CSCI 5525: Machine Learning

Homework 2

Jingxiang Li

October 22, 2015

Mercer's Condition:

A symmetric function $K(x, y)$ can be expressed as an inner product

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

for some ϕ if and only if $K(x, y)$ is positive semidefinite, i.e.

$$\int \int K(x, y)g(x)g(y)dxdy \geq 0, \quad \forall g(x) \text{ s.t. } \int g(x)^2 < \infty$$

or, equivalently:
Kernel matrix K, where $K_{ij} = K(x_i, x_j)$, is positive semidefinite for any collection $\{x_1, \ldots, x_n\}$

Problem 1.a

proof

Here we will use the Mercer's Condition to determine if $K = \sum_{j=1}^{m} w_j K_j$ is a valid kernel.

Note that K_j is a valid kernel, which suggests that

$$\int \int K_j(x, y)g(x)g(y)dxdy \geq 0, \quad \forall g(x) \text{ s.t. } \int g(x)^2 < \infty$$

Then we have

$$\sum_{j=1}^{m} w_j \int \int K_j(x, y)g(x)g(y)dxdy \geq 0$$

i.e.

$$\sum_{j=1}^{m} \int \int w_j K_j(x, y)g(x)g(y)dxdy \geq 0$$

where $w_j \geq 0, \forall j$

By Fubini's theorem, we can swap the integral sign and sum sign by assuming that function $K_j(x, y)g(x)g(y)$ is integrable, which does hold in general case.

Then we have

$$\int \int \sum_{j=1}^{m} w_j K_j(x, y)g(x)g(y)dxdy \geq 0$$

i.e.

$$\int \int K(x, y)g(x)g(y)dxdy \geq 0$$

Note that function $g(x)$ is arbitrary, by Mercer's Condition, K must be a valid kernel.

Q.E.D.

Problem 1.b

proof

Here we use the second argument of the Mercer's Condition to determine if $K = K_1 \odot K_2$ is a valid kernel.

Since K_1 and K_2 are valid kernels, for any collection $\{x_1, \ldots, x_n\}$, for any non-zero vector v, we have $v^T K_1 v \geq 0$ and $vK_2 v \geq 0$, where $K_{1,ij} = K_1(x_i, x_j)$, $K_{2,ij} = K_2(x_i, x_j)$. Which means that Kernel Matrices K_1 and K_2 are positive semidefinite.

Let $K_1 = \sum_i \lambda_i v_i v_i^T$, $K_2 = \sum_i \gamma_i w_i w_i^T$, where $\lambda_i \geq 0$ and $\gamma_i \geq 0$

Then

$$K = K_1 \odot K_2 = \sum_{i,j} \lambda_i \gamma_j (v_i v_i^T) \odot (w_i w_i^T) = \sum_{i,j} \lambda_i \gamma_j (v_i \odot w_i)(v_i \odot w_i)^T$$

Note that $(v_i \odot w_i)(v_i \odot w_i)^T$ is always positive semidefinite, since for any nonzero vector c, let $\alpha = c^T(v_i \odot w_i)$ which is a scaler, then $c^T(v_i \odot w_i)(v_i \odot w_i)^T c = \alpha^2 >= 0$.

Since $\lambda_i \geq 0$ and $\gamma_i \geq 0$, $K = K_1 \odot K_2$ must be a positive semidefinite matrix. Note that this argument holds for any collection $\{x_1, \ldots, x_n\}$, by Mercer's Condition, $K = K_1 \odot K_2$ is a valid kernel.

Q.E.D.

Problem 1.c

proof

$\forall g(x)$ s.t. $\int \int g(x)^2 dx \geq 0$,

$$\int \int K(x,y) g(x) g(y) dx dy = \int \int (xy+1)^{2015} g(x) g(y) dx dy$$

$$= \int \int (xy+1)^{2015} g(x) g(y) dx dy = \int \int (C_{2015}(xy)^{2015} + C_{2014}(xy)^{2014} + \cdots + C_0) g(x) g(y) dx dy$$

where $C_i > 0$ is some positive constant

Note that $\forall n \in \{0, 1, 2, \ldots\}$

$$\int \int (xy)^n g(x) g(y) dx dy = (\int x^n g(x) dx)^2 \geq 0$$

Hence

$$\int \int K(x,y) g(x) g(y) dx dy = \int \int (C_{2015}(xy)^{2015} + C_{2014}(xy)^{2014} + \cdots + C_0) g(x) g(y) dx dy \geq 0$$

by Mercer's Condition, K is a valid kernel function.

Q.E.D.

Problem 1.d

proof

Note that

$$K(x,y) = \exp(-\frac{1}{2}(x-y)^2) = \exp(xy)\exp(-\frac{1}{2}x^2)\exp(-\frac{1}{2}y^2)$$

Let $K_1(x,y) = \exp(xy)$, $K_2(x,y) = \exp(-\frac{1}{2}x^2)\exp(-\frac{1}{2}y^2)$, Then $K = K_1 \odot K_2$, which suggests that it's sufficient to prove K_1 and K_2 are valid kernels.

For K_2, the proof is obvious since K_2 is separable. $\forall g(x)$ s.t. $\int\int g(x)^2 dx \geq 0$,

$$\int\int K_2(x,y)g(x)g(y)dxdy = \int\int \exp(-\frac{1}{2}x^2)\exp(-\frac{1}{2}y^2)g(x)g(y)dxdy$$

$$= (\int \exp(-\frac{1}{2}x^2)g(x)dx)^2 \geq 0$$

By Mercer's Condition, K_2 is a valid kernel function.

For K_1, notice that by Taylor's expansion

$$K_1(x,y) = \exp(xy) = \sum_{j=0}^{\infty} \frac{(xy)^j}{j!}$$

By using the same arguments in problem 1.c, since

$$\int\int (xy)^j g(x)g(y)dxdy = (\int x^j g(x)dx)^2 \geq 0$$

it's easy to show that

$$\int\int K_1(x,y)g(x)g(y)dxdy \geq 0 = \int\int \sum_{j=0}^{\infty} \frac{(xy)^j}{j!} g(x)g(y)dxdy \geq 0$$

Again by Mercer's Condition, K_1 is a valid kernel function.

Using the result derived from problem 1.b, since $K = K_1 \odot K_2$ and both K_1 and K_2 are valid kernel functions, K is a valid kernel function.

Q.E.D.

Problem 2.a

Sequential Minimal Optimization (SMO) Algorithm

The objective function to be optimized is the dual form of Support Vector Machine (SVM):

$$\begin{aligned}
&\max_{\alpha \in \mathbb{R}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j K_{ij} \alpha_i \alpha_j \\
&0 \leq \alpha_i \leq C, \forall i \\
&\sum_{i=1}^{n} y_i \alpha_i = 0
\end{aligned}$$

The idea of SMO is to update Lagrange multipliers by iteratively updating two entries (α_1, α_2) in each iteration while keeping the remaining $(n-2)$ components fixed. To illustrate how SMO algorithm works, we need to answer the following two questions:

1. Subproblem Optimization: given (i, j), how to optimize (α_i, α_j)
2. Subproblem Selection: how to select (i, j) in each iteration

Subproblem Optimization

In this section we will discuss the subproblem optimization problem, which is given (i, j), how to optimize (α_i, α_j). Without loss of generality, let the two multipliers be α_1 and α_2. The objective function for the two selected multipliers can be simplified as

$$W(\alpha_1, \alpha_2) = \alpha_1 + \alpha_2 - \frac{1}{2} K_{11} \alpha_1^2 - \frac{1}{2} K_{22} \alpha_2^2 - s K_{12} \alpha_1 \alpha_2 - y_1 \alpha_1 v_1 - y_2 \alpha_2 v_2 + W_{\text{constant}}$$

Where

$$\begin{aligned}
K_{ij} &= k(x_i, x_j) \\
s &= y_1 y_2 \\
v_i &= \sum_{j=3}^{n} y_j \alpha_j K_{ij} = f(x_i) + b - y_1 \alpha_1 K_{1i} - y_2 \alpha_2 K_{2i} \\
f(x^*) &= \sum_{i=1}^{n} \alpha_i y_i K(x_i, x^*) - b
\end{aligned}$$

Because of the linear constrained on α, we must have $\alpha_1 + s\alpha_2 = \gamma$, which is some constant. Then the above objective function can be expressed in terms of α_2 alone:

$$\begin{aligned}
W(\alpha_2) = (\gamma - s\alpha_2) + \alpha_2 - \frac{1}{2} K_{11} (\gamma - s\alpha_2)^2 - \frac{1}{2} K_{22} \alpha_2^2 - s K_{12} (\gamma - s\alpha_2) \alpha_2 \\
- y_1 (\gamma - s\alpha_2) v_1 - y_2 \alpha_2 v_2 + W_{\text{constant}}
\end{aligned}$$

The stationary point of the objective function is at

$$\begin{aligned}
0 &= \frac{dW}{d\alpha_2} \\
&= s K_{11} (\gamma - s\alpha_2) - K_{22} \alpha_2 + K_{12} \alpha_2 - s K_{12} (\gamma - s\alpha_2) \\
&+ y_2 v_1 - s - y_2 v_2 + 1
\end{aligned}$$

We also need the second derivative of the objective function

$$\frac{d^2W}{d\alpha_2^2} = 2K_{12} - K_{11} - K_{22}$$

If the second derivative is negative, then the maximum of the objective function can be expressed as

$$\alpha_2^{new}(K_{11} + K_{22} - 2K_{12}) = s(K_{11} - K_{12})\gamma + y_2(v_1 - v_2) + 1 - s$$

i.e.

$$\alpha_2^{new}(K_{11} + K_{22} - 2K_{12}) = \alpha_2^{old}(K_{11} + K_{22} - 2K_{12}) + y_2(f(x_1) - y_1 - (f(x_2) - y_2))$$

Let $\eta = K_{11} + K_{22} - 2K_{12}$, $E_1 = f(x_1) - y_1$, $E_2 = f(x_2) - y_2$, we have

$$\alpha_2^{new} = \alpha_2^{old} - \frac{y_2(E_1 - E_2)}{\eta}$$

Because of the linear constrained on α, we can derive an upper bound and lower bound on α_2.

Note that $\alpha_1 + s\alpha_2 = \alpha_1^{old} + s\alpha_2^{old}$ and $0 < \alpha_1 < C$, we have

$$L = \max(0, \alpha_2^{old} - \alpha_1^{old}) < \alpha_2 < \min(C, C + \alpha_2^{old} - \alpha_1^{old}) = H, \text{ if } s = 1$$

$$L = \max(0, \alpha_1^{old} + \alpha_2^{old} - C) < \alpha_2 < \min(C, \alpha_1^{old} + \alpha_2^{old}) = H, \text{ if } s = -1$$

By this way, we can truncate α_2 by the following way:

$$\alpha_2^{new,clipped} = \begin{cases} H, & \text{if } \alpha_2^{new} \geq H \\ \alpha_2^{new}, & \text{if } L < \alpha_2^{new} < H \\ L, & \text{if } \alpha_2^{new} \leq L \end{cases}$$

Then $\alpha_1^{new} = \alpha_1^{old} + s(\alpha_2^{old} - \alpha_2^{new,clipped})$

This is the update rule when $\eta < 0$

If $\eta >= 0$, we simply check the objective function at $\alpha_2 = L$ and $\alpha_2 = H$ respectively, and choose the one that gives the larger objective value.

Subproblem Selection

To select the two multipliers in each iteration, SMO uses a two-step hierarchical selection strategy, i.e. first select α_2 and then select the corresponding α_1 in a hierarchical way.

Selecting α_2 is trivial, SMO first iterates over all non-bounded multipliers, i.e. those $0 < \alpha_i < C$, and then iterates over all multipliers. Note that the order of iteration should be randomized.

After selecting α_2, we select the corresponding α_1 using the following rule:

1. Find $0 < \alpha_i < C$ that achieves maximum $|E_i - E_2|$
2. Iterate over all $0 < \alpha_i < C$
3. Iterate over all α_i

The selection process for α_1 will continue until α_2 is successfully optimized.

The reason SMO checks α_i that maximize $|E_i - E_2|$ first is that this quantity determines the step size for each updating. Remember that the update rule for α_2 is

$$\alpha_2^{new} = \alpha_2^{old} - \frac{y_2(E_1 - E_2)}{\eta}$$

Simulation Study

Here we evaluate the optimization performance of SMO algorithm by applying it to train SVM model for dataset MNIST-13. The average training time for the model is **8.8 seconds**, and the standard deviation for the running time is **1.8 seconds**. To show the optimization performance, we record the dual objective function for each iteration till convergence and make plots for objective function values over number of iterations in figure 1.

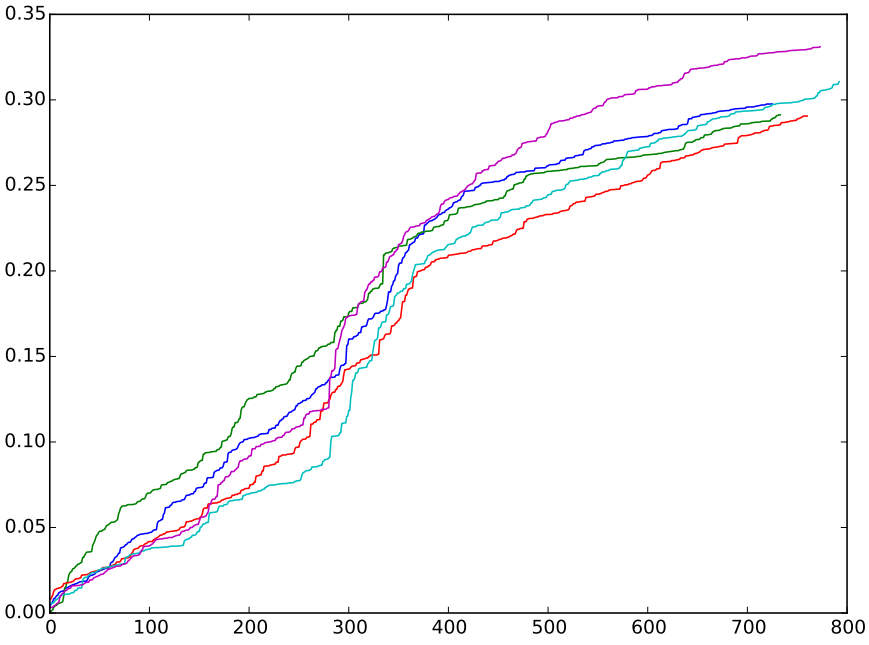

Figure 1: SMO: Dual Objective Value v.s. Number of Iterations

Problem 3.a

Primal Estimated sub-GrAdient SOlver for SVM (Pegasos)

The objective function to be optimized is the primal form of Support Vector Machine (SVM):

$$\min_w \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{(x,y)\in S} l(w;(x,y))$$

where

$$l(w;(x,y)) = max\{0, 1 - y\langle w, x\rangle\}$$

The Pegasos algorithm has two parts:

1. a stochastic gradient descent algorithm
2. a projection over weight vector to the optimal set $B = \left\{w : ||w|| \leq 1/\sqrt{\lambda}\right\}$

We will first show why B is the optimal set for w and then give the stochastic gradient descent algorithm based on it.

Optimal Set for w

In this section we will prove that the optimal w^* must be inside the set $B = \left\{w : ||w|| \leq 1/\sqrt{\lambda}\right\}$

Proof. To do so, we examine the dual form of the SVM and use the strong duality theorem. Setting $C = 1/\lambda m$, the primal objective function can be expressed as the following constrained optimization problem

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i \text{ s.t. } \forall i \in \{1,2,...m\} : \xi_i \geq 0,\ \xi_i \geq 1 - y_i\langle w, x_i\rangle$$

Then the Dual or this problem has the form:

$$\sum_{i=1}^{m}\alpha_i - \frac{1}{2}\left\|\sum_{i=1}^{m}\alpha_i y_i x_i\right\|^2 \text{ s.t. } \forall i \in \{1,2,...m\} : 0 \leq \alpha_i \leq C$$

Denote the optimal primal and dual solutions by (w^*, ξ^*) and α^*, respectively. Then the dual objective can be written as

$$||\alpha^*||_1 - \frac{1}{2}||w^*||^2$$

By strong duality, the primal objective value is equal to the dual objective value at the optimum, thus

$$\frac{1}{2}||w^*||^2 + C||\xi^*||_1 = ||\alpha^*||_1 - \frac{1}{2}||w^*||^2$$

Note that $||\alpha^*|| \leq C = \frac{1}{\lambda m}$. Therefore, $||\alpha^*||_1 \leq 1/\lambda$, hence

$$\frac{1}{2}||w^*||^2 \leq \frac{1}{2}||w^*||^2 + C||\xi^*||_1 = ||\alpha^*||_1 - \frac{1}{2}||w^*||^2 \leq \frac{1}{\lambda} - \frac{1}{2}||w^*||^2$$

which suggests that

$$||w^*|| \leq \frac{1}{\sqrt{\lambda}}$$

Q.E.D.

Knowing that the optimal w^* must be inside the set $B = \left\{w : ||w|| \leq 1/\sqrt{\lambda}\right\}$, in the stochastic gradient descent we will initialize weight vector w^0 inside the set B, and for each iteration we will project the updated w^t to the set B. In this way we can accelerate the convergence.

Stochastic Gradient Descent Algorithm

The stochastic gradient descent algorithm is straight forward. For each iteration we randomly pick k observations from the training set as the training subset A_t, and update weight vector w based on A_t. Note that when computing the gradient, we only use a subset of A_t, $A_t^+ = \{i \in A_t : \ y_i \langle w^t, x_i \rangle < 1\}$, since only observations inside A_t^+ contributes gradient to the hinge loss.

The Pegasos Algorithm

Input: Training set S, regularization parameter λ, number of iterations T, size of training subset k

Initialize: Choose w^0 s.t. $||w^0|| \leq 1/\sqrt{\lambda}$

For $t = 0, 1, \ldots, T-1$

Choose $A_t \subset S$, where $|A_t| = k$

Set $A_t^+ = \{i \in A_t : \ y_i \langle w^t, x_i \rangle < 1\}$

Set step size $\eta_t = \frac{1}{\lambda t}$

Set $w^{t+\frac{1}{2}} = (1 - \eta_t \lambda) w^t + \frac{\eta_t}{k} \sum_{(x,y) \in A_t^+} yx$ (pure gradient descent)

Set $w^{t+1} = \min\left\{1, \frac{1/\sqrt{\lambda}}{||w^{t+\frac{1}{2}}||}\right\} w^{t+\frac{1}{2}}$ (project $w^{t+\frac{1}{2}}$ into optimal set B)

Output: w^T

Simulation Study

Here we evaluate the optimization performance of Pegasos algorithm by applying it to train SVM model for dataset MNIST-13. We choose $k = 1, 20, 100, 200, 2000$, respectively. For each setting, the model is trained 5 times and the mean and the standard deviation of the training time will be recorded. The result is summarized in table 1.

Table 1: Training time for Pegasos algorithm

	$k = 1$	$k = 20$	$k = 100$	$k = 200$	$k = 2000$
Avg Time (sec)	0.228	0.059	0.053	0.056	0.106
Std Time (sec)	0.021	0.004	0.007	0.004	0.007

$***$ Note that the training time for $k = 1$ is unexpected long. This is because when k is too small, it takes time for the algorithm to find an nonempty set A_t^+.

To show the optimization performance, we record the primal objective function for each iteration till convergence and make plots for objective function values over number of iterations in figure 2.

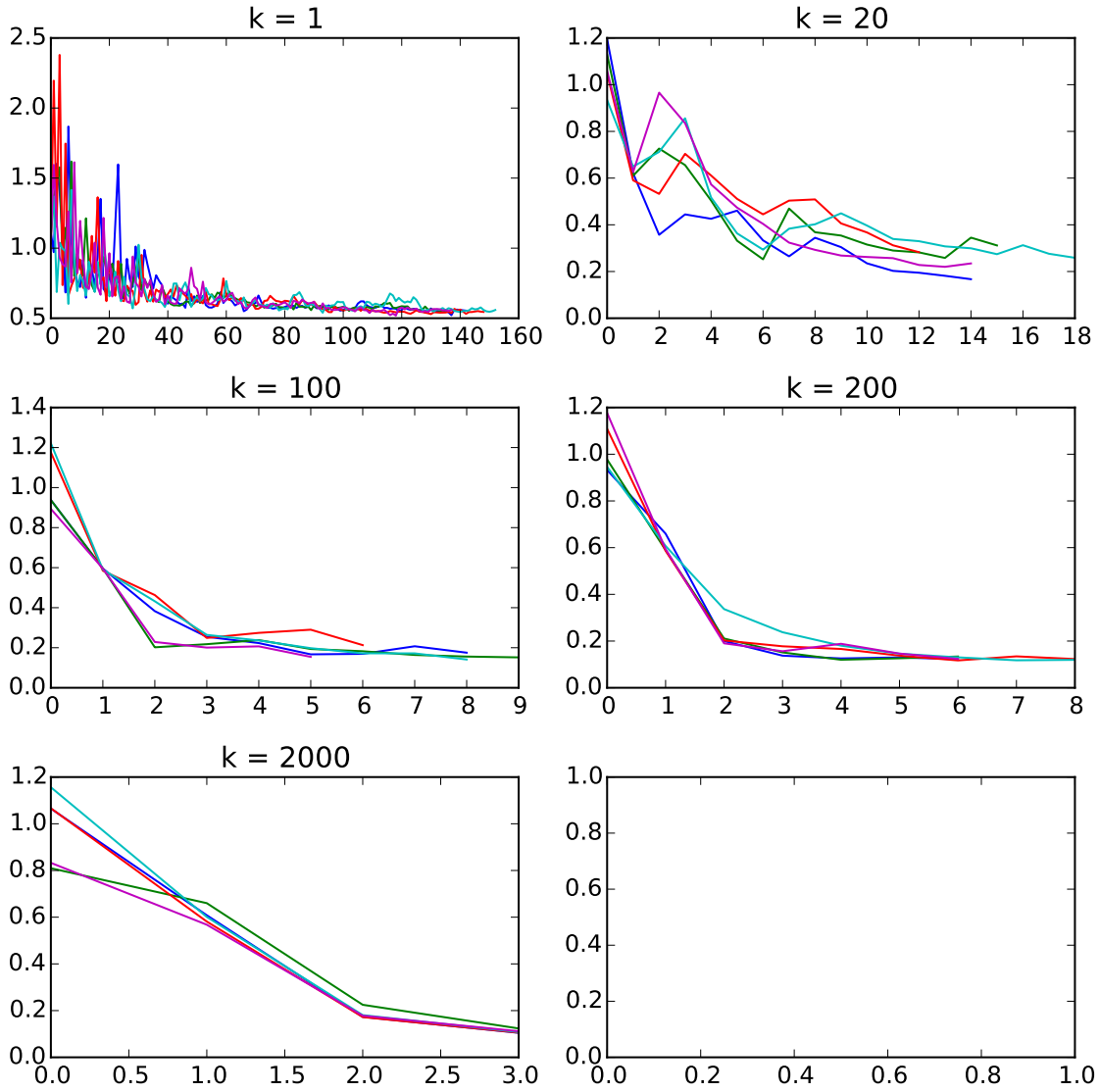

Figure 2: Pegasos: Primal Objective Value v.s. Number of Iterations