Heterogeneous Source Consensus Learning via Decision Propagation and Negotiation *

Jing Gao†, Wei Fan‡, Yizhou Sun†, and Jiawei Han†
†Dept. of Computer Science, University of Illinois at Urbana-Champaign, IL USA
‡IBM T. J. Watson Research Center, Hawthorn, NY USA
jinggao3@illinois.edu, weifan@us.ibm.com, sun22@illinois.edu, hanj@cs.uiuc.edu

ABSTRACT

Nowadays, enormous amounts of data are continuously generated not only in massive scale, but also from different, sometimes conflicting, views. Therefore, it is important to consolidate different concepts for intelligent decision making. For example, to predict the research areas of some people, the best results are usually achieved by combining and consolidating predictions obtained from the publication network, co-authorship network and the textual content of their publications. Multiple supervised and unsupervised hypotheses can be drawn from these information sources, and negotiating their differences and consolidating decisions usually yields a much more accurate model due to the diversity and heterogeneity of these models. In this paper, we address the problem of "consensus learning" among competing hypotheses, which either rely on outside knowledge (supervised learning) or internal structure (unsupervised clustering). We argue that consensus learning is an NP-hard problem and thus propose to solve it by an efficient heuristic method. We construct a belief graph to first propagate predictions from supervised models to the unsupervised, and then negotiate and reach consensus among them. Their final decision is further consolidated by calculating each model's weight based on its degree of consistency with other models. Experiments are conducted on 20 Newsgroups data, Cora research papers, DBLP author-conference network, and Yahoo! Movies datasets, and the results show that the proposed method improves the classification accuracy and the clustering quality measure (NMI) over the best base model by up to 10%. Furthermore, it runs in time proportional to the number of instances, which is very efficient for large-scale data sets.

*The work was supported in part by the U.S. National Science Foundation grants IIS-08-42769 and BDI-05-15813, Office of Naval Research (ONR) grant N00014-08-1-0565, and the Air Force Office of Scientific Research MURI award FA9550-08-1-0265. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

KDD'09, June 28–July 1, 2009, Paris, France.

Categories and Subject Descriptors

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

General Terms

Algorithms

1. INTRODUCTION

As information networks become ubiquitous, the same set of objects can be represented and categorized by different information sources, which could be either the outside knowledge or the internal information. Example scenarios include (a) user viewing interest analysis based on their ratings of movies, movie synopses and movie genres; (b) buyers profiling based on their purchase history and personal information in e-commerce; (c) advertisement campaign based on the click rates, webpage content, advertisement content and location in the web page; and (d) research paper categorization based on paper content, citations and citation contexts. In these applications, we are interested in classifying a set of objects, and the predictive information comes from multiple information sources, each of which either transfers labeled information from relevant domains (supervised classification), or derives grouping constraints from the unlabeled target objects (unsupervised clustering). Multiple hypotheses can be drawn from these information sources, and each of them can help derive the target concept. However, since the individual hypotheses are diversified and heterogeneous, their predictions could be at odds. Meanwhile, the strength of one usually complements the weakness of the other, and thus maximizing the agreement among them can significantly boost the performance. Therefore, in this paper, we study the problem of consolidating multiple supervised and unsupervised information sources by negotiating their predictions to form a final superior classification solution. We first illustrate how multiple information sources provide "complementary" expertise and why their consensus produces more accurate results through a real example.

An Example–DBLP information network.

DBLP[1] provides bibliographic information on major computer science journals and proceedings, containing 654,628 authors and 4,940 conferences/journals. It is relatively easy to identify the fields of conferences/journals by their names, but much harder to label all the authors with their research interests. Therefore, we can use the conference labels, as well

[1]http://www.informatik.uni-trier.de/∼ley/db/

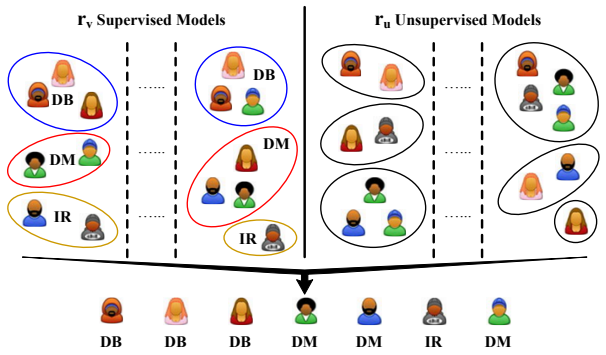

Figure 1: Consensus Learning Example

as, the connections among authors in the DBLP network to classify authors into different research fields. Specifically, the following information sources are available:

1) We can infer the area of a researcher from the conferences and journals s/he publishes in. For example, the researchers in the database community would mostly publish in the database conferences. However, this information source is not 100% reliable because it is not rare that people in the data mining field publish in database, machine learning, information retrieval or even networking, bioinformatics and software engineering conferences.

2) The co-authorship network provides useful information about the research fields of the researchers as well. If two authors often write papers together, they are more likely to share the same research field. But for researchers who have many cross-discipline cooperations, we may not be able to easily predict their research areas from this network.

3) When the information network fails to predict correctly, the textual content of the researchers' publications can play a role in the classification task. A researcher who concentrates on frequent pattern mining, graph mining and feature selection can be easily categorized into data mining community. However, some research fields may share the same keywords. For example, in papers of both database and information retrieval areas, we can find the words "query optimization", "indexing" and "retrieve", and thus a simple text classifier cannot distinguish among such areas.

For such problems, a simple concatenation of all the information sources may not be possible due to incompatible schemes, or fail because the structural information of some data sources can be lost. Furthermore, for applications in distributed computing, privacy preserving or knowledge reuse, we may only have access to the labeling or grouping results but not the raw data. Formally, for a classification task on a set of instances T, we may have r_v models that rely on outside knowledge (e.g., learn the areas of researchers from the areas of conferences), and r_u unsupervised models that group instances according to their similarity (e.g, the co-authorship network). We are aiming at consolidating predictions from all the models to boost the classification accuracy as well as improve the robustness and stability.

An example drawn from DBLP data is shown in Figure 1 where r_v supervised models predict the class labels of seven authors from information management area to be one of {Databases, Data Mining, Information Retrieval}, whereas the other r_u unsupervised models cluster the authors into cohesive groups. The objective is to find the global optimal labeling for the seven authors so that it agrees with the base models' outputs as much as possible. Existing approaches only pick the most likely label for each instance among supervised models without negotiation with the unsupervised sources, or combine the unsupervised grouping results ignoring the useful outside knowledge. As discussed above, all the information sources, no matter supervised or unsupervised, are important and only a global consolidation provides the optimal label assignments. To the best of our knowledge, this problem has not been studied before.

In this paper, we first formulate consensus learning into an optimization problem and argue that it is NP-hard. So we propose to solve the problem using an effective two-step heuristic method involving global decision propagation and local negotiation. In global propagation (Section 3.1), we first collapse test instances into groups based on the predictions of each model. We construct a belief graph where nodes represent the groups, and edge weights denote the percentage of common members they share, and start with the initial label assignments obtained from the supervised models. Each node iteratively propagates its prediction to its neighbors, and when it stabilizes, the groups that contain approximately the same set of instances would share the same predictions. In the second step (Section 3.2), to predict the class label of an instance, we make adjustments by negotiating among models locally with model weights reflecting the degree of consistency with others. We evaluated the proposed framework on four real learning tasks including 20 newsgroups categorization, Cora research paper classification, DBLP authors research areas categorization and Yahoo! movie-rating user grouping, where various learning models or information sources are available[2]. Experimental results show that the proposed method improves the classification accuracy as well as the clustering quality measure by up to 10% compared with the best base models. Moreover, both analysis and experimental results demonstrate that the running time of the solution is linear in the number of test instances, and thus it can be easily scaled to very large data sets without running into combinatorial explosions.

2. PROBLEM FORMULATION

We have a set $T = \{x_1, \ldots, x_n\}$ where x is the object ID and each object is represented in different information sources. We wish to predict the label of each example x in T: $y \in Y = \{1, \ldots, c\}$, where c is the number of classes. Suppose we have r_v classification models trained on the labeled sources, and r_u clustering methods relying on the internal structure of the test set, which can be obtained from different sources or using different algorithms. Let $r = r_v + r_u$, then we have r models: $\Lambda = \{\lambda^1, \ldots, \lambda^{r_v}, \lambda^{r_v+1}, \ldots, \lambda^r\}$, where the first r_v of them are supervised and the remaining ones are unsupervised. A supervised model λ^a $(1 \le a \le r_v)$ maps an instance x to a specific category $\lambda^a(x) \in Y$, whereas an unsupervised model maps it to a cluster and cluster ID does not directly carry any category information. In this paper, we focus on "hard" classification and clustering, i.e., x is predicted to be in exactly one class or cluster. Our aim is to use each model in Λ to find a "consolidated" solution λ^* on T, which maps $x \in T$ to one of the classes. It should agree with both the supervised and the unsupervised models as much as possible. Note that the true labels of examples

[2]At http://ews.uiuc.edu/~jinggao3/kdd09clsu.htm, there are experimental details, codes, data sets and additional experiment results.

in T are unknown, and thus the defined consensus learning problem is "unsupervised". The final predictions are derived based on the assumption that "consensus is the best", which proves to be valid in the experimental study in Section 4.

First, we favor the solution which maximizes the consensus. To define consensus, we need to first define the similarity or distance between two models' predictions on T. For the sake of simplicity, we use the following simple distance function. Consider two points x_i and x_t in T, and we define the disagreement between models λ^a and λ^b regarding their predictions on the two points as:

$$d_{x_i,x_t}(\lambda^a,\lambda^b) = \begin{cases} 0 & \text{if } \lambda^a(x_i)=\lambda^a(x_t) \text{ and } \lambda^b(x_i)=\lambda^b(x_t) \\ & \text{or } \lambda^a(x_i)\neq\lambda^a(x_t) \text{ and } \lambda^b(x_i)\neq\lambda^b(x_t) \\ 1 & \text{otherwise} \end{cases}$$

If λ^a and λ^b agree on x_i and x_t's cluster or class assignment, the distance is set to 0, otherwise to 1. Then we define the distance between λ^a and λ^b on T as the number of object pairs on which the two models disagree:

$$d(\lambda^a,\lambda^b) = \sum_{x_i,x_t\in T, i\neq t} d_{x_i,x_t}(\lambda^a,\lambda^b)$$

Therefore, one of our objectives is to minimize the disagreement with all the models: $\min_\lambda \sum_{a=1}^r d(\lambda,\lambda^a)$.

Secondly, the consolidated solution should be consistent with the predictions made by the supervised models. In other words, we need to minimize the difference between the consolidated solution and $\{\lambda^1,\ldots,\lambda^{r_v}\}$ on each x's label. Therefore, we add a penalizing term to the objective function and the consolidated solution λ^* satisfies:

$$\lambda^* = \arg\min_\lambda \left(\sum_{a=1}^r d(\lambda,\lambda^a) + \rho \sum_{a=1}^{r_v}\sum_{i=1}^n L(\lambda(x_i),\lambda^a(x_i)) \right) \quad (1)$$

where $0 \le \rho < \infty$ is the parameter to tune the contributions of the two parts, and $L(\lambda(x_i),\lambda^a(x_i))$ is the difference between the predictions made by λ and λ^a on x_i.

It can be seen that clustering consensus is a special case of the proposed framework with $\rho = 0$. Clustering consensus is shown to be NP-complete [8] based on the results of median partition problem [1]. We assume that there is at least one classifier and one clustering algorithm, and ρ is a finite number. So if the problem proposed in Eq. (1) can be solved in polynomial time, the clustering consensus problem will also be solved in polynomial time, which leads to contradiction. Hence the proposed optimization in this paper is NP-hard. Because we are tackling classification problems, the search space would be c^n, so an exhaustive search is formidable, and a greedy search would still have exponential time complexity and result in poor local maximum. For example, we would have to search 3^7 possibilities for the simple example shown in Figure 1 with 7 objects and 3 classes. Due to NP-completeness, we propose an effective heuristic in Section 3 to predict the class labels of examples in T with a linear scan of T. The solution represents the negotiation results among all the supervised and unsupervised models.

3. METHODOLOGY

We solve the problem through two steps:

- estimate $P(y|x,\lambda^a)$ $(1 \le a \le r)$, the probability of x belonging to class y according to one of the supervised or unsupervised models λ^a.

Table 1: Membership Vectors of Groups

Class/Cluster ID					Group Vectors											
					λ^1			λ^2			λ^3			λ^4		
	λ^1	λ^2	λ^3	λ^4	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}	g_{12}
x_1	1	1	2	1	1	0	0	1	0	0	0	1	0	1	0	0
x_2	1	1	2	2	1	0	0	1	0	0	0	1	0	0	1	0
x_3	1	2	1	3	1	0	0	0	1	0	1	0	0	0	0	1
x_4	2	2	3	1	0	1	0	0	1	0	0	0	1	1	0	0
x_5	3	2	3	2	0	0	1	0	1	0	0	0	1	0	1	0
x_6	3	3	1	1	0	0	1	0	0	1	1	0	0	1	0	0
x_7	2	1	3	1	0	1	0	1	0	0	0	0	1	1	0	0

- estimate $P(\lambda^a|x)$, the local weight of λ^a, which is proportional to the prediction accuracy of λ^a on x.

Based on results of the two steps, we compute $P(y|x,E)$ representing the consensus as:

$$P(y|x,E) = \sum_{a=1}^r P(y|x,\lambda^a)P(\lambda^a|x) \quad (2)$$

The predicted label for x goes to $\hat{y}$ which minimizes the risk: $\hat{y} = \arg\min_y \int_{y'\in Y} L(y',y)P(y|x,E)\mathrm{d}y'$, where $L(y',y)$ is the cost incurred when the true class label is y' but the prediction goes to y. With the most commonly used zero-one loss function, $\hat{y} = \arg\max_y P(y|x,E)$.

We hope that the final prediction $P(y|x,E)$ is close enough to the true but unknown $P(y|x)$, which we assume can be reached by consolidating the base model predictions. The challenges include: 1) When λ^a is an unsupervised model, it simply assigns x to one of the clusters but does not predict the category of x, so $P(y|x,\lambda^a)$ cannot be directly obtained. On the other hand, when λ^a is a classifier, we can set $P(y|x,\lambda^a) = 1$ when $\lambda^a(x) = y$ and 0 for all other y. However, this estimation is quite biased and we may want to modify it based on the negotiation with other models. 2) We expect that the weighting scheme can help reach the best consensus among all models. So ideally, $P(\lambda^a|x)$ should reflect the consistency of λ^a with other models on predicting x's label. We develop the following two heuristics that can solve the above problems effectively.

3.1 Model Predictions

Each model λ^a partitions T into c^a groups, and in the "hard" scenarios, an example x is a member of exactly one group if one of the supervised or unsupervised models λ^a is applied on T. Then $P(g_h^a|x,\lambda^a) = 1$ if x belongs to group g_h^a and 0 for all the other groups in λ^a. Therefore,

$$P(y|x,\lambda^a) = \sum_{l=1}^{c^a} P(y|x,g_l^a,\lambda^a)P(g_l^a|x,\lambda^a) = P(y|x,g_h^a)$$

Initially, there is a one-to-one mapping between each group g_h^a from a classifier λ^a $(1 \le a \le r_v,\ 1 \le h \le c^a)$ and each class label $y \in Y$, i.e., if λ^a predicts the label of the examples in group g_h^a to be y $(\lambda^a(x \in g_h^a) = y)$, then $\tilde{P}(y|x,g_h^a) = 1$. We treat the label information of groups from supervised models as initial labeling and estimate $P(y|x,g_h^a)$ for all the groups from all the models $(1 \le a \le r,\ 1 \le h \le c^a)$.

Altogether we have $s = \sum_{a=1}^r c^a$ groups: $\{g_1,\ldots,g_s\}$. Each group g can be represented by a length-n binary vector: $\{v_i\}_{i=1}^n$ where v_i is x_i's membership indicator [26]. For $x_i \in T$, its membership with respect to group g is 1 if $x \in g$ and 0 otherwise. The problem shown in Figure 1 is illustrated in the first four columns in Table 1 with two classifiers and

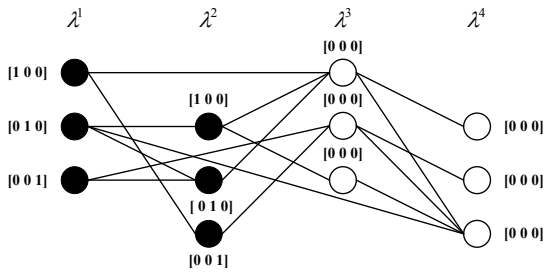

Figure 2: Illustration of the Label Propagation

two clustering methods, and {DB, DM, IR} is mapped to {1, 2, 3}. The binary membership vector of each group is shown in one of the 12 columns at the right side of Table 1. For example, the second group in λ^1 contains two examples x_4 and x_7, so the corresponding entries in g_2 are 1. Now we can measure the similarity between any two groups g_k and g_j. One commonly used measure is Jaccard coefficient:

$$J(g_k, g_j) = \frac{n_{k,j}}{n_k + n_j - n_{k,j}} \tag{3}$$

where n_k and n_j are the number of examples in g_k and g_j respectively, and $n_{k,j}$ is the number of examples in both g_k and g_j. For example, in Table 1, $J(g_4, g_8) = 2/3$ and $J(g_4, g_9) = 1/5$. Clearly, g_4 and g_8 are more similar with each other since they share more of the common members.

Now we can show the groups and their similarities using a belief graph $G = (V, E)$. Each node in V is a group g_k from a model λ^a. Each edge in E connecting two nodes g_k and g_j is weighted by the similarity between these two groups. If a group g is from a classifier λ^a $(1 \le a \le r_v)$, it has its initial class label predicted by λ^a. The labels of the groups from clustering models λ^a $(r_v + 1 \le a \le r)$ are not assigned. The graph constructed for the example in Table 1 is shown in Figure 2. For the sake of simplicity, we did not connect all the edges. Each of the six groups from λ^1 and λ^2 has its initial label (the black nodes). For example, g_1 is mapped to class 1 since the examples in g_1 are all predicted to be in class 1. So $\tilde{P}(y = 1|x, g_1) = 1$, whereas $\tilde{P}(y = 2|x, g_1)$ and $\tilde{P}(y = 3|x, g_1)$ are both 0. On the other hand, the groups from λ^3 and λ^4 are unlabeled (the white nodes), and we set their conditional distribution to [0 0 0] at first. Through this graph, we can propagate the conditional probability information from the groups in λ^1 and λ^2 (labeled nodes) to the groups in λ^3 and λ^4 (unlabeled nodes). The unlabeled nodes in turn change the predictions of their neighbors, labeled or unlabeled. When the propagation becomes stable, the groups with similar members would share similar $\hat{P}(y|x)$.

We now introduce the propagation method and analyze its optimality. Let $Q_{s\times c}$ be the matrix of conditional probability estimates we are aiming for, with each entry $Q_{kz} = \hat{P}(y = z|x, g_k)$. We set all the entries in Q to be zero initially. We define another $s \times c$ matrix F corresponding to the initial labeling from supervised models, where $F_{kz} = 1$ if $\tilde{P}(y = z|x, g_k) = 1$ and 0 otherwise. In other words, if g_k is from a supervised model, it will have 1 at the entry corresponding to its class label z. For unsupervised models, all the corresponding entries in F are 0. We construct the similarity matrix $W_{s\times s}$ with each entry W_{kj} equal to $J(g_k, g_j)$ as defined in Eq. (3), and compute a diagonal matrix D with its (k, k)-element equal to the sum of the k-th row of W. Let $H = D^{-1/2}WD^{-1/2}$ which normalizes W. Then we iterate $Q = \alpha HQ + (1-\alpha)F$ until convergence, where α is a parameter controlling the importance of the initial labeling. After the propagation stabilizes, we normalize Q so that each row of Q sums up to 1.

In fact, Q obtained from the propagation represents the minimum of the following objective function:

$$\frac{1}{2}\sum_{k,j=1}^{s} W_{kj}\sum_{z=1}^{c}\left(\frac{1}{\sqrt{D_{kk}}}Q_{kz} - \frac{1}{\sqrt{D_{jj}}}Q_{jz}\right)^2 + \mu\sum_{k=1}^{s}\sum_{z=1}^{c}(Q_{kz} - F_{kz})^2$$

In this objective function, we hope that the difference between the labels of two groups, g_k and g_j, would be as close as possible if their similarity W_{kj} is high. The second term penalizes the deviation from the initial label assignments for the groups from supervised models. We define the normalized graph laplacian as $L = D^{-1/2}(D - W)D^{-1/2} = I - H$. Due to the properties of graph laplacians, the above objective function equals to:

$$Q^T LQ + \mu(Q - F)^T(Q - F) \tag{4}$$

Differentiating the objective function in Eq. (4) with respect to Q to derive the optimal solution, we can get: $Q^* - HQ^* + \mu(Q^* - F) = 0$. By defining $\alpha = \frac{1}{1+\mu}$, we have $Q^* = (1 - \alpha)(I - \alpha H)^{-1}F$. To avoid computing a matrix inverse, we compute Q in an iterative way where $Q = \frac{1}{1+\mu}(HQ + \mu F) = \alpha HQ + (1 - \alpha)F$. It converges to Q^*, which is consistent with the initial labeling, and smoothes over the belief graph with nearby nodes sharing similar predictions.

The essence of this propagation method is that at each iteration, the conditional probability of each group g from the supervised model is the average of its close neighbors' probability estimates and the initial labeling. For example, the third group in λ^1's initial conditional probability [0 0 1] is smoothed to [0 0.06 0.94] because it is affected by the neighboring node with conditional probability [0 1 0]. On the other hand, the conditional probability of a group from the unsupervised models is only the weighted average of those from its neighbors since its initial label assignment is [0 0 0]. The propagation continues until all the nodes' predictions are stable, and then $\hat{P}(y|x, g_k)$ represents the results of negotiation among all the models with prior knowledge from the supervised models.

3.2 Model Weights

As introduced in the above section, we compute the conditional probability estimate at the group level, and we wish to adjust it using a local weight $P(\lambda^a|x)$ in the final solution. The optimal $P(\lambda^a|x)$ should reflect the prediction ability of λ^a on x where λ^a gets a higher weight if its prediction on x is closer to the true $P(y|x)$. The challenge is that the true $P(y|x)$ is not known a priori, so the optimal weights cannot be obtained. In this work, we only have access to predictions made by multiple models, but no groundtruth labels for the examples. So traditional cross-validation approaches cannot be used to compute the weights. From the "consensus" assumption, we know that the model that is more consistent with others on x's label tends to generate a more accurate prediction for x.

Therefore, we characterize the consistencies among models to approximate the model weight:

$$P(\lambda^a|x) \propto \frac{1}{r}\sum_{b=1,b\neq a}^{r} S(\lambda^a, \lambda^b|x)$$

$S(\lambda^a, \lambda^b|x)$ denotes the pair-wise similarity between λ^a and λ^b regarding x's label prediction, which represents the degree of consistencies between the two models. In other words, $\frac{1}{r}\sum_{b=1,b\neq a}^{r} S(\lambda^a, \lambda^b|x)$ is the average model accuracy of λ^a on x when we assume that $\lambda^1, \ldots, \lambda^{a-1}, \lambda^{a+1}, \ldots, \lambda^r$ is the correct model respectively. For models λ^a and λ^b, we rely on the local neighborhood structures around x with respect to the two models to compute $S(\lambda^a, \lambda^b|x)$. Suppose the sets of the examples that are in the same group with x in λ^a and λ^b are X^a and X^b respectively. If many examples are common among X^a and X^b, it is quite probable that λ^a and λ^b agree with each other on x's label. Hence the measure for the pair-wise local consistency can be defined as:

$$S(\lambda^a, \lambda^b|x) \propto \frac{|X^a \cap X^b|}{|X^a \cup X^b|} \tag{5}$$

In this computation, it doesn't matter whether λ^a and λ^b are supervised or unsupervised. We infer the label consistencies from x's neighbors according to the grouping results of λ^a and λ^b. As an example, let's compute $S(\lambda^1, \lambda^2|x_1)$ and $S(\lambda^1, \lambda^4|x_1)$ for the problem in Table 1. The neighbors of x_1 in λ^1, λ^2 and λ^4 are $\{x_2, x_3\}$, $\{x_2, x_7\}$ and $\{x_4, x_6, x_7\}$ respectively. So $S(\lambda^1, \lambda^2|x_1) \propto 1/3$ and $S(\lambda^1, \lambda^4|x_1) \propto 0/5$, which indicates that λ^1 agrees with λ^2 better on x. $P(\lambda^1|x)$ is then calculated as the average pairwise similarity between λ^1 and the other three models on x. According to the definition of $P(\lambda^a|x)$, it is obvious that when λ^a is more consistent with most of the other models on classifying x, its local weight is higher.

However, making the best local selection not always leads to global consensus. For certain examples, the most accurate label prediction may be attributed to the minority predictions. Therefore, we add a smoothing term to the model weight definition:

$$P(\lambda^a|x) \propto (1-\beta)\frac{1}{r}\sum_{b=1,b\neq a}^{r} S(\lambda^a, \lambda^b) + \beta\frac{1}{r} \tag{6}$$

Here, we assume that the model weight $P(\lambda^a|x)$ follows a mixture model of two components, where the "consensus model selector" values the local consensus among models, but the "random model selector" shows no preference to any model so that the majority and minority predictions have equal chances. β reflects our belief in this random selector compared with the consensus model selector. Finally, from Eq. (6) and the constraints that $\sum_{a=1}^{r} P(\lambda^a|x) = 1$, we can calculate the local weight of each model indicating its prediction power on x.

3.3 Time Complexity

In this part, we examine the method's time complexity. In the first step, the conditional probability of each group is learnt through propagation over the belief graph. The total number of groups is s, and each group can be represented using a binary vector, so the time to construct and normalize the similarity matrix W is simply $O(s^2)$. Suppose we have f iterations, then the propagation time is $O(fcs^2)$ where c is the number of classes. The normalization on the prediction results takes $O(s)$. The time of the second step is mainly attributed to the computation of pairwise local consistency $S(\lambda^a, \lambda^b|x)$, which can be computed in an efficient way. From Eq. (5), it can be derived that $S(\lambda^a, \lambda^b|x) = S(\lambda^a, \lambda^b|x')$ if x and x' are predicted to be in the same groups according to λ^a, and according to λ^b as well. We have $\frac{s(s-1)}{2}$ pairs of groups and we only need to calculate $S(\lambda^a, \lambda^b|x \in g_k, g_j)$ for these pairs of groups, so the time is $O(s^2)$. Note that till now, we work at the level of "groups" instead of "examples", and usually both the number of models and the number of classes or clusters are quite small (e.g., less than 10), so the computation can be very fast and the running time is independent of the number of examples. After that, we only need to check r models for each example to calculate the model weights and the weighted label prediction (combining results of step 1 and step 2). On the test set T with n examples, the complexity of this procedure is $O(rn)$, usually $n \gg r$. So the method runs in linear time with respect to the number of examples, and can scale well to large data sets.

Algorithm: Consensus Learning Algorithm
Input: (1) The classification or clustering results on a set T made by r_v classification models and r_u clustering models.
(2) Parameters α and β.
Output: Consolidated class label predictions for T.
Algorithm:

1. Construct the belief graph where each node is a group g from one of the models.
2. For each pair of groups,
   - Compute their similarity based on Eq. (3).
   - Compute their local consistency based on Eq. (5).
3. Let W be the group similarity matrix where $W_{kj} = J(g_k, g_j)$. Compute D as the diagonal matrix with (k,k) element equal to the sum of the k-th row of W. Let $H = D^{-1/2}WD^{-1/2}$.
4. Set $Q = 0$. Let F be the initial label matrix where $F_{kz} = 1$ if group g_k is predicted to be in class z by one of the classifiers.
5. Iterate $Q = \alpha HQ + (1-\alpha)F$ until convergence.
6. Normalize the row sums of Q to be 1.
7. For each example $x \in T$,
   - For each model, set $P(y|x, \lambda^a) = P(y|x, g_h^a) = Q_{ky}$ where g_h^a, indexed by k, is the group λ^a assigns x to, and compute $P(\lambda^a|x)$ based on Eq. (6).
   - Compute $P(y|x, E)$ based on Eq. (2).
   - Predict x's label as $\hat{y} = \arg\max_y P(y|x, E)$

Figure 3: Consensus Learning Algorithm

3.4 Algorithm

The proposed algorithm is summarized in Figure 3. As discussed, the two steps together help reach a consensus among r models. $P(y|x, \lambda^a)$ is calculated at a coarser level, whereas we further negotiate among models using $P(\lambda^a|x)$ targeted to each example. On the other hand, $P(y|x, \lambda^a)$ is computed globally by propagating the labeled information among all the groups. In the computation of $P(\lambda^a|x)$, only the local structure around x plays a role. So by combining different granularity of information and conducting the consolidation both globally and locally, we effectively solve the consensus learning problem. We conducted an exhaustive search among 3^7 possibilities for the problem in Table 1 and found the optimal solution is {DB, DB, DB, DM, DM, IR, DM}. The proposed method can successfully output the same solution only by one scan of the 7 instances.

4. EXPERIMENTS

We show that the proposed method is scalable, and can generate more accurate predictions compared with the baselines. Also, we can obtain conditional probability estimates for the examples even if the base models make "hard" decisions. The outputs can be used to summarize the characteristics of the underlying groups in the data.

4.1 Experiment Setup

Data Sets. We present results on four real-world applications. 1) 20 Newsgroups categorization: The 20 newsgroups data set[3] contains approximately 20,000 newsgroup documents, partitioned across 20 different newsgroups nearly evenly. We used the version where the newsgroup messages are sorted by date, and separated into training (60%) and test sets (40%). From the data sets, we construct 6 learning problems, each of which has documents from 4 different topics to distinguish. 2) Cora research paper classification [21]: The data set contains around 37,000 research papers that are classified into a topic hierarchy with 73 leaves. The citations among papers are around 75,000 entries. We conduct experiments on two top-level and two second-level classification problems where papers are from three to five different research fields. 3) DBLP network: We extracted two data sets from the DBLP network. The smaller one contains authors and conferences of four closely related areas in information management domain, whereas the larger one has seven broader areas. 4) Yahoo! Movies: The dataset is from the Yahoo! Alliance WebScope program[4] and it contains around 10,000 users, 14,000 movies and 220,000 ratings based on data generated by Yahoo! Movies on or before November 2003. We sampled two subsets from this data set where movies are from three different genres. We utilize the movie descriptive information (genre and synopsis) and movie ratings to derive the user type (which kinds of movies the user favors). Users are anonymous but the demographic information (birth-year and gender) of most users are available. More details about the above classification problems can be found in Table 2, where $|T|$ is the number of objects.

Baseline Methods. First, the proposed method is based on multiple single supervised and unsupervised models. We determine the base models according to each data set's characteristics. Since 20 Newsgroups data set only has text information, the base models are two classification (logistic regression and SVM, implemented in [11] and [4], denoted as **SC1** and **SC2**) and two clustering algorithms (K-means and min-cut, implemented in [15], denoted as **UC1** and **UC2**). In Cora, DBLP and Yahoo! Movies data sets, both the labeled set and the unlabeled target set T can be represented in two ways, which correspond to text and link information. On each of them, we train a logistic regression classifier on the labeled set and predict on T, as well as apply the K-means clustering algorithm on T. The two classification models are denoted as **SC1** (link) and **SC2** (text), and the two clustering models are **UC1** (link) and **UC2** (text). The two representations of these three data sets are as follows. Cora has paper abstracts as the text information. The link information is conveyed by the citation network where two papers are connected if one cites the other. So we can use the class labels of the neighboring nodes in the network as the link features of each paper. For the DBLP data set, we pool the titles of publications in a conference or by an author as their text features. On the other hand, we regard each conference as one dimension, and the number of papers an author published in the conference is the link feature value. In the Yahoo! Movies data sets, the movie ratings of the users act as the link information, and we collect the synopses of the movies a user rates greater than 3 out of 5 as the text features. In 20 Newsgroups and Cora, for the supervised models, outside knowledge comes from the domain the set T belongs to, whereas in DBLP and Movie data sets, supervised models are trained on a different domain (e.g., conferences vs. authors, movies vs. users).

Besides the single models, we compare the proposed method with the following ensemble methods. Note that we assume the raw data are inaccessible, and the proposed algorithm only takes outputs from multiple models as input. Therefore the baseline ensemble methods should also combine multiple models' outputs without referring to the original feature values. The output of each model is "hard", i.e., it only gives the predicted class label or cluster ID. We first map the clusters generated in one clustering model to match with those from the other model with the help of hungarian method[5]. Then, we learn two majority-voting based ensembles from the set of supervised and unsupervised base models separately, where ties are broken randomly. We denote them as Supervised Models Ensemble (**SME**) and Unsupervised Models Ensemble (**UME**) respectively. Also, we may ignore the class labels in the supervised models, regard all the base models as unsupervised clustering and try a clustering ensemble method to integrate all the partitionings. We use the Meta-Clustering Algorithm (**MCLA**) introduced in [26], where the final clustering solution is induced from the meta-clusters formed in a hyper-graph. Note that both UME and MCLA only perform clustering, and do not predict the class labels for the examples in T. We denote the proposed method which consolidates all the mod-

Table 2: Data Sets Description

Data	ID	Category Labels	$\|T\|$
News-group	1	comp.graphics comp.os.ms-windows.misc sci.crypt sci.electronics	1568
	2	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	1588
	3	sci.cypt sci.electronics sci.med sci.space	1573
	4	misc.forsale rec.autos rec.motorcycles talk.politics.misc	1484
	5	rec.sport.baseball rec.sport.hockey sci.crypt sci.electronics	1584
	6	alt.atheism rec.sport.baseball rec.sport.hockey soc.religion.christian	1512
Cora	1	Operating_Systems Programming Data_Structures_Algorithms_and_Theory	663
	2	Databases Hardware_and_Architecture Networking Human_Computer_Interaction	977
	3	Distributed Memory_Management Agents Vision_and_Pattern_Recognition	1468
	4	Graphics_and_Virtual_Reality Object_Oriented Planning Robotics Compiler_Design Software_Development	975
DBLP	1	Databases Data_Mining Machine_Learning Information_Retrieval	4236
	2	Databases_Data_Mining Networking Theory Artificial_Intelligence Information_Retrieval	21263
Yahoo! Movies	1	Drama Comedy Action_and_Adventure	7316
	2	Kids_and_Family Science_Fiction Musical	4176

[3]http://people.csail.mit.edu/jrennie/20Newsgroups/

[4]http://research.yahoo.com

[5]http://www.cs.umu.se/∼niclas/matlab/assignprob/

Table 3: Performance Comparison on a Series of Data Sets

Accuracy

Methods	20 Newsgroups						Cora				DBLP
	1	2	3	4	5	6	1	2	3	4	1
SC1	0.7966	0.8860	0.8557	0.8821	0.8756	0.8882	0.7738	0.8854	0.8617	0.8769	0.9325
SC2	0.7730	0.8615	0.8131	0.8666	0.8346	0.8571	0.7813	0.8588	0.8488	0.8821	0.8756
UC1	0.8061	0.8797	0.8652	0.8989	0.8718	0.9028	0.7647	0.8823	0.8535	0.8728	0.9379
UC2	0.7774	0.8571	0.8144	0.8477	0.8542	0.8565	0.7360	0.8598	0.7786	0.8923	0.7949
SME	0.7841	0.8739	0.8340	0.8742	0.8543	0.8727	0.7770	0.8707	0.8536	0.8792	0.9038
UME	0.7928	0.8680	0.8401	0.8736	0.8617	0.8798	0.7487	0.8713	0.8162	0.8832	0.8661
MCLA	0.7769	0.8755	0.8199	0.8619	0.8759	0.8532	0.8612	0.8724	0.8765	0.8584	0.9049
CLSU	**0.8469**	**0.9364**	**0.8856**	**0.9346**	**0.9034**	**0.9160**	**0.8854**	**0.9202**	**0.9012**	**0.9210**	**0.9525**

NMI

Methods	20 Newsgroups						Cora				DBLP
	1	2	3	4	5	6	1	2	3	4	1
SC1	0.4857	0.6736	0.6067	0.6666	0.6447	0.6898	0.3878	0.6623	0.7018	0.7021	0.7804
SC2	0.4494	0.6274	0.5270	0.6282	0.5652	0.6257	0.4241	0.6130	0.6914	0.7027	0.6306
UC1	0.5319	0.7114	0.6624	0.7270	0.7011	0.7388	0.4058	0.6541	0.7111	0.7071	0.7977
UC2	0.5028	0.7103	0.5845	0.6658	0.7016	0.6978	0.5115	0.6267	0.6927	0.7304	0.5019
SME	0.4661	0.6502	0.5646	0.6463	0.6018	0.6562	0.4021	0.6330	0.6912	0.7006	0.6979
UME	0.5010	0.6868	0.5943	0.6713	0.6725	0.7078	0.4041	0.6373	0.6504	0.7097	0.6253
MCLA	0.5267	0.7282	0.6385	0.6996	0.6943	0.7083	0.5639	0.7070	0.7384	0.7388	0.7407
CLSU	**0.5849**	**0.8028**	**0.6900**	**0.7856**	**0.7235**	**0.7618**	**0.6149**	**0.7488**	**0.7731**	**0.7856**	**0.8337**

els as Consensus Learning on Supervised and Unsupervised Models (**CLSU**). In the experiments, default parameters are used in the base packages and we set $\alpha = 0.4$, β around 0.5, and the number of iterations in the first step to be 20.

Measures. The instances in 20 Newsgroups and Cora data sets have their class labels. Moreover, we manually label the research fields of the authors in the first task of DBLP data set. For the purpose of evaluation, we restrict the number of clusters from the clustering algorithms to be the same as the number of classes. On these data sets, we evaluate the proposed method and the baselines from the following two perspectives: 1) We map the outputs of all the clustering algorithms to the best possible class predictions using hungarian method where cluster ids are matched with the class labels. Now all the methods have class label predictions for the examples in T, and thus we can evaluate their classification accuracy. Actually, this procedure on the clustering methods is "cheating" since the true class labels are used to do the mapping, and thus it should be able to generate the best accuracy from these unsupervised models. 2) All the methods, no matter classification or clustering, can group the test instances into c groups, so we can evaluate the clustering quality using the external measure–normalized mutual information (NMI) [26], averaged by the test set size. We construct a "true" model from the groundtruth labels, and compute the amount of information shared by the algorithms and the true model. A higher NMI indicates that the algorithm performs better on the data set. Due to the scale of the second DBLP data set and the anonymity of the Yahoo! Movies users, we cannot label these two test sets but simply show some examples or statistics of each group.

4.2 Empirical Results

In this section, we assess the performances of the proposed method in terms of accuracy and scalability.

Prediction Accuracy. Table 3 presents the experimental results on the 20 Newsgroups, Cora and DBLP data sets using accuracy or NMI as the performance measure. For the baseline ensemble methods (SME, UME, MCLA), the ties are randomly broken so we obtain their performance measures by averaging 50 runs. From the comparisons, we observe that: 1) On different data sets, the best single model, with respect to accuracy and NMI, can be different, which indicates that there exists large variability in the single models' predictions. 2) If only part of the information sources are used to construct an ensemble (SME, UME, MCLA), the performance may not always be improved due to the information loss. 3) The proposed CLSU method always outperforms all the base models and the baseline ensemble methods with a large margin in terms of both classification accuracy and clustering quality. We can see the consistent and often dramatic increase in performance measures on the 20 Newsgroups and Cora data sets (baselines' accuracy is mostly around 85%, but CLSU increases it to above 90%), as well as the DBLP data sets (the single models' accuracy is from 79% to 93%, and CLSU improves it to over 95%). It demonstrates the generalization accuracy and robustness of the proposed method. The success is attributed to the proposed method's wise negotiation among multiple information sources, which jointly make the accurate predictions.

Conditional Probability Estimates. The proposed consensus learning algorithm is also able to transform "hard" predictions of the base models to estimates of conditional probabilities. We selected some authors randomly sampled from the two DBLP tasks, and presented the probability of each author doing research in different areas in Table 4. The results are conducted on the subset of authors who publish in selected top conferences. The details about the research areas in the two tasks are shown in Table 2. These examples reveal that many authors are conducting research in multiple areas. It is very likely that the base models make different predictions about their areas, and thus the probability estimates generated by the proposed method are distributed among the areas they contribute to. For example, Jeffrey D. Ullman contributes to both databases and theory communities, and Andrew W. Moore's research is devoted to both data mining and machine learning areas. On the other hand, there are authors who mainly focus on one area, and the proposed method will assign a high probability to the area on which most of the models agree. Examples include Donald F. Towsley in networking and Michael Stonebraker in databases.

Group Summarization. In this experiment, we predict the distribution of a user's interests among the different movie genres in Yahoo! Movies data sets. Both the synopses and the ratings of the movies a user has watched provide use-

Table 4: Examples of $P(y|x)$ Estimates

Authors in DBLP1

NAME	DB	DM	ML	IR
Michael Stonebraker	**0.9473**	0.0107	0.0075	0.0345
Kian-Lee Tan	**0.7525**	0.1920	0.0372	0.0183
Nilesh Bansal	**0.5081**	0.1931	0.0375	0.2613
Ke Wang	0.0294	**0.8946**	0.0553	0.0207
Salvatore J. Stolfo	0.0280	**0.6583**	0.2937	0.0200
Evimaria Terzi	0.2687	**0.6558**	0.0549	0.0206
Andrew W. Moore	0.0288	0.4128	**0.5386**	0.0198
Boris Chidlovskii	0.2640	0.1931	**0.2817**	0.2612
Craig Boutilier	0.0129	0.0133	**0.9559**	0.0179
Sridhar Mahadevan	0.0299	0.1900	**0.7629**	0.0172
Barry Smyth	0.0136	0.0274	**0.4924**	0.4666
Akshay Java	0.0308	0.1896	**0.5209**	0.2587
W. Bruce Croft	0.0166	0.0284	0.0146	**0.9404**
S. K. Michael Wong	0.0155	0.0135	0.4786	**0.4924**
Xiaofei He	0.0329	0.1926	0.2844	**0.4901**

Authors in DBLP2

NAME	DB/DM	Network	AI	Theory	IR
Donald F. Towsley	0.0516	**0.8782**	0.0178	0.0257	0.0267
ChengXiang Zhai	0.0537	0.0140	0.2093	0.0048	**0.7182**
Richard M. Karp	0.0202	0.0141	0.0649	**0.8654**	0.0354
Jeffrey D. Ullman	**0.4974**	0.0134	0.0655	0.3805	0.0432
Ding-Zhu Du	0.0423	0.2152	0.0600	**0.6720**	0.0105
Hendrik Blockeel	0.3795	0.0120	**0.4778**	0.0108	0.1199
Gregory Chockler	0.0291	0.4369	0.0227	**0.4604**	0.0509
Lise Getoor	**0.6179**	0.0132	0.2371	0.0107	0.1211
Chidanand Apte	0.3779	0.0133	**0.4791**	0.0116	0.1181
Serge Abiteboul	**0.7531**	0.0201	0.0457	0.0074	0.1737
Raymond J. Mooney	0.1400	0.0121	**0.7206**	0.0116	0.1157
Judea Pearl	0.0355	0.0109	**0.7306**	0.1863	0.0367
Clement T. Yu	**0.5208**	0.0201	0.0472	0.0068	0.4051
Andrew McCallum	**0.3828**	0.0132	0.2387	0.0101	0.3552
Rong Jin	0.1478	0.0142	0.2390	0.0098	**0.5892**
Bharat K. Bhargava	0.4192	**0.4395**	0.0193	0.0210	0.1010

ful information for this task, and we build both supervised and unsupervised models over the two types of information and consolidate their predictions. We compute the probability of a user belonging to a movie genre group (such as Comedy). After that, we divide the users according to their demographic information: female or male, and age < 20, age between 20 and 40, and age > 40, and average the conditional probability estimates over the users of the same gender or age. Figure 4 reveals some interesting patterns we find in the user interest distributions. We can see that females love Drama and Comedy, whereas males' main interests are on Action movies. When people grow older, their interests gradually shift from Comedy to Drama. Regarding the distributions among Kids/Family, Science Fiction and Musical, females like the Musical movies much better than males, and people at ages 20 to 40 fall for Science Fiction movies the best, whereas teenagers have to watch Kids/Family movies a lot. Therefore, the proposed method can be applied to grouping of users for many of such services.

Scalability. As discussed in Section 3.3, the time complexity of the proposed method is quadratic in terms of the number of clusters and models, but linear with respect to the test set size. Since we usually deal with large-scale data sets which can be categorized into small groups, the running time is mostly determined by the number of instances, and thus the proposed method scales well to large data sets. We select four learning tasks, and randomly sample a subset from each set containing τ of the original instances ($\tau \in \{20\%, \ldots, 100\%\}$). The results are averaged over 50 runs and demonstrated in Figure 5. As most curves are linear especially when τ is greater than 60%, we can conclude that the results are consistent with our analysis that the proposed method has linear time complexity.

5. RELATED WORK

Many studies have shown that ensembles of multiple classifiers can usually reduce variance and achieve higher accuracy than individual classifiers [5, 13, 2]. These studies usually focus on deriving weak classifiers from data and boosting their performance by model combination. Their problem setting is different from what we discussed in this paper because they usually assume the availability of raw data. In unsupervised learning, study of clustering ensemble [26, 7, 12, 24, 18, 6] has been an active research area, whose objective is to produce a single clustering that agrees with multiple clusterings. The success of combining multiple models has been recognized when ensemble is shown to benefit from individual models as well as improve the accuracy and robustness. In fact, our method shares the same spirit as all the ensemble methods, but we extend the scope of base models to both supervised and unsupervised fields and try to find the best solution by negotiating their differences.

In recent years, an extensive body of work has crossed the boundary of supervised and unsupervised information sources. Semi-supervised or transductive learning [14, 27, 29] explores the use of unlabeled information to achieve better generalization on the unlabeled set. Particularly, label propagation [27] is used in our approach to propagate information over the belief graph. Link-based classification (i.e., collective inference, relational learning) [20, 22, 25] utilizes the link structure to classify a set of related instances simultaneously. These studies reveal that the unlabeled information, when used together with labeled instances, can produce considerable improvement in classification accuracy. However, they only take one unlabeled information source into account (e.g., manifold structure or link structure in unlabeled data set), but ignore the other possible unlabeled sources. People have investigated the problem of learning from two complementary views (co-training) [3] or multiple views (multiple view learning) [9]. Our proposed framework is more general than these studies in the sense that we do not require the labeled and unlabeled sources to be symmetric. Furthermore, we do not require access to raw data, but instead use prediction results from multiple models as input.

Some other types of information combination have also drawn researchers' attention, such as transfer learning ensemble [10, 19], webpages classification based on content and link information [28], label inference from two unlabeled data sources [17], and ensemble of relational classifiers [23]. However, all these methods only consider combining models in some specific formats. In a world with information explosions, we need a general framework that can take advantage of heterogeneous information sources. Li et al. [16] demonstrate that knowledge from the word space can be transformed to the document space to improve document clustering, however, the only information source used is the word co-occurrence matrix. In this paper, we show that for the task of knowledge transfer among variables of different types, information sources can be of many folds and a seamless consolidation of all the sources can outperforms ad-hoc combinations of part of the information sources.

6. CONCLUSIONS

In many applications, the class label of the same object can be inferred from multiple sources in different formats, such as graphs, text documents, user ratings, and click through rates. These heterogeneous information sources could be ei-

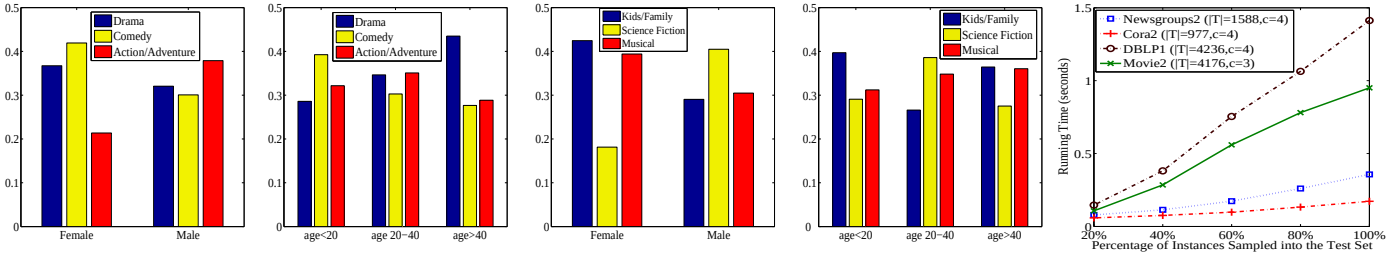

Figure 4: User Group Distributions on Yahoo! Movies Data Sets

Figure 5: Scalability of CLSU

ther supervised that contains labeled information of interest, or unsupervised that only contains structural similarity. In this work, we take advantage of different but complementary predictive powers of these heterogeneous sources to derive consolidated labels for a set of examples. This work extends the applicability of ensemble-based techniques to cross the boundary between labeled and unlabeled information by reaching and negotiating a consensus among them. This is different from traditional approaches of majority voting or model-averaging such that a minority label from supervised models or labels not even predicted by some supervised models could be the consolidated prediction. We presented a two-step heuristic method, which first uses a belief graph to propagate labeled information between supervised and unsupervised models for groups of examples with similar properties until stable predictions are reached. The final prediction is determined by negotiating among multiple models according to each example's neighborhood structure, and weighting models based on their consistencies with other models. On four data sets including 20 Newsgroups, Cora research papers, DBLP network and Yahoo! Movies, we have improved the best base model accuracy by 10%.

7. REFERENCES

[1] J. Barthelemy and B. Leclerc. The median procedure for partition. *Partitoning Data Sets, AMS DIMACS Series in Discrete Math.*, 19:3–34, 1995.

[2] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36:105–139, 2004.

[3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. of COLT' 98*, pages 92–100, 1998.

[4] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[5] T. Dietterich. Ensemble methods in machine learning. In *Proc. of MCS '00*, pages 1–15, 2000.

[6] C. Domeniconi and M. Al-Razgan. Weighted cluster ensembles: Methods and analysis. *ACM Trans. Knowl. Discov. Data*, 2(4):1–40, 2009.

[7] X. Z. Fern and C. E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proc. of ICML' 04*, pages 281–288, 2004.

[8] V. Filkov and S. Skiena. Integrating microarray data by consensus clustering. In *Proc. of ICTAI '03*, pages 418–426, 2003.

[9] K. Ganchev, J. Graca, J. Blitzer, and B. Taskar. Multi-view learning over structured and non-identical outputs. In *Proc. of UAI' 08*, pages 204–211, 2008.

[10] J. Gao, W. Fan, J. Jiang, and J. Han. Knowledge transfer via multiple model local structure mapping. In *Proc. of KDD' 08*, pages 283–291, 2008.

[11] A. Genkin, D. D. Lewis, and D. Madigan. Bbr: Bayesian logistic regression software. http://stat.rutgers.edu/~madigan/BBR/.

[12] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *ACM Trans. Knowl. Discov. Data*, 1(1), 2007.

[13] J. Hoeting, D. Madigan, A. Raftery, and C. Volinsky. Bayesian model averaging: a tutorial. *Statist. Sci.*, 14:382–417, 1999.

[14] T. Joachims. Transductive learning via spectral graph partitioning. In *Proc. of ICML' 03*, pages 290–297, 2003.

[15] G. Karypis. Cluto - family of data clustering software tools. http://glaros.dtc.umn.edu/gkhome/views/cluto.

[16] T. Li, C. Ding, Y. Zhang, and B. Shao. Knowledge transformation from word space to document space. In *Proc. of SIGIR' 08*, pages 187–194, 2008.

[17] C. X. Ling and Q. Yang. Discovering classification from data of multiple sources. *Data Min. Knowl. Discov.*, 12(2-3):181–201, 2006.

[18] B. Long, Z. Zhang, and P. S. Yu. Combining multiple clusterings by soft correspondence. In *Proc. of ICDM' 05*, pages 282–289, 2005.

[19] P. Luo, F. Zhuang, H. Xiong, Y. Xiong, and Q. He. Transfer learning from multiple source domains via consensus regularization. In *Proc. of CIKM' 08*, pages 103–112, 2008.

[20] S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *J. Mach. Learn. Res.*, 8:935–983, 2007.

[21] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval Journal*, 3:127–163, 2000.

[22] J. Neville and D. Jensen. Relational dependency networks. *J. Mach. Learn. Res.*, 8:653–692, 2007.

[23] C. Preisach and L. Schmidt-Thieme. Ensembles of relational classifiers. *Knowl. Inf. Syst.*, 14(3):249–272, 2008.

[24] W. Punch, A. Topchy, and A. K. Jain. Clustering ensembles: Models of consensus and weak partitions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1866–1881, 2005.

[25] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. Technical Report CS-TR-4905, University of Maryland, College Park, 2008.

[26] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, 2003.

[27] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. SchOlkopf. Learning with local and global consistency. In *Proc. of NIPS' 04*, pages 321–328, 2004.

[28] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *Proc. of SIGIR' 07*, pages 487–494, 2007.

[29] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.