Supervised clustering with Support Vector machine

Vishnu Kumar
B19BB066

Scope

Supervised clustering is a term use for a specific type of clustering tasks done using supervised algorithms. This paper discusses pros and cons of the paper "Supervised clustering with support vector machines." We will discuss the finding of the paper and compare it with the traditional model being used for clustering vs the finding of this paper.

Introduction

For clustering things, traditionally we use unsupervised algorithms. 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 support vector machines which is a supervised algorithm. So, the model will get the input and the output data as well. And we will analyse if the SVM does justice with the classification task or not.

Brief Discussion

This paper deals with two clustering problems. The first one is noun phrase clustering where we have to cluster all the noun phrases which refer to the same entity and the second one 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 by collecting news articles daily.

For each pair i and j and set, there is a pairwise feature vector φ_{ij}. Similarity between i and j is inner product of φ_{ij} and a learned vector w; Sim(xi, xj) = $\langle w, \varphi_{ij} \rangle$. we will proceed further by finding similarity measure between item pairs. And partition items with respect to similarity measure over an objective function. Our objective is to cluster the similar objects together so our objective function is defined as sum of similarity pairs between the clusters.

SVM^{struct} is an adaptation of SVM^{light}, for learning complex output functions [1]. Say, Ψ(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, Ψ(x, y)>; y ∈ Y.

Here, Ψ(x, y) = (1/5^2)*(sum of all the positive examples in same cluster), & <w, Ψ(x, y)> is equivalent to correlation clustering objective. And, Δ(x, y) is the measure of how unrelated two clustering are 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.
$\langle w, \Psi(x_i, y_i) \rangle \geq \langle w, \Psi(x_i, y) \rangle + \Delta(y_i, y) - \xi_i$.

For the noun-phrase problem the 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.

$SVM^{cluster}$ vs PCC

	CG	PCC	Default
Test with CG, ΔM	41.3	51.6	51.0
Test with CG, ΔP	2.89	3.15	3.59

Table: Result for NP conference.

	CG	CR	PCC	Default
Test with CG, ΔM	2.36	2.43	2.45	9.45
Test with CG, ΔP	2.04	2.08	1.96	9.45

Table: Result for News articles.

[Here CG refers to greedy approximation and CR refers to relaxation approximation.]

For NP conference the $SVM^{cluster}$ was significantly better for both MITRE and Pairwise loss training. And for news articles $SVM^{cluster}$ and PCC do not differ significantly.

Now we optimize $SVM^{cluster}$ to an arbitrary clustering loss.

Noun Phrase	Opt. to ΔM	Opt. to ΔP
Test on ΔM	41.3	42.8
Test on ΔP	4.06	2.89

This is quite a big difference between ΔM and ΔP. And it is even worse than the default.

Inclusion of Δ in finding the result: -
We know $H(y) \equiv \Delta(y_i, y) + w^T\Psi(x_i, y) - w^T\Psi(x_i, y_i)$,
compute $\hat{y} = \text{argmax}_{y \in Y} H(y)$.

Now what if we remove the Δ(yi, y) here and train the model without it.

	With loss	No loss
NP-coreference, ΔM	41.3	41.1
NP-coreference, ΔP	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.

Real Relaxation vs Greedy clustering: -

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

Neither one is significantly different from each other [2].

Pros

In regards to this paper we were able to create SVM^{struct} Which is SVM for clustering objects. This paper is itself an addition to the semi-supervised methods and for supervised methods. This method enables us to be able to learn some very complex functions. As in news phrase clustering for clustering the words referring to same entity, we actually had to find the indirect dependency and we also had to use a specific MITRE loss but still we could achieve our target with SVM^{struct}. For supervised clustering we also provided our model with some clustering example which helped it to learn better. So, if our data consists of item pairs even then we will be able to cluster them.

Cons

This method learns a weight w after getting trained on SVM^{light} and that helps us in calculate our similarity measure but some tasks say for the news clustering it has interactable exact argmax. So, with learning the approximate w one may face challenges.

Before putting our algorithm there were some constraints set. There is a pre-defined linear constraint for our model. Having these constraints and then computing optimum value can be problematic as the iterations may stop much before reaching its optimum value if we cross the constraints. {Please refer to the technical updates paper}

For a supervised method we need labelled data i.e., we need item pairs along with some previous clustering so the data collection is more rigorous in this process but still we were able to get better results as compared to default values.

Conclusion

So, after comparing all the results this model seems to be better as we learnt the complex function easily as also produced similar results for the non-complex functions. This also can be used for many more purposes in future.

References

[1] Tsochantaridis, I. H. (2004). Support vector machine learning for interdependent and structured output spaces. ICML.

[2] Ng, V. &. (2002). Improving machine learning approaches to coreference resolution. ACL-02 (pp. 104– 111).