Class-Distribution Regularized Consensus Maximization for Alleviating Overfitting in Model Combination

Sihong Xie[†] Jing Gao[§] Wei Fan[‡] Deepak Turaga[¶] Philip S. Yu[†]
[†]Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA
[§]Department of Computer Science, University at Buffalo, Buffalo, NY, USA
[¶]IBM T.J Watson Research Yorktown Height, NY, USA
[‡]Huawei Noah's Ark Lab, Hong Kong

ABSTRACT

In data mining applications such as crowdsourcing and privacy-preserving data mining, one may wish to obtain consolidated predictions out of multiple models without access to features of the data. Besides, multiple models usually carry complementary predictive information, model combination can potentially provide more robust and accurate predictions by correcting independent errors from individual models. Various methods have been proposed to combine predictions such that the final predictions are maximally agreed upon by multiple base models. Though this maximum consensus principle has been shown to be successful, simply maximizing consensus can lead to less discriminative predictions and overfit the inevitable noise due to imperfect base models. We argue that proper regularization for model combination approaches is needed to alleviate such overfitting effect. Specifically, we analyze the hypothesis spaces of several model combination methods and identify the trade-off between model consensus and generalization ability. We propose a novel model called Regularized Consensus Maximization (RCM), which is formulated as an optimization problem to combine the maximum consensus and large margin principles. We theoretically show that RCM has a smaller upper bound on generalization error compared to the version without regularization. Experiments show that the proposed algorithm outperforms a wide spectrum of state-of-the-art model combination methods on 11 tasks.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

Keywords

Ensemble; Large Margin; Generalization Error

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
KDD'14, August 24–27, 2014, New York, NY, USA.
ACM 978-1-4503-2956-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2623330.2623676.

1. INTRODUCTION

Combining multiple supervised and unsupervised models can be desirable and beneficial, or sometimes even a must. For example, in crowdsourcing, privacy-preserving data mining or big data applications, there could be only predictions from multiple models available, with raw features of the data being withheld or discarded. One has to merge the output of these models to obtain the final classification or clustering results. On the one hand, there are various new consensus-based solutions, such as those proposed in [16, 34, 30, 13, 18, 27, 37]. One common idea that these algorithms share is to learn a model that has highest prediction consensus among base models. On the other hand, simple model combination algorithms, such as majority voting [15], that do not pursue model consensus are portrayed as baselines inferior to the algorithms seeking consensus. These comparisons give the illusion that the more consensus one can achieve, the more likely the consolidated predictions will be accurate. One might ask: are the consolidated predictions that achieve maximal consensus the best choice? Could these consensus-based methods overfit the noisy and limited *observed* data, leading to results inconsistent with the *true* data distribution? After all, the goal of classification/clustering is to produce discriminative predictions [4, 29].

In this paper, we study the above questions based on the Consensus Maximization framework [16] (CM for short in the sequel), due to its generality and effectiveness. We first present a running example of CM in Table 1 to demonstrate that solely maximizing the consensus can lead to undesirable results. Suppose we have 5 instances in 2 classes, whose ground truth labels are shown in the first column of the table. There are 2 supervised models (M_1 and M_2) and 2 unsupervised model (M_3 and M_4). A supervised (resp. unsupervised) model predicts the class (resp. cluster) labels of all instances. The predictions from a model are shown under the header with the model's name. Note that neither the correspondence between class labels and cluster labels, nor the correspondence between cluster labels from different clustering models is known. We describe the details of CM later and for the moment, one can think of CM as a black box that consolidates the predictions of base models and outputs predictive posteriors $p(y = 1|\mathbf{x})$ that achieve maximal consensus among base models. For majority voting (MV for short in the sequel), it simply averages the predictions from supervised models (predictions of unsupervised models cannot be used by MV because the correspondence between classes and cluster labels is unknown). The consolidated

predictions produced by CM and MV are shown in the last two columns of the table.

From this running example, one can see that CM makes more correct predictions than MV does. However, the posteriors $p(y = 1|\mathbf{x})$ produced by CM tend to be closer to the decision boundary and the margins between $p(y = 1|\mathbf{x})$ and $p(y = 0|\mathbf{x})$ are quite small. We have two observations. First, according to the margin-based generalization error analysis [24], a smaller margin of posterior class distributions between different classes leads to a higher *empirical margin risk*, which contributes to the overall generalization error. If one can produce consolidated predictions with a large posterior margin, a tighter upper bound on the generalization error can be obtained. Second, if the hypothesis space for a model combination algorithm has large capacity (measured by VC-dimension, growth function or covering number, etc.), then the upper bound of generalization error is also higher. One may incorporate certain relevant prior knowledge of the data to shrink the size of the hypothesis space. For example, for multi-class single-label classification, desirable consensus predictions should be discriminative in the sense that an instance belongs to one and *only* one class. Our goal is to reduce empirical margin risk and the capacity of the hypothesis space of model combination methods such as CM, and obtain a smaller upper bound on the generalization error.

We propose a family of regularization objectives over class distribution to reduce generalization errors. As a solid instance, we add regularization objectives to CM to obtain Regularized Consensus Maximization (RCM). In terms of algorithmic effectiveness, though the regularization introduces many tuning parameters and makes the optimization problem not jointly convex, we develop a simple yet efficient approximation to the regularization term without introducing additional parameters. An alternative optimization procedure is developed to find a local minimum with reasonably good empirical results. In terms of theoretical effectiveness of learning, we give a detailed analysis of the algorithm and formally prove that, comparing to the original version, RCM achieves a smaller upper bound on generalization error. In summary, we make the following contributions:

- To the best of our knowledge, this is the first work to consider, address and theoretically analyze (Section 2.2 and 4) the overfitting issue in model combination methods.
- We propose a class distribution regularization framework and formulate it as an optimization problem to trade-off between maximal consensus and large margin principles. The optimization problem can be solved conveniently using gradient descent.
- We compare the performance of the proposed algorithm to a wide range of state-of-the-art model combination methods. Extensive experiments on 11 datesets demonstrate the theory and the effectiveness of the proposed framework.

2. OVERFITTED CONSENSUS MAXIMIZATION

In this section, we recapitulate some basic concepts used in the CM framework, which is followed by an analysis of why it tends to overfit the data.

Table 1: Running CM Example

y	Predictions				MV Results	CM Results
	M_1	M_2	M_3	M_4	$P(y=+\|\mathbf{x})$	$P(y=+\|\mathbf{x})$
+	+	+	C1	R0	1	0.5073
+	+	+	C0	R0	1	0.5097
+	−	+	C0	R0	.5	0.5024
−	+	−	C1	R1	.5	0.4946
−	−	−	C1	R1	0	0.4873

Table 2: Notations

$\mathbf{x}^i \in \mathcal{X} = \mathbb{R}^d$	An instance of data
$\mathcal{D} = \{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$	Collection of instances
$Y = [y_1, \ldots, y_n]^\top$	Ground truth labels of $\mathcal{D}$
$U_{n\times c}$	Membership indicators of data
$Q_{v\times c}$	Membership indicators of groups
$A_{n\times v}$	Affinity matrix
c	Number of classes
$\mathbf{u}^i$	The i-th row of matrix U
$\mathbf{u}_j$	The j-th column of matrix U

2.1 Preliminaries

CM combines the output of multiple supervised and unsupervised base models (as shown in Table 1). Let the set of instances be $\mathcal{D} = \{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$, each of which belongs to one of c classes $\{1, \ldots, c\}$. Suppose we have m models $\{M_1, \ldots, M_m\}$. Without loss of generality, we assume that the first $0 \le r \le m$ models are supervised models, and the rest are unsupervised models. A supervised model predicts the class label of each instance, while an unsupervised model predicts the cluster label and partitions $\mathcal{D}$ into c clusters. Given the predictions of m models, without access to the data $\mathcal{D}$, CM computes consolidated predictions to achieve maximal consensus among base models.

We use the example in Table 1 to demonstrate how CM constructs a bipartite graph and obtains consensus predictions. The bipartite graph contains group nodes and instance nodes, where a group node represent a class or cluster from a supervised or unsupervised model, and an instance node represent a data instance. Therefore, there are $c \times m$ group nodes and n instance nodes. We number the group nodes such that class/cluster ℓ from the j-th model is labeled as the $((j-1)\times c + \ell)$-th group. The bipartite graph constructed using the toy example is shown in Figure 1. An instance node is connected to a group node iff the corresponding instance is classified or clustered into the corresponding class or cluster of a model. A group node of a supervised model is also connected to an external node representing its ground truth label (the left-most nodes in the figure). As $\mathbf{x}^1$ is predicted to have label 1 by M_1, instance node $\mathbf{x}^1$ is linked to the group nodes g^1 representing class 1 of M_1. Similarly, $\mathbf{x}^5$ is linked to group node g_8 to indicate that the instance belongs to the second group (R_1) by M_4.

CM consolidates these base models into a single model whose predictions achieve maximal consensus among the predictions of base models. Formally, we denote the membership distribution of an instance node for $\mathbf{x}^i$ by a row probabilistic vector $\mathbf{u}^i = [u^i_1, \ldots, u^i_c]$. Similarly, the membership distribution of a group node is given by a row probabilistic vector $\mathbf{q}^j = [q^j_1, \ldots, q^j_c]$. These vectors can be collectively denoted by two matrices: $U = [\mathbf{u}^{1\top}, \ldots, \mathbf{u}^{n\top}]^\top$ and

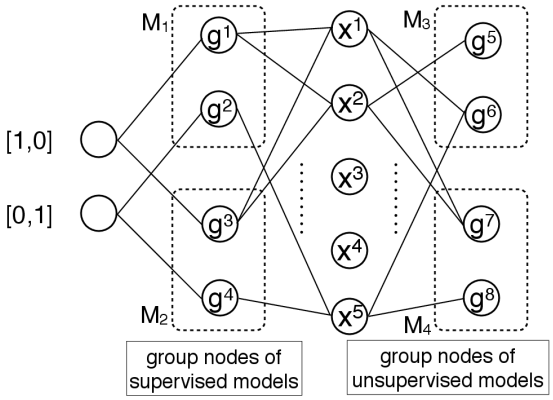


Figure 1: Bipartite graph representation in CM

$Q = [\mathbf{q}^{1\top}, \ldots, \mathbf{q}^{v\top}]^\top$ where $v = c \times m$. CM seeks smooth probability distributions U and Q, such that *any* connected nodes in the graph have similar probabilistic distributions. CM solves the following optimization problem:

$$\begin{aligned} \min_{U,Q} \quad & \sum_{i=1}^{n}\sum_{j=1}^{v} a_j^i \|\mathbf{u}^i - \mathbf{q}^j\|^2 + \alpha \sum_{j=1}^{v} b_j \|\mathbf{q}^j - \bar{\mathbf{y}}^j\|^2 \\ \text{s.t.} \quad & u_\ell^i \geq 0, \textstyle\sum_{\ell=1}^{c} u_\ell^i = 1, i = 1, \ldots, n \\ & q_\ell^j \geq 0, \textstyle\sum_{\ell=1}^{c} q_\ell^j = 1, j = 1, \ldots, v \end{aligned}$$

where $a_j^i = 1$ if the i-th instance node is connected to the j-th group nodes, and is equal to 0 otherwise. b_j indicates if the j-th group node is from a classification model ($b_j = 1$) or a clustering model ($b_j = 0$). $\bar{\mathbf{y}}^j$ is a vector indicating the class label of the j-th group node from a classification model. For example, $\bar{\mathbf{y}}^1 = [1, 0]$ and $\bar{\mathbf{y}}^2 = [0, 1]$ in the above running example. $\bar{\mathbf{y}}^j$ is an all zero vector if the corresponding group node is from a clustering model.

We denote the objective by $\mathcal{L}(U, Q)$ and is defined as **consensus loss**. It measures the level of disagreement between the consolidated predictions and the outputs of base models. By minimizing $\mathcal{L}(U, Q)$, consensus among base models is maximized. The second term accounts for the initial predictions of supervised models, and does not play a role if all base models are unsupervised. The whole optimization is solved via block coordinate descent:

$$Q = (D_v + \alpha K_v)^{-1}(A^\top U + \alpha K_v Y) \quad (1)$$

$$U = D_n^{-1} A Q \quad (2)$$

Here $A = (a_j^i)_{n\times v}$, $D_v = \text{diag}(A^\top \times \mathbb{1}_n)$ and $D_n = \text{diag}(A \times \mathbb{1}_v)$, $K_v = \text{diag}(\bar{\mathbf{Y}} \times \mathbb{1}_c)$. Here $\mathbb{1}_k$ is an all one column vector of length k. Upon convergence, the final posterior distributions are given in the rows of U and one can use Bayes' optimal decision rule to decide the most likely label.

2.2 CM overfits

The hypothesis space of a learning algorithm is the set of all feasible solutions of the algorithm. A larger hypothesis space has more expressive power comparing to a smaller one, leading to less training errors. However, models with a larger hypothesis space is more complicated and can lead to less generalization ability and more predicting errors [29]. Therefore, one needs to trade-off between minimizing training error and model complexity. We compare the hypothesis spaces of two model combination methods, MV and CM, leading to some insights into the overfitting issue of CM.

Suppose there are m base models (for ease of presentation, assume all models are supervised models). For each instance and each class, a model outputs 1 (a vote) or 0 (no vote). A model combination method is a function f that maps predictions of base models to posterior distributions on the probabilistic simplex:

$$f : X \to S \quad (3)$$

$$S = \{p \subset \mathbb{R}^\ell | p = \sum_{\ell=1}^{c} \theta_\ell \mathbf{e}_\ell = [\theta_1, \ldots, \theta_\ell], \sum_{\ell=1}^{c} \theta_\ell = 1, \theta_\ell \geq 0\}$$

where we abuse the notation X, such that X is the collection of base model predictions. $\mathbf{e}_\ell$ is the standard basis having 1 in its ℓ-th position and 0 anywhere else, representing the distribution of class ℓ, θ_ℓ is the probability that an instance belongs to class ℓ. When $c = 3$, an example of 2-simplex is shown in Figure 2(a). Various model combination methods can be seen as ways of searching a suitable mapping f in the hypothesis space $\mathcal{F}$ of all such mappings, to optimize certain objectives. Existing methods differ in their hypothesis spaces $\mathcal{F}$ and the way they searches, but the capacity of the hypothesis space is directly related to the generalization ability of a method. Note that the domain of all model combination methods are the same, so the capacities of their hypothesis spaces are completely determined by the images of the maps $f(X) \subset S$.

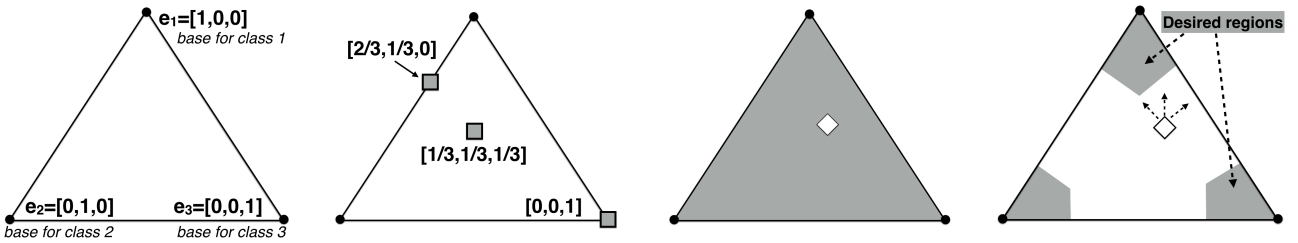
Majority voting simply sums up the number of votes for each class and assigns an instance to the class having the most votes. Formally, given the output of r models for an instance, say, $[\hat{y}_1, \ldots, \hat{y}_r]$, $\hat{y}_k \in \{1, \ldots, c\}$, the decision of majority voting is made based on the the vector:

$$\frac{1}{r}\left[\sum_{k=1}^{r} \mathbb{1}[\hat{y}_k = 1], \ldots, \sum_{k=1}^{r} \mathbb{1}[\hat{y}_k = c]\right] \in S \quad (4)$$

Note that majority voting maps the predictions of base models to rational vectors on the simplex, with denominators equal to the number of models. For example, if an instance receives two votes for class 1, one votes for class 2 and 0 vote for class 3, from a total of three classifiers, then the output of f is $[2/3, 1/3, 0]$, shown as the square with an arrow in Figure 2(b).

CM maps predictions of base models of an instance to a posterior distribution in S, and the image of the map is the whole simplex S. The relaxation from rational vectors to real vectors allows a larger hypothesis space such that CM can find an f to attain low consensus loss (see Section 5.2). However, it also allows CM to pick an f that outputs predictions close to uniform distribution with small margin (like the diamond in Figure 2(c)), leading to higher empirical margin risk (see Section 4) It is verified in Section 5 that CM does tend to output predictions that have small consensus losses and small margins. Here we define "overfitting" in model combination in a vague sense, and defer the formal analysis to Section 4.

Definition 1 (Overfitting in model combination). *A model combination method consolidates predictions of base models to achieve a high degree of model consensus but with higher generalization error upper bound.*



(a) 2-Simplex (b) Hypothesis spaces of MV (only part of the output space is shown) (c) Hypothesis space of CM (d) Hypothesis space of RCM

Figure 2: Hypothesis space of various methods: the tips of the triangles represent the bases $\mathbf{e}_\ell$

3. CLASS-DISTRIBUTION REGULARIZED CONSENSUS MAXIMIZATION

3.1 Regularization over class distributions

According to the above analysis, if we adopt a reasonably small but rich enough hypothesis space for CM, then we could avoid over-fitting and achieve better performance. How can we specify a suitable hypothesis space for CM? Note that the predictions lying near the corners of the simplex (shadows in Figure 2(d)) have a more dominating component θ_{ℓ_0} for some class ℓ_0. On the one hand, when the difference between $p(y = \ell_0|\mathbf{x})$ and any other $p(y = \ell|\mathbf{x})$ is larger, the prediction is more discriminative to reflect the true class distribution. On the other hand, if the number of dominating entries in $p(y|\mathbf{x})$ is greater than 1, then those dominating classes are correlated since they co-occur, conflicting the multi-class distribution assumption. This observation indicates that when searching for solutions in the hypothesis space, CM should penalize solutions that lie too far away from any corner of the simplex and encourage solutions that lie close to the corners. For CM to reduce the penalized consensus loss $\mathcal{L}(U, Q)$, it must move its predictions towards one of the corners on the simplex, as shown by the arrows in Figure 2(d). The above intuition suggests that the consolidated predictions should exhibit some sort of independence between classes, given the problem is a multi-class single label problem.

Specifically, recall that U is the consolidated prediction, with the $\ell-$th column being the posterior probabilities $p(y = \ell|\mathbf{x})$, we can compute the empirical class correlation matrix $\Sigma = U^\top U$. We want the matrix Σ to be close to a $c\times c$ matrix D, which represents the ideal class correlations. For example, to enforce independence between classes in multi-class classification problems, we can set the diagonal elements of D to a positive number whose scale is comparable to the empirical correlations, and set the off-diagonal elements to a positive number much smaller than the diagonal elements.

By adopting the Frobenius norm, we obtain the following regularization term

$$\Delta_F = \frac{1}{2}\|\Sigma - D\|_F \tag{5}$$

or by adopting the relative entropy [5]

$$\Delta_E = \frac{1}{2}\sum_{i,j=1}^{c} \Sigma_{ij} \log \frac{\Sigma_{ij}}{D_{ij}} \tag{6}$$

Adding any of the above regularization terms to the objective of CM, we obtain the following optimization problem:

$$\begin{aligned} \min_{U,Q} \quad & \sum_{i=1}^{n}\sum_{j=1}^{v} a_{ij}\|\mathbf{u}^i - \mathbf{q}^j\|^2 + \alpha \sum_{j=1}^{v} b_j \|\mathbf{q}^j - \bar{\mathbf{y}}^j\|^2 + \lambda\Delta \\ \text{s.t.} \quad & u^i_\ell \geq 0, \|\mathbf{u}^i\|_1 = 1, i = 1, \ldots, n \\ & q^j_\ell \geq 0, \|\mathbf{q}^j\|_1 = 1, j = 1, \ldots, v \end{aligned}$$

where $\Delta = \Delta_F$ or Δ_E. The parameter λ controls the trade-off between model consensus and class independence. We will see that the regularization helps reduce the capacity of hypothesis space and also the empirical margin risk.

3.2 Optimization of the Class-distribution Regularized Model

Our plan for solving the optimization problem Eq.(7) is to first ignore the constraints that $\mathbf{u}^i$ and $\mathbf{q}^j$ are probability distributions and solve the unconstrained optimization problem using gradient descent, then we address the probabilistic constraints on $\mathbf{u}^i$ and $\mathbf{q}^j$ in the next section. The gradient descent steps for the first two terms in the above objective function are given in Eq.(1) and Eq.(2), the gradients of the regularization term Δ with respect to column $\mathbf{u}_j$ are as follows:

$$\begin{aligned} \frac{\partial \Delta_F}{\partial \mathbf{u}_j} &= \sum_{i=1}^{c} (\Sigma_{ij} - D_{ij})\mathbf{u}_i \\ \frac{\partial \Delta_E}{\partial \mathbf{u}_j} &= \sum_{i=1}^{c} (1 + \log \frac{\Sigma_{ij}}{D_{ij}})\mathbf{u}_i \end{aligned}$$

Thus a gradient descent step for the regularization term with respect to column $\mathbf{u}_j$ are:

$$\mathbf{u}_j \leftarrow \mathbf{u}_j - \eta_t \sum_{i=1}^{c} (\Sigma_{ij} - D_{ij})\mathbf{u}_i \tag{7}$$

$$\mathbf{u}_j \leftarrow \mathbf{u}_j - \eta_t \sum_{i=1}^{c} (1 + \log \frac{\Sigma_{ij}}{D_{ij}})\mathbf{u}_i \tag{8}$$

where $\leftarrow$ indicates the assignment of an updated $\mathbf{u}_j$ to itself. η_t is the learning rate in the t-th iteration with $\eta_t = \eta_0/\sqrt{t}$ and η_0 is the initial learning rate. We let the trade-off parameter λ in the RCM objective be absorbed in η_0. Eq.(7) and Eq.(8) have a quite intuitive meaning: for each column $\mathbf{u}_i$ representing the i-th class, depending on whether the empirical class correlation Σ_{ij} exceeds the ideal class correlation D_{ij}, $\mathbf{u}_j$ is moved away from ($\Sigma_{ij} > D_{ij}$) or towards

($\Sigma_{ij} < D_{ij}$) $\mathbf{u}_i$, and the amount of displacement is proportional to the distance between the empirical and ideal class correlation. In practice, it is not easy to specify the ideal class correlation matrix D, and the scaling parameters $\beta_{ij} = \Sigma_{ij} - D_{ij}$ (or $1 + \log \frac{\Sigma_{ij}}{D_{ij}}$) may be sensitive to the choice of D. Simply setting all the β_{ij} to be 1 will actually hurt the performance, as we ignore the information about the class correlations.

We propose an approximation of Eq.(7) and Eq.(8) to avoid specifying the parameters D and to maintain the effect of the regularization, namely, a large margin between class distributions. Note that in Eq.(8), for $i \neq j$, D_{ij} should be some small number and if $\Sigma_{ij} \gg D_{ij}$, the scaling parameter $1 + \log \frac{\Sigma_{ij}}{D_{ij}}$ will be large; on the other hand, if Σ_{ij} is about the same as D_{ij}, $1 + \log \frac{\Sigma_{ij}}{D_{ij}}$ will be close to 1. According to this observation, when computing the gradient for the column $\mathbf{u}_j$, we can set β_{ij} as follows:

$$\beta_{ij} = \begin{cases} 1 & \text{if } i = \operatorname{argmin}_{k \neq j} \|\mathbf{u}_k - \mathbf{u}_j\|_2 \\ -1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The resulting regularization term is

$$\Delta_A = \frac{1}{2} \sum_{j=1}^{c} \|\mathbf{u}_j - \mathbf{u}_{d(j)}\|_2^2 \tag{10}$$

where

$$d(j) = \operatorname*{argmin}_{k \neq j} \|\mathbf{u}_j - \mathbf{u}_k\|_2 \tag{11}$$

Eq.(7) and Eq.(8) become

$$\mathbf{u}_j \leftarrow \mathbf{u}_j - \eta_t(\mathbf{u}_{d(j)} - \mathbf{u}_j) \tag{12}$$

So far we have specified all necessary gradient descent steps for RCM. Nonetheless, the original CM gradient descent steps involve the rows of the matrices U and Q, while to minimize the regularization term Δ, one has to work with the columns of U. It is non-trivial to derive gradient descent steps involving both rows and columns of a matrix. We adopt an alternative optimization procedure that first minimizes the consensus loss $\mathcal{L}(U, Q)$ through Eq.(1) and Eq.(2), then minimizes Δ_A through Eq.(12). These two steps are alternatively repeated until it converges.

3.3 Projection to the Probabilistic Simplex

The converted unconstrained optimization problem ignores the constraints:

$$\begin{aligned} &u_\ell^i \geq 0, \|\mathbf{u}^i\|_1 = 1, i = 1, \ldots, n \\ &q_\ell^j \geq 0, \|\mathbf{q}^j\|_1 = 1, j = 1, \ldots, v \end{aligned} \tag{13}$$

Although Eq.(1) and (2) maintain rows of U and Q as probability distributions, Eq.(12) might bring any entry of U to be greater than 1 or less than 0, and a row in U or Q might not sum up to 1. We propose to perform *probabilistic projection* for all $\mathbf{u}^i$ after all gradient descent steps in each iteration. More formally, the following optimization problem finds $\mathbf{v}$, the projection of $\mathbf{u}^i$ onto the probabilistic simplex

$$\begin{aligned} \min_{\mathbf{v}} \quad & \|\mathbf{v} - \mathbf{u}^i\|_2 \\ \text{s.t.} \quad & \|\mathbf{v}\|_1 = 1, v_\ell \geq 0, \ell = 1, \ldots, c \end{aligned}$$

The optimal solution $\mathbf{v}^*$ serves as the new $\mathbf{u}^i$ for the next iteration, with the probabilistic constraints satisfied. An efficient algorithm (in $O(cn)$) with implementation to solve the above problem can be found in [12]. The complete algorithm is described in Algorithm 1.

Algorithm 1 Regularized Consensus Maximization (RCM)

Input: Affinity matrix A, initial learning rate η_0
Set $\mathbf{u}^j$ to uniform distribution.
for $t = 1 \rightarrow$ MaxIterNum **do**
  $Q = (D_v + \alpha K_v)^{-1}(A^\top U + \alpha K_v Y)$
  $U = D_n^{-1} A Q$
  $\eta_t = \eta_0 / \sqrt{t}$
  for $j = 1 \rightarrow c$ **do**
    $d(j) = \arg\min_{k \neq j} \|\mathbf{u}_k - \mathbf{u}_j\|$
    $\mathbf{u}_j \leftarrow \mathbf{u}_j - \eta_t(\mathbf{u}_{d(j)} - \mathbf{u}_j)$
  end for
  Project $\mathbf{u}^i$ to the probabilistic simplex.
end for

4. GENERALIZATION ERROR OF RCM

In this section, we prove that, compared to CM, the proposed regularization leads to a smaller upper bound on generalization error. The generalization error bound consists of two terms: the empirical margin risk on training data and a term measuring the capacity of the hypothesis space explored by a learning algorithm. Regarding the empirical margin risk, we first define the multi-class margin [24].

Definition 2 (Canonical Function). *Given a function $f \in \mathcal{F}$ that maps predictions of base models to posterior distribution (see Section 2.2). For the instance $\mathbf{x}$, $f(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_c(\mathbf{x})] \in S$ where $f_\ell(\mathbf{x})$ is the probability that $\mathbf{x}$ belongs to class ℓ, according to f. Let M_1 be the smallest index ℓ such that $f_\ell(\mathbf{x}) = \max_k f_k(\mathbf{x})$, and M_2 be the smallest index ℓ such that $f_\ell(\mathbf{x}) = \max_{k \neq M_1} f_k(\mathbf{x})$. The canonical function $\Delta f : X \rightarrow [-1, 1]^c$, with the ℓ-th component being:*

$$\Delta f_\ell(\mathbf{x}) = \begin{cases} f_\ell(\mathbf{x}) - f_{M_2}(\mathbf{x}) & \text{if } \ell = M_1 \\ f_\ell(\mathbf{x}) - f_{M_1}(\mathbf{x}) & \text{otherwise} \end{cases} \tag{14}$$

M_1 is the label selected by Bayes decision rule and M_2 is the closest runner-up. Δf_ℓ measures how far away the selected label is from the other competitors. Based on the canonical function, we define the multi-class empirical margin risk

Definition 3 (Empirical Margin Risk). *For $\gamma > 0$ and training set $s = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^m$, the empirical margin risk $R_s^\gamma(f)$ of the function f is*

$$R_s^\gamma(f) = \frac{1}{m} |\{\mathbf{x}_i | \exists \ell \in \{1, \ldots, c\}, y_{i\ell} \cdot \Delta f_\ell(\mathbf{x}_i) < \gamma\}| \tag{15}$$

where $y_{i\ell}$ is the ℓ-th component of the true label vector $\mathbf{y}_i$.

Next we define necessary concepts to measure the capacity of hypothesis spaces.

Definition 4 (Supremum Metric for functions). *[24, 3] Suppose $\mathcal{F}$ is the collection of functions mapping from X to S, and $s = \{\mathbf{x}_i\}_{i=1}^m \subset X$ is a given set of instances. Define the metric (distance measure) for functions $d(\cdot, \cdot) : \mathcal{F} \times \mathcal{F} \rightarrow [0, +\infty)$ on s by*

$$d_s(f, \tilde{f}) = \max_{\mathbf{x}_i \in s} \sum_{\ell=1}^{c} |f_\ell(\mathbf{x}_i) - \tilde{f}_\ell(\mathbf{x}_i)| \tag{16}$$

Note that the metric such defined depends on the set of instances s.

DEFINITION 5 (COVERING NUMBER). *Let $(\mathcal{F}, d_s)$ be the space of functions equipped with the supremum metric, where $s \subset X$ a finite set of instances. Define $B_s(f, r)$ the closed ball centered at f with radius r:*

$$B_s(f, r) = \{g \in \mathcal{F} | d_s(f, g) \leq r\} \tag{17}$$

The covering number $\mathcal{N}(\epsilon, \mathcal{H}, d_s)$ of a set $\mathcal{H} \subset \mathcal{F}$ is defined as

$$\mathcal{N}(\epsilon, \mathcal{H}, d_s) = \inf_T \{|T|\} \ s.t. \ \mathcal{H} \subset \cup_{f \in T} B_s(f, \epsilon) \tag{18}$$

The set T is called an ϵ-cover of the subset $\mathcal{H}$.

The following bound on generalization error for multi-class classification is given in [24]:

THEOREM 1. *Let $\mathcal{F}$ be a set of functions from X to S and $\Delta\mathcal{F}$ be the set of canonical functions Δf. Let s be a learning set of size m drawn iid. from a probability distribution P. Let $0 < \gamma < 1$. With probability $1 - \delta$, $\forall f \in \mathcal{F}$,*

$$R(f) \leq R_s^\gamma(f) + \sqrt{\frac{1}{2m} \ln \left(\frac{2\mathcal{N}_\infty(\gamma/2, \Delta\mathcal{F}^\gamma)}{\delta} \right)} \tag{19}$$

where

$$\mathcal{N}_\infty(\gamma, \mathcal{F}) = \sup_{s:|s|=2m} \mathcal{N}(\gamma, \mathcal{F}, d_s) \tag{20}$$

$\Delta\mathcal{F}^\gamma = \{\pi_\gamma \circ \Delta f : \Delta f \in \Delta\mathcal{F}\}$ *where π_γ is the truncation function applied to each of the c components of Δf*

$$\pi_\gamma(f_\ell(\mathbf{x})) = \begin{cases} \gamma \cdot sign(f_\ell(\mathbf{x})) & if \ |f_\ell(\mathbf{x})| \geq \gamma \\ f_\ell(\mathbf{x}) & otherwise \end{cases} \tag{21}$$

Given the bound in Eq.(19), we want to prove that both terms in the bound for the regularized CM are smaller than those for the original CM, and obtain the following theorem:

THEOREM 2. *RCM has a smaller upper bound on generalization error compared with that of CM.*

The above theorem is proved in two steps in the following two lemmas.

LEMMA 1. *RCM achieves a lower empirical margin risk if we use Δ_E as our regularization term and the matrix D is such set that the scaling parameters $\beta_{ij} = \beta_{ji}$ and $\beta_{ii} = 1$.*

PROOF. Given training data s, $0 < \gamma < 1$, $1 - R_s^\gamma(f)$ is the proportion of correctly classified instances with margin greater than γ. Suppose f is the prediction function found by CM and $\tilde{f}$ is that found by RCM. In other words, $\tilde{f}$ is obtained by applying Eq.(8) to f. Note that $R_s^\gamma(\tilde{f}) \leq R_s^\gamma(f) \iff 1 - R_s^\gamma(\tilde{f}) \geq 1 - R_s^\gamma(f)$, we need to prove, for any correctly classified instance with margin greater than γ, its margin under $\tilde{f}$ is not smaller than that under f.

Let $\mathbf{u} = [f_1, \ldots, f_c]$ and $\tilde{\mathbf{u}} = [\tilde{f}_1, \ldots, \tilde{f}_c]$ be the evaluations of f and $\tilde{f}$ at some point $\mathbf{x}$ that is correctly classified with margin larger than γ (we ignore the arguments of f and $\tilde{f}$). Assume $1 = \text{argmax}_\ell f_\ell$ and $2 = \text{argmax}_{\ell \neq 1} f_\ell$. Then $y_1 \cdot \Delta f_1 \geq \gamma$. But $y_1 = 1$, so $\Delta f_1 = f_1 - f_2 \geq \gamma$. The gradients Eq.(8) at $\mathbf{x}$ be

$$g_j = \eta_t \sum_{i=1}^{c} (1 + \log \frac{\Sigma_{ij}}{D_{ij}}) f_i > 0, j = 1, 2 \tag{22}$$

Assume that proper values are set to matrix D, such that $\Sigma_{ii} = D_{ii}$ but $\Sigma_{ij} \gg D_{ij}$ for $i \neq j$. Then the gradients are

$$g_j = \eta_t \sum_{i=1}^{c} \beta_{ij} f_i, j = 1, 2 \tag{23}$$

where $\beta_{ii} \ll \beta_{ij}, i \neq j$. That is, for a given j, f_i has a much larger weight than f_j in g_j for $i \neq j$. If $\beta_{ij} = \beta_{ji}$, then by $f_1 > f_2$, we have $g_2 > g_1$,

$$\Delta\tilde{f}_1 = \tilde{f}_1 - \tilde{f}_2 = (f_1 - g_1) - (f_2 - g_2) = \Delta f_1 - (g_1 - g_2) > \Delta f_1 \tag{24}$$

□

LEMMA 2. *The hypothesis space of RCM has smaller covering number than the hypothesis space of CM.*

PROOF. Let $\Delta\mathcal{F}^\gamma = \{\pi_\gamma \circ \Delta f : \Delta f \in \Delta\mathcal{F}\}$ and $\Delta\tilde{\mathcal{F}}^\gamma = \{\pi_\gamma \circ \Delta\tilde{f} : \Delta f \in \Delta\tilde{\mathcal{F}}\}$ where $\mathcal{F}$ is the collection of functions $f : X \to S$ and $\tilde{\mathcal{F}}$ are their large margin version as defined in Lemma 1, Δf is the canonical function and π_γ is the truncation function Eq.(21). Then $\Delta\tilde{\mathcal{F}}^\gamma \subset \Delta\mathcal{F}$ since for any $f \in \mathcal{F}$, its large margin version $\tilde{f} \in \mathcal{F}$, thus we have $\Delta\tilde{\mathcal{F}} \subset \Delta\mathcal{F}$. After truncation, $\Delta\tilde{\mathcal{F}}^\gamma \subset \Delta\mathcal{F}^\gamma$.

Given any training data s of size $2m$, any $\gamma/2$-cover of $\Delta\mathcal{F}^\gamma$ is also a $\gamma/2$-cover of $\Delta\tilde{\mathcal{F}}^\gamma$. Therefore by definition Eq.(18),

$$\mathcal{N}(\gamma/2, \Delta\mathcal{F}^\gamma, s) = \inf\{|T|\} \geq \inf\{|T'|\} = \mathcal{N}(\gamma/2, \Delta\tilde{\mathcal{F}}^\gamma, s) \tag{25}$$

where $T \in \{\gamma/2\text{-covers of } \Delta\mathcal{F}^\gamma\}$ and $T' \in \{\gamma/2\text{-covers of } \Delta\tilde{\mathcal{F}}^\gamma\}$. By the definition Eq.(20), we conclude that

$$\begin{aligned} \mathcal{N}_\infty(\gamma/2, \Delta\mathcal{F}^\gamma) &= \sup_s \mathcal{N}(\gamma/2, \Delta\mathcal{F}^\gamma) \\ &\geq \sup_s \mathcal{N}(\gamma/2, \Delta\tilde{\mathcal{F}}^\gamma) = \mathcal{N}_\infty(\gamma/2, \Delta\tilde{\mathcal{F}}^\gamma) \end{aligned}$$

□

5. EXPERIMENTAL RESULTS

In this section, we first summarize the experimental settings, including evaluation benchmarks and model combination baselines. Then we demonstrate how CM overfits the data and how the proposed RCM resolves the issue.

5.1 Experimental Settings

Benchmarks A model consolidation method consolidates the predictions of multiple supervised and/or unsupervised models to come up with improved predictive performance. Therefore, to evaluate the performance, we need the predictions from multiple base models for the datasets, whose information are summarized in Table 3. The dataset[1] contains 11 text classification tasks. Each task contains the predictions given by the output of 2 classification and 2 clustering models. For details of how they processed the data, please refer to [16].

We compare RCM with CM in order to verify the effectiveness of the large margin constraint. CM and RCM share most of the parameters such as number of iterations, importance of supervised models, etc.. For the shared parameters, we adopt the parameter settings of CM [16]. In addition, we set the initial learning rate η_0 to be 0.1. We also compare RCM with other state-of-the-art cluster ensemble methods:

[1]available at `http://www.cse.buffalo.edu/~jing/`

Table 4: Overall Performance on Text Classification Tasks

Methods	Newsgroups						Cora				DBLP
	1	2	3	4	5	6	1	2	3	4	1
MCLA	0.7574	0.8345	0.7816	0.8225	0.8039	0.8332	0.8522	0.8009	0.8442	0.8262	0.8604
HBGF	0.721	0.636	0.7677	0.6885	0.6421	0.7482	0.7966	0.6574	0.7655	0.7912	0.8146
SNNMF	0.5980	0.6904	0.6384	0.5733	0.6245	0.6753	0.7407	0.6492	0.7051	0.6989	0.6307
BCE	0.6639	0.2544	0.7082	0.7230	0.7247	0.7474	0.6546	0.8915	0.5565	0.2482	0.2887
ECMC	0.5599	0.6215	0.6294	0.6759	0.6338	0.4530	0.5973	0.6428	0.5252	0.8513	0.7771
CM	**0.8131**	**0.9106**	0.8608	0.9117	0.8857	0.9094	0.8688	0.9151	0.8951	0.9036	0.9412
RCM	**0.8131**	0.9030	**0.8735**	**0.9232**	**0.8927**	**0.9134**	**0.8703**	**0.9222**	**0.9203**	**0.9128**	**0.9429**

Table 3: Datasets and Base Models

Datasets		# Instances	# Classes	Predictors
20NG	1	1568	4	Apply SVM, Logistic Regression, K-means and mini-cut to texts. 4 Predictors in total.
	2	1588	4	
	3	1573	4	
	4	1484	4	
	5	1584	4	
	6	1512	4	
Cora	1	663	3	Apply Logistic Regression K-means to citation/publication network and texts. 4 Predictors in total.
	2	977	4	
	3	1468	5	
	4	975	5	
DBLP		4236	4	

MCLA [27], HBGF [27], SNNMF [18], BCE [30] ECMC [37]. MCLA and HBGF are graph partition based approaches, which use spectral clustering [23, 11] to partition the bipartite or hyper graph constructed from the predictions of base models. There is no parameter to tune for these two methods. SNNMF is a matrix factorization based method, which derives clustering of instances using the similarity matrix constructed from base models' predictions. We run SNNMF to its convergence to obtain the final predictions. BCE is a Bayesian approach to consensus maximization. We set its parameters as follows: LDA parameters $\alpha = 0.5, \beta = 0.1$, number of iterations for Gibbs sampling is set to 50,000, the topic distributions of the words in documents are randomly initialized. We observe that performance the Gibbs sampling for BCE is sensitive to the initialization of the parameters and unstable, we run the BCE for 10 times and report its best performance. We also implemented BCE using variational inference, but the procedure did not converge after long runs, so we do not report the corresponding results. ECMC is a matrix factorization method with a de-noising step, we adopt the implementations of robust PCA and matrix completion packages[2], with $d_0 = 0.4, d_1 = 0.6$ and other parameters being the default values (see [37] for details).

5.2 Overfitting in Consensus Maximization

In Section 2.1 and 2.2 we theoretically showed that, CM produces predictions that minimize the consensus loss but overfit the data, and therefore might not generalize well, and in Section 3.1, we proposed RCM to solve the issues. By comparing CM and RCM in consensus loss, prediction margins and accuracy (next section), we verify that CM does have the overfitting issue and RCM can effectively mitigate overfitting.

On the one hand, one can see from Figure 3 that CM has a lower consensus loss than RCM does across all datasets.

[2] `http://perception.csl.illinois.edu/matrix-rank/sample_code.html`

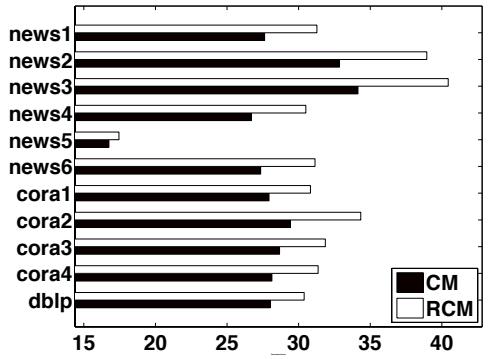


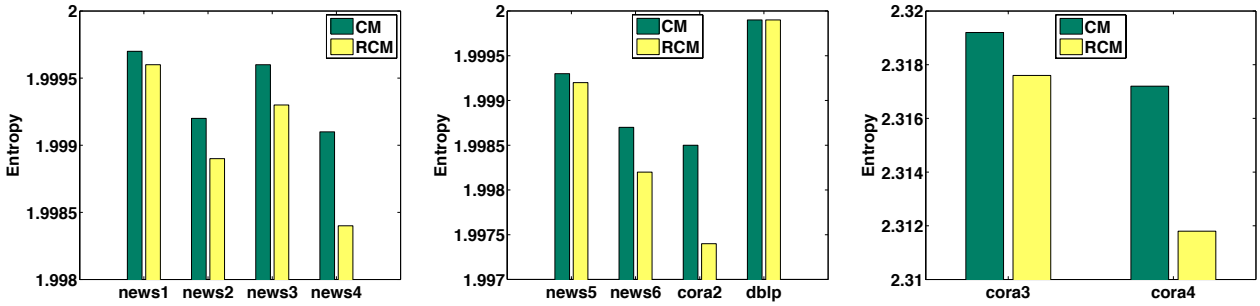
Figure 3: Consensus Loss

This is because CM solely minimizes the consensus loss while RCM minimizes a regularized consensus loss and has a smaller hypothesis space. On the other hand, we use entropy of $\mathbf{u}^i$ ($h^i = -\sum_{\ell=1}^{c} u_\ell^i \log u_\ell^i$) as a measure of prediction margin: the higher the entropy, the smaller the margin $\mathbf{u}^i$ has and the less discriminative $\mathbf{u}^i$ is. We show the averaged entropy $\frac{1}{n}\sum_{i=1}^{n} h^i$ for each dataset in Figure 4(a), 4(b), 4(c). From the figures, we can see that the entropy is higher in the predictions of CM across all datasets except on the *dblp* dataset. (the result on the *cora1* dataset is not shown due to the scale). Therefore on average, the predictions of CM have smaller margins than those of RCM. Since margin is used as an indicator of generalization performance of a learning algorithm [4], CM might overfit the data while RCM should improve the generalization ability and accuracy of CM.

5.3 Accuracy

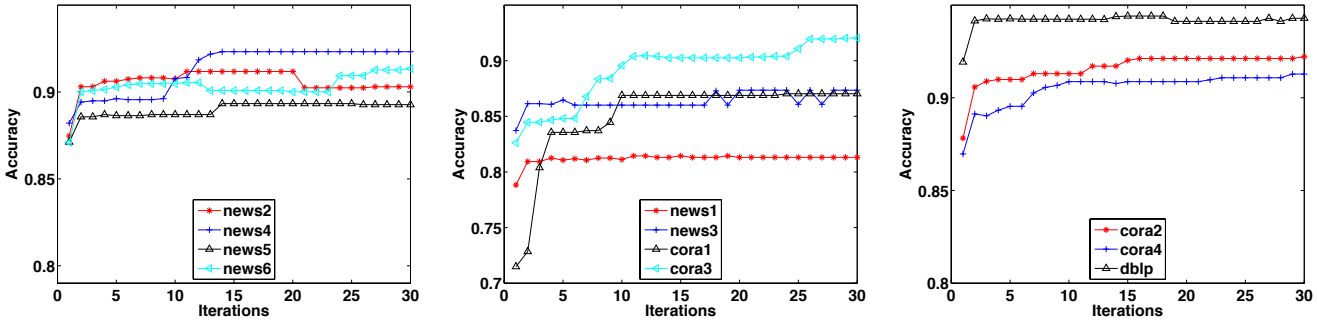
In Table 4, we compare the accuracies of RCM and the baselines on 11 text classification tasks. From the table, we can see that BCE is very unstable and there are two main reasons for this. First, similar to LDA, BCE needs a lot of observed data to infer the consolidated labels, yet usually we have only a couple of base models. Second, Gibbs sampling is too sensitive to initial conditions while variational inference does not converge given only a handful of data. ECMC and SNNMF sometimes give reasonable performance, such as ECMC on the *cora4* task. However, their optimization are also sensitive to initialization, and their solutions are unstable. Both MCLA and HBGF in general have better performance than ECMC and SNNMF, though they are still outperformed by both CM and RCM.

The comparison between CM and RCM is more interesting. Using the proposed regularization over the class distributions, RCM controls the size of its hypothesis space and focuses on the more discriminative predictions. As we can see from the table, RCM outperforms CM on 10 out of 11



(a) Prediction Entropy (b) Prediction Entropy (c) Prediction Entropy

Figure 4: Consensus loss and entropy of CM and RCM



(a) (b) (c)

Figure 5: Convergence of RCM

datasets. These evidences, together with the comparisons of consensus loss and entropy in Section 5.2, clearly demonstrate that CM overfits the data to produce highly consensus predictions, while RCM is able to trade-off between two objectives and achieves better accuracy.

Statistical significance of the results We verify that the improvements brought by the proposed method is statistically significant. According to [10], one can compare the effectiveness of different algorithms based on their performance on multiple datasets. Since among all baselines, CM has the closest performance to RCM, we compare these two methods using the Wilcoxon signed-ranks test. For the details of how to carry out the test, please refer to [10]. The test shows that RCM is statistically significantly superior to CM with $\alpha = 0.05$, where α is the probability that RCM is *not* better than CM.

5.4 Convergence Study

For each of the text classification tasks, we record the accuracy at the end of each iteration of RCM. In Figure 5, we plot the accuracies against the number of iterations. From the figure, we can see that, except for the *news3* and *dblp* tasks, RCM converges to some fixed accuracies. Even for those two exceptions where there are some zigzag's at the tails of the curves, we notice that the lowest accuracies obtained after the 25th iteration are at least the same as the best baseline (CM in both cases). Therefore, we conclude that given a big enough number of iterations, the algorithm performs better than or comparable with the baselines.

6. RELATED WORK

Graph partition based methods [27, 14]. In the pioneering work of [27], they presented three methods for cluster ensemble. For example, HGPA in [27] constructs a hypergraph consisting of membership indicators from clustering models as hyperedges, then a hypergraph partition algorithm partitions the hypergraph. Another method in [27], MCLA, partitions the hypergraph into k subgraphs, each of which consists of membership indicators from clustering models. These subgraphs are then used to calculate the association strength of an instance with subgraphs. HBGF is proposed in [14]. HBGF constructs the same bipartite graph as that in [16], then it partitions the graph using spectral clustering or METIS. Since the graph contains both group nodes and instance nodes (See Section 2.1), the results is a partition of both types of nodes. The partition of instance nodes is taken as the aggregated clustering. All these methods do not take discriminative constraints into account in their optimization formula.

Matrix factorization methods [18, 19]. They solve the consensus clustering problem through symmetric nonnegative matrix factorization (SNNMF). Using the predictions of base models, a similarity matrix is derived and factorized into orthonormal cluster membership indicators. The indicator matrices play the role of the consolidated predictions in CM. The orthonormality constraint on the indicators acts as the large margin regularization. Note that the orthonormality and the non-negative constraint are more restrictive than the large margin regularization proposed in this paper, as the consolidated predictions can have only

one entry as 1, with all the other entries being 0. We compare the proposed algorithm with the SNNMF formulation in Section 5.

Probabilistic methods [30, 1, 32, 20]. In [30], they solve the consensus clustering problem using a LDA like model. Predictions from base models are treated as documents in LDA, and the consolidated predictions as the latent topics of the documents. The method differs from LDA in that different models are assumed to have different topic-word distributions. In [1], they extend the above method in order to combine both supervised and unsupervised predictions in a transfer learning setting, which is different from the problem we are addressing here. In [32], they propose a non-parametric Bayesian method to select the number of clusters in consensus clustering. This algorithm is best employed as a parameter selection step before applying the method proposed in the paper.

In [2], they propose to combine supervised and unsupervised models as a way of knowledge transfer, where unsupervised models in the target domain serve as additional constraints while supervised models provide initial labeling. It might be important to give different weights to different base models, which might have different importance for the clustering task. In [34], they improve the performance of CM via functional space sampling. They impose weights on base models, where the weights are learned iteratively while seeking consensus results. In [18], weighted consensus ensemble is formulated as a sparse learning problem where the most important base models can be selected for consolidation. In [13], they provide a general framework for cluster ensemble called "Generalized Weighted Cluster Aggregation". Their goal is to find an affinity matrix that is close to all other affinity matrices derived from predictions of base models. Note that the proposed large margin formulation can be adopted by the above algorithms as a building block, and thus is not directly comparable.

Combining structural predictions has also been studied. For example, in multilabel classification, label correlations provide important information to achieve better classification performance. In [35], they propose a novel consensus classification algorithm to combine multi-label predictions from multiple base models, taking both label correlation and model consensus into account. Learning to rank is an important research problem in information retrieval, and recently, aggregating multiple ranking results is attracting more and more attention, due to its potential to improve ranking performance over single ranking model. The oldest ranking aggregation method called "Borda Couting", which can be traced back to 1770. The modern statistical ranking aggregation started with the Bradley-Terry (BT) model [7, 6], which infers the underlying ranking via maximum likelihood estimation. There are also many extensions of the BT model. For example, in [9], they extended the BT model by adding and learning weights on the base ranking models. In [33], they propose an online Bayesian learning algorithm for the BT model. In [25], they study the theoretical aspect of combining multiple ranking results into one ranking list. They present conditions under which certain popular ranking aggregation algorithms converge. Aggregating multiple ranking results has also found its place in gaming, such as XBox platform [17] In [31], they propose a Bayesian model to aggregate multiple visual trackers' output for reliable visual tracking. The proposed method in the paper focuses on combining flat predictions instead of structural predictions.

This work is also related to crowdsourcing, which aims at design mechanisms and algorithms to collect useful information from massive human population. Aggregating the data collected from multiple human beings is similar to combining the predictions given by multiple base models considered in this paper. In [26], they use crowdsourcing to obtain cheap annotations for NLP tasks, such as affect recognition, word similarity, etc. In [22], they propose to carry out multiple related crowdsourcing tasks simultaneously to alleviate data sparsity for a single crowdsourcing task. In [36], they propose to actively select annotator-instance pairs for human labeling. The idea is that by identifying the most uncertain instance and the corresponding most reliable annotator for the instance, one can learn underlying labels of the instances more effectively.

The algorithm proposed here is motivated by the maximum margin principle widely adopted in previous works [38, 28, 39]. In these works, to encourage discriminability, models are trained with the constraint that the prediction of a labeled instance should be closer to its true label than to other labels by some distance. However, these works focus on supervised learning, which is quite a different setting from the unsupervised setting here. There are a few works addressing overfitting in clustering [8, 21]. In [8], they analyze the issue from a learning theory perspective, and propose a general algorithm called "nearest neighbor clustering" to restrict the hypothesis space and avoid overfitting. The algorithm learns the similarity matrix for better spectral clustering results, and might be used to construct the affinity matrix in CM. These algorithms focus on clustering and do not address the model combination problem directly.

7. CONCLUSIONS

In this paper, we consider the overfitting issue in consensus maximization for model combination. The problem is analyzed by inspecting the hypothesis space and margin-based generalization error of CM. To solve the problem, we develop a model called class-distribution regularized CM that trades off two objectives, namely, consensus among base models and margins in predictions. The resulting optimization problem is challenging to solve since it involves many regularization parameters and is not jointly convex. We propose a simple and efficient approximation of the original problem, which can be solved using gradient descent. In the experiments, we compared the proposed method with CM and other baselines on 11 datasets, demonstrating the improvement due to the large margin regularization.

Acknowledgements

This work is supported in part by China 973 Fundamental R&D Program (No.2014CB340304), NSF grants (CNS-1115234, DBI-0960443, OISE-1129076, and IIS-1319973) and Huawei grant.

8. REFERENCES

[1] Acharya Ayan, Hruschka Eduardo, R., Ghosh Joydeep, Sarwar Badrul, and Ruvini Jean-David. Probabilistic combination of classifier and cluster ensembles for non-transductive learning. In *SDM*, 2013.

[2] Acharya Ayan, R. Hruschka Eduardo, Ghosh Joydeep, and Acharyya Sreangsu. An optimization framework for semi-supervised and transfer learning using multiple classifiers and clusterers. In *CoRR*, 2012.

[3] P. L. Bartlett. The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Trans. Inf. Theor.*, 2006.

[4] Asa Ben-Hur, David Horn, Hava T. Siegelmann, and Vladimir Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2002.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[6] Ralph Allan Bradley. Rank analysis of incomplete block designs: Ii. additional tables for the method of paired comparisons. *Biometrika*, 41(3/4):pp. 502–537, 1954.

[7] Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):pp. 324–345, 1952.

[8] Sébastien Bubeck and Ulrike von Luxburg. Nearest neighbor clustering: A baseline method for consistent clustering with arbitrary objective functions. *Journal of Machine Learning Research*, 2009.

[9] Xi Chen, Paul N. Bennett, Kevyn Collins-Thompson, and Eric Horvitz. Pairwise ranking aggregation in a crowdsourced setting. WSDM, 2013.

[10] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 2006.

[11] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *SIGKDD*, 2004.

[12] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l1-ball for learning in high dimensions. ICML, 2008.

[13] Wang Fei, Wang Xin, and Li Tao. Generalized cluster aggregation. In *IJCAI*, 2009.

[14] Xiaoli Zhang Fern and Carla E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. ICML, 2004.

[15] Jing Gao, Wei Fan, Deepak Turaga, Olivier Verscheure, Xiaoqiao Meng, Lu Su, and Jiawei Han. Consensus extraction from heterogeneous detectors to improve performance over network traffic anomaly detection. In *INFOCOM*, 2011.

[16] Jing Gao, Feng Liang, Wei Fan, Yizhou Sun, and Jiawei Han. Graph-based consensus maximization among multiple supervised and unsupervised models. In *NIPS*, 2009.

[17] Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill(tm): A bayesian skill rating system. NIPS.

[18] Tao Li and Chris Ding. Weighted Consensus Clustering. SDM, 2008.

[19] Tao Li, Chris Ding, and Michael I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. ICDM, 2007.

[20] Xudong Ma, Ping Luo, Fuzhen Zhuang, Qing He, Zhongzhi Shi, and Zhiyong Shen. Combining supervised and unsupervised models via unconstrained probabilistic embedding. IJCAI, 2011.

[21] Meila Marina and Shortreed Susan. Regularized spectral learning. *Journal of Machine Learning Research*, 2006.

[22] Kaixiang Mo, Erheng Zhong, and Qiang Yang. Cross-task crowdsourcing. KDD, 2013.

[23] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.

[24] H. Paugam-Moisy, A. Elisseeff, and Y. Guermeur. Generalization performance of multiclass discriminant models. In *Neural Networks, 2000. IJCNN 2000*, 2000.

[25] Arun Rajkumar and Shivani Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *ICML*, 2014.

[26] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. EMNLP, 2008.

[27] Alexander Strehl and Joydeep Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 2003.

[28] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. In *NIPS*. 2004.

[29] Vladimir Vapnik. *Statistical learning theory*. Wiley, 1998.

[30] Hongjun Wang, Hanhuai Shan, and Arindam Banerjee. Bayesian cluster ensembles. In *SDM*, 2009.

[31] Naiyan Wang and Dit-Yan Yeung. Ensemble-based tracking: Aggregating crowdsourced structured time series data. In *ICML*, 2014.

[32] Pu Wang, Carlotta Domeniconi, and Kathryn Blackmond Laskey. Nonparametric bayesian clustering ensembles. In *ECML PKDD*, 2010.

[33] Ruby C. Weng and Chih-Jen Lin. A bayesian approximation method for online ranking. *J. Mach. Learn. Res.*, 12, 2011.

[34] Sihong Xie, Wei Fan, and Philip S. Yu. An iterative and re-weighting framework for rejection and uncertainty resolution in crowdsourcing. In *SDM*, 2012.

[35] Sihong Xie, Xiangnan Kong, Jing Gao, Wei Fan, and Philip S. Yu. Multilabel consensus classification. In *ICDM*, 2013.

[36] Yan Yan, Romer Rosales, Glenn Fung, and Jennifer Dy. Active learning from crowds. ICML, 2011.

[37] Jinfeng Yi, Tianbao Yang, Rong Jin, A.K. Jain, and M. Mahdavi. Robust ensemble clustering by matrix completion. ICDM, 2012.

[38] Yi Zhang and Jeff Schneider. Maximum margin output coding. ICML, 2012.

[39] Jun Zhu, Amr Ahmed, and Eric P. Xing. Medlda: maximum margin supervised topic models for regression and classification. ICML, 2009.