ECSE-4962
Matthew Caulfield
**Monday November 11, 2020**

# Homework 3

1. Recall that a function $K : X \times X \to \mathbb{R}$ is a valid kernel function if it is symmetric and positive semi-definite function. For the current problem, we assume that the domain $X = \mathbb{R}$.

   (a) Let $K_1, ..., K_m$ be valid kernel functions. Let $w_j > 0$ for $j = 1, ..., n$ be the corresponding weight for each kernel function. Because each kernel $K_j$ is valid then the following holds.

   $$K_j = K_j^T, \qquad j = 1, .., m$$
   $$w_j K_j = w_j K_j^T$$
   $$\sum_{j=1}^{M} w_j K_j = \sum_{j=1}^{M} w_j K_j^T$$

   Thus $K = \sum_{j=1}^{M} w_j K_j$ is symmetric because symmetry is maintained when multiplying by a scalar and adding symmetric matrices. Because each $K_j$ is valid and therefor semipositive definite then

   $$\forall \mathbf{x}, \qquad \mathbf{x^T K_j x} \geq \mathbf{0}, \qquad \mathbf{j = 1, ..., m}$$
   $$\mathbf{x^T w_j K_j x} \geq \mathbf{0}$$
   $$\mathbf{x^T} \sum_{\mathbf{j=1}}^{\mathbf{M}} \mathbf{w_j K_j x} \geq \mathbf{0}$$

   So $K = \sum_{j=1}^{M} w_j K_j$ is semi-positive definite. Because $K = \sum_{j=1}^{M} w_j K_j$ is both symmetric and semipositive definite it is a valid kernel function.

   (b) Let $K_1, K_2$ be valid kernel functions and K be defined as the Hadamard product $K = K_1 \odot K_2$ such that $K(x_i, x_j) = K_1(x_i, x_j) K_2(x_i, x_j)$. Then because $K_1, K_2$ are symmetric,

   $$K_1(x_i, x_j) = K_1(x_j, x_i), \qquad K_2(x_i, x_j) = K_2(x_j, x_i) \qquad \forall x_i, x_j \in K_1, K_2$$
   $$K_1(x_i, x_j) K_2(x_i, x_j) = K_1(x_j, x_i) K_2(x_j, x_i)$$
   $$K(x_i, x_j) = K(x_j, x_i).$$

   Therefore the kernel K is symmetric. Because $K_1, K_2$ are semipositive definite then by definition,

   $$\sum_{i,j} c_i K_1(x_i, x_j) c_j \geq 0, \qquad \sum_{i,j} d_i K_2(x_i, x_j) d_j \geq 0$$
   $$\sum_{i,j} c_i K_1(x_i, x_j) c_j \sum_{i,j} d_i K_2(x_i, x_j) d_j \geq 0$$
   $$\sum_{i,j} d_i c_i K_1(x_i, x_j) K_2(x_i, x_j) c_j d_j \geq 0$$
   $$\sum_{i,j} b_i K(x_i, x_j) b_j \geq 0$$

   So K is semipositive definite. Because K is both semipositive definite and symmetric it is a valid kernel function.

(c) Let $K(x, x') = (xx' + 1)^{2015}$ where $x, x' \in \mathbb{R}$.

$$K(x, x') = (xx' + 1)^{2015}$$
$$(xx' + 1)^{2015} = (x'x + 1)^{2015}$$
$$(x'x + 1)^{2015} = K(x', x)$$
$$K(x, x') = K(x', x)$$

So, K is symmetric. Assume that $K_1$ is a valid kernel. Then $K_1$ is semipositive definite. $K_1 + 1$ is also a valid kernel and therefore semipositive definite. $(K_1 + 1)^{2015}$ is also semipositive definite so K is semipositive definite. Because K is both symmetric and semipositive definite K is a valid kernel.

(d) Let $K(x, x') = \exp(-(x - x')^2/2)$.

$$(x - x')^2 = (x' - x)^2$$
$$\frac{-(x - x')^2}{2} = \frac{= (x' - x)^2}{2}$$
$$\exp(\frac{-(x - x')^2}{2}) = \exp(\frac{-(x' - x)^2}{2})$$
$$K(x, x') = K(x', x)$$

So K is symmetric. We can write the kernel $K(x, x')$ as a function $f(t) = exp(-t^2/2) = E[e^{it\mathcal{N}(0,1)}], t = x - x'$, where $\mathcal{N}(0, 1)$ is the Gaussian distribution with parameters 0 and 1. Then for $x_1, ...x_n \in \mathbb{R}, a_1, ..a_n \in \mathbb{R}$,

$$\sum_{i,j} a_i h(x_i, x_k) a_j = \sum_{i,j} a_i E[e^{i(x_i - x_j)\mathcal{N}(0,1)}] a_j =$$

$$\sum_{i,j} a_i E[e^{i(x_i - x_j)\mathcal{N}(0,1)} e^{i(-x_j)\mathcal{N}(0,1)}] a_j = E[\left| \sum_{i,j} a_i e^{i(x_i - x_j)\mathcal{N}(0,1)} \right|^2] \geq 0$$

So K is positive semidefinite. Because K is both symmetric and positive semidefinite it is a valid kernal function.

2. The Pegasos algorithm from paper Pegasos: Primal Estimated sub-GrAdient SOlver for SVM [1] was implemented in this homework assignment to evaluate its performance on the MNIST-13 dataset to classify handwritten 1s and 3s.

**Algorithm 1:** Pegasos Algorithm

input: S, $\lambda$, T, k initialize $W_1$ ;
**for** *t=1,2,...,T* **do**
  Choose $A_t \subseteq$ S, $|A_t| = k$;
  $A_t^+ = \{(x, y) \in A_t : y < \mathbf{w_t}, \mathbf{x} < \mathbf{1}\}$;
  $\eta_t = \frac{1}{\lambda t}$;
  $\mathbf{w_{t+\frac{1}{2}}} = (\mathbf{1} - \eta_t \lambda)\mathbf{w_t} + \frac{\eta_t}{\mathbf{k}} \sum_{(\mathbf{x},\mathbf{y}) \in \mathbf{A_t^+}} \mathbf{yx}$;
  $\mathbf{w_{t+1}} = \min\{\mathbf{1}, \frac{\frac{\mathbf{1}}{\sqrt{\lambda}}}{\|\mathbf{w_{t+\frac{1}{2}}}\|}\}\mathbf{w_{t+\frac{1}{2}}}$;
**end**
output: $\mathbf{W_{T+1}}$;

The goal of this homework was to optimize the Pegasos algorithm on the dataset. The primal objective function in equation 1 was used to determine the optimization of the the algorithm.

$$f(\mathbf{w}; \mathbf{A_t}) = \frac{\lambda}{\mathbf{2}}\|\mathbf{w}\|^{\mathbf{2}} = \frac{\mathbf{1}}{\mathbf{k}} \sum_{(\mathbf{x},\mathbf{y})\in\mathbf{A_t}} \max\{\mathbf{0}, \mathbf{1} - \mathbf{y} < \mathbf{w}, \mathbf{x} >\} \tag{1}$$

The parameters that were tuned to optimize this algorithm where the batch size, the regularization parameter, $\lambda$,and the number of iterations. To optimize this algorithm it was run over batch sizes of 1, 20, 100, 200, and 200 and number of iterations were set to 200. The regularization parameter was then found so that the primal objective function value would converge to 0 before 200 iterations were occurred. Experimentally, regularization parameter was chosen to be $\lambda = 30$. The average run time and standard deviation of the run time over 5 runs are shown in the table below for each batch size.

| Batch Size | Average run time | std of run time |
|---|---|---|
| 1 | 0.0034 sec | 0.0016733 sec |
| 20 | 0.0016733 sec | 0.0011402 sec |
| 100 | 0.2158 sec | 0.011987 sec |
| 200 | 0.6088 sec | 0.0094181 sec |
| 2000 | 5.784 sec | 0.039894 sec |

As the batch size increases so does the run time. The following are plots of the primal objective function value compared to the number of iterations for each run of each batch.

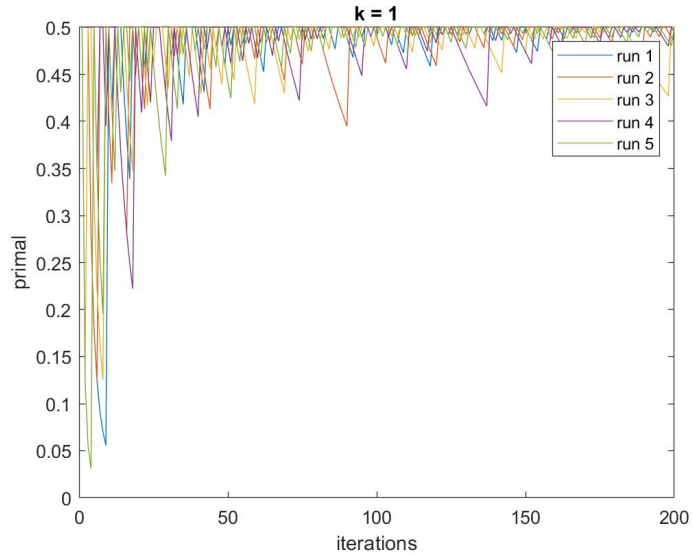


Figure 1: Plot of the Primal Objective Function Values for mini-batch size 1

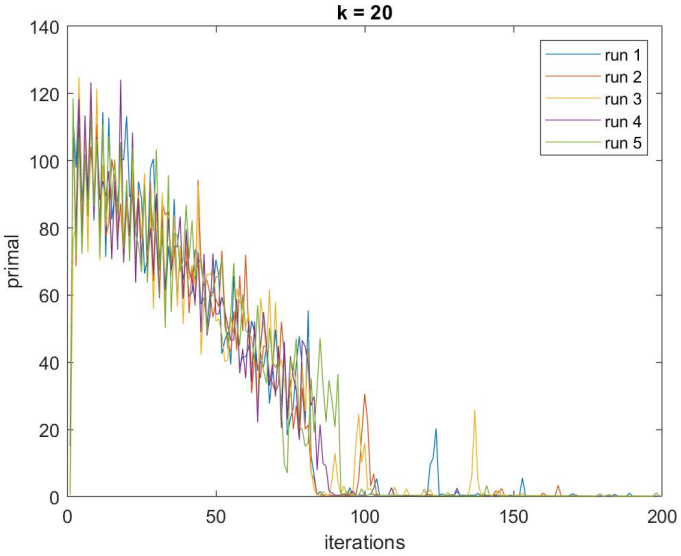


Figure 2: Plot of the Primal Objective Function Values for mini-batch size 20

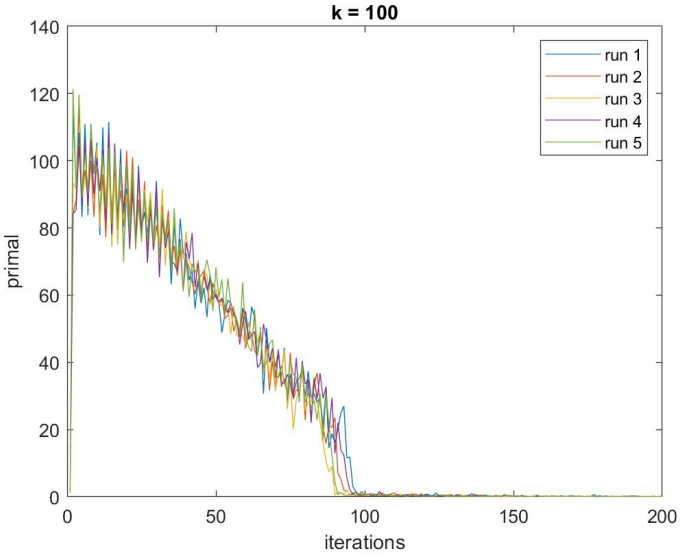


Figure 3: Plot of the Primal Objective Function Values for mini-batch size 100

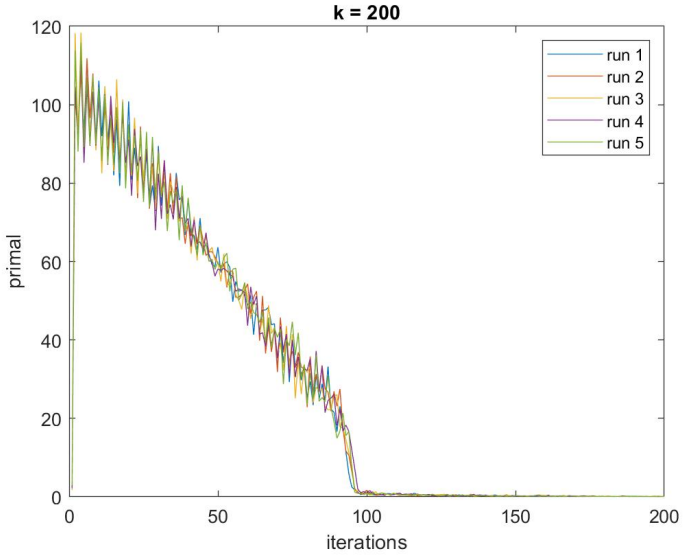


Figure 4: Plot of the Primal Objective Function Values for mini-batch size 200

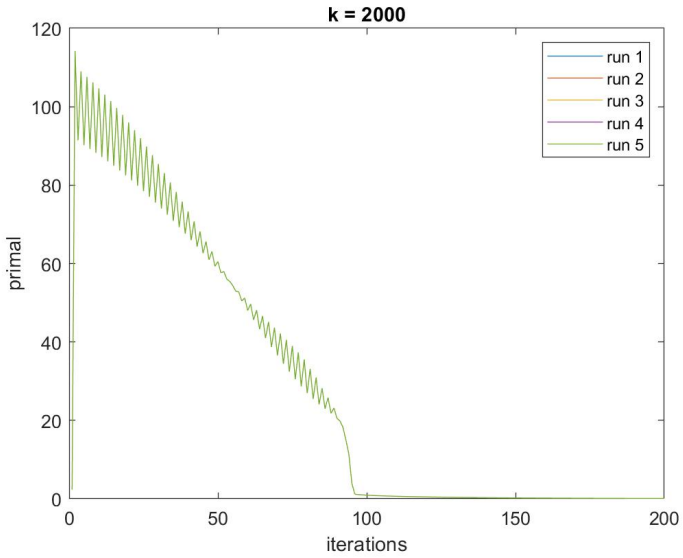


Figure 5: Plot of the Primal Objective Function Values for mini-batch size 2000

For a batch size of $k = 1$ the primal objective function value converges to 0.5 after around 50 iterations. For the other batch sizes the primal objective function values converge to 0 at around 100 iterations for $\lambda = 30$. As the batch size increases the runs overlap more. This makes sense because the mini-batch for each run will overlap more as the mini-batch size increases.

## References

[1] Shalev-Shwartz, S., Singer, Y.; Srebro, N. (2007). *Pegasos: Primal Estimated sub-GrAdient SOlver for SVM.* Proceedings of the 24th International Conference on Machine Learning - ICML ′07. doi:10.1145/1273496.1273598