Can we do better than picking a kernel matrix a priori?

Abstract

Our main objective is to learn the Kernel matrix from the Dataset which start with transductive approach and conclude it with inductive approach it is possible but computaional expensive unless we figure out iterative SDP solver.

0.1 Introduction

0.1.1 Motivation for Problem

Kernel Method have changed the era of machine learning due to their capability of working with linear inference models and identifying nonlinear relationships among input data.These method can be applied to both Regresssion and Classification problem.
It was interesting problem for pre machine learning era to obtain method which are good enough to classfy non linear data and have ability to work with heterogenous data.Multi-layer percepton was another method which was in line with Kernel method.

Deep learning era have changed the scientific community significanlty due to their unexplainable performnace.Some people found them as black box which is generating the result based on Universal Approximation Theorem.Deep Learning is still more of Research tool rather than workhorse of market.People still try to first find the solution in Machine learning due to their computational efficiency.They have built on more rigorous mathematics. So people have started looking into it again How can we take advantages of kernel method in this deep Learning era. To understand and make our Neural network robust.

0.2 Literature Review

0.2.1 SVM and Representer theorem

The information specifying the inner products between each pair of points in the embedding space is contained in the so-called kernel matrix, which is symmetric and positive semidefinite (If all points are linearly independent). This matrix essential describes the geometry of embedding space.The imporatance of kernel method is that it extracts all information needed from inner products of training data points. So no need to learn a kernel function over entire sample space which could be infinte in some case.
Representer Theorem: Let Ω : [0,∞)$\rightarrow \mathcal{R}^+$ be a strictly monotonically increasing function. Consider minimization of empirical risk over $\mathcal{H}$.Then any minimizer of the regularized risk

$$C(X_i, y_i, g(X_i)), i = 1, ..., n) + \Omega(||g||^2)$$

admits a representation

$$g(X) = \sum_{i=1}^{n} \alpha_i K(X_i, X)$$

where K is positive semidefinite Kernel and $\mathcal{H}$ is Reproducing Kernel Hilbert Space.

Functions in $\mathcal{H}$ are linear combination of kernels centered at all points of dataset.Though we are searching over this space,the minimizer can always be expressed as a linear combination of kernels centered on data points only.Irrespective of the dimension of $\mathcal{H}$, we can solve the optimization problem by searching for only n real numbers.

0.2.2 Self Learning

If we have huge collection of documents and very few labelled.Naive approach train the model and try it out incrementally by self training. We select documets labelled that are labelled high confidence by the smaller model and use them to buil a training data set for a larger SVM Model. Even we choose high confidence regime still it will make lot of error amd accuracy can drastically decrease if some of data is mislabeled.

0.2.3 Transductive Learning

Transduction:The problem of completing the labeling of partially unlabeled data.The Learning algorithm in this paper need only to learn a set of entries of gram matrix not complete kernel function.One example that the authors discuss in detail is the support vector machine (SVM), where they show this method yields a polynomial time algorithm in the number of test examples, whereas Vapnik's original method for transduction scales exponentially in the number of test examples.

Given: labeleled training data (x_i, y_i) i= 1,..,n and unlabeled test data (x_j^*) j=1,....,m

Estimate : class labels y^* =$(y_1^*, ..., y_m^*)$at these test points

Goal of Learning: minimization of risk on the test set:

$$R(y^*) = \frac{1}{m}\sum_{j=1}^{m}\int_y L(y, y_j^*) dP(y/x_j^*)$$

$$where y^* = (f(x_1^*, \omega), ...f(x_m^*, \omega))$$

Optimization formulation for TSVM(Joachims, 1998) :

The main Goal is separating labeled training data using large margin hyperplane and separating working data set using a large margin hyperplane. Slack variable ζ_i for training ,ζ_j^* for unlabeled test data Minimize

$$R(w, b) = \frac{1}{2}(w \cdot w) + C\sum_{i=1}^{n}\zeta_i + C^*\sum_{j=1}^{m}\zeta_j^*$$

subject to

$$y_i[(w \cdot x_i) + b] \geq 1 - \zeta_i$$

$$y_i^*[(w \cdot x_i) + b] \geq 1 - \zeta_i^*$$

$$\zeta_i, \zeta_j^* \geq 0.i = 1, .., n, j = 1, ..., m$$

$$y_j^* = sign(w \cdot x_j + b), j = 1, ..., m$$

Solution: Support vector S(x) = $(w^* \cdot x) + b$

One additional constraint for appropriate accuracy.

$$\frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{m}\sum_{j=1}^{m}[(w \cdot x_j) + b]$$

Hyperparameter C and C^* control the trade off between explantion and margin size. Soft margin inductive SVM is a special case of soft margin transdcution with zero slacks $\zeta_j^* = 0$ The above problem is not convex due to additional term of loss of unlabeled data.So different Optimization heuristic results in different solutions.Exact solution is possible by Exhaustive search. During implementaion I have used the Label switch heuristic optimization.Algorithm described in next section in next section.

This problem of Non convex optimization is solved out by adding some constraint In Kernel Learning algorithm(Lankreit).

0.2.4 Learning the Kernel matrix

Training set S_{n_tr} = $\{(x_1, y_1), , (x_{n_{tr}}, y_{ntr})\}$ is labeled, and the remainder (i.e., test set) $T_{nt} = \{x_{n_{tr+1}}, , x_{n_{tr+nt}}\}$ are unlabeled, and the aim is to predict the labels for the test samples. In such a setting, the optimizing the kernel is equivalent to choosing a kernel matrix of form

K= $\begin{pmatrix} K_{tr} & K_{tr,t} \\ K_{tr,t} & K_t \end{pmatrix}$

where K_{ij} = $\Phi(x_i)\Phi(x_j)$ By setting up the optimization problem to optimize over the "training-data block" K_{tr}, we want to learn the optimal "mixed-data block" $K_{tr,t}$ and "test-data block" K_t. What we mean to say is that training and test data blocks must somehow be entangled: tuning the entries in K that correspond to training data (i.e., optimizing their embedding) should automatically tune the test-data entries in some way as well. This can be achieved by constraining the search for optimal kernel to only a certain category of kernels to prevent overfitting the training data, and generalize well to unlabeled test data.

General optimization result:

$$\omega_{C,\tau} = \max_{\alpha} 2\alpha^T e - \alpha^T(G(K) + \tau I)\alpha : C \geq \alpha \geq 0, \alpha^T y = 0$$

formulatting above equation as Semi definite optimization problem using constraints

$\min_{K \in \kappa} = \omega_{C,\tau}(K_{tr})$, s.t trace(K) $\geq$ c

$\omega_{C,\tau}$ is convex function:

$$\min_{\mu,t,\lambda,\nu,\delta} t$$

$$subject\ to. \quad trace\left(\sum_{i=1}^{m}\mu_i K_i\right)$$

$$\Sigma_{i=1}^{m}\mu_i K_i \succeq 0$$

$$\begin{pmatrix} G(\Sigma_{i=1}^{m}\mu_i K_{i,tr}) + \tau I_{n_{tr}} & e + \nu - \delta + y \\ (e + \nu - \delta + \lambda y)^T & t - 2C\delta^T e \end{pmatrix} \succeq 0$$

$$\nu \succeq 0 \quad \delta \succeq 0$$

To solve this we need to use packages like sedumi or cvx and use interior point method.These method are polynomial time. Using one more constraint.

$$K = \Sigma_{i=1}^{m}\mu_i K_i, \ \ \mu \geq 0$$

For this restricted linear subspace of positive semidefinite cone P

The author formulated the QCQP problem. This is a major improvement over the worst-case complexity of the previous example that didn't assume positive linear combinations, and over the general SDP form where K $\succeq$ 0. Additionally this constraint results in improved numerical stability -it prevents the algorithm from finding solutions that use large weights $_i$ with opposite sign that cancel.

$$\max_{\alpha,t} 2\alpha^T e - \tau\alpha^T\alpha - ct$$

$$subject\ to \quad t \geq \frac{1}{r_i}\alpha^T G(K_{i,tr})\alpha \ \ i = 1, ..., m$$

,

$$\alpha^T y = 0 \ \ C \geq \alpha \geq 0$$

Similarly,Author formulated QCQP framework for Hard margin ,Soft margin L1 norm and L2 norm,Alignment.Also shown How can we learn the parameter C in L2 norm.

Bound on transduction:

The author estimated the performance of support vector machines for transduction using properties of the class ., (where, for convenience, we suppose that training and test data have the same size $n_t r = n_t = n$) is, with probability 1 (over the random draw of the training set Sn and test set T_n), bounded by

$$\frac{1}{n}\Sigma_{i=1}^{m} \max\{T, 0\} + \frac{1}{\sqrt{n}}\left(4 + \sqrt{2\log\frac{1}{d}} + \sqrt{\frac{C(\kappa)}{n\gamma^2}}\right)$$

where T= $1 - y_i f(x_i)$ i=1 where is the 1-norm soft margin on the data and C(κ) is a certain measure of the complexity of the kernel class K. For instance, for the class K of positive linear combinations defined above, C(κ) $\geq$ mc, where m is the number of kernel matrices in the combination and c is the bound on the trace. So, the proportion of errors on the test data is bounded by the average error on the training set and a complexity term, determined by the richness of the class K and the margin γ . Good generalization can thus be expected if the error on the training set is small, while having a large margin and a class κ that is not too rich.

0.3 Method Description

Preprocessing of data: Datasets Moviereview,Sonar,BreastCancer,HeartDisease.All data is provided in raw format.I removed the missing labeled data and removed the redundancy and unused information while training.

I have started with tools of SVM of python package sklearn on movie review dataset. I have done scratch implementation of SVM for kernel function(Radial Linear and Polynomial) using cvxopt tool.

Training Algorithm for TSVM:

```
Input : training data $(x_i, y_i)$ i= 1,..,n and unlabeled test data $(x_j^*)$j=1,....,m
Parameters : C and $^*$, $no^+$
Output : class labels $y^*$ =$(y_1^*, ...., y_m^*)$at these test points

(w,b,$\zeta$) = QP([Labeled data],C);
Classify the test samples using(w,b). The $no^+$ test data whose (W $\cdot$ x + b )value is highest are assigned positive label and other are assigned to negative label.
$C_-^*$ =MIN
$C_+^*$ =MIN*$\frac{no_+}{k-no_+}$
    while(($C_-^* < C^*$) OR ($C_+^* < C^*$));
    (w,b,$\zeta$,$\zeta^*$) = QP([labeled data],[unlabeled data and their predicted label],C,$C_+^*$,$C_-^*$);
while ($\exists a, b\ such\ that\ (y_a^* \times y_b^* < 0)(\zeta_a^* > 0)\ \&\ \ (\zeta_b^* > 0)$ $(\zeta_a^* + \zeta_b^* > 2))$ (
$y_a^*$=-$y_a^*$ ;
$y_b^*$=-$y_b^*$ ;
(w,b,$\zeta$,$\zeta^*$) = QP([labeled data],[unlabeled data and their predicted label],C,$C_+^*$,$C_-^*$);
)
$C_-^*$ = min($C_-^*$*2,$C^*$);
$C_+^*$ = min($C_+^*$*2,$C^*$);
)
return($y_1^*, ...., y_k^*$);
```

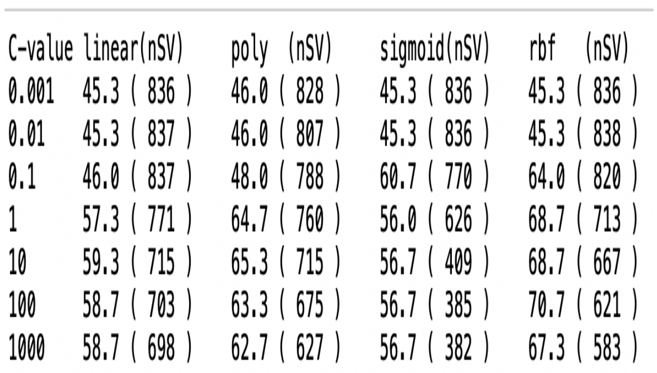
To implement the TSVM and SDP SVM used CVX solver of MATLAB.

So done scratch implementation of SVM and TSVM for kernel funtion(Radial,Sigmoid,Linear and polynomial) using MATLAB optimization toolbox for Quadratic problem.

Algorithm for SDPSVM:

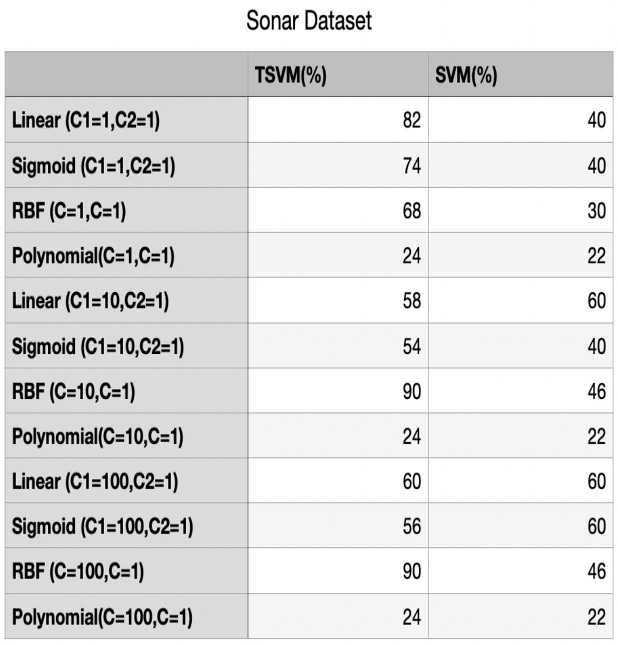
Three kernel matrix of different kernel funtion is used in it(RBF,Sigmoid,Linear).You can Initialize N of kernel matrix by trying different combination of Kernel function and there parameter. So the objective in this case is to learn the kernel matrix in embedding dimension which can give us accurate result.Our search space is constrained to Convex subset of positive definite cone.So we are giving weight to every Kernel matrix. We need to learn the appropriate combination of above kernel matrix. Once we find the weight. The problem become simple.

0.4 Results



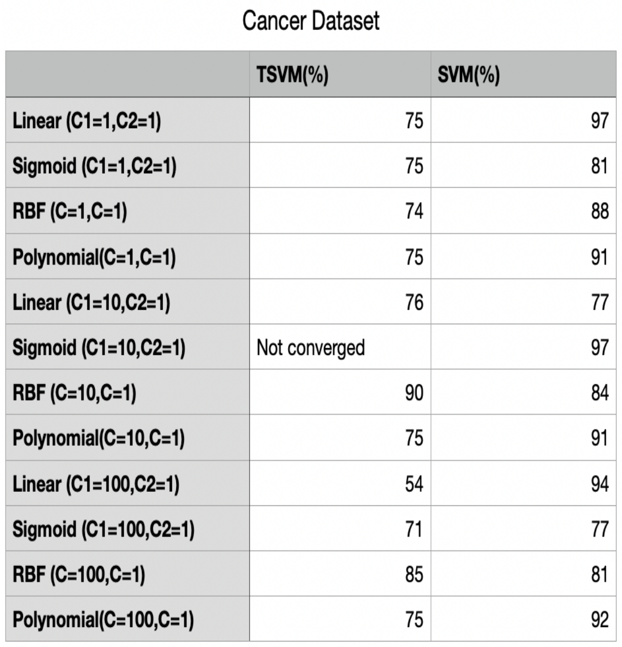
```
C-value linear(nSV)   poly  (nSV)    sigmoid(nSV)   rbf   (nSV)
0.001   45.3 ( 836 )  46.0 ( 828 )   45.3 ( 836 )   45.3 ( 836 )
0.01    45.3 ( 837 )  46.0 ( 807 )   45.3 ( 836 )   45.3 ( 838 )
0.1     46.0 ( 837 )  48.0 ( 788 )   60.7 ( 770 )   64.0 ( 820 )
1       57.3 ( 771 )  64.7 ( 760 )   56.0 ( 626 )   68.7 ( 713 )
10      59.3 ( 715 )  65.3 ( 715 )   56.7 ( 409 )   68.7 ( 667 )
100     58.7 ( 703 )  63.3 ( 675 )   56.7 ( 385 )   70.7 ( 621 )
1000    58.7 ( 698 )  62.7 ( 627 )   56.7 ( 382 )   67.3 ( 583 )
```

As we increase the C value Model will penalize the summation of slack variable over all dataset accuracy is increasing.One problem with this approach is it wont give us qualitative information that are more no of data points are missclassified with small ζ or one outlier (incorrect labeled data) is making the summmation large.It is resolved in ν-SVM



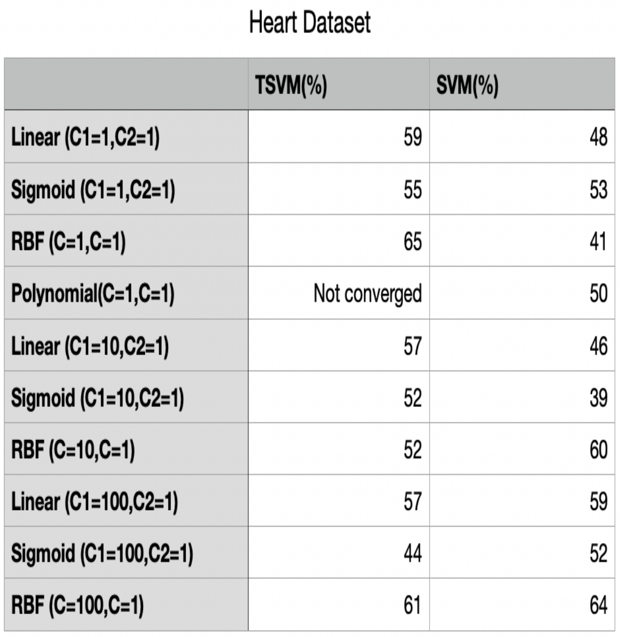
Sonar Dataset

	TSVM(%)	SVM(%)
Linear (C1=1,C2=1)	82	40
Sigmoid (C1=1,C2=1)	74	40
RBF (C=1,C=1)	68	30
Polynomial(C=1,C=1)	24	22
Linear (C1=10,C2=1)	58	60
Sigmoid (C1=10,C2=1)	54	40
RBF (C=10,C=1)	90	46
Polynomial(C=10,C=1)	24	22
Linear (C1=100,C2=1)	60	60
Sigmoid (C1=100,C2=1)	56	60
RBF (C=100,C=1)	90	46
Polynomial(C=100,C=1)	24	22



Cancer Dataset

	TSVM(%)	SVM(%)
Linear (C1=1,C2=1)	75	97
Sigmoid (C1=1,C2=1)	75	81
RBF (C=1,C=1)	74	88
Polynomial(C=1,C=1)	75	91
Linear (C1=10,C2=1)	76	77
Sigmoid (C1=10,C2=1)	Not converged	97
RBF (C=10,C=1)	90	84
Polynomial(C=10,C=1)	75	91
Linear (C1=100,C2=1)	54	94
Sigmoid (C1=100,C2=1)	71	77
RBF (C=100,C=1)	85	81
Polynomial(C=100,C=1)	75	92



Heart Dataset

	TSVM(%)	SVM(%)
Linear (C1=1,C2=1)	59	48
Sigmoid (C1=1,C2=1)	55	53
RBF (C=1,C=1)	65	41
Polynomial(C=1,C=1)	Not converged	50
Linear (C1=10,C2=1)	57	46
Sigmoid (C1=10,C2=1)	52	39
RBF (C=10,C=1)	52	60
Linear (C1=100,C2=1)	57	59
Sigmoid (C1=100,C2=1)	44	52
RBF (C=100,C=1)	61	64

0.4.1 Observation:

1. TSVM provide better results in most of the test cases.

2. As we increase the C1 value in TSVM we penalize theζ (labeled data) it will converge towards SVM.

3. RBF kernel function perform better than other kernel function.

0.5 Conclusion

Started with basic SVM which provide good accuracy, when we have large no of labeled dataset.To get more insight about the classfied data. we should use ν-SVM.
If we have large partially labelled dataset. Transduction is the correct approach because we have the test data in our hand. So we can utilise it to learn the classfier.
I have found two Optimization technique to deal the Non convexity in Transduction problem:
1:Deterministic Annealing
2:Label Switching Heuristic
Another approach is to Learn the Kernel Matrix by contraining it to subset of positive semidefinite cone. Here we need to solve the SDP problem.We can generalize this to online learning case(where data is come one by one,In every iteration Size of kernel matrix increase by one) If we able to solve the SDP problem iteratively than only we can make it computationally efficient.

0.6 Citations and References

- Kernel based support vector machine via semidefinite programming: Application to medical diagnosis
- Learning the Kernel Matrix with Semidefinite Programming
- “Learning by transduction,” in Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence
- Transductive inference for text classification using support vector machines