CS709 Convex Optimization

Project Report

On

Constrained Quadratic Programming

Submitted by:

Suraj Kumar
Avnish Bhagwate

Introduction:

It is a special type of non-linear programming problem in which we solve quadratic objective function under linear constraint. It can be applied to various domains like regression analysis, decision analysis, demand-supply response, portfolio analysis and quadratic approximations.

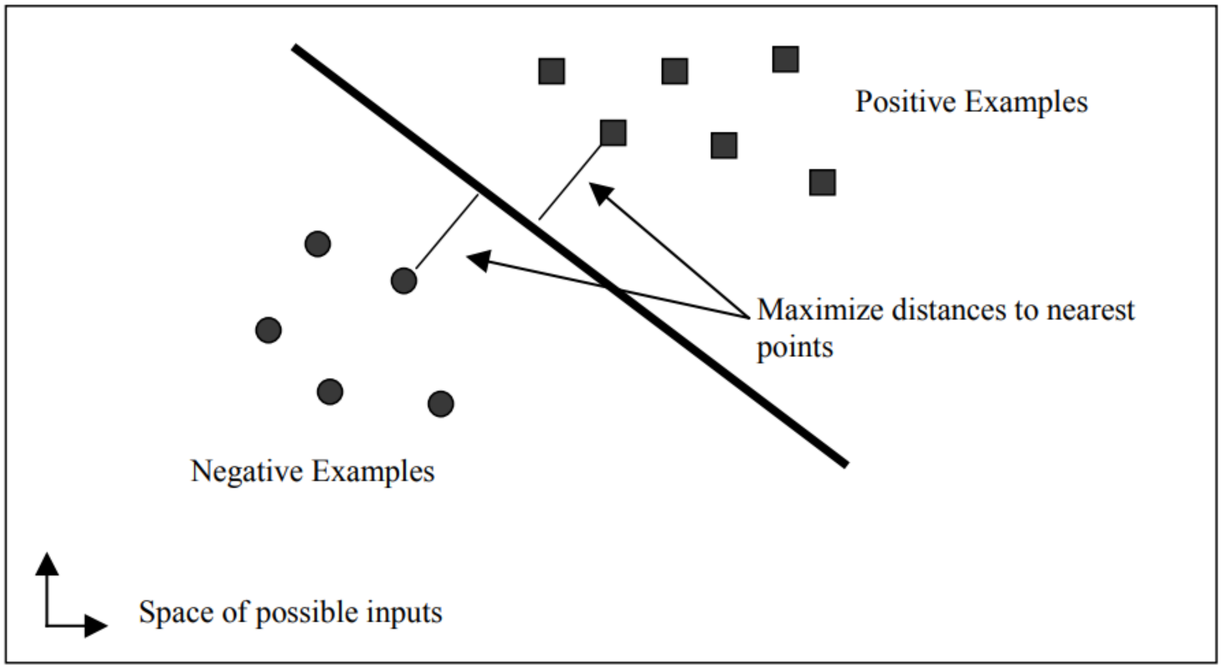
Definition and Formulation:

Quadratic programming deals with the problem of optimizing an objective function which is quadratic and the constraints are linear. The constraints can be equality or inequality.

$$\begin{aligned} &\text{minimize} \quad \tfrac{1}{2}\mathbf{x}^{\mathrm{T}} Q\mathbf{x} + \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ &\text{subject to} \quad A\mathbf{x} \leq \mathbf{b}, \end{aligned}$$

Under the quadratic programming domain, we have chosen to solve SVM classification problem.

Support Vector Machines



Following is the optimization formulation of SVM

$$\min_{\vec{w},b} \tfrac{1}{2}||\vec{w}||^2 \text{ subject to } y_i(\vec{w}\cdot\vec{x}_i - b) \geq 1, \forall i,$$

Using Lagrangian, this optimization problem can be converted into a dual form which is a Quadratic Problem as shown below:

$$\min_{\vec{\alpha}} \Psi(\vec{\alpha}) = \min_{\vec{\alpha}} \tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j (\vec{x}_i \cdot \vec{x}_j)\alpha_i\alpha_j - \sum_{i=1}^{N}\alpha_i,$$

Subject to the following constraints

$$\alpha_i \geq 0, \forall i, \qquad \sum_{i=1}^{N} y_i\alpha_i = 0.$$

Above optimization formulation is valid only for linearly separable dataset. For non-separable case, we have the following set of equations:

Primal

$$\min_{\vec{w},b,\vec{\xi}} \tfrac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{N}\xi_i \text{ subject to } y_i(\vec{w}\cdot\vec{x}_i - b) \geq 1-\xi_i, \forall i,$$

Dual

$$\min_{\vec{\alpha}} \Psi(\vec{\alpha}) = \min_{\vec{\alpha}} \tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j K(\vec{x}_i, \vec{x}_j)\alpha_i\alpha_j - \sum_{i=1}^{N}\alpha_i,$$
$$0 \leq \alpha_i \leq C, \forall i,$$
$$\sum_{i=1}^{N} y_i\alpha_i = 0.$$

Both primal and dual solutions are quadratic optimization problems. To solve these problems we can either use “off-shelve” numerical QP optimization tool or use some efficient algorithms, to be discussed next.

There are various optimization algorithms to solve the SVM classification problem.

1. Sequential Minimal Optimization (SMO)
2. Modified Gradient Projection
3. Decomposition Method
4. Active Set Method
5. Cutting Plane Method

Out of the methods mentioned above, we will be implementing SMO due to the following reasons:

- Generally faster
- Scalable
- Simple
- Easy to implement
- No storage for Kernel matrix

Sequential Minimal Optimization

Sequential Minimal Optimization (SMO) is a simple algorithm that can quickly solve the SVM QP problem without any extra matrix storage and without using numerical QP optimization steps at all. SMO decomposes the overall QP problem into QP sub-problems and chooses to solve these problems at every step. For the standard SVM QP problem, the smallest possible optimization problem involves two Lagrange multipliers, because the Lagrange multipliers must obey a linear equality constraint. At every step, SMO chooses two Lagrange multipliers to jointly optimize, finds the optimal values for these multipliers, and updates the SVM to reflect the new optimal values.

There are two components to SMO:

- An analytic method for solving for two Lagrange multipliers.
- A heuristic for choosing which multipliers to optimize.

We were selecting Lagrangian Parameters based on the following heuristic:

- Iterate over entire training set.
- If training example does not fulfill KKT conditions within tolerance, we take its Lagrangian as one of the Lagrangian parameters.
- Second Lagrangian parameter is chosen from following in order:
    - Repeated pass over all non-bound examples
    - If we can't any alpha_j from above then we iterate over all training set.

Solution using CVXOPT

Since we have lot of off-shelve Quadratic Optimization solvers available and the dual SVM is a quadratic problem, it becomes imperative to take one of these for performance comparison. We are using CVXOPT module of Python. The following table is under 100% accuracy.

No. of samples	Time units (CVXOPT)	Time units (SMO)
1000	3.5486	9.1620
2000	17.1108	15.8780
3000	43.4945	40.6005

New implementation based on the feedback

Based on the feedback to improve the algorithm using the fact to choose the alphas based on some criteria rather that sequentially choosing alphas, we implemented the algorithm whilst choosing the alphas in a way that difference of errors between the corresponding alphas is the maximum.

For comparison with an already implemented algorithm, we chose the RasseLegin implementation.

For a dummy dataset of 3000 points and a 100% accuracy:

The new implementation took 0.13043199999999988 unit time

RasseLegin implementation took 0.3835519999999999 unit time

Which shows the gain in time of the new implementation over the already implemented algorithm.

References

http://luthuli.cs.uiuc.edu/~daf/courses/optimization/Papers/smoTR.pdf

https://www.csie.ntu.edu.tw/~cjlin/papers/bottou_lin.pdf

http://web.cs.iastate.edu/~honavar/smo-svm.pdf

http://cvxopt.org/userguide/coneprog.html#quadratic-programming

https://github.com/LasseRegin/SVM-w-SMO/blob/master/SVM.py