introduction to machine learning

DR.Amiri

electrical engineering department

Ahmadreza Majlesara 400101861

assignment 4

July 11, 2024

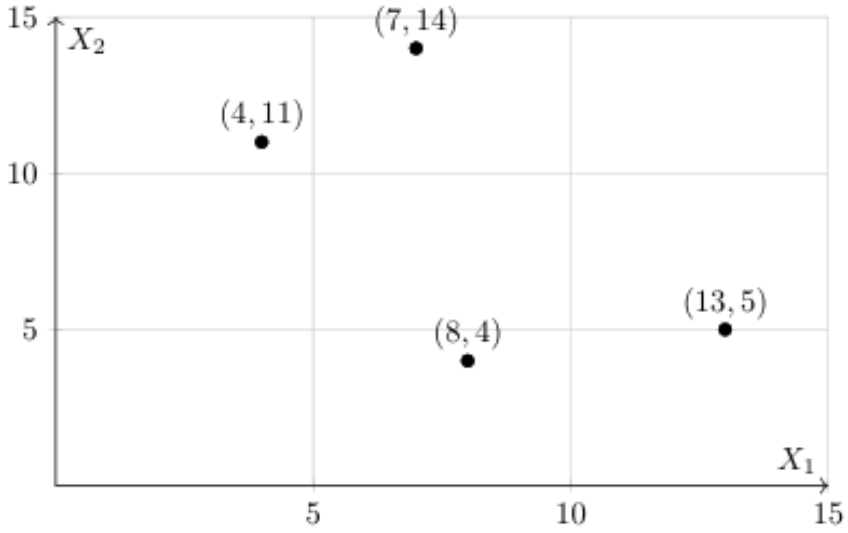
introduction to machine learning

assignment 4

Ahmadreza Majlesara 400101861

Dimensionality Reduction using PCA

For the following data points, find the first principal component and then project the data points onto it.



solution

first we define X as the matrix of data points:

$$X = \begin{bmatrix} 4 & 11 \\ 7 & 14 \\ 8 & 4 \\ 13 & 5 \end{bmatrix}$$

we can define the covariance matrix as:

$$S = \frac{1}{n}(X^T X - \frac{1}{n} X^T 1_n 1_n^T X)$$

where 1_n is a vector of ones of size n.

so we have:

$$S = \frac{1}{4}\begin{bmatrix} 4 & 7 & 8 & 13 \\ 11 & 14 & 4 & 5 \end{bmatrix}\begin{bmatrix} 4 & 11 \\ 7 & 14 \\ 8 & 4 \\ 13 & 5 \end{bmatrix} - \frac{1}{4}\begin{bmatrix} 4 & 7 & 8 & 13 \\ 11 & 14 & 4 & 5 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} 4 & 11 \\ 7 & 14 \\ 8 & 4 \\ 13 & 5 \end{bmatrix}$$

so we have:

$$S = \begin{bmatrix} 10.5 & -8.25 \\ -8.25 & 17.25 \end{bmatrix}$$

now we can find the eigenvalues and eigenvectors of S:

$$det(S - \lambda I) = 0$$

$$det\begin{bmatrix} 10.5 - \lambda & -8.25 \\ -8.25 & 17.25 - \lambda \end{bmatrix} = 0$$

$$\Rightarrow \lambda_1 = 22.79, \lambda_2 = 4.68$$

$$\Rightarrow v_1 = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}, v_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$$

so the first principal component is v_1 and the projection of the data points onto it is:

$$Xv_1 = \begin{bmatrix} 4 & 11 \\ 7 & 14 \\ 8 & 4 \\ 13 & 5 \end{bmatrix}\begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix} = \begin{bmatrix} -6.9037 \\ -7.7224 \\ 1.138 \\ 3.0947 \end{bmatrix}$$

Reconstruction Error

We want to perform PCA. Each sample $x_i \in \mathbb{R}^p$ is projected onto the new coordinate system using $z_i = V_{1:k}^T x_i$. Here, $V_{1:k}$ is the matrix of the first k principal components ($V_{1:k} = [v_1|v_2|...|v_k]$). We can reconstruct x_i from z_i using the equation $\hat{x}_i = V_{1:k} z_i$.

part 1

Prove:

$$\|\hat{x}_i - \hat{x}_j\|_2 = \|z_i - z_j\|_2$$

solution

from the given information, we have:

$$\hat{x}_i = V_{1:k} z_i$$

$$\hat{x}_j = V_{1:k} z_j$$

so we can write:

$$\hat{x}_i - \hat{x}_j = V_{1:k} z_i - V_{1:k} z_j$$

$$\hat{x}_i - \hat{x}_j = V_{1:k}(z_i - z_j)$$

thus we can conclude that:

$$\|\hat{x}_i - \hat{x}_j\|^2 = \|V_{1:k}(z_i - z_j)\|^2$$

so if we expand the right side of the equation by the fact that $\|x\|^2 = x^T x$ we get:

$$\|V_{1:k}(z_i - z_j)\|^2 = (V_{1:k}(z_i - z_j))^T (V_{1:k}(z_i - z_j))$$

$$= (z_i - z_j)^T V_{1:k}^T V_{1:k}(z_i - z_j)$$

Since $V_{1:k}$ consists of the first k principal components, its columns are orthonormal.

$$\Rightarrow V_{1:k}^T V_{1:k} = I$$

$$\Rightarrow (z_i - z_j)^T V_{1:k}^T V_{1:k}(z_i - z_j) = (z_i - z_j)^T I (z_i - z_j)$$

$$= (z_i - z_j)^T (z_i - z_j)$$

$$= \|z_i - z_j\|^2$$

thus we approve that:

$$\Rightarrow \|\hat{x}_i - \hat{x}_j\|^2 = \|z_i - z_j\|^2$$

part 2

Prove:

$$\sum_{i=1}^{n} \|x_i - \hat{x}_i\|_2^2 = (n-1) \sum_{i=k+1}^{p} \lambda_i$$

What can be inferred from this equation regarding the reconstruction error?

solution

again from the given information, we have:

$$\hat{x_i} = V_{1:k} V_{1:k}^T x_i$$

so

$$x_i - \hat{x_i} = x_i - V_{1:k} V_{1:k}^T x_i$$

this can be written as:

$$x_i - \hat{x_i} = (I - V_{1:k} V_{1:k}^T) x_i$$

and if we get norm of the above equation we get:

$$\|x_i - \hat{x_i}\|^2 = \|(I - V_{1:k} V_{1:k}^T) x_i\|^2$$

Sum of Squared Reconstruction Errors can be written as:

$$\sum_{i=1}^{n} \|x_i - \hat{x_i}\|^2 = \sum_{i=1}^{n} \|(I - V_{1:k} V_{1:k}^T) x_i\|^2$$

$$\sum_{i=1}^{n} \|(I - V_{1:k} V_{1:k}^T) x_i\|^2 = \sum_{i=1}^{n} \|V_{k+1:p} V_{k+1:p}^T x_i\|^2$$

$$= \sum_{i=1}^{N} (V_{k+1:p} V_{k+1:p}^T x_i)^T (V_{k+1:p} V_{k+1:p}^T x_i)$$

$$= \sum_{i=1}^{N} x_i^T V_{k+1:p} V_{k+1:p}^T V_{k+1:p} V_{k+1:p}^T x_i$$

so cause $V_{k+1:p}$ is orthogonal matrix, we have:

$$= \sum_{i=1}^{N} x_i^T V_{k+1:p} V_{k+1:p}^T x_i$$

$$= \sum_{i=1}^{N} Tr(x_i^T V_{k+1:p} V_{k+1:p}^T x_i)$$

$$= \sum_{i=1}^{N} Tr(V_{k+1:p}^T x_i x_i^T V_{k+1:p})$$

so as we know we can define $S = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T$ and as $\bar{x} = 0$ this can be written as:

$$S = \frac{1}{N-1} \sum_{i=1}^{N} x_i x_i^T$$

so we can write:

$$\sum_{i=1}^{N} Tr(V_{k+1:p}^T x_i x_i^T V_{k+1:p}) = Tr(V_{k+1:p}^T S V_{k+1:p})$$

which is equal to:

$$(n-1)Tr(V_{k+1:p}^T S V_{k+1:p}) = (n-1)\sum_{i=k+1}^{p}\lambda_i$$

so the sum of squared reconstruction errors is equal to:

$$\sum_{i=1}^{n}\|x_i-\hat{x}_i\|_2^2 = (n-1)\sum_{i=k+1}^{p}\lambda_i$$

conclusion:

The equation $\sum_{i=1}^{n}\|x_i-\hat{x}_i\|^2 = (n-1)\sum_{i=k+1}^{p}\lambda_i$ provides insights into how PCA minimizes reconstruction error by focusing on the largest eigenvalues (principal components) that capture the most variance in the data. It highlights the trade-offs involved in dimensionality reduction and reconstruction accuracy in PCA.The reconstruction error is directly related to the eigenvalues that correspond to the discarded principal components. The larger these eigenvalues, the greater the reconstruction error. If we use more principal components (larger k) to reconstruct the data, the reconstruction error will decrease.

Mixture Models

In this question we get introduced to mixture models.

Introduction to Mixture Models

In your own words, explain how the MM algorithm can deal with non convex optimization objective functions by considering simpler convex objective functions.

solution

the MM algorithm can deal with non convex optimization objective functions by considering simpler convex objective functions by using the Expectation-Maximization (EM) algorithm. in this algorithm we define a simpler function like $g(\theta)$ and we try to maximize it so if we maximize this function the $\bar{\theta}$ is actually the maximizer of the original non-convex function $f(\theta)$.
the function we define as $g(\theta)$ is a lower bound of the original function $f(\theta)$ like solving a dual problem in optimization.this works as the equation below:

$$f(\theta) \geq g(\theta) \quad \forall \theta$$

$$\text{if } \theta = \bar{\theta} \text{ then } f(\bar{\theta}) = g(\bar{\theta})$$

Mixture Models for specific distribution

You are given a data set $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, $x_i \in \mathbb{R}^d, y_i \in \mathbb{R}, i = 1, \ldots, N$. The data points accumulate on m different lines, $a_j^T x_i = y_i$, for $a_j \in \mathbb{R}^d, j = 1, \ldots, m$.

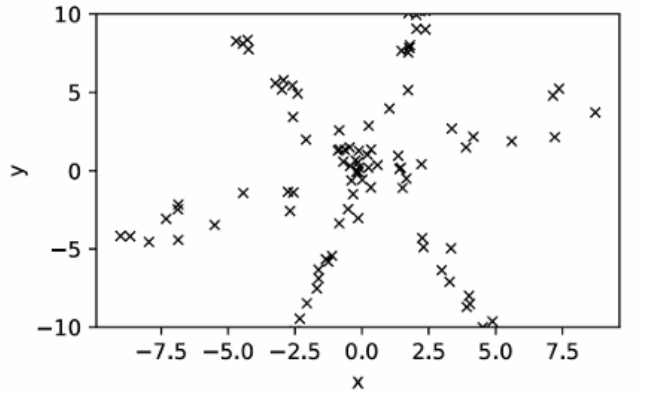


Figure 1. d=1 and m=3

Then

$$p(x, y \mid \theta) = \sum_{j=1}^{m} \pi_j \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(a_j^T x - y\right)^2}{2\sigma^2}\right),$$

where $\theta = (\pi_{1:m}, a_{1:m})$, $\sum \pi_j = 1$, $\pi_j \geq 0$ and $\sigma > 0$ is given and fixed.

- Find the responsibilities in the E-step of Soft EM,

$$r_{nk}^{(t)} = p\left(z_n = k \mid x_n, y_n, \theta^{(t-1)}\right).$$

- Write down the class predictions $z_n^{(t)}$ for (x_n, z_n) in the E-step of Hard EM in terms of $r_{nk}^{(t)}$.
- Assume that we observe the true labels $z_1, \ldots, z_\ell$ for the first ℓ datapoints, $\ell < N$. How can we modify the E-step of Soft EM to incorporate the additional information?

solution

we know that:

$$p(z_n = k|x_n, y_n, \theta) = \frac{p(x_n, y_n|z_n = k, \theta)p(z_n = k|\theta)}{\sum_{j=1}^{m} p(x_n, y_n|z_n = j, \theta)p(z_n = j|\theta)}$$

we define $p(z_n = j, \theta)$ as π_j and from the given information $r_{nk}^{(t)} = p\left(z_n = k \mid x_n, y_n, \theta^{(t-1)}\right)$ so we can wtirite:

$$p(x, y|z = k) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(a_k^T x - y\right)^2}{2\sigma^2}\right)$$

$$\Rightarrow r_{nk}^{(t)} = \frac{\pi_k^{(t-1)} \exp\left(-\frac{\left(a_k^T x_n - y_n\right)^2}{2\sigma^2}\right)}{\sum_{j=1}^{m} \pi_j^{(t-1)} \exp\left(-\frac{\left(a_j^T x_n - y_n\right)^2}{2\sigma^2}\right)}$$

for the second part we have:

$$z_n^{(t)} = \arg\max_k \log r_{nk}^{(t)}$$

we calculated the $r_{nk}^{(t)}$ in the previous part so we can write:

$$z_n^{(t)} = \arg\max_k \log r_{nk}^{(t)} = \arg\max_k \log \left(\frac{\pi_k^{(t-1)} \exp\left(-\frac{\left(a_k^T x_n - y_n\right)^2}{2\sigma^2}\right)}{\sum_{j=1}^{m} \pi_j^{(t-1)} \exp\left(-\frac{\left(a_j^T x_n - y_n\right)^2}{2\sigma^2}\right)} \right)$$

the dominator is the same for all k so we can ignore it and we can write:

$$z_n^{(t)} = \arg\max_k \log \pi_k^{(t-1)} - \frac{\left(a_k^T x_n - y_n\right)^2}{2\sigma^2}$$

and for the third part we can use the true labels to modify the E-step of Soft EM by setting the $r_{nk}^{(t)}$ to 1 for the true label and 0 for the others. so the new $r_{nk}^{(t)}$ will be:

for the labeled data:

$$r_{nk}^{(t)} = \begin{cases} 1 & \text{if } z_n = k \\ 0 & \text{otherwise} \end{cases}$$

for the unlabeled data:

$$r_{nk}^{(t)} = \frac{\pi_k^{(t-1)} \exp\left(-\frac{\left(a_k^T x_n - y_n\right)^2}{2\sigma^2}\right)}{\sum_{j=1}^{m} \pi_j^{(t-1)} \exp\left(-\frac{\left(a_j^T x_n - y_n\right)^2}{2\sigma^2}\right)}$$

EM Algorithm for GMM and CMM

This part, we want you to apply EM algorithm to learn (estimate) parameters for two different mixture model and classify them according to some of their parameters.

EM for Gaussian Mixture Model

Compute estimate of parameters for Gaussian Mixture Models for N observed data $\{x_n\}_{n=1}^{N}$:

- Determine model parameters and initialize them.
- Compute complete dataset likelihood.
- Compute model parameter updates using EM algorithm.

solution

for first step we need to determine model parameters and initialize them.

$$p(x_n|\theta) = \sum_{k=1}^{K} p(x_n|z_n = k, \theta_k) p(z_n = k|\theta)$$

$$= \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \quad \pi_k = \pi_k^{(0)}, \mu_k = \mu_k^{(0)}, \Sigma_k = \Sigma_k^{(0)}$$

now for the second part we have:

$$p(D|\theta) = \prod_{n=1}^{N} p(x_n, z_n|\theta) = \prod_{n=1}^{N} p(x_n, z_n|\theta) p(z_n|\theta)$$

$$= \prod_{n=1}^{N} \pi_{z_n} \mathcal{N}(x_n|\mu_{z_n}, \Sigma_{z_n})$$

$$\log p(D|\theta) = \sum_{n=1}^{N} \log \pi_{z_n} + \log \mathcal{N}(x_n|\mu_{z_n}, \Sigma_{z_n})$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \delta_{z_n,k} (\log \pi_k + \log \mathcal{N}(x_n|\mu_k, \Sigma_k))$$

and for the third part we have:

$$q^{(t)}(z_n) = p(z_n|x_n, \theta^{(t)}) = \frac{p(x_n|z_n = k, \theta^{(t)}) p(z_n = k|\theta^{(t)})}{\sum_{k=1}^{K} p(x_n|z_n = k, \theta^{(t)}) p(z_n = k|\theta^{(t)})}$$

$$\Rightarrow q^{(t)}(z_n) = \frac{\pi_k^{(t-1)} \mathcal{N}(x_n|\mu_k^{(t-1)}, \Sigma_k^{(t-1)})}{\sum_{k=1}^{K} \pi_k^{(t-1)} \mathcal{N}(x_n|\mu_k^{(t-1)}, \Sigma_k^{(t-1)})}$$

now we define our loss function as:

$$\mathscr{L}(\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} q^{(t)}(z_n)$$

we want to maximize tuhis function subject to $\sum_{k=1}^{K} \pi_k = 1$ so we define the lagrangian function as:

$$L = \sum_{i=1}^{N} \sum_{k=1}^{K} q^{(t)}(z_n) + \lambda(1 - \sum_{k=1}^{K} \pi_k)$$

$$\frac{\partial L}{\partial \pi_k} = \sum_{n=1}^{N} q^{(t)}(z_n) - \lambda = 0$$

$$\Rightarrow \pi_k = \frac{1}{\lambda} \sum_{n=1}^{N} q^{(t)}(z_n)$$

$$\sum_{k=1}^{K} \pi_k = 1 \Rightarrow \lambda = N$$

$$\pi_k^{t+1} = \frac{1}{N} \sum_{n=1}^{N} q^{(t)}(z_n)$$

to compute μ_k we have:

$$\frac{\partial L}{\partial \mu_k} = \sum_{n=1}^{N} q^{(t)}(z_n) \Sigma_k^{-1}(x_n - \mu_k) = 0$$

$$\mu_k^{t+1} = \frac{\sum_{n=1}^{N} q^{(t)}(z_n) x_n}{\sum_{n=1}^{N} q^{(t)}(z_n)}$$

and for Σ_k we have:

$$\frac{\partial L}{\partial \Sigma_k} = \sum_{n=1}^{N} q^{(t)}(z_n) \Sigma_k^{-1} - \frac{1}{2} \sum_{n=1}^{N} q^{(t)}(z_n) \Sigma_k^{-1}(x_n - \mu_k)(x_n - \mu_k)^T \Sigma_k^{-1} = 0$$

$$\Sigma_k^{t+1} = \frac{\sum_{n=1}^{N} q^{(t)}(z_n)(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} q^{(t)}(z_n)}$$

EM for Categorical Mixture Model

Compute estimate of parameters for Categorical Mixture Models for N observed data $\{x_n\}_{n=1}^{N}$:

- Determine model parameters and initialize them.
- Compute complete dataset likelihood.
- Compute model parameter updates using EM algorithm.
- Find closed-form solution for parameters using EM algorithm.

solution

for first step we need to determine model parameters and initialize them.

$$p(x_n|\theta) = \sum_{k=1}^{K} p(x_n|z_n = k, \theta_k) p(z_n = k|\theta)$$

$$= \sum_{k=1}^{K} \pi_k \prod_{c=1}^{C} \theta_{k,c}^{x_{n,c}} \quad \pi_k = \pi_k^{(0)}, \theta_k = \theta_k^{(0)}$$

now for the second part we have:

$$p(D|\theta) = \prod_{n=1}^{N} p(x_n, z_n|\theta) = \prod_{n=1}^{N} p(x_n, z_n|\theta) p(z_n|\theta)$$

$$= \prod_{n=1}^{N} \pi_{z_n} \prod_{c=1}^{C} \theta_{z_n,c}^{x_{n,c}}$$

$$\log p(D|\theta) = \sum_{n=1}^{N} \log \pi_{z_n} + \sum_{c=1}^{C} x_{n,c} \log \theta_{z_n,c}$$

and for the third part we have:

$$q^{(t)}(z_n) = p(z_n|x_n, \theta^{(t)}) = \frac{p(x_n|z_n = k, \theta^{(t)}) p(z_n = k|\theta^{(t)})}{\sum_{k=1}^{K} p(x_n|z_n = k, \theta^{(t)}) p(z_n = k|\theta^{(t)})}$$

$$\Rightarrow q^{(t)}(z_n) = \frac{\pi_k^{(t-1)} \theta_{k,x_n}}{\sum_{k=1}^{K} \pi_k^{(t-1)} \theta_{k,x_n}}$$

now we define our loss function as:

$$\mathscr{L}(\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} q^{(t)}(z_n)$$

we have to maximize it subject to $\sum_{k=1}^{K} \pi_k = 1$ so we define the lagrangian function as:

$$L = \sum_{i=1}^{N} \sum_{k=1}^{K} q^{(t)}(z_n) + \lambda (1 - \sum_{k=1}^{K} \pi_k)$$

$$\frac{\partial L}{\partial \pi_k} = \sum_{n=1}^{N} q^{(t)}(z_n) - \lambda = 0$$

$$\Rightarrow \pi_k = \frac{1}{\lambda} \sum_{n=1}^{N} q^{(t)}(z_n)$$

$$\sum_{k=1}^{K} \pi_k = 1 \Rightarrow \lambda = N$$

$$\pi_k^{t+1} = \frac{1}{N} \sum_{n=1}^{N} q^{(t)}(z_n)$$

now to update $\theta_{k,c}$ we have:

$$\mathscr{L}(\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} q^{(t)}(z_n) \log \theta_{k,x_n}$$

$$\frac{\partial \mathscr{L}(\theta)}{\partial \theta_{k,c}} = \sum_{n=1}^{N} q^{(t)}(z_n) \frac{x_{n,c}}{\theta_{k,c}} = 0$$

$$\theta_{k,c}^{t+1} = \frac{\sum_{n=1}^{N} q^{(t)}(z_n) x_{n,c}}{\sum_{n=1}^{N} q^{(t)}(z_n)}$$

Advanced Hard-Margin SVM with Dual Problem and Kernel Methods

Primal Formulation and Geometric Interpretation

Define the primal optimization problem for a hard-margin SVM. Clearly state the objective function and the constraints. Provide a geometric interpretation of the margin and explain why maximizing the margin is important in the context of classification.

solution

The primal optimization problem for a hard-margin SVM can be defined as follows:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1, \quad i = 1,2,\ldots,n \end{aligned}$$

where $\mathbf{w}$ is the weight vector, b is the bias term, $\mathbf{x}_i$ are the input vectors, and y_i are the corresponding class labels. The objective function aims to minimize the norm of the weight vector, which corresponds to maximizing the margin between the two classes. The constraints ensure that all data points are correctly classified with a margin of at least 1.

Geometrically, the margin represents the distance between the decision boundary and the closest data point from either class. Maximizing the margin ensures that the decision boundary is as far away as possible from the data points, which leads to better generalization and robustness of the classifier. By maximizing the margin, the SVM aims to find the hyperplane that best separates the two classes while maintaining a safe margin of separation.

Derive the Dual Problem from the Primal Formulation

Starting from the primal formulation, derive the dual optimization problem for the hard-margin SVM. Introduce Lagrange multipliers and formulate the Lagrangian. Show detailed steps to obtain the dual problem by minimizing the Lagrangian with respect to the primal variables $\mathbf{w}$ and b. Ensure to explain the mathematical properties and assumptions used in the derivation.

solution

To derive the dual problem from the primal formulation of the hard-margin SVM, we introduce Lagrange multipliers $\alpha_i \geq 0$ for each constraint. The Lagrangian is defined as:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left[y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1\right]$$

where $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$.
To derive the dual problem, we minimize the Lagrangian with respect to $\mathbf{w}$ and b by setting the derivatives to zero:

$$\nabla_{\mathbf{w}} L(\mathbf{w}, b, \alpha) = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0$$
$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0$$

Substituting these back into the Lagrangian, we obtain the dual Lagrangian:

$$\mathscr{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

subject to the constraints $\alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$. The dual problem is then given by:

$$\text{maximize} \quad \mathscr{L}(\alpha)$$

with the constraints $\alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$.

KKT Conditions and Support Vectors

State the Karush-Kuhn-Tucker (KKT) conditions for the hard-margin SVM. Use these conditions to explain the significance of support vectors in the context of the dual problem.

solution

The Karush-Kuhn-Tucker (KKT) conditions for the hard-margin SVM are as follows:

$$\nabla_{\mathbf{w}} L(\mathbf{w}, b, \alpha) = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0$$
$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] = 0$$
$$\alpha_i \geq 0$$
$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0$$
$$\alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] = 0$$

The KKT conditions state that the Lagrange multipliers α_i are non-negative and are zero for all data points that are not support vectors. The support vectors are the data points that lie on the margin or violate the margin constraint, and they have non-zero Lagrange multipliers. These support vectors play a crucial role in defining the decision boundary of the SVM, as they contribute to the computation of the weight vector $\mathbf{w}$ and the bias term b.
In the context of the dual problem, the support vectors are the data points that define the hyperplane that separates the two classes. By having non-zero Lagrange multipliers, these support vectors influence the decision boundary and the margin of the SVM. The KKT conditions ensure that the support vectors are correctly classified and lie on the margin, thereby defining the optimal separating hyperplane.

Kernel Trick and Dual Problem Reformulation

Using the kernel trick, replace the inner product in the dual problem with a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$. Derive the dual optimization problem for a hard-margin SVM with a specific kernel, such as the Gaussian RBF kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$. Discuss the computational benefits of using the kernel trick and how it allows handling non-linear separable data.

solution

To incorporate the kernel trick into the dual problem, we replace the inner product $\mathbf{x}_i \cdot \mathbf{x}_j$ with the kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$. For the Gaussian RBF kernel, the kernel function is defined as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

The dual optimization problem with the Gaussian RBF kernel is then given by:

$$\text{maximize} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

subject to the constraints $\alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$. The kernel trick allows us to operate in a higher-dimensional feature space without explicitly computing the transformation $\phi(\mathbf{x})$. By using the kernel function, we can efficiently compute the dot products in the feature space without explicitly mapping the data points.

This computational benefit is crucial for handling non-linearly separable data, as it allows us to find complex decision boundaries in high-dimensional spaces without the need for explicit feature mapping. The kernel trick enables SVMs to capture non-linear relationships between data points by implicitly transforming them into a higher-dimensional space, where the data becomes linearly separable.

Solve the Dual Problem and Interpret the Results

Solve the dual problem and determine the support vectors. Explain how the support vectors are used to construct the decision boundary in both the original and transformed feature spaces. Provide a geometric interpretation of the decision boundary in the context of the kernelized SVM.

solution

we can solve the dual problem using QP. if we call the solution to that $\alpha*$ we can write:

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i$$

the original decision boundary can be written as:

$$\mathbf{w}^{*^T} \cdot \mathbf{x} + b^* = 0$$

using the solution to QP we can find b^* by:

$$b^* = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{n} \alpha_j^* y_j \mathbf{x}_j^T \cdot \mathbf{x}_i \right)$$

so the decision boundary in the original feature space can be written as:

$$\sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i^T \cdot \mathbf{x} + \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{n} \alpha_j^* y_j \mathbf{x}_j^T \cdot \mathbf{x}_i \right) = 0$$

using same method we can write the decision boundary in the transformed feature space as:

$$\sum_{i=1}^{n} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{n} \alpha_j^* y_j K(\mathbf{x}_j, \mathbf{x}_i) \right) = 0$$

The support vectors are the data points that have non-zero Lagrange multipliers α_i^*. These support vectors lie on the margin or violate the margin constraint and play a crucial role in defining the decision boundary of the SVM.

In the original feature space, the support vectors are used to construct the hyperplane that separates the two classes. In the transformed feature space, the support vectors define the decision boundary in the higher-dimensional space, allowing the SVM to capture non-linear relationships between the data points.

Advanced Soft-Margin SVM with Regularization and Kernel Methods

Primal Formulation with Regularization and Slack Variables

Define the primal optimization problem for a soft-margin SVM, incorporating regularization and slack variables.

solution

The primal optimization problem for a soft-margin SVM with regularization and slack variables can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1,2,\ldots,n \\ & \xi_i \geq 0, \quad i = 1,2,\ldots,n \end{aligned}$$

where $\mathbf{w}$ is the weight vector, b is the bias term, ξ are the slack variables, C is the regularization parameter, n is the number of training samples, $\mathbf{x}_i$ is the i-th input vector, and y_i is the corresponding class label.

Derive the Dual Problem with Regularization

Derive the dual form of the optimization problem for the soft-margin SVM. Introduce Lagrange multipliers for both the margin constraints and the slack variables. Formulate the Lagrangian and show the detailed steps to obtain the dual problem by minimizing the Lagrangian with respect to the primal variables.

solution

To derive the dual form of the optimization problem for the soft-margin SVM, we introduce Lagrange multipliers α and β for the margin constraints and the slack variables, respectively. The Lagrangian is given by:

$$\mathscr{L}(\mathbf{w},b,\xi,\alpha,\beta) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{n}\beta_i\xi_i$$

where $\alpha = [\alpha_1,\alpha_2,\ldots,\alpha_n]$ and $\beta = [\beta_1,\beta_2,\ldots,\beta_n]$ are the Lagrange multipliers. To obtain the dual problem, we minimize the Lagrangian with respect to the primal variables $\mathbf{w}$, b, and ξ by setting the derivatives to zero:

$$\frac{\partial\mathscr{L}}{\partial\mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i\mathbf{x}_i = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i\mathbf{x}_i$$

$$\frac{\partial \mathscr{L}}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$
$$\frac{\partial \mathscr{L}}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \quad \Rightarrow \quad \alpha_i + \beta_i = C$$

so we can rewrite the Lagrangian as:

$$\mathscr{L}(\alpha, \beta) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

so the dual problem is defined as:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{subject to} \quad & 0 \le \alpha_i \le C, \quad i = 1, 2, \ldots, n \\ & \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned}$$

the kkt conditions are:

- primal feasibility: $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \ge 1 - \xi_i$
- dual feasibility: $\begin{cases} 0 \le \alpha_i \le C \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \\ \xi_i \ge 0 \end{cases}$
- stationary: $\begin{cases} \nabla_{\mathbf{w}} \mathscr{L} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \\ \frac{\partial}{\partial b} \mathscr{L} = -\sum_{i=1}^{N} \alpha_i y_i = 0 \\ \frac{\partial}{\partial \xi_i} L = C - \alpha_i - \beta_i = 0 \end{cases}$
- Complementary slackness: $\begin{cases} \alpha_i(1 - \xi_i - y_i(\mathbf{w}^\top \mathbf{x}_i + b)) = 0 \\ \xi_i \alpha_i = 0 \end{cases}$

Real-World Application and Parameter Selection

Discuss how a soft-margin SVM can be applied to a real-world classification problem, such as spam email detection. Describe the process of selecting the regularization parameter C and kernel parameters through cross-validation. Explain the impact of these parameters on the model's performance and generalization ability.

solution

A soft-margin SVM can be applied to spam email detection by using features extracted from email content, sender information, and metadata.

The model can learn to distinguish between spam and non-spam emails based on these features. To select the regularization parameter C and kernel parameters, we can use cross-validation to evaluate the model's performance on a validation set. By varying C and kernel parameters, such as the degree of a polynomial kernel or the bandwidth of a Gaussian kernel, we can find the values that optimize the model's performance.
The regularization parameter C controls the trade-off between maximizing the margin and minimizing the classification error. A larger C value penalizes misclassifications more heavily, leading to a more complex decision boundary that may overfit the training data. On the other hand, a smaller C value allows for more misclassifications, resulting in a simpler decision boundary that may underfit the data.
Kernel parameters, such as the degree of a polynomial kernel or the bandwidth of a Gaussian kernel, determine the complexity of the decision boundary in the feature space. Higher degrees or bandwidths can capture more complex patterns in the data but may lead to overfitting. Lower degrees or bandwidths result in simpler decision boundaries that may underfit the data.
By tuning these parameters through cross-validation, we can find the optimal values that balance model complexity and generalization ability, leading to better performance on unseen data.

Kernelized Soft-Margin SVM with Polynomial Kernel

Formulate the dual problem for a kernelized soft-margin SVM using the polynomial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$. Derive the necessary mathematical expressions and constraints.

solution

The dual problem for a kernelized soft-margin SVM using the polynomial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$ can be formulated as:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, 2, \ldots, n \\ & \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned}$$

where α are the Lagrange multipliers, C is the regularization parameter, n is the number of training samples, $\mathbf{x}_i$ is the i-th input vector, and y_i is the corresponding class label.

Model Evaluation and Interpretation

Explain how to evaluate the performance of a soft-margin SVM model on a test dataset. Discuss metrics such as accuracy, precision, recall, and F1-score. Provide an interpretation of the

model's decision boundary and the influence of support vectors in the context of the chosen kernel and regularization parameters.

solution

Accuracy: The ratio of correctly predicted instances to the total number of instances. it can be calculated as:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}}$$

Precision: The ratio of correctly predicted positive instances to the total predicted positive instances. it can be calculated as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall (Sensitivity): The ratio of correctly predicted positive instances to the total actual positive instances. it can be calculated as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

F1-Score: The harmonic mean of precision and recall. it can be calculated as:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Understanding the model's decision boundary and the role of support vectors requires analyzing how the chosen kernel and regularization parameters impact the margin and the model's generalization capability. Support vectors, which are the data points nearest to the decision boundary, possess non-zero α_i values. These values dictate the boundary's position and orientation.