Learning with Support Vector Machines for Query-By-Multiple-Examples

Dell Zhang
SCSIS
Birkbeck, University of London
London WC1E 7HX, UK
dell.z@ieee.org

Wee Sun Lee
Department of Computer Science
National University of Singapore
Singapore 117590
leews@comp.nus.edu.sg

ABSTRACT

We explore an alternative Information Retrieval paradigm called Query-By-Multiple-Examples (QBME) where the information need is described not by a set of terms but by a set of documents. Intuitive ideas for QBME include using the centroid of these documents or the well-known Rocchio algorithm to construct the query vector. We consider this problem from the perspective of *text classification*, and find that a better query vector can be obtained through learning with Support Vector Machines (SVMs). For online queries, we show how SVMs can be learned from *one-class examples* in *linear* time. For offline queries, we show how SVMs can be learned from *positive and unlabeled examples* together in *linear* or *polynomial* time. The effectiveness and efficiency of the proposed approaches have been confirmed by our experiments on four real-world datasets.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Performance, Experimentation

Keywords

Support Vector Machine, One-Class Learning, PU Learning.

1. PROBLEM

We explore an alternative Information Retrieval paradigm called Query-By-Multiple-Examples (QBME): given a set of documents $P = \{\mathbf{x}_1, \dots, \mathbf{x}_l\}$ as query, retrieve/rank the documents in a corpus $U = \{\mathbf{x}_{l+1}, \dots, \mathbf{x}_{l+u}\}$ according to their relevance to P. The problem of QBME occurs frequently in practice. This is because only relevant documents are usually stored, and it is often desirable to find more relevant documents. For example, a researcher may have saved in her computer some journal articles on a subtopic in bioinformatics (P), and she wants to find more materials on that subtopic from the PubMed Central digital library (U).

In this paper, we distinguish between two types of queries: *online* queries where response is required immediately and *offline* queries where the user is willing to wait in order to get better results. For online queries, we restrict ourselves to using only P because of efficiency consideration. For offline queries, it may be feasible to use U in addition to P to improve retrieval performance.

2. APPROACHES

Taking the classic Vector Space Model [5], we represent every document as a *normalized* document vector, and seek to best describe P as a query vector $\mathbf{w}$ such that all the documents in U can be ranked appropriately by a linear function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. The intuitive idea to solve the problem of QBME is to use the *centroid* of P, $\mathbf{w} = \frac{1}{l}\sum_{i=1}^{l}\mathbf{x}_i$, as the query vector for online processing, or to use the simplified *Rocchio* algorithm [5] to construct the query vector $\mathbf{w} = \frac{1}{l}\sum_{i=1}^{l}\mathbf{x}_i - \frac{1}{u}\sum_{j=1}^{u}\mathbf{x}_j$ for offline processing. However, given that we have a set of relevant documents which should be more informative than just keyword queries, we may hope to obtain a better query vector by considering this problem from the perspective of *text classification*. Actually QBME crosses the traditional boundary of retrieval and classification. Support Vector Machine (SVM) [6] in its simplest form, linear SVM, consistently provides state-of-the-art performance for various text classification tasks.

2.1 Learning from P Only

We propose to use the following one-class SVM formulation SVM_{1c}^{struct} to construct the query vector $\mathbf{w}$ taking P as training examples. It can be shown by adapting the proof from [3] that SVM_{1c}^{struct} is equivalent to the standard linear SVM using only positive examples. Moreover, unlike the original one-class SVM formulation [6] that requires quadratic time for training, SVM_{1c}^{struct} can be trained by the *cutting-plane algorithm* in linear time w.r.t. $|P| = l$ [2].

OP 1. SVM_{1c}^{struct}

$$\begin{aligned} \min_{\mathbf{w},\xi\geq 0} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\xi \\ s.t. \quad & \forall \bar{\eta} \in \{0,1\}^l : \\ & \frac{1}{l}\mathbf{w}^T\sum_{i=1}^{l}\eta_i\mathbf{x}_i \geq \frac{1}{l}\sum_{i=1}^{l}\eta_i - \xi \end{aligned}$$

SIGIR'08, July 20–24, 2008, Singapore.
ACM 978-1-60558-164-4/08/07.

2.2 Learning from P and U

If we have not only a set of relevant documents P, but also a set of irrelevant documents N, we can use the standard linear SVM to construct the query vector $\mathbf{w}$ taking P and N as *positive* and *negative* examples respectively. However, what we have in addition to P is only the *unlabeled* corpus U where relevant and irrelevant documents are mixed. This problem of training a classifier with positive and unlabeled examples is called *PU learning* [4]. Our solution is to take the relevant documents in U as noise thus U can be considered as a very *noisy* set of negative examples. Denote the *observed* label of an example $\mathbf{x}$ by y, i.e., $\forall \mathbf{x}_i \in P : y_i = 1$ and $\forall \mathbf{x}_i \in U : y_i = -1$. Denote the *actual* label of an example $\mathbf{x}$ by z that indicates its true relevancy. Unfortunately, the standard linear SVM that minimizes the observed error rate does not guarantee to achieve the minimal actual error rate [1]. Nevertheless, under a reasonable assumption that the documents in P are *randomly* sampled from the class of relevant documents with a certain probability μ, we can prove that optimizing the observed values of the following two performance measures is in fact equivalent to optimizing their corresponding actual values.

Balanced Accuracy. The *balanced accuracy* a.k.a. the *AUC for just one run* of a classifier is the average of its *sensitivity* and *specificity*. With some simple calculation [1] we can see that the observed balanced accuracy $\widehat{B}$ is connected to the actual balanced accuracy B via $\widehat{B} - \frac{1}{2} \propto B - \frac{1}{2}$. It can be shown by adapting the proof from [3] that the following SVM formulation $\mathrm{SVM}_{ba}^{struct}$ minimizes the cost function $\Delta_{ba}(\bar{h}(\bar{\mathbf{x}}), \bar{y}) = 1 - \widehat{B}$. Moreover, $\mathrm{SVM}_{ba}^{struct}$ can be trained by the *cutting-plane algorithm* in linear time w.r.t. $|P \cup U| = l + u$ [2].

OP 2. $\mathrm{SVM}_{ba}^{struct}$

$$\begin{aligned} \min_{\mathbf{w},\xi\geq 0} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\xi \\ s.t. \quad & \forall \bar{\eta} \in \{0,1\}^n \setminus \bar{0} : \\ & \frac{1}{n}\mathbf{w}^T \sum_{i=1}^{n} \eta_i y_i \mathbf{x}_i \geq \frac{1}{n}\sum_{i=1}^{n} \eta_i \lambda_i - \xi \\ & \lambda_i = 1/(4l) \text{ if } y_i > 0 \text{ and } \lambda_i = 1/(4u) \text{ if } y_i < 0 \end{aligned}$$

Precision-Recall Product. An IR system is usually evaluated in terms of its *precision* and *recall* [5]. We can prove that the observed recall $\hat{r}$ is equal to the actual recall r and the observed precision $\hat{p}$ is proportional to the actual precision p, consequently their product satisfies $\widehat{pr} \propto pr$. It is noteworthy that the *precision-recall product* closely correlates with the popular F_1 measure [5]: $pr \leq F_1 \leq \sqrt{pr}$. The cost function $\Delta_{pr}(\bar{h}(\bar{\mathbf{x}}), \bar{y}) = 1 - \widehat{pr}$ can be minimized using the following SVM formulation SVM_{pr}^{perf} according to [2]. Moreover, SVM_{pr}^{perf} can be trained by a *sparse-approximation algorithm* in *polynomial* time w.r.t. $|P \cup U| = l + u$ because the loss function Δ_{pr} can be computed from the contingency table [2].

OP 3. SVM_{pr}^{perf}

$$\begin{aligned} \min_{\mathbf{w},\xi\geq 0} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\xi \\ s.t. \quad & \forall \bar{y}' \in \{+1,-1\}^n \setminus \bar{y} : \\ & \frac{1}{2n}\mathbf{w}^T \sum_{i=1}^{n} (y_i - y_i{}')\mathbf{x}_i \geq \frac{1}{2n}\Delta_{pr}(\bar{y}', \bar{y}) - \xi \end{aligned}$$

3. EXPERIMENTS

We conduct experiments on four real-world datasets which are pre-processed and publicly available[1] — **(1) news20**, **(2) siam-competition2007**, **(3) mediamill-exp1** with its top 5 topics and **(4) reuters-21578** with its top 65 topics. Given a retrieval topic, the query P consists of the relevant documents before the split point, while the corpus U consists of all (relevant and irrelevant) documents after the split point. We simply set the SVM parameter $C = 100$ throughout all our experiments. The code for the proposed SVM algorithms is available on the 1st author's homepage. The effectiveness of QBME methods evaluated by Mean Average Precision (MAP) [5] is shown in Table 1. The efficiency of QBME methods evaluated by the average CPU seconds of training on a PC with Pentium 4 (3GHz) processor and 2GB memory is shown in Table 2.

Table 1: The effectiveness of QBME methods.

dataset	(1)	(2)	(3)	(4)
Centroid	0.4011	0.2115	0.4747	0.6833
$\mathrm{SVM}_{1c}^{struct}$	0.3436	0.2239	0.4835	0.6974
Rocchio	0.6867	0.4627	0.5555	0.7138
$\mathrm{SVM}_{ba}^{struct}$	0.8200	0.5635	0.6735	0.7194
SVM_{pr}^{perf}	0.8236	0.5565	0.6727	0.7289

Table 2: The efficiency of QBME methods.

dataset	(1)	(2)	(3)	(4)
$\mathrm{SVM}_{1c}^{struct}$	0.0250	0.0327	0.1560	0.0169
$\mathrm{SVM}_{ba}^{struct}$	0.8905	1.0409	3.6140	1.8042
SVM_{pr}^{perf}	7.1325	43.0641	731.3120	1.9352

As we have anticipated learning from both P and U leads to much higher performances than learning from P only. Using just P, SVM_{1c} works as effectively as the Centroid algorithm, but it provides sparser query vectors which is beneficial to further similarity computation based on *inverted index* [5]. Using both P and U, $\mathrm{SVM}_{ba}^{struct}$ and SVM_{pr}^{perf} work significantly better than the Rocchio algorithm, with $\mathrm{SVM}_{ba}^{struct}$ being orders of magnitude faster than SVM_{pr}^{perf}.

4. REFERENCES

[1] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT)*, pages 92–100, Madison, WI, 1998.

[2] T. Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 377–384, Bonn, Germany, 2005.

[3] T. Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 217–226, Philadelphia, PA, 2006.

[4] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*. Springer, 2006.

[5] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[6] B. Scholkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002.

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
http://www.cs.cmu.edu/~hustlf/r21578_vec_download.html