Supervised Clustering with support Vector machines

Thomas Finley
tomf@cs.cornell.edu
Thorsten Joachims
tj@cs.cornell.edu
Department of Computer Science,
Cornell University,
Ithaca, NY 14853, USA

Introduction

Clustering is generally done using unsupervised methods. For unsupervised model we use only the input feature. We don't provide the output data for training such a model. But this paper uses SVM which is a supervised algorithm. So, the model will get the input and the corresponding output data features as well. For the supervised clustering task we will distribute the whole work under 4 topics namely Supervised clustering motivation, Problem statement difficulties,learning to cluster with SVM^{struct} and application to real problems.

Supervised clustering motivation

Let's take an example of clustering different marbles together. What should be the property on which they should be clustered together, e.g., colour, size, transparency, etc. And to adjust this we have several options like; doing it manually, provide constraints on item pairs and using algorithms to satisfy these constraints (semi-supervised clustering) or provide a series of tuples having item set and cluster on those sets, or learning a hoe to cluster (supervised clustering). Here we look at two tasks. The first is noun-phrase clustering where we have to cluster all the noun phrases which refer to the same entity and the second is to cluster news articles of same kind together. The data for noun-phrase clustering has been taken from noun-cluster conference and the data for news article clustering has been taken from google news and collecting news articles daily.

Problem statement and difficulties

We take a set of n training examples, and we learn how to cluster future sets of items; (x1, y1), . . . , (xn, yn). For simple clustering we find similarity measure between item pairs, and partition items with respect to similarity measure over an objective function. For clustering objective function, we assign items in such a way that it will maximize an objective function. Here the objective function is sum of similarity of pairs in same cluster [1]. For each pair i and j and set, there is a pairwise feature vector Φij. Similarity between i and j is inner product of Φij and a learned vector w; $Sim(xi,xj) =< w,\Phi ij >$. If we consider Naïve training on a set x with partitioning y, we will learn by a simple classifier then. [2]

The problems that we have here are the performance measure e.g., MITRE F-measure for NP conference which is decided on the basis of imbalanced positive/ negative ratio. There is one more problem suppose in the noun phrase problem if there are two names for one person then it should refer to same cluster, is such clustering possible with SVM, or we can say are we able to learn with indirect dependency.

Learning to cluster with SVM^{struct}

First of all, SVM^{struct} is an adaptation of SVM^{light}, for learning complex output functions [4]. Say,$\Psi(x,y)$is the relationship between input x and output y then SVM^{struct} is able to learn function from x to y. $f(x;w) = argmax < w,\Psi(x,y) >$; y$\in Y$. Here, $\Psi(x,y) = (1/5^2) * (\phi 12 + \phi 34 + \phi 35 + \phi 45)$; obtained by Naïve training.$< w,\Psi(x,y) >$ is equivalent to correlation clustering objective. And, $\Delta(x,y)$ is the measure of how unrelated two clustering are. The term delta is also called pairwise loss. MITRE loss is specific to NP conference. Let's formulate a linear constraint for all training examples and for all possible wrong clustering, we will check the value of objective function for correct clustering greater than the value of objective function for every wrong clustering and we also make sure that they differ by the loss between a wrong and a correct clustering. We will allow slack an upper bound on loss.

$< w,\Psi(xi,yi) > \geq < w,\Psi(xi,y) > +\Delta(yi,y) - \varepsilon i.$

Now we will use the algorithm to set the constraints. For this we will iteratively find clustering ŷ associated with the most violated constraints, then we will just ignore this constant and this will become our clustering objective plus the loss. Now we can find ŷ for the argmax with a clustering function variant.

1: Input: (x1, y1), . . . ,(xn, yn), C,∈
2:$Si \leftarrow \phi$ for all i = 1, . . . , n
3: repeat
4: for i = 1, . . . , n do
5: H(y) $\equiv \Delta(yi,y) + wT\Psi(xi,y) - wT\Psi(xi,yi)$
6: compute $\hat{y}$ = $\text{argmax}_{y\in Y}$ H(y)
7: compute εi = max{0, $\text{max}_{y\in Si}$ H(y)}
8: if H($\hat{y}$) >$\varepsilon i + \varepsilon$ then
9: Si ← Si ∪{$\hat{y}$}
10: w ← optimize primal over S = ∪i SI
11: end if
12: end for
13: until no Si has changed during iteration

Application to real problems

For the noun-phrase problem our approach is to use greedy correlation clustering after training all the pairs with SVM^{light}; denoted as PCC (pairwise classifier clustering). We will also use SVM^{struct} with greedy correlation clustering. And for the news article clustering we will first build pair waise (N=2) feature vector which gave us 31 features of the following kind cos sim. of unigrams in title, cos sim. of unigrams in headline, cos sim. of porter stemmed trigrams in quoted article text, etc. [Here CG refers to greedy approximation and CR refers to relaxation approximation.]

$SVM^{cluster}$ vs PCC

	CG	PCC	Default
Test with CG, Δ M	41.3	51.6	51.0
Test with CG, Δ P	2.89	3.15	3.59

table: Result for NP conference.

	CG	CR	PCC	Default
Test with CG, Δ M	2.36	2.43	2.45	9.45
Test with CG, Δ P	2.04	2.08	1.96	9.45

table: Result for News article.

Inclusion of Δ in finding the result: -

We know that H(y) ≡ Δ(yi, y)+ wT Ψ(xi, y) - wT Ψ(xi, yi) now we will compute $\{\hat{y}\}$ = $\text{argmax}_{y\in Y}$ H(y).

Now what if we remove the $\Delta(yi,y)$ here and train the model without it. Applying a similar margin maximizing framework to perform collective classifications. [3]

	With loss	No loss
Np - conference, Δ M	41.3	41.1
Np - conference, Δ P	2.89	2.81
News, train CG, test CG	2.36	2.42
News, train CR, test CR*	2.08	2.12

It is very significant to each other. But can we do better.

Real Relaxation vs Greedy clustering: -

	Train CG	Train CR
Test CG	2.36	2.43
Test CG	2.04	2.08

Conclusion: Neither one is significantly different from each other. So, we have achieved a satisfactory result here.

[1] Blum A. Chawla S. (2002) Bansal, N. Correlation clustering. machine learning. page 89–113, 2002.

[2] Cardie Ng, V. Improving machine learning approaches to coreference resolution. *ACL-02*, page 104–111, 2002.

[3] Chatalbashev V. Koller D. Taskar, B. Learning associative markov networks. *ICML, Banff, Alberta, Canada: ACM Press*, 2004.

[4] Hofmann T. Joachims T. Altun Y Tsochantaridis, I. Support vector machine learning for interdependent and structured output spaces. *ICML*, 2004.