Instance- and Bag-Level Manifold Regularization for Aggregate Outputs Classification*

Shuo Chen, Bin Liu, Mingjie Qian, Changshui Zhang
State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Automation, Tsinghua University, Beijing 100084, China
{chenshuo07, liubin07} @mails.tsinghua.edu.cn
qian.mingjie@gmail.com, zcs@mail.tsinghua.edu.cn

ABSTRACT

Aggregate outputs learning differs from the classical supervised learning setting in that, training samples are packed into bags with only the aggregate outputs (labels for classification or real values for regression) known. This setting of the problem is associated with several kinds of application background. We focus on the aggregate outputs classification problem in this paper, and set up a manifold regularization framework to deal with it. The framework can be of both instance level and bag level for different testing goals. We propose four concrete algorithms based on our framework, each of which can cope with both binary and multi-class scenarios. The experimental results on several datasets suggest that our algorithms outperform the state-of-art technique.

Categories and Subject Descriptors

H.3.m [**Information Storage and Retrieval**]: Miscellaneous—*Learning*; I.2.6 [**Artificial Intelligence**]: Learning—*Concept Learning*; I.5.2 [**Pattern Recognition**]: Design Methodology —*Classifier design and evaluation*

General Terms

Algorithms, Experimentation, Theory

Keywords

Aggregate Outputs Classification, Manifold Regularization

1. INTRODUCTION

The concept of aggregate outputs learning (AOL) was first proposed by [4]. It is a problem elicited from the application of analysis of single particle mass spectrometry (SPMS) data. The goal is training to predict quantity of *black carbon* (BC) for a certain given single particle. However, the available instrument is not precise enough and can only measure the aggregated BC (sum of BC) for bags of training particles. The problem becomes how to make use of the training samples and the aggregated BC to train an accurate predictor for single particle. The SPMS problem is of regression nature, and the authors of [4] also extended it to the classification setting which is what we concern in this paper. It is different from classical supervised classification in that, the training samples are packed into several bags, and only aggregated labels, which indicate the numbers of samples from different classes in each bag, are provided. We show an example of training dataset for this setting in Table 1.

Bag ID	Height	Weight	Aggregated Label
1	169cm	60kg	2 F, 3 M
	160cm	50kg	
	185cm	95kg	
	175cm	70kg	
	155cm	45kg	
2	170cm	50kg	1 F, 2 M
	188cm	100kg	
	162cm	65kg	

Table 1: An example of training dataset for aggregate outputs classification. There are 8 instances, each with two features of height and weight of a person. The instances are packed into 2 bags. The aggregated labels show how many males and females in each bag.

There is scarce literature of AOL. The original work of [4] is the only one that strictly follows the setting as far as we know. In [4], the authors adapt three classical supervised learning algorithms to solve the aggregate outputs learning problem, namely k-nearest-neighbor (kNN), artificial neural network (ANN) and support vector machine (SVM). Both classification and regression versions of the algorithm are provided. We are only interested in solving the aggregate outputs classification (AOC) problem in this paper, and we denote the three classification algorithms as AOC-kNN, AOC-ANN and AOC-SVM.

In this paper, we set up a manifold regularization framework [2] to cope with the AOC problem, both instance- and bag-level versions of algorithm are proposed. We aim to use

*This work is supported by NSFC (Grant No. 60835002 and No. 60675009).

CIKM'09, November 2–6, 2009, Hong Kong, China.

a function in a Reproducing Kernel Hilbert Space (RKHS) to predict the label of any sample or the aggregated label of any bag. Our algorithms can serve the binary classification as well as the multi-class scenario, which is not covered in [4] in detail. The experimental results are promising.

2. PROBLEM FORMULATION

For simplicity, we first introduce in the binary classification scenario, where each sample belongs to one of the two classes. Suppose we are given a training dataset with n samples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. $\mathbf{x}_i$ is the vector representation of the ith sample. We assume that each of the samples $\mathbf{x}_i$ is from the same input space χ and $\chi \in \mathbb{R}^d$. We also assume that each $\mathbf{x}_i$ is associated with a hidden y_i that takes the value of ± 1 for two different classes. The label space is denoted by ψ. However, these labels cannot be observed directly. The n samples are aggregated into m disjunct bags, and an $m \times n$ aggregating matrix A is provided. The element of A is given by

$$A_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_j \text{ belongs to the } i\text{th bag} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

We denote the m bags as $\mathscr{B}_1, \mathscr{B}_2, \ldots, \mathscr{B}_m$, with each $\mathscr{B}_i \in \mathfrak{B}$. $\mathfrak{B}$ is the space of the bags. We are also provided with an m-dimensional column aggregated labels vector b, with its each element b_i equaling the sum of the hidden labels associated with the samples in the ith bag. For example, given the ith bag containing 5 positive samples and 3 negative samples, the value of b_i is 2. Our goals for the AOC problem can be both transductive and inductive. For the transductive goal, we hope to give the label of each training $\mathbf{x}_i$ that is as close to y_i as possible. Very often, our inductive goal is simplified to get a good predictive accuracy on a test dataset with known labels that is from the same distribution as that of the training set.

The above goals are of instance level, which means that our trained classifier is supposed to act on single instance. We also propose to train bag-level classifiers in this paper. Our bag-level goal can only be inductive, since all the aggregated labels of the training bags are given. We thus expect our bag-level classifier to give as accurate aggregated labels on test bags as possible.

Moreover, the bag-level classifier can also serve the goal of instance classification. All we need is to deem the instances for testing as bags with only one instance. As shown in our experiments, they also yield competitive results of error rate.

When considering the case of multi-class classification, we only need to add a modification to the formulation mentioned above. We suppose c classes are taken into account in the problem, where c is an integer and $c \geq 3$. Rather than a scalar of value ± 1, any of the hidden labels is now represented by a c-dimensional 0-1 vector $\mathbf{y}$ with its ith element indicating whether the associated sample belongs to the ith class or not. The vector b is replaced by an $m \times c$ matrix B with B_{ij} equaling the number of the samples in the ith bag that belongs to the jth class.

3. INSTANCE-LEVEL MANIFOLD REGULARIZATION FOR AOC

We start with binary classification case first. After choosing a Mercer kernel $K : \chi \times \chi \to \mathbb{R}$, an RKHS ensues. We denote it as $\mathcal{H}_K$ and it is a space of functions $\chi \to \mathbb{R}$. There is also an associated norm $\|\cdot\|_K$ for $\mathcal{H}_K$. (One can refer to [5] for more details of Mercer kernel, RKHS and the associated norm). We thus want to learn a soft label function $f \in \mathcal{H}_K$ so that

$$f^* = \arg\min_{f \in \mathcal{H}_K} L(A, b, f) + \gamma_1 \|f\|_K^2 + \gamma_2 \|f\|_I^2 \quad (2)$$

where $L(A, b, f)$ is the loss function concerning the aggregate outputs. $\|f\|_K^2$ is the square of norm in $\mathcal{H}_K$. It penalizes the complexity of f in $\mathcal{H}_K$. $\|f\|_I^2$ is a term that penalizes the unsmoothness of f on the given samples. γ_1 and γ_2 are two positive parameters that control the tradeoff between the terms. We explain each of the three terms as follows.

For the convenience in the later optimization process, we use two kinds of loss function $L(A, b, f)$ in this paper

$$L_1(A, b, f) = \sum_{i=1}^{m} U_\epsilon((A\hat{f})_i - b_i) \quad (3)$$

$$L_2(A, b, f) = (A\hat{f} - b)^T (A\hat{f} - b) \quad (4)$$

where

$$U_\epsilon(t) = H_{-\epsilon}(t) + H_{-\epsilon}(-t) \quad (5)$$

$$H_\theta(t) = \max(0, \theta - t) \quad (6)$$

$H_\theta(t)$ is the widely used hinge loss function and $L_1(A, b, f)$ can be comprehended as a vector version of the double sided hinge loss. $L_2(A, b, f)$ is a quadratic loss.

By manipulating the expanded version of representor theorem in [2], the learned function f shall have the form of

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}) \quad (7)$$

As a result,

$$\|f\|_K^2 = \alpha^T \mathcal{K} \alpha \quad (8)$$

where $\mathcal{K}$ is the kernel matrix defined on the training dataset. $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$ is the vector of representing weights. Using the vector form one could write

$$\hat{f} = \mathcal{K}\alpha \quad (9)$$

where $\hat{f}$ is the vector of the values by imposing f on the training samples.

$\|f\|_I^2$ term is used to penalize the unsmoothness of function f. We use

$$\|f\|_I^2 = \hat{f}^T \mathcal{R} \hat{f} \quad (10)$$

where $\mathcal{R}$ is the graph Laplacian regularizer ([2]).

Thus, when using L_1 as loss function, (2) becomes the following optimization problem

$$\min_\alpha \sum_{i=1}^{m} U_\epsilon((A\mathcal{K}\alpha)_i - b_i) + \gamma_1 \alpha^T \mathcal{K} \alpha + \gamma_2 \alpha^T \mathcal{K}\mathcal{R}\mathcal{K}\alpha \quad (11)$$

when L_2 is used, it becomes

$$\min_\alpha (A\mathcal{K}\alpha - b)^T (A\mathcal{K}\alpha - b) + \gamma_1 \alpha^T \mathcal{K} \alpha + \gamma_2 \alpha^T \mathcal{K}\mathcal{R}\mathcal{K}\alpha \quad (12)$$

We denote the two algorithms corresponding to (11) and (12) as AOC-manifold-il-L1 and AOC-manifold-il-L2.

For the optimization problem (11), it can be reformulated as

$$\min_{\alpha,\eta} \sum_{i=1}^{m} \eta_i + \alpha^T(\gamma_1\mathcal{K} + \gamma_2\mathcal{K}\mathcal{R}\mathcal{K})\alpha \quad (13)$$

subject to

$$\eta_i \geq 0, \forall i = 1, 2, \ldots, m \quad (14)$$

$$-\epsilon - \eta_i \leq (A\mathcal{K}\alpha - b)_i \leq \epsilon + \eta_i, \forall i = 1, 2, \ldots, m \quad (15)$$

This is a standard quadratic optimization setting and can be solved efficiently.

The solution of the optimization problem (12) can be written in an analytical form

$$\alpha = (\mathcal{K}A^T A\mathcal{K} + \gamma_1\mathcal{K} + \gamma_2\mathcal{K}\mathcal{R}\mathcal{K})^\dagger \mathcal{K}A^T b \quad (16)$$

Now, let us consider the multi-class classification scenario. We need to modify our formulation (2) to

$$\begin{aligned} F &= [f_1, f_2, \ldots, f_c] \\ &= \underset{f_i \in \mathcal{H}_K, \forall i=1,2,\ldots,c}{\arg\min} \Big\{ L(A, B, F) \\ &\quad + \gamma_1 \sum_{i=1}^{c} \|f_i\|_K^2 + \gamma_2 \sum_{i=1}^{c} \|f_i\|_I^2 \Big\} \end{aligned} \quad (17)$$

Similar with (3) and (4), we adapt the two loss functions as

$$L_1(A, B, F) = \sum_{i=1}^{m}\sum_{j=1}^{c} U_\epsilon((A\hat{F})_{ij} - B_{ij}) \quad (18)$$

$$L_2(A, B, F) = tr\big((A\hat{F} - B)^T(A\hat{F} - B)\big) \quad (19)$$

By using the representor theorem, the optimization problems using the two different loss functions become

$$\min_{\alpha} \sum_{i=1}^{m}\sum_{j=1}^{c} U_\epsilon((A\mathcal{K}\alpha)_{ij} - B_{ij}) + \gamma_1\, tr(\alpha^T\mathcal{K}\alpha) + \gamma_2 tr(\alpha^T\mathcal{K}\mathcal{R}\mathcal{K}\alpha) \quad (20)$$

$$\begin{aligned} \min_{\alpha} tr((A\mathcal{K}\alpha - B)^T(A\mathcal{K}\alpha - B)) + \gamma_1 tr(\alpha^T\mathcal{K}\alpha) \\ + \gamma_2 tr(\alpha^T\mathcal{K}\mathcal{R}\mathcal{K}\alpha) \end{aligned} \quad (21)$$

These two optimization problems can be solved similarly to the binary case, and we omit the details for briefness.

4. BAG-LEVEL MANIFOLD REGULARIZATION FOR AOC

For bag-level classifier in binary classification case, we want to learn a function g

$$g = \underset{g \in \mathcal{H}_{K_b}}{\arg\min}\, L(A, b, g) + \gamma_1\|g\|_{K_b}^2 + \gamma_2\|g\|_I^2 \quad (22)$$

Where $K_b : \mathfrak{B} \times \mathfrak{B} \to \mathbb{R}$ is a Mercer kernel. $\mathcal{H}_{K_b}$ is the associated RKHS with a norm $\|\cdot\|_{K_b}$. Similar with the instance-level scenario, we can use two different loss functions

$$L_1(A, b, g) = \sum_{i=1}^{m} U_\epsilon(g_i - b_i) \quad (23)$$

$$L_2(A, b, g) = (g - b)^T(g - b) \quad (24)$$

The two optimization problems guided by representor theorem can be written as

$$\min_{\beta} \sum_{i=1}^{m} U_\epsilon((\mathcal{K}_b\beta)_i - b_i) + \gamma_1\beta^T\mathcal{K}_b\beta + \gamma_2\beta^T\mathcal{K}_b\mathcal{R}_b\mathcal{K}_b\beta \quad (25)$$

$$\min_{\beta} (\mathcal{K}_b\beta - b)^T(\mathcal{K}_b\beta - b) + \gamma_1\beta^T\mathcal{K}_b\beta + \gamma_2\beta^T\mathcal{K}_b\mathcal{R}_b\mathcal{K}_b\beta \quad (26)$$

Algorithms associated with (25) and (26) will be mentioned as AOC-manifold-bl-L1 and AOC-manifold-bl-L2. Here $\mathcal{K}_b$ is the kernel matrix on the training bags. We use the set kernel from [3] and write

$$\mathcal{K}_b = A\mathcal{K}A^T \quad (27)$$

where $\mathcal{K}$ is the instance-level kernel matrix we used before. $\mathcal{R}_b$ can be calculated similarly.

The solutions of (25) and (26) can be given by an optimization problem and an analytical form, similarly to case of instance-level classifier.

The bag-level classifiers for the multi-class scenario can be constructed analogously. We omit the derivation here due to the lack of space.

5. EXPERIMENTS AND DISCUSSION

In this section, we conduct our experiments on *ionosphere* from UCI datasets ([1]). It is originally collected for supervised learning. There are no natural bags constructed specifically for the AOC problem. Thus we need to create the aggregation ourselves. There are two parameters r and n_b we vary for the aggregation in our experiments. r is the "randomness" within bags ([4]) and n_b is the number of samples per bag.

Our algorithms include AOC-manifold-il-L1, AOC-manifold-il-L2, AOC-manifold-bl-L1 and AOC-manifold-bl-L2. We have realized AOC-kNN and AOC-ANN in [4] for comparison. We did not compare with AOC-SVM mainly because it is too time-consuming. For the *ionosphere* dataset with 351 samples for training, the problem cannot be solved in one day.

We evaluate the performances of the six algorithms on both instance level and bag level. For the instance-level evaluation, we conduct both transductive and inductive experiments. We list these settings and descriptions in Table 2, where "norm of error" criterion for bag-level setting is $\|g - b\|_2$ for binary case and $\|G - B\|_F$ for multi-class case.

Each setting of our experiments is repeated 100 times to give statistical results. The difference between each running time is due to the random swap indicated by r and the randomly divided dataset for training and testing.

Ionosphere is a dataset targeting free electrons in the ionosphere. There are 351 samples in the dataset, and each one has 34 features. The samples are labeled with "Good" or "Bad", indicating whether the data reveals the structure of the ionosphere or not. The results are reported in Fig. 1–3 and Table 3. AOC-manifold-bl-L1 and AOC-manifold-bl-L2 have very similar performances on this dataset, and their curves overlap for the most part.

From the results of the experiments, we reach the following conclusions:

1. Our four algorithms based on manifold regularization are generally better in error rate for instance-level setting and norm of error for bag-level setting than AOC-kNN and AOC-ANN.

Experiment Setting	Training Samples	Test Samples	Criterion for Evaluation
Instance-Level Transduction	Whole dataset	Whole dataset	Error Rate
Instance-Level Induction	80% of the whole dataset	The rest 20% of the dataset	Error Rate
Bag-Level Induction	80% of the whole dataset	The rest 20% of the dataset	Norm of Error

Table 2: Three experiment settings and their descriptions

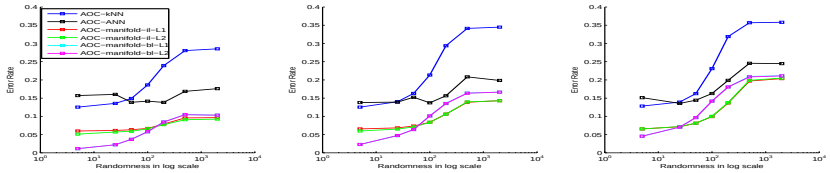

Figure 1: Results of experiments on *ionosphere*, with the instance-level transduction setting. $n_b = 5, 10, 20$ **respectively.**

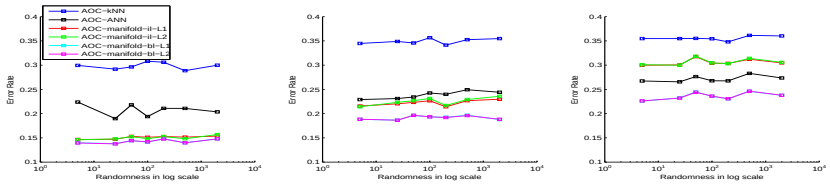

Figure 2: Results of experiments on *ionosphere*, with the instance-level induction setting. $n_b = 5, 10, 20$ **respectively.**

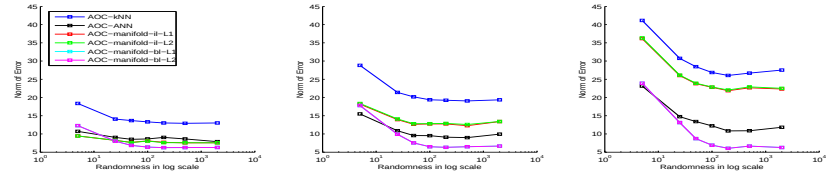

Figure 3: Results of experiments on *ionosphere*, with the bag-level induction setting. $n_b = 5, 10, 20$ **respectively.**

Algorithm	Time/s
AOC-kNN	0.0244 ± 0.0113
AOC-ANN	6.0082 ± 0.5178
AOC-manifold-il-L1	0.1733 ± 0.0088
AOC-manifold-il-L2	0.3176 ± 0.0117
AOC-manifold-bl-L1	0.0513 ± 0.0175
AOC-manifold-bl-L2	0.0421 ± 0.0165

Table 3: Average run time for the six algorithms on *ionosphere*, with the instance-level transduction setting.

2. For the run time of the algorithms, AOC-kNN is the fastest. Our four algorithms follow closely, with bag-level ones better. AOC-ANN is much slower.

3. There is no definite conclusion about which of the instance- and bag-level algorithms is better. It depends on the dataset and setting.

4. Algorithms using L_1 and L_2 loss function yield similar results in error rate or norm of error. There could be some differences in the run time.

6. CONCLUSION

In this paper, we set up a manifold regularization framework for the AOC problem, and propose four new algorithms for both instance- and bag-level settings. Our algorithms can be used for both binary and multi-class cases, while the latter one is not specified in the former works. Also, we conduct experiments on several datasets, with the results suggesting the advantage of our methods.

7. REFERENCES

[1] A. Asuncion and D. Newman. UCI machine learning repository, 2007.

[2] M. Belkin, P. Niyogi, and V. Sindhwani. On manifold regularization. In *AISTAT*, 2005.

[3] T. Gartner, P. Flach, A. Kowalczyk, and A. Smola. Multi-instance kernels. In *ICML*, 2002.

[4] D. Musicant, J. Christensen, and J. Olson. Supervised learning by training on aggregate outputs. In *ICDM*, pages 252–261, 2007.

[5] B. Scholkopf and A. Smola. *Learning with kernels*. MIT press Cambridge, Mass, 2002.