COMP 5212 Machine Learning (2024 Spring)
Homework 1 (Solution):
Hand out: Feb 23, 2024
Due: March 8, 2024, 11:59 PM
Total Points: 86

Your solution should contain below information at the top of its first page.

1. Your name
2. Your student id number

Some Notes:

- Homeworks will not be easy, please start early.
- For this homework, please submit a single PDF file to Canvas. Make sure to prepare your solution to each problem on a separate page.
- The total points of this homework is 86, and a score of 86 already gives you full grades on this homework. There are possibly bonus questions, you can optionally work on them if you are interested and have time.
- You can choose either using LaTeX by inserting your solutions to the problem pdf, or manually write your solutions on clean white papers and scan it as a pdf file – in the case of handwriting, please write clearly and briefly, we are not responsible to extract information from unclear handwriting. **We highly recommend you use LaTeX for the sake of any misunderstandings about the handwriting.** If your submission is a scan of a handwritten solution, make sure that it is of high enough resolution to be easily read. At least 300dpi and possibly denser.
- We encourage students to work in groups for homeworks, but the students need to write down the homework solutions or the code independently. In case that you work with others on the homework, please write down the names of people with whom you've discussed the homework. You are not allowed to copy, refer to, or look at the exact solutions from previous years, online, or other resources.
- **Late Policy:** 3 free late days in total across the semester, for additional late days, 20% penalization applied for each day late. **No assignment will be accepted more than 3 days late**.
- Please refer to the Course Logistics page for more details on the honor code and logisitcs. We have zero tolerance — in the case of honor code violation for a single time, you will fail this course directly.

1 Linear Regression (30 pts)

Let $D = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)\}$ where $\mathbf{x_i} \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ be the training data that you are given. As you have to predict a continuous variable, one of the simplest possible models is linear regression, i.e. to predict y as $\mathbf{w^T x}$ for some parameter vector $\mathbf{w} \in \mathbb{R}^d$.[1] We thus suggest minimizing the following loss

$$\underset{\mathbf{w}}{\operatorname{argmin}} \hat{R}(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \mathbf{w^T x_i})^2. \tag{1}$$

Let us introduce the $n \times d$ matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ with the $\mathbf{x_i}$ as rows, and the vector $\mathbf{y} \in \mathbb{R}^n$ consisting of the scalars y_i. Then, (1) can be equivalently re-written as

$$\underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{Xw} - \mathbf{y}\|^2. \tag{2}$$

We refer to any $\mathbf{w}^*$ that attains the above minimum as a solution to the problem.

1. [**4 points**] Show that if $\mathbf{X^T X}$ is invertible, then there is a unique $\mathbf{w}^*$ that can be computed as $\mathbf{w}^* = (\mathbf{X^T X})^{-1} \mathbf{X^T y}$.

Note that

$$\hat{R}(w) = \|Xw - y\|^2 = (Xw - y)^T (Xw - y) = w^T X^T X w - 2 w^T X^T y + y^T y.$$

The gradient of this function is equal to

$$\nabla \hat{R}(w) = 2X^T X w - 2X^T y.$$

Because $\hat{R}(w)$ is convex (formally proven in (4)), its optima are exactly those points that have a zero gradient, i.e. those w^* that satisfy $X^T X w^* = X^T y$. Under the given assumption, the unique minimizer is indeed equal to $w^* = (X^T X)^{-1} X^T y$.

2. [**4 points**] Please give at least two cases under which $\mathbf{X^T X}$ is not invertible, explain your answers. How many solutions does Eq. 2 have in these cases?

1. Case 1: n < d, $\mathbf{X}$ has at most rank n as it is a $n \times d$ matrix and hence at most n of its singular values are nonzero. This means that there is at least one index j such that $\sigma_j = 0$ and hence any $z_j \in \mathbb{R}$ is a solution to the optimization problem. As a result the set of optimal solutions for z is a linear subspace of at least one dimension. By rotating this subspace using V, i.e. $\mathbf{w} = V\mathbf{z}$, it is evident that the optimal solution to the optimization problem in terms of $\mathbf{w}$ is also a linear subspace of at least one dimension and that thus no unique solution exists and result in infinite solutions. Furthermore, since $\mathbf{X}$ has at most rank n, $\mathbf{X}^T\mathbf{X}$ is not of full rank. As a result $(\mathbf{X}^T\mathbf{X})^{-1}$ does not exist, so $(\mathbf{X}^T\mathbf{X})$ is not invertible.

2. Case 2: n ≥ d and $\mathbf{X}$ does not have full column rank. In this case, rank of $(\mathbf{X}^T\mathbf{X})$ < d and thus it is not invertible. Then it will have infinitely many solutions.

3. [**4 points**] Consider the case $n \geq d$. Under what assumptions on $\mathbf{X}$ does Eq. 2 admit a unique solution $\mathbf{w}^*$? Give an example with $n = 3$ and $d = 2$ where these

[1] Without loss of generality, we assume that both $\mathbf{x_i}$ and y_i are centered, i.e. they have zero empirical mean. Hence we can neglect the otherwise necessary bias term b.

assumptions do not hold.
The optimization problem admits a unique solution only if all the singular values of $\mathbf{X}$ are nonzero. For $n \geq d$, this is the case if and only if $\mathbf{X}$ is of full rank, i.e., all the columns of $\mathbf{X}$ are linearly independent. As an example for a matrix not satisfying these assumptions, any matrix with linearly dependent dependent suffices, e.g.

$$\mathbf{X} = \begin{pmatrix} 1 & -2 \\ 0 & 0 \\ -2 & 4 \end{pmatrix}.$$

The ridge regression optimization problem with parameter $\lambda > 0$ is given by

$$\operatorname*{argmin}_{\mathbf{w}} \hat{R}_{\text{ridge}}(\mathbf{w}) = \operatorname*{argmin}_{\mathbf{w}} \left[\sum_{i=1}^{n} (y_i - w^T \mathbf{x_i})^2 + \lambda \mathbf{w}^T \mathbf{w} \right]. \tag{3}$$

4. [**4 points**] Show that $\hat{R}_{\text{ridge}}(w)$ is convex with regards to $\mathbf{w}$. You can use the fact that a twice differentiable function is convex if and only if its Hessian $\mathbf{H} \in \mathbb{R}^{d \times d}$ satisfies $\mathbf{w}^T \mathbf{H} \mathbf{w} \geq 0$ for all $\mathbf{w} \in \mathbb{R}^d$ (is positive semi-definite).

Because convex functions are closed under addition, we will show that each term in the objective is convex, from which the claim will follow. Each data term $(y_j - \mathbf{w}^T \mathbf{x}_j)^2$ has a Hessian $\mathbf{x}_j \mathbf{x}_j^T$, which is positive semi-definite because for any $\mathbf{w} \in \mathbb{R}^d$ we have $\mathbf{w}^T \mathbf{x}_j \mathbf{x}_j^T \mathbf{w} = (\mathbf{x}_j^T \mathbf{w})^2 \geq 0$ (note that $\mathbf{x}_j^T \mathbf{w} = \mathbf{w}^T \mathbf{x}_j$ are scalars). The regularizer $\lambda \mathbf{w}^T \mathbf{w}$ has the identity matrix λI_d as a Hessian, which is also positive semi-definite because for any $\mathbf{w} \in \mathbb{R}^d$ we have $\mathbf{w}^T \lambda I_d \mathbf{w} = \lambda \|\mathbf{w}\|^2 \geq 0$, and this completes the proof.
5. [**4 points**] Derive the closed form solution to Eq. 3.

The gradient of $\hat{R}_{\text{Ridge}}(\mathbf{w})$ with respect to $\mathbf{w}$ is given by

$$\nabla \hat{R}_{\text{Ridge}}(\mathbf{w}) = 2\mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{y}) + 2\lambda \mathbf{w}.$$

Similar to (a), because $\hat{R}_{\text{Ridge}}(\mathbf{w})$ is convex, we only have to find a point $\mathbf{w}^*_{\text{Ridge}}$ such that

$$\nabla \hat{R}_{\text{Ridge}}(\mathbf{w}^*_{\text{Ridge}}) = 2\mathbf{X}^T(\mathbf{X}\mathbf{w}^*_{\text{Ridge}} - \mathbf{y}) + 2\lambda \mathbf{w}^*_{\text{Ridge}} = 0.$$

This is equivalent to

$$(\mathbf{X}^T\mathbf{X} + \lambda I_d)\mathbf{w}^*_{\text{Ridge}} = \mathbf{X}^T\mathbf{y},$$

which implies the required result

$$\mathbf{w}^*_{\text{Ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda I_d)^{-1}\mathbf{X}^T\mathbf{y}.$$

6. [**5 points**] Show that Eq. 3 admits the unique solution $\mathbf{w}^*_{\text{ridge}}$ for any matrix $\mathbf{X}$. Show that this even holds for the cases in (b) and (c) where Eq. 2 does not admit a unique solution w^*.
Note that $\mathbf{X}^T\mathbf{X}$ is a positive semi-definite matrix since $\forall \mathbf{w} \in \mathbb{R}^d : \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} = \sum_{i=1}^{n} [(\mathbf{X}\mathbf{w})_i]^2 \geq 0$, which implies that it has non-negative eigenvalues. But then, $\mathbf{X}^T\mathbf{X} + \lambda I_d$ has eigenvalues bounded from below by $\lambda > 0$, which means that it is invertible and thus the optimum is uniquely defined.

7. [**5 points**] What is the role of the term $\lambda \mathbf{w}^T \mathbf{w}$ in $\hat{R}_{\text{ridge}}(w)$? What happens to w^*_{ridge} as $\lambda \to 0$ and $\lambda \to \infty$?

The term $\lambda \mathbf{w}^T \mathbf{w}$ "biases" the solution towards the origin, i.e. there is a quadratic penalty for solutions $\mathbf{w}$ that are far from the origin. The parameter λ determines the extend of this effect: As $\lambda \to 0$, $\hat{R}_{\text{Ridge}}(\mathbf{w})$ converges to $\hat{R}(\mathbf{w})$. As a result the optimal solution $\mathbf{w}^*_{\text{Ridge}}$ approaches the solution of (1). As $\lambda \to \infty$, only the quadratic penalty $\mathbf{w}^T \mathbf{w}$ is relevant and $\mathbf{w}^*_{\text{Ridge}}$ hence approaches the null vector $(0, 0, \ldots, 0)$.

2 Generalized Linear Models (25 pts)

In class, we have discussed generalized linear models (GLMs). GLMs are models for data where the response variable y follows a distribution from the exponential family, and the mean of the response variable is related to the predictors via a link function. Consider the following form of GLM:

$$p(y|\mathbf{x}, \eta) = b(y) \exp\left(y\eta - a(\eta)\right), \tag{4}$$

where p is the probability mass function or probability density function of y, $\eta := \mathbf{w}^T\mathbf{x}$ is the natural parameter, $a(\eta)$ is the log normalizer.

1. [**7 points**] Show that linear regression and logistic regression are special cases of the GLM in 4. Identify $b(y)$ and $a(\eta)$.

linear regression:

$$\begin{aligned} P(y|\mu) &= \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}(y-\mu)^2) \\ &= \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}y^2) \exp(\mu y - \frac{1}{2}\mu^2) \end{aligned} \tag{5}$$

Thus:

$$\begin{aligned} \eta &= \mu \\ a(\eta) &= \frac{1}{2}\eta^2 \\ b(y) &= \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}y^2) \end{aligned}$$

logistic regression:

$$\begin{aligned} P(y|\mu) &= (1-\mu)^{1-y}\mu^y \\ &= \exp\left[(1-y)\log(1-\mu) + y\log(\mu)\right] \\ &= \exp\left[y\log\frac{\mu}{1-\mu} + \log(1-\mu)\right] \end{aligned} \tag{6}$$

Thus:

$$\begin{aligned} \eta &= \log\frac{\mu}{1-\mu} \\ \log(1-\mu) &= -\log(1+e^\eta) \\ a(\eta) &= \log(1+e^\eta) \\ b(y) &= 1 \end{aligned}$$

2. [**6 points**] Consider the Poisson regression model, which is defined by the following probability mass function:

$$p(y|\mathbf{x}, \mathbf{w}) = \text{Pois}(y|\exp(\mathbf{w}^T\mathbf{x})).$$

Here, $\text{Pois}(y|\mu) = \frac{\mu^y e^{-\mu}}{y!}$ is the Poisson distribution parameterized by μ, with support $y \in \mathbb{N}$. Show that Poisson regression belongs to the family of GLMs defined in Equation 4,

and identify $a(\eta)$ for this model.

$$Pois(y|\mu) = \frac{1}{y!}\exp(y\log(\mu)-\mu)$$

Thus:

$$\begin{aligned}\eta &= \log\mu\\ a(\eta) &= e^{\eta}\\ b(y) &= \frac{1}{y!}\end{aligned}$$

3. [**6 points**] Consider a dataset $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$, where $\mathbf{x}_n \in \mathbb{R}^d$ and y_n is the response variable. Derive the negative log-likelihood (NLL) loss function for the Poisson regression model. Simplify the expression as much as possible
By definition. the NLL for (1) should be

$$-\sum_{n=1}^{N} p\left(y_n|\mathbf{x}_n,\eta\right) = \sum_{n=1}^{N}\left(a(\mathbf{w}^\top\mathbf{x}_n) - y_n\mathbf{w}^\top\mathbf{x}_n\right) - Nb(y)$$

Thus, the answer is:

$$\sum_{n=1}^{N}(\exp(\mathbf{w}^T x) - y_n(\mathbf{w}^T x))$$

4. [**6 points**] Prove that in Equation 4, $\mathbb{E}[y|\mathbf{x}] = a'(\eta)$ and $\mathrm{Var}[y|\mathbf{x}] = a''(\eta)$.

$$1 = \int_{y\in\mathbb{R}} p(y|x,\eta) = \int_{y\in\mathbb{R}} b(y)\exp(y\eta - a(\eta))dy$$

take both sides derivates:

$$0 = \int_{y\in\mathbb{R}} b(y)(\eta - a'(\eta))\exp(y\eta - a(\eta))dy$$

therefore:

$$a'(\eta)\int_{y\in\mathbb{R}} b(y)\exp(y\eta - a(\eta))dy = a'(\eta)dy = \int_{y\in\mathbb{R}} b(y)\eta\exp(y\eta - a(\eta))dy = \mathbb{E}[y|x]dy$$

Thus:

$$a'(\eta) = \mathbb{E}(y|x)$$

On the other hand:

$$a''(\eta) = \int_{y\in\mathbb{R}} b(y)y(y\eta - a'(\eta))\exp(y\eta - a(\eta))dy$$

Thus:

$$a''(\eta) = \mathbb{E}[y^2|x] - a'(\eta)\mathbb{E}[y|x] = \mathrm{Var}[y|x]$$

3 Support Vector Machines (31 pts)

In binary classification, we are interested in finding a hyperplane that separates two clouds of points living in, say, $\mathbb{R}^p$. The support vector machine (SVM), which we talked about in class, is a pretty popular method for doing binary classification; to this day, it's (still) used in a number of fields outside of just machine learning and statistics.

One issue arises with the standard SVM, though, when the data points are not linearly separable in $\mathbb{R}^p$, i.e., we cannot find a hyperplane which separates the two classes of points. In such cases, it is often useful to map the data points to a different space (potentially of higher dimension than $\mathbb{R}^p$) where the points become separable. Such maps are called nonlinear feature maps.

In this problem, you will develop a SVM with the RBF kernel to address the nonlinearly separable problem of the standard SVM. You will implement your own RBF-SVM in part (b) of this question, but as a starting point, we will first investigate the SVM dual problem in part (a) of this question.

Throughout, we assume that we are given n data samples, each one taking the form (x_i, y_i), where $x_i \in \mathbb{R}^p$ is a feature vector and $y_i \in \{-1, +1\}$ is a class. In order to make our notation more concise, we can transpose and stack the x_i vertically, collecting these feature vectors into the matrix $X \in \mathbb{R}^{n\times p}$; doing the same thing with the y_i lets us write $y \in \{-1, +1\}^n$.

The primal problem of SVM with slack variables is

$$\begin{array}{ll} \underset{\beta\in\mathbb{R}^p,\ \beta_0\in\mathbb{R},\ \xi\in\mathbb{R}^n}{\text{minimize}} & \frac{1}{2}\|\beta\|_2^2 + C\sum_{i=1}^n \xi_i \\ \text{subject to} & \xi_i \geq 0, \quad i = 1, \ldots, n, \\ & y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i, \quad i = 1, \ldots, n, \end{array} \tag{7}$$

where $\beta \in \mathbb{R}^p$, $\beta_0 \in \mathbb{R}$, $\xi = (\xi_1, \ldots, \xi_n) \in \mathbb{R}^n$ are our variables, and C is a positive margin coefficient chosen by the implementer.

(i, 2pts) Does strong duality (i.e. zero duality gap) hold for problem (7)? Why or why not? (Your answer to the latter question should be short.)

Yes, the strong duality holds, because it meets the Slater's condition: the primal is a convex problem, and all the constraints are affine.

(ii, 8pts) Derive the Karush-Kuhn-Tucker (KKT) conditions for problem (7). Please use $\alpha \in \mathbb{R}^n$ for the dual variables (i.e., Lagrange multipliers) associated with the constraints "$y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i,\ i = 1, \ldots, n$", and $\mu \in \mathbb{R}^n$ for the dual variables associated with the constraints "$\xi_i \geq 0,\ i = 1, \ldots, n$".

$$L(\beta, \beta_0, \xi, \alpha, \mu) = \frac{1}{2}\|\beta\|_2^2 + C\sum_{i=1}^n \xi_i - \mu^T\xi - \sum_{i=1}^n \alpha_i\big(y_i(x_i^T\beta + \beta_0) + \xi_i - 1\big) \quad \alpha \geq 0, \mu \geq 0$$

stationarity :

$$\nabla_\beta L = 0, \nabla_{\beta_0} L = 0, \nabla_\xi L = 0$$
$$\Rightarrow \beta = \sum_{i=1}^n \alpha_i y_i x_i, \quad \sum_{i=1}^n \alpha_i y_i = 0, \quad \alpha = C\mathbf{1} - \mu$$

complementary slackness :

$$\mu_i \xi_i = 0,\ \alpha_i(1 - \xi_i - y_i(x_i^T\beta + \beta_0)) = 0, \quad \text{for all } i$$

primal feasibility :

$$\xi \geq 0,\ y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i, \quad \text{for all } i$$

dual feasibility :

$$\mu \geq 0, \alpha \geq 0$$

(iii, 8pts) Show that the SVM dual problem can be written as

$$\begin{array}{ll} \underset{\alpha \in \mathbb{R}^n}{\text{maximize}} & -(1/2)\alpha^T \tilde{X}\tilde{X}^T\alpha + 1^T\alpha \\ \text{subject to} & \alpha^T y = 0, \\ & 0 \leq \alpha \leq C1, \end{array} \tag{8}$$

where $\tilde{X} \in \mathbb{R}^{n \times p} = \text{diag}(y)X$, α is the dual variable, and the 1's here are vectors (of the appropriate and possibly different sizes) containing only ones.

The Lagrangian dual function $g(\mu, \alpha)$ is

$$g(\mu, \alpha) = \min_{\beta, \beta_0, \xi} L,$$

to minimize L, we set $\nabla_{\beta,\beta_0,\xi} L = 0$ to solve, (this is the same to stationarity condition). $\nabla_{\beta,\beta_0,\xi} L = 0$ requires $\beta = \sum_{i=1}^n \alpha_i y_i x_i, \sum_i \alpha_i y_i = 0, \alpha = C\mathbf{1} - \mu$, where

$$\begin{aligned} g(\mu, \alpha) &= \frac{1}{2}(\sum_i \alpha_i y_i x_i)^T(\sum_i \alpha_i y_i x_i) - \sum_i \alpha_i y_i x_i^T \beta + \mathbf{1}^T\alpha \\ &= -\frac{1}{2}(\sum_i \alpha_i y_i x_i)^T(\sum_i \alpha_i y_i x_i) + \mathbf{1}^T\alpha \\ &= -(1/2)\alpha^T\tilde{X}\tilde{X}^T\alpha + \mathbf{1}^T\alpha \end{aligned}$$

when $\sum_i \alpha_i y_i = 0, \alpha = C\mathbf{1} - \mu$ cannot be met, $g(\mu, \alpha) = -\infty$, thus

$$g(\mu, \alpha) = \begin{cases} -(1/2)\alpha^T\tilde{X}\tilde{X}^T\alpha + \mathbf{1}^T\alpha & \text{if } \sum_i \alpha_i y_i = 0, \alpha = C\mathbf{1} - \mu \\ \infty \quad \text{otherwise} \end{cases}$$

remove the slack varible μ we obtain the dual problem:

$$\begin{array}{ll} \underset{\alpha \in \mathbb{R}^n}{\text{maximize}} & -(1/2)\alpha^T \tilde{X}\tilde{X}^T\alpha + 1^T\alpha \\ \text{subject to} & \alpha^T y = 0, \\ & 0 \leq \alpha \leq C1, \end{array}$$

(iv, 5pts) Give an expression for the optimal β in terms of the optimal α variables and explain how.

Since strong duality holds for the primal problem, thus the optimal solutions for primal and dual problem β and α must satisfy KKT conditions. By the stationarity condtion we have $\beta = \sum_{i=1}^{n} \alpha_i y_i x_i$.

Now we are going to take a glimpse of the "magic" of kernels. Let's first recall what is a kernel we learned in the lectures: Given a feature map $\phi : \mathbb{R}^d \to \mathcal{K}$, where $\mathcal{K}$ is a Hilbert space (i.e., a vector space with inner product), the kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the corresponding inner product function

$$K(x_i, x_j) \coloneqq \langle \phi(x_i), \phi(x_j) \rangle. \tag{9}$$

Here the feature map, as we mentioned earlier, is used to "embed" the original data into a higher dimensional space such that they become separable. Recall the objective of the dual SVM, and it can be rewritten as

$$\begin{aligned} & -\tfrac{1}{2}\alpha^T \tilde{X}\tilde{X}^T \alpha + 1^T \alpha && (10)\\ \Leftrightarrow & -\tfrac{1}{2}\alpha^T Y X X^T Y \alpha + 1^T \alpha && (11)\\ \Leftrightarrow & -\tfrac{1}{2}\alpha^T Y G Y \alpha + 1^T \alpha, && (12)\\ & && (13) \end{aligned}$$

where $Y = \mathrm{diag}(y)$, and $G = XX^T$ is the so called Gram matrix (this is also called the Kernel Matrix in our lectures), $G_{ij} = \langle x_i, x_j \rangle$. One nice property of the Gram matrix of a kernel K is that

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = G_{ij}. \tag{14}$$

Hence, the kernel builds a bridge between the feature maps and the original dual problem.

(vi, 8pts) Show that the Gram matrix of a kernel K is positive semidefinite. Let the dimension of the feature space after the feature map be p'. If $p << p'$, which one is more efficient to solve, the primal or the dual? Why?

For any vector z,

$$\begin{aligned} z^T G z &= \sum_{i,j} z_i z_j \langle \phi(x_i), \phi(x_j) \rangle \\ &= \langle \sum_i z_i \phi(x_i), \sum_j z_j \phi(x_j) \rangle \\ &= \langle \sum_i z_i \phi(x_i), \sum_i z_i \phi(x_i) \rangle \\ &\geq 0, \end{aligned}$$

thus the Gram matrix of a kernel K is positive semidefinite. If $p << p'$, solving the dual problem would be more efficient since from the problem definition the dual problem is only related to Gram matrix G, which is a $n \times n$ matrix that does not grow with p', and the parameters to be optimized $\alpha \in \mathbb{R}^n$, whose dimension is a constant. In contrast, the primal problem directly optimizes parameter β that have p' dimension.

Now we are going to probe into the infinite dimensional space. We have seen so far how to build a kernel from a given feature map, but can we do the opposite? Suppose a map K is a kernel, can we find the corresponding feature map ϕ such that

$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{K}}$? In the lectures, we have learned that for $K(x_i, x_j)$ to be a valid kernel function, it is *sufficient and necessary* if the corresponding Kernel matrix (Gram Matrix) is symmetric positive semidefinite.

There is no need to go into such difficulty of finding the feature maps, however, since we have the kernel-feature map equivalence (14). We only need to compute the value of the kernel function, avoiding the complexity of computing the inner product of high dimensional feature maps.

Given this intuition, we consider the radial basis function (RBF) kernel

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = \exp\left(-\gamma \|x_i - x_j\|^2\right). \tag{15}$$

In Eq. 15, γ controls the bandwidth of the kernel. For RBF kernel, the corresponding feature maps have infinite dimensional feature spaces. The RBF kernel is a reasonable measure of x_i and x_j's similarity, and is close to 1 when x_i and x_j are close, and near 0 when they are far apart. In the following problems, you are going to use the RBF kernel in SVM.

Part (b) (Bonus Questions)

Please submit your code as an appendix to this problem.

(i, 5pts, Bonus) Implement the dual SVM in problem (3) with the RBF kernel using a standard Quadratic Program solver (typically available as "quadprog" function in Matlab, R, or in `Mathprogbase.jl` in Julia; you may also refer `CVXOPT` in Python, `GORUBI`, or `MOSEK`). Load a small synthetic toy problem with inputs $X \in \mathbb{R}^{863 \times 2}$ and labels $y \in \{-1, 1\}^{863}$ from `data.txt` and solve the dual SVM with $\gamma = \{10, 50, 100, 500\}$ and $C = \{0.01, 0.1, 0.5, 1\}$. Report the optimal objective values of the dual. **Hint for Python users:** you can use the function `cvxopt.solvers.qp` detailed in this page.

```
gamma 100, C 0.10, obj -28.3742
gamma 100, C 0.50, obj -60.7804
gamma 100, C 0.01, obj -6.7948
gamma 100, C 1.00, obj -83.4114
gamma 10, C 0.10, obj -49.4808
gamma 10, C 0.50, obj -187.5612
gamma 10, C 0.01, obj -7.1303
gamma 10, C 1.00, obj -326.3421
gamma 50, C 0.10, obj -31.2606
gamma 50, C 0.50, obj -79.7071
gamma 50, C 0.01, obj -6.7163
gamma 50, C 1.00, obj -116.6115
gamma 500, C 0.10, obj -42.4175
gamma 500, C 0.50, obj -64.3648
gamma 500, C 0.01, obj -7.3065
gamma 500, C 1.00, obj -73.9632
```

Figure 1: Objective values.

```python
def calc_gram(gamma_, X):
    n = X.shape[0]
    gram_ = np.zeros((n, n))
    for i in range(n):
        for j in range(n):
            gram_[i, j] = np.exp(-gamma * np.sum((X[i] - X[j]) ** 2))

    return gram_
```

```python
gamma_list = {10,50,100,500}
C_list = {0.01, 0.1, 0.5, 1}

sol_dict = {}
for gamma in gamma_list:
    for C in C_list:
        # gram matrix
        G = matrix(calc_gram(gamma, X))
        P = matrix(np.matmul(np.matmul(np.diag(Y), G), np.diag(Y)))
        q = matrix(np.ones(ns) * (-1))

        # equality constraint
        A = matrix(Y).trans()
        b = matrix(0.0)

        # inequality constraint
        Q1 = np.diag(np.ones(ns))
        Q2 = np.diag(np.ones(ns) * -1)
        Q = np.concatenate([Q1, Q2], axis=0)
        h = np.zeros(2 * ns)
        h[:ns] = C
        Q = matrix(Q)
        h = matrix(h)

        sol = solvers.qp(P, q, Q, h, A, b)
        sol_dict[(gamma, C)] = sol
```

Figure 2: Main code for solver.

The objective values is shown in Figure 1, and the code is shown in Figure 2, where we only show the main solver code, full code can be found in Seciton **??**.

(ii, 5pts, Bonus) For each of the parameter pairs, show a scatter plot of the data and plot the decision border (where the predicted class label changes) on top. How and why does the decision boundary change with different pair of parameters?

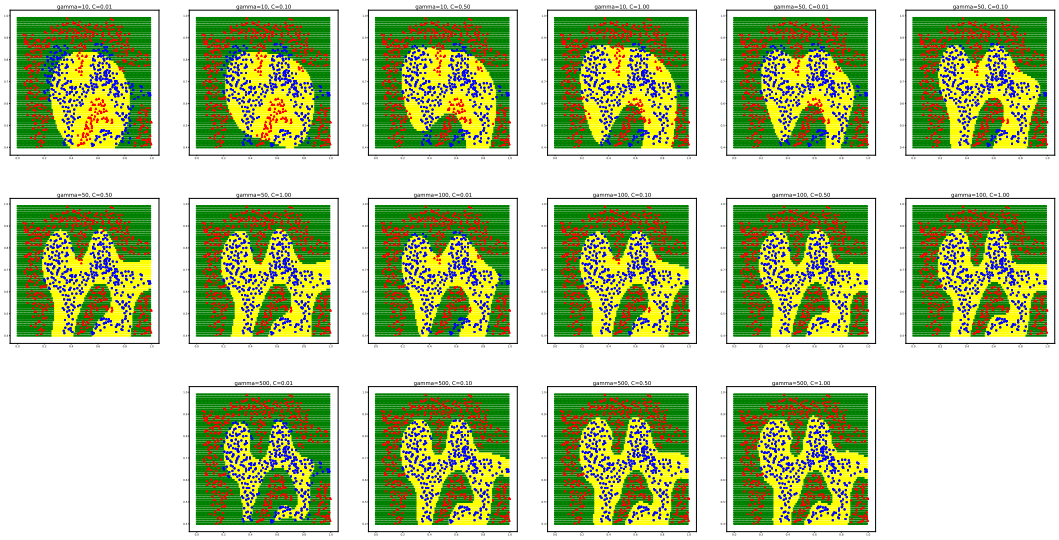

Figure 3: Scatter plot.

The figure is shown in Figure 3.