Asian Journal of Information Technology 5 (3) : 312-316, 2006

Web Information Clustering by Personal Search Engine Based on SVM

[1-3]Wang deji, [3]Li mincheng and [1]Xiong fanlun
[1]Institute of Intelligent Machines of Chinese Academy of Science, Hefei 230031,China
[2]Institute of Information Science, University of Science and Technology of China, Hefei 230026, China
[3]Training Centre for Staff and Workers of CNTC, Zhengzhou 450008, China

Abstract: Web information is scaling more than exponentially with time. How to acquire information efficiently by personal search engine is staring us in our faces. Personal preference can not be easily described but can be learned quickly from the examples. Although PCC (pairwise classification clustering) is a powerful tool for learning the examples, but transitive dependences dwarf it. In this paper, we introduce clustering with SVM and define semantic cosine similarity based ontology to solve this problem. Experiments proof that it is efficient and powerful.

Key words: SVM, PCC, information acquisition, ontology

INTRODUCTION

How to get information by your preference is an important thing because too much information is on the web. For example, a person wants to get news either by topic, or author, or by language, etc. However, the current information acquisition research, which includes the SVM-decision tree and unsupervised clustering based SVM, can not satisfy this requirement. Of course, supervised clustering is the best way to tackle the problem, but it may not produce desirable clustering without additional information by the user. What we can do is to adjust algorithm or similarity measure. Compared with the adjusting algorithm, modifying the similarity measure has some intuitive appeal. Unfortunately, a person often can not easily specify the similarity measure but can give some example. So to learn the similarity measure for clustering is a wise choice.

The common way is to use a binary classifier (PCC). Take all pairs of items in all training sets and describe each pair in terms a feature vector. Let positive examples be the same class and negative examples be the different. When a new set of items is run though the classifier, whether a pair should or should not be in the same class can be decided by the output value (positive or negative). But the approach assumes that all the pairs are i.i.d and can not take advantages of dependencies between item pairs. To overcome this kind of problem, some research has employed heuristics to train the classifier. However, this approach is built with expert domain knowledge and is not applicable to other tasks. To avoid this problem, some researchers have adopt CRFs(Conditional Random Fields), which uses a variety of clustering functions and does not require the independence of attributes, but can not optimize the clusters with respect to loss function. Our supervised clustering with SVM is closely to this, except ours is motivated by a maximum margin approach rather CRFs.

Supervised Learning and Support Vector Machines: Given some examples we wish to predict certain properties, in the case where there are available a set of examples whose properties have already been characterized the task is to learn the relationship between the two. One common early approach was to present the examples in turn to a learner. The learner makes a prediction of the property of interest, the correct answer is presented and the learner adjusts its hypothesis accordingly. This is known as learning with a teacher, or supervised learning.

In this method, we want to find the approach to get the desirable clustering by the complete clustering of the example WebPages.

The learning algorithm receives a set S of n training examples $(x_1, y_1), \ldots, (x_n, y_n) \in X \times Y$, all drawn from a distribution P(X, Y). X is the set of all possible sets of items and Y is the set of all possible clusterings (partitionings) of these sets. For any (x, y), $x = \{x_1, x_2, \ldots, x_m\}$ is a set of m WebPages. And $y = \{y_1, y_2, \ldots, y_c\}$ with y_i ? x is the partitioning of x into c clusters. The goal is to learn a clustering function h that can accurately cluster new WebPages.

$$h: X \rightarrow Y \quad (1)$$

Given a loss function that compares two clusterings

Corresponding Author: Wang deji, Institute of Intelligent Machines of Chinese Academy of Science, Hefei 230031,China

$$\Delta: X \times Y \rightarrow R, \quad (2)$$

the training error for a clustering function h on an example (x, y) is $\Delta(h(x), y)$.
The goal is to find h to minimize risk

$$Err_p(h) = \int_{X^*Y} \Delta(h(x),y)dP(x,y) \quad (3)$$

So the goal is consistent with the SVM and we can introduce it into the clustering.

The Support Vector Machine (SVM), is a training algorithm for learning classification and regression rules from data, for example the SVM can be used to learn polynomial, Radial Basis Function (RBF) and Multi-layer Perception (MLP) classifiers. SVMs are based on the structural risk minimization principle, closely related to regularization theory. This principle incorporates capacity control to prevent over-fitting and thus is a partial solution to the bias-variance trade-off dilemma. The standard SVM is 2 outputs, but our clustering is a multi outputs. The $SVMs_{truct}$ is suitable for it. Before introduce it, we describe the clustering index-semantic cosine similarity for the clustering and its models.

Semantic cosine similarity: Cosine similarity and Vector space model (TFIDF) are often employed as the indexes for clustering. But they ignore the semantic relation of key words. Now we present a novel method named semantic cosine similarity (SCS) based on ontology.

Step 1: We Setup the specific knowledge domain with knowledge categorization.

The knowledge domain is task-oriented and can be modeled by a task-oriented ontology. There exist standard domain vocabularies that are familiar to the major classes. Fragmented exemplar of task-oriented ontology is as fellows:

Step 2: Identify in x_i (the ith news article) the occurrences of the concept nodes x_{ip} (p-numbered concept node of the task oriented ontology; x_{ip}-the pth node encounters in the news article xi). Translate xi with the approach of semantic imposition into a term vector

$$x_i = (x_{0i}, x_{0i}, x_{0i} \ldots . x_{0i})$$

For a given task-oriented ontology, assume there are m nodes e.g., 20 nodes (Fig. 1.) Therefore, any news document (x_i) is to be represented as a sized-20 vector of. This vector is initialized with a vector of zeros. The value of is increased by 1 when encountering an existence of the p-numbered concept node in the news document. For instance, the term vector of a news document (containing only node 20) unfolds as (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1). When imposing the semantic (implied by the ontology) into a vector, all of the parent nodes of 1-valued are accounted as existence and thus their values are increased by 1 as well, unfolding a vector of more nonzero terms. Using the last example of document of node 20, the resulting vector becomes (0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,1,0,1,1) as the parent nodes of node 20 include node 4, 8, 12, 17, 19, 20. Afterward, we name this method of imposing semantic as the semantic imposition. Term value for the p-numbered concept, the number is

$$sum = X_{ip} {}^{*} g_P {}^{*} S_p \quad (4)$$

x_{ip}: a number resulting from the approach of semantic imposition (applied to the document x_i) for the p-numbered concept node of the task oriented ontology.

$$g_p = 1, p = 0; 1; 2; \ldots; m$$

Normalization component $S_i = (\sum (x_{ip}g_p)^2)^{-\frac{1}{2}}$ (5)

Step 3: Repeat Step 2) until all of the input documents are translated.

Step 4: Compute the similarity value between and two term vectors (x_a, x_b) with the measure of cosine

similarity: $\Phi(x_a, x_b) = x_a * x_b / \| x_a \|_2 * \| x_b \|_2$ (6)

Step 5: Repeat Step 3) until the similarity values of all the pairs of documents are calculated.

Model: In this supervised clustering method, we hold the clustering algorithm constant and modify the similarity measure so that the clustering algorithm produces desirable clustering

Each article has 3 Semantic cosine vectors and each stands for the similarity for the headline ,article text and article text in quotations of the two article, The pairwise feature vector $\varphi_{1,2}$ for two articles x_1, $x_2 \in x$ are the 3 semantic cosine similarities between these entities corresponding vectors in x_1 and x_2, plus one feature which is always the constant

Define Sim_w-similarity measure, maps pairs of items to a real number, which indicate how similar the pair is positive values indicate the pair is alike, negative values, unlike.

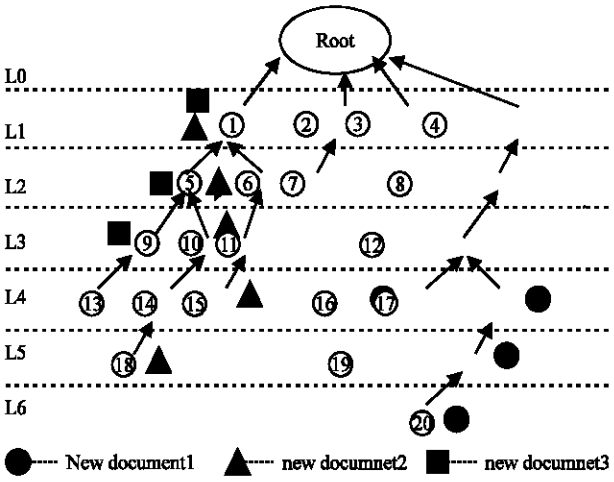

Fig. 1: Fragmented exepler of task-oriented ontology

Define: $\Phi(x_1, x_2) \equiv \Phi_{1,2} \in x_1, x_2 \in X$ they are a pair.
Define: W-weight parameters vector
$Sim_w(x_1, x_2) = W^T\Phi_{1,2}$ (7)

Rules: the correlation clustering of a set of items X is the clustering Y maximizing the sum of similarities for item pairs in the same cluster.

$$\arg\max_{Y} \sum_{y \in Y} \sum_{x_1, x_2 \in y} Sim_W(x_1, x_2) \quad (8)$$

$$= \arg\max_{Y} \sum_{y \in Y} \sum_{x_1, x_2 \in y} W_T \phi(x_1, x_2), \quad (9)$$

$$= \arg\max_{Y} W_T \left(\sum_{y \in Y} \sum_{x_1, x_2 \in y} \phi(x_1, x_2) \right) \quad (10)$$

Algorithm for clustering with SVM: The $SVMs_{truct}$ satisfies the below:

$$\min_{W, \xi} \frac{1}{2} \| W \|^2 + C \sum_{i=1}^{n} \xi_i \quad \text{s.t.} \quad \forall i : \xi_i \geq 0 \quad (11)$$

$$\forall i, \forall y \in Y \setminus y_i :$$

$$W^T\Phi(x_i, y_i) \geq W^T\Phi(x_i, y) + \Delta(y_i, y) - \xi_i \quad (12)$$

Expression (11) is typical SVM quadratic objective and slack constraints.

Expression (12) expresses the set of constraints that allow us to learn the desired hypothesis.

Where lack norm is 1. Loss acts as the Margin.

$\Delta(\hat{y}, y)$: Loss between a true cluster y and a predicted one $\hat{y}$

Where, we chose $\Delta = 100(W/T)$. T is the total number of pair of item in the set partitioned by y and $\hat{y}$. W is the total number of a pairs. Where y and $\hat{y}$ disagree about their cluster membership.

$W^T\Phi(x_i, y)$ is the correlation clustering objective.

SVM finds the vector W to make $W^T\Phi(x_i, y)$ is max for the correct y.

For

, so $\quad Errs(h) = \frac{1}{n} \sum_{i=1}^{n} \Delta(y_i, h(x_i)) \leq \frac{1}{n} \sum_{i=1}^{n} \xi_i$

$\sum_{i=1}^{n} \xi_i$

upper bounds the training loss.

Input: $(x_1, y_1) \ldots (x_n, y_n)$, C, ε
S_i (i = 1 ... n) ←, w, ξ←0
Repeat
for i = 1, . . . , n
$H(y) = \Delta(y_i, y) - W^T\Phi(x_i, y_i) + W^T\Phi(x_i, y)$

Compute $\hat{y} = \arg\max_{y \in Y} H(y)$ // find the most violated constraint
compute $\xi_i = \max\{0, \max_{y \in S_i}(H(y))\}$

if $H(\hat{y}) > \xi_i + \varepsilon$ then //violated by more than ε

$S_i \leftarrow S_i \cup \{\hat{y}\}$ //add constraint to working set
w← optimize primal over S
end if
end for
Until no S_i has changed during iteration
By solving ,

$$\hat{y} = \arg\max_{y \in Y} H(y)$$

the algorithm finds the clustering $\hat{y}$ associated with the most violated constraints for (x_i, y_i). Since H is the minimum necessary

slack $\hat{y}$ for under the current W◦

if $H(\hat{y}) > \xi_i + \varepsilon$, the constraint is violated by more than ε, so we introduce the constraint and re-optimize. The algorithm repeats this process until no new constraints are introduced◦

EXPERIMENTS

SVM cluste r versus PCC: Parameters: C=1, ε=0.01◦
Computer: CPU-Pentium4 2.0GHz/memory-1GB/video

Table 1: Results for No transitive dependences news articles

	Sport news		Political news		Economical news	
Approach	precision	CPU consuming	precision	CPU consuming	precision	CPU consuming
PCC	98.60%	130 Min.	89.48%	145 Min.	89.60%	140 Min.
SVM clustering	96.53%	135 Min.	91.97%	150 Min.	92.91%	143 Min.

Table 2: Results for transitive dependences news articles

	Sport news		Political news		Economical news	
Approach	precision	CPU consuming	precision	CPU consuming	precision	CPU consuming
PCC	78.64%	133 Min.	69.33%	142 Min.	69.64%	136 Min.
SVM clustering	95.53%	140 Min.	90.97%	150 Min.	90.48%	158 Min.

card-GeForce6600GT/ hard disk-80GB.The news article clustering data set is a new data set we derived by trawling DMRESEARCH News. DMRESEARCH News itself works by clustering news articles, but presumably their clustering method is sufficiently sophisticated that teaching an unsophisticated clustering method how to cluster in the same fashion is interesting. For each day for 30 days, at most 10 topics from the sports politics economics category were selected and from each topic at most 15 articles were selected. The topics form our true reference clusters. We have various simple heuristics for extracting the article text, quoted article text, headline and title. The first 15 days are the training set and the last 15 days are the test set.

From Table1 and Table 2 ,we can see that SVM cluster is more effective than the PCC approach when the data contains transitive Supervised Clustering with Support Vector Machines and that both methods perform comparably when not. We also can see that the sport news can be easily classified than the others because the sport news terms have less different meanings than those of the others.

Efficiency of SVM cluster: Of all the reported experiments, the time that SVMcluster took to converge was between2 and 3 hours, while the PCC used less times.

CONCLUSION

We formulated a supervised clustering method SVM cluster based on an SVM framework for learning structured outputs. The algorithm accepts a series of training clusters, a series of sets of items and clusterings over that set. The method learns a similarity measure between item pairs to cluster future sets of items in the same fashion as the training clusters. Supervised Clustering with Support Vector Machines The learning algorithm`s correctness depends on an ability to iteratively find and introduce the most violated constraint.

Overall, it suggests that SVM cluster is more effective than the nave PCC approach when the data contains transitive Supervised Clustering with Support Vector Machines and that both methods perform comparably when not.

REFERENCES

1. Finley, T. and T. Joachims, 2005. Supervised clustering with support vector machines, Proceedings of the International Conference on Machine Learning (ICML).
2. Bamshad, M., C. Robert, S. Jaideep, 2003. Automatic Personalization Based on Web Usage Mining¬ Communications of the ACM.
3. http://www.dmresearch.net
4. Mitchell. T., 1997. Machine Learning. McGraw-Hill International.
5. Basu, S., M. Bilenko and R.J. Mooney, 2004. A probabilistic framework for semi-supervised clustering. ACM SIGKDD-2004. Seattle, WA, pp: 59-68
6. Bilenko, M., S. Basu, and R.J. Mooney, 2004. Integrating Constraints and Metric Learning in Semisupervised Clustering. ICML. New York, NY, USA:ACM Press.
7. Cohen, W. and J. Richman, 2001. Learning to match and cluster entity names. ACM SIGIR workshop on Mathematical/Formal Methods in IR.
8. De Bie, T., M. Momma and N. Cristianini, 2003. Efficiently learning the metric using side-information. ALT. Sapporo, Japan: Springer, pp: 175-189.
9. Demaine, E. and N. Immorlica, 2003. Correlation clustering with partial information. RANDOMAPPROX. Princeton, New Jersey, pp: 1-13
10. Joachims, T., 2003. Learning to align sequences: A Maximum-margin Approach (Technical Report).

11. Kamishima, T. and F. Motoyoshi, 2003. Learning from cluster examples. Mach. Learn., 53: 199-233.
12. Lanckriet, G.R.G., N. Cristianini, P. Bartlett, L.E. Ghaoui and M.I. Jordan, 2004. Learning the kernel matrix with semidefinite programming. J. Mach. Learn. Res., 5: 27-72.
13. Dagan I., L. Lee and P. Pereira, 1999. Similarity-based models of word cooccurrence probabilities, Machine Learning, Special issue on Machine Learning and Natural Language.
14. http://www.keenage.com
15. Agirre E. and G. Rigau, 1995. A proposal for word sense disambiguation using conceptual distance, in International Conference Recent Advances in Natural Language Processing RANLP'95, Tzigov Chark, Bulgaria.
16. Zhu, lijun, 2004. 6 dissertation of Chinese agriculture university , Research of Domain Knowledge Based Information Resource Management Pattern In World Wide Web Environment.