Master Thesis Second Progress Report

Alhasan Abdellatif

March 2019

1 Brief Overview

1.1 The Idea

1 - Use Cholesky decomposition to decompose RBF sparse kernel K, such that $K = LL^T$, where L is an $m \times m$ lower triangular matrix. We then observe that $\boldsymbol{\phi(x)} \in \mathbb{R}^{\boldsymbol{m}}$.
2- Use L to solve the primal problem using **LibLinear**, which is faster than LibSVM with RBF Kernel, often $O(m \times n)$ vs $O(m^2 \times n)$ or $O(m^3 \times n)$.

1.2 Classifying unseen data

We observe that that a test point $x \in \mathbb{R}^n$, whereas the coefficients obtained $w \in \mathbb{R}^m$. **Hence, we can not pass a test set directly to LibLinear to predict**.
We have to use $x \mapsto sgn(w^T\phi(x) - b)$ but we don't have the explicit expression of $\phi(x)$, **one idea** is to compute the dual variables λ , using (1), and use them to classify new points, in a similar way the dual problem does:

$$(L^T D)\lambda = W \tag{1}$$

$$x \mapsto sgn(\sum_{i=1}^{m} \lambda_i y_i k(x_i, x) - b) \tag{2}$$

where D is a diagonal matrix for labels, y_i is the label of point x_i and $k(x_i, x)$ is the kernel value between point x_i and test point x.

Two observation can be made:

1- The linear system in (1) is triangular and sparse which is fast to solve .

2- The kernel used in (2), $k(x_i, x)$, can be RBF kernel or the sparse kernel, I believe using RBF kernel here is better, since we have finish training using the sparse kernel, and also in terms of speed and not degrading information.

2 Results

Key Results : **Accuracy**, **Training Time** and **Sparsity of both K and L.**

Key parameters : σ**, C, l** and **kernel used in (2)**, either RBF or sparse RBF.

The tests are done on 4 data sets with binary classes , with different σ values and $C = 1$ and kernel used in (2) was RBF. The accuracy and the Sparsity are reported below.

datasets (train/test)	σ	**RBF Kernel Accuracy**	**Sparse Kernel Accuracy**	**Sparsity on K**	**Sparsity on L**
svmguide1(3,089/ 4,000)	15	0.96	0.93	0.88	0.73
	8	0.96	0.95	0.96	0.78
Magic Gamma(11,411/ 7,608)	25	0.84	0.86	0.82	0.57
	8	0.73	0.788	0.99	0.81
Parabola(1000/ 10000)	0.07	0.97	0.97	0.88	0.51
	0.03	0.97	0.97	0.97	0.65
	0.02	0.96	0.96	0.99	0.81
Cancer (525/ 174)	2	0.97	0.99	0.66	0.66
	0.7	0.89	0.95	0.92	0.88

The parameter l in the sparse kernel is set to the minimum odd number satisfying $l \geq \lfloor n/2 \rfloor + 1$.

I implemented the prediction rule in (2) for binary classes only, that why the datsets are binary, I will further develop it to multiclasses using one-vs-all concept.

2.1 Time comparison

Time is not reported yet but the experiments were done using regular matrices without exploiting the sparsity of K and L yet. That's why, the RBF kernel was faster in all cases than the sparse kernel. I have been working on applying best practice and using sparsity to reduce the training time of very large datasets and soon I will report the training time too. The **training time for the sparse kernel** will include the time of the following:

- Computation of the sparse RBF kernel
- Cholesky decomposition
- Solving Optimization problem , LibLinear
- Solving the Linear system (2) to obtain λ.

2.2 Results conclusion

The accuracy of the sparse RBF kernel seems to be matching with that of the RBF kernel and for the appropriate choice of σ, more than 90% of the kernel K is sparse while maintaining the accuracy.

L is less sparser than K, I have applied AMD-similar method from a python package and it results in a more sparser L but I did not check yet how this affects the accuracy.

I tried both kernels RBF and Sparse RBF in (2), and their accuracy sounds reasonable, however RBF kernels is faster in all cases, that is why I chose it .