DDA3020: Homework II December 11, 2023

Problem 1 (10 points)

Regularizing separate terms in 2 d logistic regression (Exercise 8.7 of Murphy's book) (1 point) (1) Consider the data in Figure 1, where we fit the model

$$p(y=1 \mid \mathbf{x}, \mathbf{w})=\sigma\left(2 w_0+w_1 x_1+w_2 x_2\right).$$

Suppose we fit the model by maximum likelihood, i.e.,

$$J(\mathbf{w})=-\ell\left(\mathbf{w}, \mathcal{D}_{\text {train }}\right),$$

where $\ell\left(\mathbf{w}, \mathcal{D}_{\text {train }}\right)$ is the log likelihood on the training set. Sketch a possible decision boundary corresponding to $\hat{\mathbf{w}}$. (Copy the figure first (a rough sketch is enough), and then superimpose your answer on your copy, since you will need multiple versions of this figure). Is your answer (decision boundary) unique? How many classification errors does your method make on the training set?

(2) Now suppose we regularize only the w_0 parameter, i.e., we minimize

$$J_0(\mathbf{w})=-\ell\left(\mathbf{w}, \mathcal{D}_{\text {train }}\right)+\lambda w_0^2.$$

Suppose λ is a very large number, so we regularize w_0 all the way to 0 , but all other parameters are unregularized. Sketch a possible decision boundary. How many classification errors does your method make on the training set? Hint: consider the behavior of simple linear regression, $2w_0+w_1 x_1+w_2 x_2$ when $x_1=x_2=0$.

(3) Now suppose we heavily regularize only the w_1 parameter, i.e., we minimize

$$J_1(\mathbf{w})=-\ell\left(\mathbf{w}, \mathcal{D}_{\text {train }}\right)+\lambda\left|w_1\right|$$

Sketch a possible decision boundary. How many classification errors does your method make on the training set?

(4) Now suppose we heavily regularize only the w_2 parameter. Sketch a possible decision boundary. How many classification errors does your method make on the training set?

1 Solution

(1) Logistic Regression Model: Given the logistic regression model:

$$p(y = 1|\mathbf{x}, \mathbf{w}) = \sigma(2w_0 + w_1x_1 + w_2x_2)$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$.

Decision Boundary: The decision boundary occurs where $p(y = 1|\mathbf{x}, \mathbf{w}) = 0.5$, which implies:

$$2w_0 + w_1x_1 + w_2x_2 = 0$$

From this equation, we can express x_2 as a function of x_1:

$$x_2 = -\frac{2w_0}{w_2} - \frac{w_1}{w_2}x_1$$

Sketching the Decision Boundary: The decision boundary will be a straight line in the 2D plane, since it's defined by a linear combination of x_1 and x_2. The slope and intercept of this line will depend on the learned values of $\mathbf{w}$

Uniqueness of Decision Boundary: there are multiple lines that can separate the '+' and 'o' points with the same minimal classification error, thus not unique. The classification error is 0.

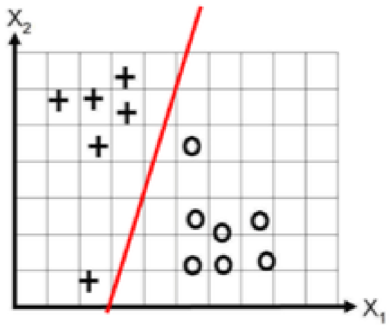

Figure 1: Visualization of data

(2) Regularization: In this problem, the cost function is:

$$J_0(\mathbf{w}) = -\ell\left(\mathbf{w}, \mathcal{D}_{\text{train}}\right) + \lambda w_0^2$$

where λ is a large regularization parameter and ℓ is the log likelihood.

If λ is very large, it'll push w_0 towards 0 because of the λw_0^2 term in the cost function.

Behavior of the Logistic Regression Model: Given that w_0 is 0, our logistic model becomes:

$$p(y = 1|\mathbf{x}, \mathbf{w}) = \sigma(w_1x_1 + w_2x_2)$$

where the decision boundary is when:

$$p(y = 1|\mathbf{x}, \mathbf{w}) = 0.5$$

which implies:

$$w_1 x_1 + w_2 x_2 = 0$$

Rearranging for x_2 we get:

$$x_2 = -\frac{w_1}{w_2} x_1$$

Decision Boundary: The decision boundary will still be a straight line, but now it will pass through the origin (since the constant term w_0 is 0). The slope of this line will depend on the ratio $-\frac{w_1}{w_2}$. Given the hint, when $x_1 = x_2 = 0$, the logistic regression output becomes 0.5, which means the decision boundary should pass through the origin.

The classification error is 1 because the boundary can cross (0,0) which made one point misclassified.

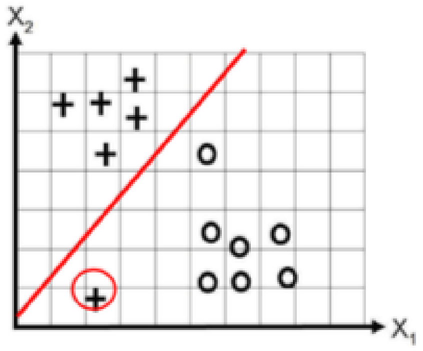

Figure 2: Visualization of data

(3) Regularization: In this problem, the cost function is:

$$J_1(\mathbf{w}) = -\ell\left(\mathbf{w}, \mathcal{D}_{\text{train}}\right) + \lambda |w_1|$$

where λ is a large regularization parameter and ℓ is the log likelihood.

When λ is very large and we're using L1 regularization (as indicated by the absolute value), it encourages sparsity in the weights. Specifically, it will push w_1 towards 0.

Behavior of the Logistic Regression Model: Given that w_1 is approximately 0 due to heavy regularization, our logistic model simplifies to:

$$p(y = 1|\mathbf{x}, \mathbf{w}) = \sigma(w_0 + w_2 x_2)$$

The decision boundary is when:

$$p(y = 1|\mathbf{x}, \mathbf{w}) = 0.5$$

which implies:

$$w_0 + w_2 x_2 = 0$$

Rearranging for x_2 we get:

$$x_2 = -\frac{w_0}{w_2}$$

Decision Boundary: The decision boundary will be a horizontal line since the term x_1 (the horizontal coordinate) no longer influences the decision. The position of this horizontal line along the x_2 axis will depend on the values of w_0 and w_2. We can't predict the exact position of the boundary, but its **orientation (horizontal)** is determined by the heavy regularization on w_1. Visualizing this on the figure, we can see the classification error is 2.

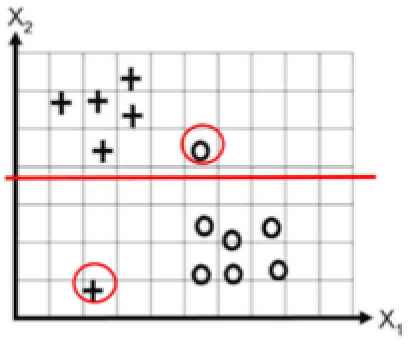

Figure 3: Visualization of data

(4) Very similar to question (3). For this question, the boundary is a vertical line, the classification error is 0.

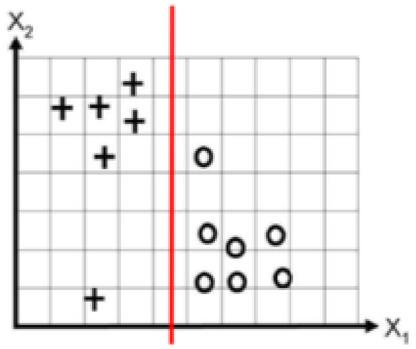

Figure 4: Visualization of data

Problem 2 (15 points)

Manual Construction of a Non-Linear SVM Classifier (Inspired by Exercise 14.1 in Murphy's book)

Consider a dataset with two points in 1D: ($x_1 = -1, y_1 = -1$) and ($x_2 = 2, y_2 = 1$). We will transform each point into 3D using the mapping $\phi(x) = \left[1, x, x^2\right]^\top$. This is analogous to employing a quadratic kernel. The optimal margin classifier is described by:

$$\begin{aligned} &\min \|\mathbf{w}\|^2 \quad \text{subject to} \\ &y_1 \left(\mathbf{w}^T \phi\left(\mathbf{x}_1\right) + w_0\right) \geq 1 \\ &y_2 \left(\mathbf{w}^T \phi\left(\mathbf{x}_2\right) + w_0\right) \geq 1. \end{aligned}$$

(1) Propose a vector that could be perpendicular to the optimal orientation of **w** in the transformed space.

(2) Calculate the margin accomplished with your suggested w. Remember, the margin reflects the distance of each support vector to the decision boundary. For guidance, consider the spatial relationship between two points and a plane dividing them.

(3) Deduce the exact **w** by leveraging the principle that the margin equates to $1/\|\mathbf{w}\|$.

2 Solution

(1) Given the mapping function $\phi(x) = [1, x, x^2]^\top$:

For $x_1 = -1$, $\phi(x_1) = [1, -1, 1]^\top$

For $x_2 = 2$, $\phi(x_2) = [1, 2, 4]^\top$

In the 3D space, the transformed points $\phi(x_1)$ and $\phi(x_2)$ are not just points but represent two classes. The optimal margin classifier aims to find a hyperplanethat separates these two classes with the maximum margin.

The vector **w** is perpendicular to this hyperplane. Hence, if we find the direction vector joining the two transformed points, that vector will be parallel to **w**.

Using the transformed points $\phi(x_1)$ and $\phi(x_2)$ to calculate a direction vector joining the two transformed points. The direction vector **d** is (parallel to w):

$$\mathbf{d} = \begin{bmatrix} 1 - 1 \\ 2 - (-1) \\ 4 - 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 3 \end{bmatrix}$$

Therefore, a vector that could be perpendicular to the optimal orientation of **w** in the transformed space is perpendicular to $\begin{bmatrix}0\\3\\3\end{bmatrix}$. Any vector that makes sense can be the answer, for example $\begin{bmatrix}0\\-1\\1\end{bmatrix}$

(2) Because we have 2 points, the margin should be half of the distance between $\phi(\mathbf{x}_1)$ and $\phi(\mathbf{x}_2)$:

As the direction vector is $\begin{bmatrix}0\\3\\3\end{bmatrix}$, we have:

margin $= 1/2 * \|\mathbf{d}\| = \sqrt{0+9+9} = \frac{3\sqrt{2}}{2}$

(3) Deduce the exact **w** using the principle that the margin equates to $\frac{1}{\|\mathbf{w}\|}$.

Given that the margin equates to $\frac{1}{\|\mathbf{w}\|}$, we can set the two expressions equal to each other:

$$\frac{3\sqrt{2}}{2} = \frac{1}{\|\mathbf{w}\|}$$

$$\|\mathbf{w}\| = \frac{\sqrt{2}}{3}$$

Give the direction vector **d** is parallel to w, we can assume $\mathbf{w} = [0, a, a]^T$. We can solve $|a| = \frac{1}{3}$.

Given

$$y_1\left(\mathbf{w}^T\phi(\mathbf{x}_1) + w_0\right) \geq 1$$
$$y_2\left(\mathbf{w}^T\phi(\mathbf{x}_2) + w_0\right) \geq 1.$$

we will get:

$$-w_0 \geq 1$$
$$6a + u_0 \geq 1$$

so $a = \frac{1}{3}$, $\mathbf{w} = [0, \frac{1}{3}, \frac{1}{3}]^T$

Problem 3 (15 points)

Given a binary data set: (2 points)

$$\text{Class - 1}: \begin{bmatrix} (1 & 0) \\ (0 & 1) \\ (-1 & 0) \\ (0 & -1) \end{bmatrix} \quad \text{Class } +1: \begin{bmatrix} (2 & 0) \\ (0 & 2) \\ (-2 & 0) \\ (0 & -2) \end{bmatrix}$$

(1) Can you find a svm linear classifier for this data set? If not, how to find a nonlinear classifier?

(2) Explain the concept of the RBF kernel and how it allows for non-linear separation in the feature space. Discuss its advantages over linear SVM, especially in relation to this specific dataset.

3 Solution

(1) Given the provided binary dataset, we can first visualize the data to determine if it's linearly separable.

The data points for Class -1 are: (1, 0), (0, 1), (-1, 0), (0, -1)

The data points for Class +1 are: (2, 0), (0, 2), (-2, 0), (0, -2)

From the given data points, it's evident that the Class -1 points are closer to the origin, while the Class +1 points are further away from the origin. The data points for both classes form a square shape around the origin.

Given this configuration, the data is not linearly separable because there's no straight line that can separate the two classes without misclassifying at least one point.

However, the data is radially separable, meaning that a circle (or in higher dimensions, a hypersphere) can be used to separate the two classes. The Class -1 points lie inside this circle, while the Class +1 points lie outside of it.

To find a nonlinear classifier for this dataset, we can use the SVM with a radial basis function (RBF) kernel, which is commonly used for radially separable data. The RBF kernel can transform the data into a higher-dimensional space where it becomes linearly separable, allowing the SVM to find a hyperplane that separates the two classes.

In summary: 1) The data is not linearly separable. 2) To classify this data, we can use an SVM with an RBF kernel.

(2) The Radial Basis Function (RBF) kernel, often referred to as the Gaussian kernel, is a popular kernel used in SVM and other kernelized algorithms. The formula for the RBF kernel between two data points x and z is given by:

$$K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right)$$

Where: $\|x - z\|$ is the Euclidean distance between the two data points.

σ is a parameter that determines the width of the Gaussian function.

Non-linear Separation in Feature Space:

The RBF kernel implicitly maps the input data into a higher-dimensional feature space where the data might become linearly separable, even if it's not in the original space. This is achieved without explicitly computing the transformation or even knowing what the transformed space looks like. The kernel trick allows us to compute dot products in this higher-dimensional space using only the original data points.

The RBF kernel measures the similarity between data points. Points that are close in the input space will have a kernel value close to 1, while points that are far apart will have a value close to 0. This creates a nonlinear boundary in the original space based on these similarities.

Advantages over Linear SVM:

1. Handling Non-linear Data: The primary advantage of the RBF kernel over a linear SVM is its ability to handle non-linearly separable data. While a linear SVM can only find a linear decision boundary, the RBF kernel allows SVM to find complex, non-linear decision boundaries.

2. Flexibility: The RBF kernel has a parameter σ that determines the width of the Gaussian function. By tuning σ, one can control the flexibility of the decision boundary. A small σ results in a more flexible boundary, while a large σ results in a smoother boundary.

For the provided dataset, the data points from the two classes form squares that are centered around the origin. This configuration is radially separable, meaning the data can be separated by a circle (or hypersphere in higher dimensions).

A linear SVM would fail to separate this data because there's no straight line that can segregate the two classes. However, the RBF kernel SVM can capture the radial nature of the separation. The decision boundary in the original space would resemble a circle around the origin, with Class -1 points inside and Class +1 points outside.

Problem 4 (20 points)

Analyzing Margin Width in Relation to Dual Variables in SVM.

In SVM, the margin width γ plays a crucial role in understanding the geometry of the decision boundary. Given the dual formulation of the SVM optimization problem, we seek to establish a direct relationship between γ and the Lagrange multipliers $\{\alpha_n\}$.

Consider a different perspective on the SVM's optimization, focusing on an alternate representation of the decision function involving a transformation ϕ applied to the input vectors $\mathbf{x}$, such that the dual problem now reads:

$$\begin{aligned}
&\max_{\alpha} \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \phi\left(\mathbf{x}_n\right)^{\top} \phi\left(\mathbf{x}_m\right) \\
&\text{s.t. } \sum_{n=1}^{N} \alpha_n y_n = 0 \\
&\qquad \alpha_n \geq 0 \quad \forall n = 1, 2, \ldots, N
\end{aligned}$$

Your task involves two primary objectives:

(1) Derive an expression for γ in terms of $\{\alpha_n\}$, assuming that the transformation ϕ influences the inner product between feature vectors. Substantiate your derivation by exploring how ϕ affects the geometry of the decision boundary.

(2) Prove that the original relationship, $\frac{1}{\gamma^2} = \sum_{n=1}^{N} \alpha_n$, holds true even under the transformation ϕ. In your proof, emphasize any new mathematical insights or properties that emerge due to the involvement of ϕ.

Hint: You might need to revisit the concept of kernels and how they implicitly apply transformations on input vectors, influencing the SVM's decision function.

4 Solution

(1) The objective function of the support vector machine is:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \tag{1}$$

subject to $y_i(w^T x_i + b) \geq 1, \forall i$.

Then, the Lagrangian function becomes:

$$L(\alpha, w, b) = \frac{1}{2} \|w\|^2 - \sum_{n=1}^{N} \alpha_n \left[y_n \left(w^T x_n + b \right) - 1 \right] \tag{2}$$

Since the second term (including the negative sign) is always negative, the Lagrange primal function p is:

$$p(\alpha) = \frac{1}{2}\|w\|^2 \tag{3}$$

Then, the primal problem becomes:

$$\min_{\alpha \geq 0} \frac{1}{2}\|w\|^2 \tag{4}$$

Next, the Lagrange dual function $d(\alpha)$ is:

$$d(\alpha) = \min L(\alpha, w, b) \tag{5}$$

By KKT conditions:

$$\frac{\partial L}{\partial w} = w - \sum_{n=1}^{N} \alpha_n y_n x_n = 0 \tag{6}$$

$$\Rightarrow w = \sum_{n=1}^{N} \alpha_n y_n x_n \tag{7}$$

$$\frac{\partial L}{\partial b} = -\sum_{n=1}^{N} \alpha_n y_n b = 0 \tag{8}$$

$$\Rightarrow \sum_{n=1}^{N} \alpha_n y_n = 0 \tag{9}$$

Then, we get

$$-\sum_{n=1}^{N} \alpha_n [y_n(w^T x_n + b) - 1] = -\sum_{n=1}^{N} \alpha_n y_n \left(\sum_{m=1}^{N} \alpha_m y_m x_m \right)^T x_n + \sum_{n=1}^{N} \alpha_n \tag{10}$$

$$= -\sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m + \sum_{n=1}^{N} \alpha_n \tag{11}$$

Therefore, the dual function becomes:

$$d(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m \tag{12}$$

$$= \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\|w\|^2 \tag{13}$$

subject to

$$\sum_{n=1}^{N} \alpha_n y_n = 0 \tag{14}$$

Influence of ϕ: The transformation ϕ maps the input vectors $\mathbf{x}$ into a higher-dimensional space. The inner product in this space is given by $\phi(\mathbf{x}_n)^\top \phi(\mathbf{x}_m)$. The choice of ϕ determines the geometry of the decision boundary in the transformed space. For instance, a linear ϕ will result in a linear decision boundary, while a polynomial or radial basis function (RBF) ϕ can result in non-linear decision boundaries.

The kernel trick in SVM allows us to compute the inner products in the transformed space without explicitly computing the transformation ϕ. The kernel function $K(\mathbf{x}_n, \mathbf{x}_m)$ is defined as:

$$K(\mathbf{x}_n, \mathbf{x}_m) = \phi(\mathbf{x}_n)^\top \phi(\mathbf{x}_m)$$

By using the kernel trick, we can handle high-dimensional spaces efficiently, even infinite-dimensional spaces in the case of the RBF kernel.

(2) Since both primal and dual problems have the same optimal value.

$$\sum_{n=1}^{n} \alpha_n - \frac{1}{2}\|w\|^2 = \frac{1}{2}\|w\|^2 \tag{15}$$

Then, by $\gamma = \|w\|$:

$$\sum_{n=1}^{n} \alpha_n = \|w\|^2 = \frac{1}{\gamma^2} \tag{16}$$

Hence, the kernel function doesn't not influence the final results, it has been cancelled out during the calculation.