually composed of two different components — feature extraction [21, 22, 23, 24, 25, 26, 27, 28] and classification [29, 30]. Feature extraction refers to the procedure of extracting an effective and discriminant feature vector from a data sample, so that different samples of different classes could be separated easily. This procedure is usually highly domain-specific. For example, for the face recognition problem, the visual feature should be extracted using some image processing technologies, whereas for the problem of predicting zinc-binding sites from protein sequences, the biological features should be extracted using some biological knowledge [31]. On the other hand, different from feature extraction, classification is a much more general problem. We usually design a class label prediction function as a classifier for this purpose. To learn the parameter of a classifier function, we usually try to minimize the classification error of the training samples in a training set and simultaneously reduce the complexity of the classifier. For example, the most popular classifier is support vector machine (SVM), which minimizes the hinge losses to reduce the classification error, and at the same time minimizes the $\ell_2$ norm of the classifier parameters to reduce the complexity [32]. In this paper, we focus on the classification aspect.

Mutual information [33] is defined as the information shared between two sets of variables. It has been used as a criterion of feature extraction for pattern classification problems [34]. However, surprisingly, it has never been directly explored in the problem of classifier learning. Actually, mutual information has a strong relation to Kullback-Leibler divergence, and there are many works using KL-divergence for classifiers [35, 36]. Moreno et al. [35] proposed a novel kernel function for support vector classification based on Kullback-Leibler divergence, while Liu and Shum [36] proposed to learn the most discriminating feature that maximizes the Kullback-Leibler divergence for the Adaboost classifier. However, both these methods do not use the KL-divergence based criterion to learn parameters of linear classifiers. To bridge this gap, in this paper, for the first time, we try to investigate using mutual information as a criterion of classifier learning. We propose to learn a classifier by maximizing the mutual information $I(f; y)$ between the classification response variable $f$ and the true class label variable $y$. The classification response variable $f$ is a function of classifier parameters and data samples. The insight is that mutual information is defined as the information shared between $f$ and $y$. From the viewpoint of information theory, if the two variables are not mutually independent, and one variable is known, it usually reduces the uncertainty about the other one. Then mutual information is used to measure how much uncertainty is reduced in this case. To illuminate how the mutual information can be used to measure the classification accuracy, we consider the two extreme cases:

- On one hand, if the classification response variable $f$ of a data sample is randomly given, and it is independent of its true class label $y$, then knowing $f$ does not give any information about $y$ and vice versa, and the
mutual information between them could be zero, i.e., $I(f; y) = 0$. 

- On the other hand, if $f$ is given so that $y$ and $f$ are identical, knowing $f$ can help determine the value of $y$ exactly as well as reduce all the uncertainty about $y$. This is the ideal case of classification, and knowing $f$ can reduce all the uncertainty about $y$. In this case, the mutual information is defined as the uncertainty contained in $f$ (or $y$) alone, which is measured by the entropy of $f$ or $y$, denoted by $H(f)$ or $H(y)$ respectively, where $H(\cdot)$ is the entropy of a variable. Since $f$ and $y$ are identical, we can have $I(f; y) = H(f) = H(y)$.

Naturally, we hope that the classification response $f$ can predict the true class label $y$ as accurately as possible, and knowing $f$ can reduce the uncertainty about $y$ as much as possible. Thus, we propose to maximize the mutual information between $f$ and $y$ with regard to the parameters of a classifier. To this end, we proposed a mutual information regularization term for the learning of classifier parameters. An objective function is constructed by combining the mutual information regularization term, a classification error term and a classifier complexity term. The classifier parameter is learned by optimizing the objective function with a gradient descend method in an iterative algorithm.

The rest parts of this paper are organized as follows: in Section 2, we introduce the proposed classifier learning method. The experiment results are presented in section 3. In section 4 the paper is concluded.

2. Proposed Method

In this section, we introduce the proposed classifier learning algorithm to maximize the mutual information between the classification response and the true class label.

2.1. Problem Formulation

We suppose that we have a training set denoted as $X = \{x_i\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ is the $d$-dimensional feature vector for the $i$-th training sample, and $n$ is the number of training samples. The class label set for the training samples is denoted as $Y = \{y_i\}_{i=1}^n$, where $y_i \in \{+1, -1\}$ is the class label of the $i$-th sample. To learn a classifier to predict the class label of a given sample with its feature vector $\mathbf{x}$, we design a linear function as a classifier,

$$g(\mathbf{x}; w) = \text{sign}(f) = \text{sign}(\mathbf{w}^\top \mathbf{x}),$$

(1)

where $\mathbf{w}$ is the classifier parameter vector, $f = \mathbf{w}^\top \mathbf{x}$ is the classification response of $\mathbf{x}$ given the classifier parameter $\mathbf{w}$, and $\text{sign}(\cdot)$ is the signum function which transfers the classification response to the final binary classification result. We also denote the classification response set of the training samples as $F = \{f_i\}_{i=1}^n$, where $f_i = \mathbf{w}^\top \mathbf{x}_i \in \mathbb{R}$ is the classification response of the $i$-th training sample. To learn the optimal classification parameter $\mathbf{w}$ for the classification problem, we consider the following three problems:
2.1.1. Classification Loss Minimization

To learn the optimal classification parameter \( w \), we hope the classification response \( f \) of a data sample \( x \) obtained with the learned \( w \) can predict its true class label \( y \) as accurately as possible. To measure the prediction error, we use a loss function to compare a classification response against its corresponding true class label. Given the classifier parameter \( w \), the loss function of the \( i \)-th training sample \( x_i \) with its classification response \( f_i = w^\top x_i \) and true class label \( y_i \) is denoted as \( L(f_i, y_i; w) \). There are a few different loss functions which could be considered.

**Hinge Loss** is used by the SVM classifier [37, 38, 39], and it is defined as

\[
L(f_i, y_i; w) = \max(0, 1 - y_i f_i) = \tau_i \times (1 - y_i w^\top x_i),
\]

where \( \tau_i \) is defined as

\[
\tau_i = \begin{cases} 
1, & \text{if } y_i w^\top x_i \leq 1 \\
0, & \text{otherwise.}
\end{cases}
\]

**Squared Loss** is usually used by regression problems [40, 41, 42], and it is defined as

\[
L(f_i, y_i; w) = (1 - y_i f_i)^2 = (1 - y_i w^\top x_i)^2.
\]

**Logistic Loss** is defined as follows, and it is also popular in regression problems [43, 44],

\[
L(f_i, y_i; w) = \log [1 + \exp(-y_i f_i)] = \log \left[1 + \exp(-y_i w^\top x_i)\right].
\]

**Exponential loss** is another popular loss function which could be used by both classification and regression problems [40, 45], which is defined as

\[
L(f_i, y_i; w) = \exp(-y_i f_i) = \exp(-y_i w^\top x_i).
\]

Obviously, to learn an optimal classifier, the average loss of all the training samples should be minimized with regard to \( w \). Thus the following optimization problem is obtained by applying a loss functions to all training samples,

\[
\min_w \frac{1}{n} \sum_{i=1}^{n} L(f_i, y_i; w).
\]
2.1.2. Classifier Complexity Reduction

To reduce the complexity of the classifier to prevent the over-fitting problem, we also regularize the classifier parameter by a $\ell_2$ norm term as

$$\min_w \frac{1}{2} \|w\|^2_2. \quad (8)$$

2.1.3. Mutual Information Maximization

We also propose to learn the classifier by maximizing the mutual information $I(f; y)$ between the classification response variables $f \in F$ and the true class labels $y \in Y$. The mutual information between two variables $f \in F$ and $y \in Y$ is defined as

$$I(f; y) = H(f) - H(f|y), \quad (9)$$

where $H(f)$ is the marginal entropy of $f$, which is used to measure the uncertainty about $f$, and $H(f|y)$ is the entropy of $f$ conditional on $y$, which is used as the measure of uncertainty of $f$ when $y$ is given. To use the mutual information as a criterion to learn the classifier parameters, we first need to estimate $H(f)$ and $H(f|y)$.

**Estimation of $H(f)$** We use the training samples to estimate $H(f)$, and according to the definition of entropy, we have

$$H(f) = -\int p(f) \log p(f) df \approx -\sum_{i=1}^n p(f_i) \log p(f_i), \quad (10)$$

where $p(f)$ is the probability density of $f$. It could be seen that the entropy of $f$ is the expectation of $-\log p(f)$ [46]. The non-parametric kernel density estimation (KDE) [47] is used to estimate the probability density function $p(f)$,

$$p(f) = \frac{1}{n} \sum_{j=1}^n K(f - f_j; \sigma), \quad (11)$$

where $K(z; \sigma) = \exp \left(-\frac{z^2}{2\sigma^2}\right)$ is the Gaussian kernel function [48] and $\sigma$ is the bandwidth parameter [47].

**Estimation of $H(f|y)$** We also use the training samples to estimate $H(f|y)$, and according to its definition, we have

$$H(f|y) = \sum_{c \in \{+1, -1\}} p(c)H(f|y = c), \quad (12)$$
where \( p(c) = \frac{n_c}{n} \) is the probability density of class label \( c \), \( n_c = \# \{ \mathbf{x}_i \in X | y_i = c \} \) is the number of samples with the class label equal to \( c \), and

\[
H(f|y = c) = -\int p(f|y = c) \log p(f|y = c) df
\approx -\sum_{i: y_i = c} p(f_i|y = c) \log p(f_i|y = c)
\]

(13)
is the conditional entropy of \( f \) given the class label \( y = c \) \([49, 50, 51]\). We also use the KDE to estimate the conditional probability density function

\[
p(f|y = c) = \frac{1}{n_c} \sum_{i: y_i = c} K(f - f_i, \sigma)
\]

(14)

Substituting it to (12), we have the estimated \( H(f|y) \),

\[
H(f|y) \approx -\sum_{c \in \{+1, -1\}} \frac{n_c}{n} \left( \frac{n_c}{n} \sum_{i: y_i = c} p(f_i|y = c) \log p(f_i|y = c) \right)
\]

(15)

With the estimated entropy \( H(f) \) and the conditional entropy \( H(f|y) \), the mutual information between the variable \( f \) and \( y \) could be rewritten as the function of parameter \( w \) by substituting \( f_i = w^\top \mathbf{x}_i \),

\[
\tilde{I}(f, y; w) = H(f) - H(f|y)
\]

\[
= -\sum_{i=1}^{n} p(f_i) \log p(f_i)
+ \sum_{c \in \{+1, -1\}} \frac{n_c}{n} \left( \frac{n_c}{n} \sum_{i: y_i = c} p(f_i|y = c) \log p(f_i|y = c) \right)
\]

(16)

\[
= -\sum_{i=1}^{n} p(w^\top \mathbf{x}_i) \log p(w^\top \mathbf{x}_i)
+ \sum_{c \in \{+1, -1\}} \frac{n_c}{n} \left( \frac{n_c}{n} \sum_{i: y_i = c} p(w^\top \mathbf{x}_i|y = c) \log p(w^\top \mathbf{x}_i|y = c) \right)
\]

To learn the classifier parameter \( w \), we maximize the mutual information with regard to \( w \),

\[
\max_w \tilde{I}(f, y; w)
\]

Remark: It should be noted that similar to our method, the algorithm proposed in [36] maximizes KL-divergence between the class PDF, \( p(f|y = c) \),
and the total PDF, \( p(f) \), therefore, [36] has relation to method in Kullback-Leibler boosting. However, different from our method, it uses KL-divergence as a criterion to select the most discriminating features, whereas our method uses mutual information as a criterion to learn the classifier parameter.

2.1.4. Overall Optimization Problem

By combining the optimization problems proposed in (7), (8) and (17), the optimization problem for the proposed classifier parameter learning method is obtained as

\[
\min_w \left\{ O(w) = \frac{1}{n} \sum_{i=1}^{n} L(f_i, y_i; w) + \alpha \frac{1}{2} \|w\|_2^2 - \beta I(f; y; w) \right\}, \tag{18}
\]

where \( \alpha \) and \( \beta \) are tradeoff parameters. In the objective function, there are three terms. The first one is optimized so that the prediction error is minimized, the second term is used to control the complexity of the classifier, and the last term is introduced so that the mutual information between the classification response and the true class label can be maximized.

2.2. Optimization

Direct optimization to (18) is difficult. Instead of seeking a closed-form solution, we try to optimize it using gradient descent method in an iterative algorithm [52]. In each iteration, we employ the gradient descent method to update \( w \). According to the optimization theory, if \( Q(w) \) is defined and differentiable in a neighborhood of a point \( w^{old} \), then \( Q(w) \) decreases faster if \( w \) goes from \( w^{old} \) in the direction of the negative gradient of \( Q(w) \) at \( w^{old} \),

\[
-\nabla Q(w)|_{w=w^{old}} \quad \text{new} \leftarrow \text{old} - \eta \nabla Q(w)|_{w=w^{old}}, \tag{19}
\]

where \( \eta \) is the descent step.

The key step is to compute the gradient of \( Q(w) \), which is calculated as

\[
\nabla Q(w) = \frac{1}{n} \sum_{i=1}^{n} \nabla L(f_i, y_i; w) + \alpha w - \beta \nabla I(f; y; w), \tag{20}
\]

where \( \nabla L(f_i, y_i; w) \) and \( \nabla I(f; y; w) \) are the gradient of \( L(f_i, y_i; w) \) and \( I(f; y; w) \) respectively. They are given analytically as follows.

2.2.1. Computation of \( \nabla L(f_i, y_i; w) \)

We give the analytical gradients of different definitions of \( L(f_i, y_i; w) \) as follows:

Hinge Loss is not a smooth function, but we can first update \( \tau_i \) using previous \( w \) as in (3), and then fix it when we derive \( \nabla L(f_i, y_i; w) \),

\[
\nabla L(f_i, y_i; w) = -\tau_i \times (y_i x_i). \tag{21}
\]
**Squared Loss** is a smooth function, and its gradient is

$$\nabla L(f_i, y_i; w) = -(1 - y_i w^\top x_i) \times (y_i x_i).$$  \hspace{1cm} (22)

**Logistic Loss** is also smooth with its gradient as

$$\nabla L(f_i, y_i; w) = -\frac{\exp(-y_i w^\top x_i)}{1 + \exp(-y_i w^\top x_i)} \times (y_i x_i).$$  \hspace{1cm} (23)

**Exponential loss** is also smooth, and its gradient can be obtained as

$$\nabla L(f_i, y_i; w) = -\exp(-y_i w^\top x_i) \times (y_i x_i).$$  \hspace{1cm} (24)

2.2.2. **Computation of** $\nabla \tilde{I}(f, y; w)$

The gradient of $\tilde{I}(f, y; w)$ is computed as

$$\nabla \tilde{I}(f, y; w) = -\sum_{i=1}^{n} \left( \nabla p(w^\top x_i) \log p(w^\top x_i) + \nabla p(w^\top x_i) \right)$$

$$+ \sum_{c \in \{+1, -1\}} \frac{n_c}{n} \left[ \sum_{i: y_i = c} \left( \nabla p(w^\top x_i | y = c) \log p(w^\top x_i | y = c) + \nabla p(w^\top x_i | y = c) \right) \right]$$

$$= -\sum_{i=1}^{n} \left( \log p(w^\top x_i) + 1 \right) \nabla p(w^\top x_i)$$

$$+ \sum_{c \in \{+1, -1\}} \frac{n_c}{n} \left( \sum_{i: y_i = c} \left( \log p(w^\top x_i | y = c) + 1 \right) \nabla p(w^\top x_i | y = c) \right),$$  \hspace{1cm} (25)

where the gradients of $p(w^\top x_i)$ and $p(w^\top x_i | y = c)$ are computed as

$$\nabla p(w^\top x_i) = \frac{1}{n \sigma^2} \sum_{j=1}^{n} \exp \left( -\frac{(w^\top x_i - w^\top x_j)^2}{2 \sigma^2} \right) (w^\top x_i - w^\top x_j) (x_j - x_i),$$

$$\nabla p(w^\top x_i | y = c) = \frac{1}{n_c \sigma^2} \sum_{j: y_j = c} \exp \left( -\frac{(w^\top x_i - w^\top x_j)^2}{2 \sigma^2} \right) (w^\top x_i - w^\top x_j) (x_j - x_i).$$  \hspace{1cm} (26)

3. **Experiments**

In this section, we evaluate the proposed classification method on two real world pattern classification problems.

8
3.1. Experiment I: Zinc-binding Site Prediction

Zinc is an important element for many biological processes of an organism, and it is closely related to many different diseases. Moreover, it is also critical for proteins to play their functional roles [31, 53, 54]. Thus functional annotation of zinc-binding proteins is necessary to biological process control and disease treatment. To this end, predicting zinc-binding sites of proteins shows its importance in bioinformatics problems. In the first experiment, we evaluate the proposed classification method on the problem of predicting zinc-binding sites.

3.1.1. Data Set and Protocol

For the purpose of experiment, we collected a set of amino acids of four types, which are CYS, HIS, GLU and ASP (CHED). These four types are the most common zinc-binding site types, which take up roughly 96% of the known zinc-binding sites. In the collected data set, there are 1,937 zinc-binding CHEDs and 11,049 non-zinc-binding CHEDs, resulting a data set of 13,986 data samples. Given a candidate CHED, the problem of zinc-binding site prediction is to predict if it is a zinc-binding site or a non-zinc-binding site. In this experiment, we treated a zinc-binding CHED as a positive sample, and a non-zinc-binding CHED as a negative sample. To extract features from a CHED, we computed the position specific substitution matrices (PSSM) [55], the relative weight of gapless real matches to pseudocounts (RW-GRMTP) [56], Shannon entropy [31], and composition of k-spaced amino acid pairs (CKSAAP) [57], and concatenated them to form a feature vector for each data sample. Please note that the value of each feature was scaled to the range between -1 and 1, so that the performance does not depend on the selection of scaling.

To conduct the experiment, we used the 10-fold cross validation protocol [58]. The entire data set was split into ten non-overlapping folds, and each fold was used as a test set in turn, while the remaining nine folds were combined and used as a training set. The proposed algorithm was performed to the training set to learn a classifier from the feature vectors of the training samples, and then the learned classifier was used to predict the class labels of the test samples. Please note that the tradeoff parameters of the proposed algorithm was tuned within the training set. The averaged value of the hyper-parameters $\alpha$ and $\beta$ are 5.8 and 44.8. The parameter $\sigma$ was computed as $\sigma = \varsigma \times \text{dist}$, where $\text{dist}$ was the median value of distances between pairs of training samples, and the averaged value of $\varsigma$ was 0.451. The classification performance was measured by comparing the predicted labels against the true labels. The receiver operating characteristic (ROC) and recall-precision curves were used as performance metrics. The ROC curve was obtained by plotting true positive rates (TPRs) against false positive rates (FPRs), while recall-precision curve was obtained by plotting precision against recall values. TPR, FPR, recall, and precision are defined as
\[ TPR = \frac{TP}{TP + FN}, FPR = \frac{FP}{FP + TN}, \]
\[ \text{recall} = \frac{TP}{TP + FN}, \text{precision} = \frac{TP}{TP + FP}. \]  

(27)

where \( TP, FP, FN \) and \( TN \) represent the number of true positives, false positives, false negatives and true negatives, respectively. Moreover, area under ROC curve (AUC) [59] was used as a single performance measure. A good classifier should achieve a ROC curve close to the top left corner of the figure, a recall-precision curve close to the top right corner, and also a high AUC value.

3.1.2. Results

In this experiment, we compared the proposed mutual information regularized classifier against the original loss functions based classifier without mutual information regularization, so that the improvement achieved by maximum mutual information regularization could be verified. The four different loss functions listed in Section 2 were considered, and the corresponding classifiers were evaluated here. The ROC and recall-precision curves of four loss functions based classification methods are given in Fig. 1. The proposed maximum mutual information regularized method is denoted as “MaxMutInf” after a loss function title in the figure. It turns out that maximum mutual information regularization improves all the four loss functions based classification methods significantly. Although various loss functions achieved different performances, all of them could be boosted by reducing the uncertainty about true class labels, which could be measured by the mutual information between class labels and classification responses. Therefore, the results show that maximizing mutual information is highly effective in reducing uncertainty of true class labels, and hence it can significantly improve the quality of classification.

Moreover, we also plotted AUCs of different methods in Fig. 2. Again, we observe that maximum mutual information regularization improves different loss functions based classifiers. We also can see that among these four loss functions, hinge loss achieves the highest AUC values, while squared loss achieves the lowest. The AUC value of classifiers regularized by both hinge loss and mutual information is 0.9635, while that of squared loss and mutual information is even lower than 0.95. The performances of logistic and exponential loss functions are similar, and they are between the performances of hinge loss and squared loss.

Since the mutual information is used as a new regularization technique, we are also interested in how the proposed regularization alone works. We therefore compared the following three cases.

1. Conventional case which only uses the classification loss regularization. This case is corresponding to setting \( \beta = 0 \) in (18). In this case, we only used the hinge loss since it has been shown that this loss function obtains better accuracy than other loss functions.
Figure 1: ROC and recall curves of experiments on zinc-binding site prediction. “Loss” stands for combination of classification loss and $\ell_2$-regularization, and “Loss - MaxMutInf” stands for combination of classification loss, $\ell_2$ and maximum mutual information-regularization.
2. **Mutual information regularization case** which is corresponding to the problem in (18) when the first term is ignored.

3. **Hybrid regularization case** which is the proposed framework which combines the classification loss minimization and mutual information regularization.

The comparison results are given in Fig. 3. It can be seen that the conventional case which only uses the hinge loss function achieved better results than the method with only mutual information regularization, and the hybrid regularization achieved the best results. This means mutual information regularization cannot obtain good performance by itself and should be used with traditional loss functions.

3.2. **Experiment II: HEp-2 Cell Image Classification**

Antinuclear Autoantibodies (ANA) test is a technology used to determine whether a human immune system is creating antibodies to fight against infec-
3.2.1. Data Set and Protocol

In this experiment, we used the database of HEp-2 cell images of the ICIP 2014 competition of cell classification by fluorescent image analysis [61]. In this data set, there are 13,596 cell images, and they belong to six cell classes, which are namely Centromere, Golgi, Homogeneous, Nucleolar, NuclearMembrane, and Speckled. Each cell image is segmented by a mask image showing the boundary of the cell. Moreover, the entire data set is composed of two groups of different intensity types, which are Intermediate and Positive. Overall, the intermediate group outnumbers the positive group, with an exception that, for the cases of Centromere and Speckled, the latter marginally outnumbers the former. The number of images in different classes of two groups are given in Fig. 4. To present each image for the classification problem, we extracted shape and texture features and concatenated them to form a visual feature vector [60].

Experiments were conducted in two groups respectively. We also adopted the 10-fold cross validation for the experiment. To handle the problem of multiple class problem, we used the one-against-all strategy. Each class was treated as a positive class in turn, while all remaining five classes were combined to form a negative class. A classifier was learned for each class to discriminate it from other classes. A test sample was assigned to a class with the largest classification response. The classification accuracy was used as a classification performance.
3.2.2. Results

The boxplots of accuracies of the 10-fold cross validation on the two groups of HEp-2 cell image data set are given in Fig. 5. From this figure, it could be observed that for both two groups of data sets, the proposed regularization method can improve the classification performances significantly, despite of the variety of loss functions. It can also be seen that the performances on the second group (Positive) is inferior to that of the first group (Intermediate). This indicates that it is more difficult to classify cell images when their contrast is low. However, the improvement achieved by mutual information regularization is consistent over these two groups.

4. Conclusions

Can knowing the classification response of a data sample reduce uncertainty about its true class label? In this paper, we proposed this question and tried to answer it by learning an optimal classifier to reduce such uncertainty.

Insighted by the fact that the reduced uncertainty can be measured by the mutual information between classification responses and true class labels, we proposed a new classifier learning algorithm, by maximizing the mutual information when learning the classifier. Particularly, our algorithm adds a maximum mutual information regularization term. We investigated the classification performances when maximum mutual information was used to regularize the classifier learning based on four different loss functions. The experimental results show that the proposed regularization can improve the classification performances of all these four loss function based classifiers. In the future, we will study how to apply the proposed algorithm on large scale dataset based on some distributed big data platforms [62, 63, 64, 65].

References

Figure 5: Experiment results on HEp-2 cell image data set. Please note that “Loss” stands for combination of classification loss and $\ell_2$-regularization, and “Loss - MaxMutInf” stands for combination of classification loss, $\ell_2$ and maximum mutual information-regularization.
svd-based nmr spectra peak picking method, Bioinformatics 25 (12) (2009)
i268–i275.

maximizing correntropy for cancer clustering, BMC Bioinformatics 14 (1)

[6] J. J. Wang, H. Bensmail, X. Gao, Multiple graph regularized protein do-

similarity measures and hierarchical context coherently for protein-protein
comparison in protein database retrieval, BMC bioinformatics 13 (Suppl 7)

[8] Z. Liu, A. Abbas, B.-Y. Jing, X. Gao, Wavpeak: picking nmr peaks through
wavelet-based smoothing and volume-based filtering, Bioinformatics 28 (7)

[9] P. Wang, Intelligent pattern recognition and applications to biometrics in
an interactive environment, in: GRAPP 2009 - Proceedings of the 4th
International Conference on Computer Graphics Theory and Applications,

based on level set method, genetic algorithms and adaptive asymmetrical
svms, Engineering Applications of Artificial Intelligence 24 (3) (2011) 458–
475.

leg and arm movements, Engineering applications of artificial intelligence

[12] J. Yang, B. Payne, M. Hitz, Z. Fei, L. Li, T. Wei, Location aided energy bal-

malicious websites detection, in: 2014 IEEE Conference on Communica-
tions and Network Security (CNS) (IEEE CNS 2014), San Francisco, USA,
2014.

[14] L. Xu, Z. Zhan, S. Xu, K. Ye, Cross-layer detection of malicious websites,
in: Proceedings of the third ACM conference on Data and application

action recognition, in: Multimedia and Expo (ICME), 2013 IEEE Interna-
tional Conference on, 2013, pp. 1–6. doi:10.1109/ICME.2013.6607606.


