Online multiple instance regression*

Wang Zhi-Gang(王志岗)[a], Zhao Zeng-Shun(赵增顺)[b)c)], and Zhang Chang-Shui(张长水)[a)†]

[a)] *Department of Automation, Tsinghua University, State Key Laboratory of Intelligent Technologie and Systems, Tsinghua National Laboratory for Information Science and Technology, Beijing* 100084, *China*

[b)] *College of Information and Electrical Engineering, Shandong University of Science and Technology, Qingdao* 266590, *China*

[c)] *School of Control Science and Engineering, Shandong University, Jinan* 250061, *China*

(Received 16 January 2013; revised manuscript received 8 March 2013)

The multiple instance regression problem has become a hot research topic recently. There are several approaches to the multiple instance regression problem, such as Salience, Citation KNN, and MI-ClusterRegress. All of these solutions work in batch mode during the training step. However, in practice, examples usually arrive in sequence. Therefore, the training step cannot be accomplished once. In this paper, an online multiple instance regression method "OnlineMIR" is proposed. OnlineMIR can not only predict the label of a new bag, but also update the current regression model with the latest arrived bag. The experimental results show that OnlineMIR achieves good performances on both synthetic and real data sets.

Keywords: mutiple instance, regression, online learning

PACS: 87.19.lo, 87.15.Qt **DOI:** 10.1088/1674-1056/22/9/098702

1. Introduction

One shortcoming of traditional supervised learning is that the label information of training examples is not always available. Multiple instance learning (MIL) is a variant version of the supervised learning when some labels are uncertain. This concept was proposed in Ref. [1]. In multiple instance (MI) setting, a set of example-label pairs are provided for training. Each example contains more than one instance (feature vectors). The labels are assigned to bags of instances, which are different from the traditional supervised learning. The goal is to acquire a learner to predict a new bag's label. According to the label value, MIL can be divided into two subfields—classification and regression. When the label is an integer, it becomes a classification problem.[1] If the label takes a real number, it turns into a regression problem.[2] Drug activity prediction[1] is the first application of MIL. More than one structural conformations of the drug molecule are given, but only part of conformations contribute to the molecule label, where the value of label is "active" or "inactive". The contributed confirmations in each molecule are unknown. It is required to design a classifier to predict a new molecule's label.

Many studies of the multiple instance classification can be found in Refs. [3]–[7]. The online version of MI classification[8–10] has also been designed for incrementally arrived MI data. But some applications require not only the class label of the bag, but also the likelihood of membership. This is strongly desirable in some applications, but MI classification has difficulties in satisfying this requirement. For example, if the multiple instance classification is applied to the drug activity prediction,[1,11] only the information on whether the molecule is active or not is available. However, we have no certainty about how active the molecule is. To overcome the drawback of MI classification, MI regression is applied. Besides, in some tasks, the bag label is real-valued, which means that the MI classification cannot work. MI regression has such advantages over MI classification.

In recent years, some studies have been done on the MI regression. The first multiple instance regression algorithm is proposed in Ref. [2]. It takes a linear regression model for granted and picks one primary instance from each bag to get a predictive function. This solution cannot deal with the nonlinear cases and ignores the relevances between instances within the same bag. References [12] and [13] modify kNN classifier and some other traditional classifiers to solve the multiple instance regression. Reference [14] alternates the optimizations on two subproblems to solve this problem. This method cannot predict one bag's label directly, only obtain the label of a single instance. The MI-ClusterRegress[15] is proposed to obtain both the predictive function and the contribution of each instance to the bag label. MI-ClusterRegress can predict the bag label without other prior knowledge and obtain the contribution of each instance to the bag label. These methods share the same one assumption: all training bags would be provided for once. However, in practice, the bags usually arrive in sequence. The most intuitive solution is to discard the current

*Project supported by the China State Key Science and Technology Project on Marine Carbonate Reservoir Characterization (Grant No. 2011ZX05004-003), the National Basic Research Program of China (Grant No. 2013CB329503), the Beijing Municipal Education Commission Science, Technology Development Plan key project, China (Grant No. KZ201210005007), and China Postdoctoral Science Foundation (Grant No. 2012M521336).

†Corresponding author. E-mail: zcs@mail.tsinghua.edu.cn

© 2013 Chinese Physical Society and IOP Publishing Ltd

model and repeat training step on both previous and new coming bags. However, this leads to a high computational cost, which is impractical when the bag number increases dramatically. Therefore, the existing MI regression approaches cannot deal with sequentially arrived training bags efficiently. It is our motivation to propose the online MI regression algorithm "OnlineMIR".

Our proposed OnlineMIR has such advantages. Firstly, our online multiple instance regression algorithm can use the latest arrived bag with its label to update our current predictive model. Secondly, our method can predict the bag label directly. In comparison, many previous solutions can only obtain a single instance's label. Thirdly, our algorithm can obtain the contribution of each instance to the bag label. This characteristics is very useful to analyze the bag structure. The experiments show that our algorithm achieves good performances in both synthetic and real data sets.

2. Model and method

2.1. Problem definition

Here, the multiple instance regression problem is defined as follows. A collection of bags and their labels $\{B_i, y_i\}, i = 1,2,\ldots,N$, arrive at $t = t_1,\ldots,t_i,\ldots,t_N$ respectively, $t_i < t_{i+1}$, $i < N$, N denotes the number of B_i, B_i contains N_i instances and B_{ij} denotes the j-th instance of B_i, B_{ij} is a d-dimensional feature vector and y_i is the real-valued true label of B_i. We use $\widehat{y}_i$ to indicate the predicted label of B_i by the regression model. The predictive model f_n is obtained from training example pairs (B_i, y_i), $i = 1,2,\ldots,n$, $n < N$. Our task is to improve the current predictive function f_n to f_{n+1} based on the latest arrived bag B_{n+1} and its label y_{n+1}.

2.2. OnlineMIR

Online single instance regression[16] has been studied carefully in Ref. [17], in which the online support vector regression(SVR) problem is proposed. When a new feature vector and its label arrive, they can update the current predictive model. Online SVR works with steps similar to that of online SVM.[18]

Our idea is to reduce the online multiple instance regression to an online single instance regression problem, by using the existing online SVR method.[17] However, the new arrived example is a bag instead of a single instance, where online SVR cannot be applied directly. The intuitive solution is to decompose the instances in the latest arrived bag and use the online SVR on each instance. Reference [15] has demonstrated that this behavior got a bad result. Reference [12] assumes that the bag label are only decided by one instance in each bag and is independent of others. However, there are more complicated cases than this assumption. More than one instance usually are associated with a bag label and each instance has different contributions. If we update the regression model incrementally with the instances, the differences between their contributions to the bag label will be ignored. Therefore, we try to convert each bag into an virtual instance. We use b_i to denote the virtual instance converted from bag B_i. Here, a coefficient vector is applied to convert instances in one bag into a virtual instance

$$\begin{aligned} &b_i = B_i \cdot \lambda_i, \\ &\lambda_i \in \mathbb{R}^{N_i},\ 0 \le \lambda_i^k \le 1,\ \sum_{k=1}^{N_i} \lambda_i^k = 1, \\ &B_i \in \mathbb{R}^{N_i \times d}. \end{aligned} \tag{1}$$

Here, λ_i is the coefficient vector that converts B_i to the virtual single instance b_i, λ_i^k is the k-th element of λ_i. The value of λ_i^k indicates the contribution of B_{ik} to the label y_i.

If λ_i is known, this problem has been solved. In this way, the most important problem is how to get λ_i. In most cases, not every instance in a bag contribute much to the bag label. Some instances have relatively little effect on the label value. When a new bag and its label arrive, the current regression model is used to predict the label of each instance. We assume that if the difference between the predicted label of an instance in one bag and the corresponding bag label is small, this instance has more effects on the bag label and vice versa. Therefore, the label difference

$$\mathrm{dif}(B_{ij}) = |f(B_{ij}) - y_i| \tag{2}$$

becomes an indicator that measures the effect of instance B_{ij} on the bag label y_i. The function $f(x)$ denotes the predictive model. Equation (2) is applied to divide instances into two clusters. The first cluster contains instances with large influence on the bag label. The second cluster has almost nothing to do with the bag label. Now, a bag with M instances $\{x_i\}$, $i = 1,\ldots,M$ arrives. We sorts the label differences $\mathrm{dif}(x_i), i = 1,\ldots,M$ into an ascend sequence $\{a_1,\ldots,a_M\}$, $a_i \le a_{i+1}$, $i = 1,\ldots,M-1$, with $\{x_i\}$ a set of instances. Then, the next step is to solve the optimization problem

$$\begin{aligned} &\min_L \frac{v_1 + v_2}{|m_1 - m_2|}, \\ &m_1 = \frac{1}{L}\sum_{i=1}^{L} a_i,\ \ m_2 = \frac{1}{M-L}\sum_{i=L+1}^{M} a_i, \\ &v_1 = \frac{1}{L}\sum_{i=1}^{L}(a_i - m_1)^2,\ \ v_2 = \frac{1}{M-L}\sum_{i=L+1}^{M}(a_i - m_2)^2, \\ &1 < L < M,\ \ L \in \mathbb{Z}^+. \end{aligned} \tag{3}$$

Here, L is a positive integer, m_1 and v_1 are the mean and variance of the first L label differences respectively, m_2 and v_2 are the mean and variance of the remaining $M-L$ differences. We can compute the objective function value of Eq. (3) for all L's

candidate values and select the number that minimizes the objective function of Eq. (3). The first L instances dominate the bag label value. However, the remaining M–L instances have less effect on the label.

We assume that the instances with much contribution to the bag label come from a normal distribution $\mathcal{N}(\mu,\Sigma)$, where, μ is the mean and Σ is covariance matrix, respectively. Given a set of instances $\{x_i\}$, $i=1,\ldots,M$, μ and Σ can be estimated as

$$\begin{aligned}\hat{\mu} &= \frac{1}{R}\sum_{i=1}^{R} x_i,\\ \hat{\Sigma} &= \frac{1}{R}\sum_{i=1}^{R}(x_i-\mu)^{\mathrm{T}}(x_i-\mu).\end{aligned} \tag{4}$$

Here, R is the number of instances contributing much to the bag label, μ and Σ in $\mathcal{N}(\mu,\Sigma)$ need updating as the new bags arrive. Assume that the newly arrived bag contains only one instance denoted as x_{new}. Given the normal distribution $\mathcal{N}(\mu,\Sigma)$ currently and new instance x_{R+1}, μ is updated as

$$\mu_{\mathrm{new}} = \frac{1}{R+1}(\mu_{\mathrm{old}}\times R + x_{R+1}), \tag{5}$$

μ_{old}, μ_{new} are the mean values of R, $R+1$ instances respectively, Σ_{old} and Σ_{new} are the covariance matrixes of R, $R+1$ instances respectively, $x(i)$ is the i-th element of the vector x. According to the definition of covariance, the element $\Sigma_{\mathrm{old}}(i,j)$ at i-th row and j-th column of Σ_{old} is

$$\begin{aligned}\Sigma_{\mathrm{old}}(i,j) &= \frac{1}{R}\sum_{k=1}^{R}\big[x_k(i)-\mu_{\mathrm{old}}(i)\big]\times\big[x_k(j)-\mu_{\mathrm{old}}(j)\big]\\ &= \frac{1}{R}\sum_{k=1}^{R}\big[x_k(i)x_k(j)-\mu_{\mathrm{old}}(i)x_k(j)-\mu_{\mathrm{old}}(j)x_k(i)\\ &\quad+\mu_{\mathrm{old}}(i)\mu_{\mathrm{old}}(j)\big]\\ &= \frac{1}{R}\sum_{k=1}^{R}\big[x_k(i)x_k(j)\big]-\mu_{\mathrm{old}}(i)\mu_{\mathrm{old}}(j)-\mu_{\mathrm{old}}(j)\mu_{\mathrm{old}}(i)\\ &\quad+\mu_{\mathrm{old}}(i)\mu_{\mathrm{old}}(j),\\ \Sigma(i,j) &= \frac{1}{R}\sum_{k=1}^{R}x_k(i)x_k(j)-\mu_{\mathrm{old}}(i)\mu_{\mathrm{old}}(j),\\ \Sigma_{\mathrm{old}} &= \frac{1}{R}\sum_{k=1}^{R}x_kx_k^{\mathrm{T}}-\mu_{\mathrm{old}}\mu_{\mathrm{old}}^{\mathrm{T}}.\end{aligned} \tag{6}$$

From Eq. (6), Σ is updated as

$$\begin{aligned}\Sigma_{\mathrm{new}}(i,j) &= \frac{1}{R+1}\sum_{k=1}^{R+1}x_k(i)x_k(j)\\ &\quad-\mu_{\mathrm{new}}(i)\mu_{\mathrm{new}}(j)\\ &= \frac{1}{R+1}\left[\sum_{k=1}^{R}x_k(i)x_k(j)+x_{\mathrm{new}}(i)x_{R+1}(j)\right]\\ &\quad-\mu_{\mathrm{new}}(i)\mu_{\mathrm{new}}(j)\\ &= \frac{R}{R+1}\left[\Sigma_{\mathrm{old}}(i,j)+\mu_{\mathrm{old}}(i)\mu_{\mathrm{old}}(j)\right]\\ &\quad+\frac{1}{R+1}x_{R+1}(i)x_{R+1}(j)-\mu_{\mathrm{new}}(i)\mu_{\mathrm{new}}(j),\\ \Sigma_{\mathrm{new}} &= \frac{R}{R+1}\left(\Sigma_{\mathrm{old}}+\mu_{\mathrm{old}}\mu_{\mathrm{old}}^{\mathrm{T}}\right)\\ &\quad+\frac{1}{R+1}x_{R+1}x_{R+1}^{\mathrm{T}}-\mu_{\mathrm{new}}\mu_{\mathrm{new}}^{\mathrm{T}}.\end{aligned} \tag{7}$$

It can be seen from Eq. (7) that only three parameters μ,Σ, and the instance number R are required to update the distribution $\mathcal{N}(\mu,\Sigma)$.

If there are r instances to update the distribution $\mathcal{N}(\mu,\Sigma)$ in new arrived bag B_{n+1}, μ, and Σ are updated as

$$\mu_{n+1} = \frac{1}{R+r}(\mu_n\times R+\sum_{i=R+1}^{R+r}x_i), \tag{8}$$

$$\begin{aligned}\Sigma_{n+1} &= \frac{R}{R+r}\left(\Sigma_n+\mu_n\mu_n^{\mathrm{T}}\right)\\ &\quad+\frac{1}{R+r}\sum_{i=R+1}^{R+r}x_ix_i^{\mathrm{T}}-\mu_{n+1}\mu_{n+1}^{\mathrm{T}}.\end{aligned} \tag{9}$$

Here, $x_{R+1},\ldots,x_{R+r}$ are r instances in the new bag B_{n+1} contributing much to the bag label, μ_n, Σ_n, and μ_{n+1}, Σ_{n+1} are the parameters before and after the bag B_{n+1} arrives.

Now, λ_i is obtained from the following normal distribution:

$$\lambda_i(j) = \frac{1}{\sqrt[d/2]{2\pi\Sigma}}\exp\left[-\frac{1}{2}(B_{ij}-\mu)^{\mathrm{T}}\Sigma^{-1}(B_{ij}-\mu)\right]. \tag{10}$$

To satisfy the condition $\sum_{j=1}^{N_i}\lambda_i(j)=1$, the normalization operation is needed.

In the following step, we discuss how to get the initial regression function. Any one static multiple instance regression method can be used for initialization. In the following experiments, MI-ClusterRegress[15] is used to get the initial regression function. Equation (3) will select instances from initially arrived bags to obtain current $\mathcal{N}(\mu,\Sigma)$. This initialization step is described as Algorithm 1.

The online training algorithm is summarized as Algorithm 2. For a bag B with an unknown label, we obtain its λ by Eq. (10) and normalize it. We predict the label of B is $f(B\cdot\lambda)$. The running time of Algorithm 2 is dominated by the arrived bag B_i. It can be found that the running time of online SVR is independent of B_i. The λ needs N_i Gaussian function computations. The optimization of Eq. (3) requires N_i-1 computations of the objective function. Therefore, the time complexity of Algorithm 2 is $\mathcal{O}(N_i)$ if the new bag B_i arrives. If N bags that contain totally N_s instances arrive continuously, the time complexity becomes $\mathcal{O}(N_s-N)$. The linear time complexity ensures that OnlineMIR can deal with the latest arrived bags quickly.

Algorithm 1 Initialization step for OnlineMIR

Input: initial predictive function $f_0(x)$, training instance set D for $f_0(x)$

1 Extract the dif function value on each instance in D by Eq. (2).

2 Solve Eq. (3) to get the instances which contribute a lot to their bag labels.

3 Compute μ and Σ by Eq. (4).

Output: normal distribution $\mathcal{N}(\mu,\Sigma)$ and instance number m

Algorithm 2: OnlineMIR

Input: the current predictive function $f_n(x)$, the latest arrived bag B_{n+1} and its label y_{n+1}

1 Extract the feature of each instance in bag B_{n+1} by Eq. (2).

2 Solve Eq. (3) to get the instances $\{x_i,\ i=1,\ldots,M\}$ with large contributions on their bag labels

3 Add instances $\{x_i\}$ to $\mathcal{N}(\mu,\Sigma)$; update parameter μ and Σ by Eqs. (8) and (9).

4 Compute λ_{n+1} by Eq. (10) for B_{n+1}

5 $\lambda_{n+1}=\dfrac{\lambda_{n+1}}{||\lambda_{n+1}||_1}$.

6 Set $b_{n+1}=B_{n+1}\lambda_{n+1}$ and input b_{n+1} to the onlineSVR algorithm.

Output: predictive function $f_{n+1}(x)$, normal distribution $\mathcal{N}(\mu,\Sigma)$ and instance number n

3. Results and discussion

In this section, we will show the performances of OnlineMIR both on synthetic and real data sets. The toolkit OnlineSVR[17] is used to realize the online single instance regression. MI-ClusterRegress[15] is applied to initialize the online algorithm both on synthetic and real data sets.

3.1. Experiments on a synthetic data set

In this section, two regression models are used to validate our method. In the first model, a one-dimensional linear regression model

$$y=x \tag{11}$$

is the ground truth for convenience. We set the variable with twenty values of $x=1,2,3,\ldots,20$. For each x value, 10 random numbers are generated subject to normal distribution $\mathcal{N}(x,5)$ to form one bag. Therefore, there are 20 bags with 10 instances in each one. Here, only the first three bags with $y=1,2,3$ are used to initialize OnlineMIR. The remaining bags will be added one by one with their labels. The true regression function and the result of our method are shown in Fig. 1. The comparison with the true regression function should be more convincing than with other methods. To our knowledge, OnlineMIR is the first algorithm to solve the online multiple instance regression problem and no similar method for this new problem can be used for comparison. We find that the result of OnlineMIR gets close to the ground truth $y=x$ mostly. When the value of x is small (for example $x\in[1,10]$), our result and the ground truth line overlap. As x increases, the result of OnlineMIR begins to deviate to the ground truth line slightly. But the difference is very small. Generally speaking, such error is acceptable.

The second regression model is

$$y=0.5x^2+x-2. \tag{12}$$

The settings are the same as the first data set. Figure 2 shows the result of OnlineMIR on this synthetic data set. The result of OnlineMIR gets close to the ground truth mostly though a very small deviation appears with x increasing.

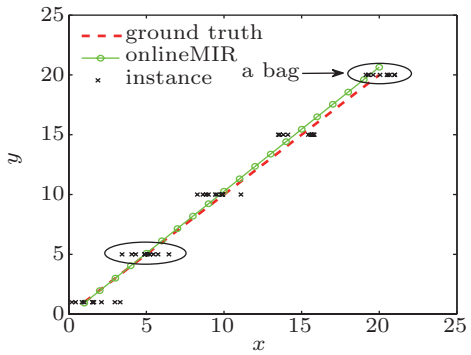


Fig. 1. (color online) Comparison between the ground truth $y=x$ and the result of OnlineMIR. Each bag's instances stand horizontally. Two bags are enclosed by a black ellipse curve.

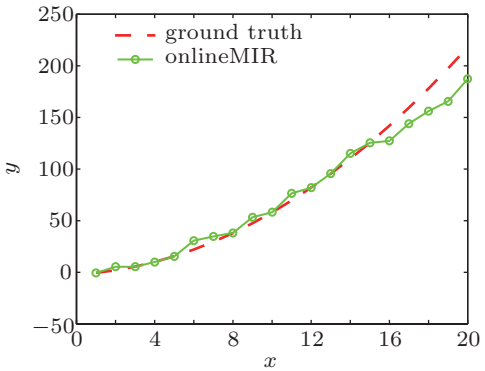


Fig. 2. (color online) Comparison between the ground truth $y=0.5x^2+x-2$ and the OnlineMIR result.

3.2. Experiments on a real data set

In this section, a real data set is used to show the performances of OnlineMIR. This data set comes from the USDA National Agricultural Statistics Service (NASS),[19] which contains many remote sensing observations from satellites.

The task is to predict crop yield before the growing season according to these observations. The original data can be used directly and are transformed into an index named "NDVI (Normalized Difference Vegetation Index)". NDVI is a real number. More details about NDVI chould be found in Refs. [15] and [19]. Each feature vector contains 46 NDVIs, which are transformed from the observations within an area of one county in California state. One bag contains the NDVI vectors from one county. NASS records observations from each area of all California counties and provides the observations during five years from 2001 to 2005. Here, observations from 2001 to 2003 are used to be the training set. We rearrange these bags in line for online training according to thte bag ID and time. The observations from 2004 to 2005 are the testing set. There are two tasks, which predict corn and wheat yields in every counties, respectively.

Table 1. The number of the training and testing bags in corn and wheat predictions.

	Crop	Wheat
Training bag number	59	69
Testing bag number	23	38

Figures 3 and 4 show the results of OnlineMIR for crop and wheat yields predictions. Meanwhile, the results from MI-ClusterRegress (a state-of-the-art batch-mode multiple instance regression algorithm) are provided for comparison. For the yield value varying within a large range, we apply the relative error ratio (R_{re}) to compare these two methods

$$R_{\mathrm{re}} = \frac{|y-\hat{y}|}{y}. \tag{13}$$

The parameters in OnlineMIR and MI-ClusterRegress are acquired by five-fold cross validation. The initial predictive function $f_0(x)$ in Algorithm 1 is the output of MI-ClusterRegress on the first 5 training bags. Based on this initial predictive function, we insert bags one by one using OnlineMIR to get the red curves in Figs. 3 and 4. Because the true regression functions are unknown, the regression curves (the blue ones in Figs. 3 and 4) of MI-ClusterRegress on all training bags are provided for comparison with OnlineMIR results.

From Fig. 3, we can see that the highest R_{re} value is about 0.18. Most R_{re} values locate under or around 0.1. The result of OnlineMIR gets close to that of MI-ClusterRegress. The largest difference between them is smaller than 0.01 from the 10th bag. In most testing bags, R_{re} of OnlineMIR is little lower than that of MI-ClusterRegress. However, OnlineMIR gets slight improvement on several relative higher R_{re} from MI-ClusterRegression. Figure 4 shows similar trends for the wheat yield prediction. However, both methods obtain higher R_{re} values than the crop yield prediction. Seven bags get R_{re} values larger than 0.4. For these bags, OnlineMIR acquires slightly lower R_{re} values than MI-ClusterRegress.

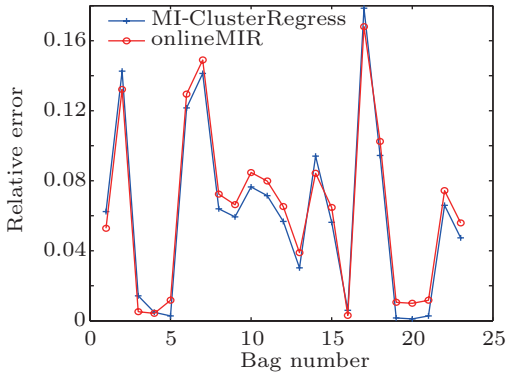


Fig. 3. (color online) Comparison between OnlineMIR and MI-ClusterRegress for the corn yield prediction.

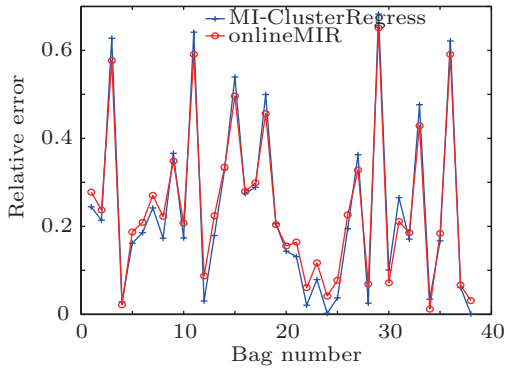


Fig. 4. (color online) Comparison between OnlineMIR and MI-ClusterRegress for the wheat yield prediction.

However, for other testing bags, MI-ClusterRegress behaves better.

We assume that the performance difference between the two methods is caused by two reasons. Firstly, OnlineMIR magnifies the errors accumulated previously with a high likelihood when inserting new bag(s) to the current prediction model. Secondly, OnlineMIR divides the instances in each bag into two parts: one part contributes to the bag label and the other part does not. In comparison, MI-ClusterRegress usually divides each bag into more than two clusters. Hence, more instances in each bag contribute the bag label in OnlineMIR usually. When MI-ClusterRegress ingnores some instances that contribute much to the bag label, OnlineMIR achieves improvement. Otherwise, OnlineMIR has a slightly lower performance. The average time of one bag insertion on crop and wheat data set is 0.366 second, 0.439 second, respectively.

OnlineMIR requires an initial regression function, which is obtained with MI-ClusterRegress on the first 5 training bags. The initial MI-ClusterRegress needs a parameter k, which is the number of Gaussian components. We set $k = 2,3,4,\ldots,10$ to get nine R_{re} values (Eq. (13)) $\{l_k,\ k = 2,\ldots,10\}$ on crop and wheat predictions, l_k contains the predicted R_{re} values on all testing bags. We compute

$$l_{\mathrm{mean}} = \frac{1}{9}\sum_{k=2}^{10} l_k. \tag{14}$$

And $\bar{R}_{\rm re}$

$$\bar{R}_{\rm re} = \frac{1}{\#\{l_k\}} \|l_k - l_{\rm mean}\|_1 \tag{15}$$

is used to measure the effect of k on the result of OnlineMIR. The $\bar{R}_{\rm re}$ denotes the mean deviation of $R_{\rm re}$ on all testing bags. In Eq. (15), $\#\{l_k\}$ is the length of l_k, which is also the number of testing bags. In crop and wheat data sets, $\#\{l_k\} = 23,\ 38$ respectively. Figure 5 shows the effects of k value in initial MI-ClusterRegress on predicted $R_{\rm re}$ results of OnlineMIR on both crop and wheat predictions. Whatever value k takes, $\bar{R}_{\rm re}$ keeps below about 0.02 and 0.05 for crop and wheat predictions, respectively. It indicates that the effect of the k value on the predicted result is very small.

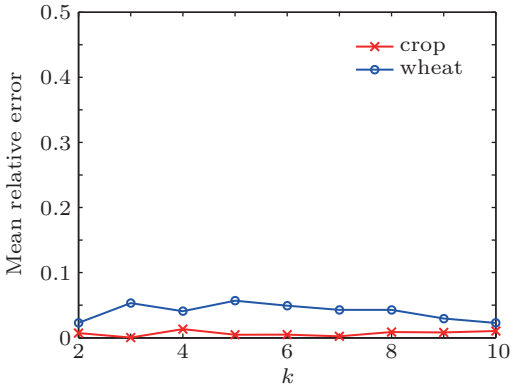


Fig. 5. (color online) Effect of k in the initial MI-ClusterRegress of OnlineMIR.

4. Conclusion

This paper proposes an online multiple instance regression algorithm "OnlineMIR". Given a current regression model, we construct a normal distribution model to describe the instances with high effect on the bag label. When a new bag with its label arrives, it can be used to improve the current regression model. For one bag with an unknown label, the bag label can be obtained without prediction on each instance's label. In addition, the contribution of each instance to the bag can be obtained by using our algorithm. The experiments show that OnlineMIR achieves results compared to the art-of-state batch-mode method MI-ClusterRegress. We will try to improve the optimization strategy in Eq. (3) in our future work. In addition, we will also design better methods to find those instances with a larger contribution to the bag label for better performance.

References

[1] Dietterich T G, Lathrop R H and Lozano-Pérez T 1997 *Artif. Intell.* **89** 31
[2] Ray S and Page D 2001 *Workshop of International Conference on Machine Learning*, June 28–July 1, 2001, Williamstown, USA, pp. 425–432
[3] Andrews S, Tsochantaridis I and Hofmann T 2003 *Advances in Neural Information Processing Systems*, December 9, Vancouver, Canada, pp. 577–584
[4] Chen Y X and Wang J Z 2004 *J. Mach. Learn. Res.* **5** 939
[5] Chen Y X, Bi J and Wang J Z 2006 *IEEE Trans. Pattern Anal. Mach. Intell.* **28** 1931
[6] Fung G, Dundar M, Krishnapuram B and Rao R B 2007 *Advances in Neural Information Processing Systems*, December 4, Vancouver, Canada, pp. 425–433
[7] Zhou Z H, Sun Y Y and Li Y F 2009 *Proceedings of the* 26*th Annual International Conference on Machine Learning*, June 14–18, Montreal, Canada, pp. 1249–1256
[8] Babenko B, Yang M H and Belongie S 2011 *IEEE Trans. Pattern Anal. Mach. Intell.* **33** 1619
[9] Li M, Kwok J T and Lu B L 2010 *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, June 13–18, San Francisco, USA, pp. 1395–1401
[10] Zeisl B, Leistner C, Saffari A and Bischof H *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, June 13–18, 2010, San Francisco, USA, pp. 1879–1886
[11] Zhou Z H 2010 *Signal Process* **2** 2
[12] Amar R A, Dooly D R, Goldman S A and Zhang Q 2001 *Workshop of International Conference on Machine Learning*, June 28–July 1, Williamstown, USA, pp. 3–10
[13] Dooly D R, Zhang Q, Goldman S A and Amar R A 2003 *J. Mach. Learn. Res.* **3** 651
[14] Wagstaff K L and Lane T 2007 *Workshop of International Conference on Machine Learning*, June 20–24, 2007, Corvallis, USA, pp. 444–451
[15] Wagstaff K L, Lane T and Roper A 2008 *Workshop of IEEE International Conference on Data Mining*, December 15–19, 2008, Pisa, Italy, pp. 291–300
[16] Sun Z H and Jiang F 2010 *Chin. Phys. B* **19** 110502
[17] Parrella F 2007 *A Thesis Presented for the Degree of Information Science*
[18] Cauwenberghs G and Poggio T 2001 *Advances in Neural Information Processing Systems*, December 3, 2001, Vancouver, Canada, pp. 409–415
[19] Galford G L, Mustard J F, Melillo J, Gendrin A, Cerri C C and Cerri C E P 2008 *Remote Sens. Environ.* **112** 576
[20] Meng Q F, Chen Y H and Peng Y H 2009 *Chin. Phys. B* **18** 2194