

Pattern Recognition and Machine Learning: Homework 3

Qingru Hu 2020012996

March 14, 2023

Problem 1

The loss function for LDA (using Fisher's condition) is:

$$l_{\text{LDA}} = \frac{1}{2}(\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2 \quad (1)$$

The loss function for logistic regression is:

$$l_{\text{LG}}(x) = \begin{cases} -\log(\theta(x)) & y_i = 1 \\ -\log(1 - \theta(x)) & y_i = 0 \end{cases}$$

where $\theta(x) = \frac{e^{\mathbf{w}^\top \mathbf{x}_i + b}}{1 + e^{\mathbf{w}^\top \mathbf{x}_i + b}}$. The loss function of soft SVM is:

$$l_{\text{hinge}}(x) = \max(1 - y_i \cdot (\mathbf{w}^\top \mathbf{x}_i + b), 0) \quad (2)$$

For a concrete class $y = 1$, plot curves of the three loss functions in Fig.1.

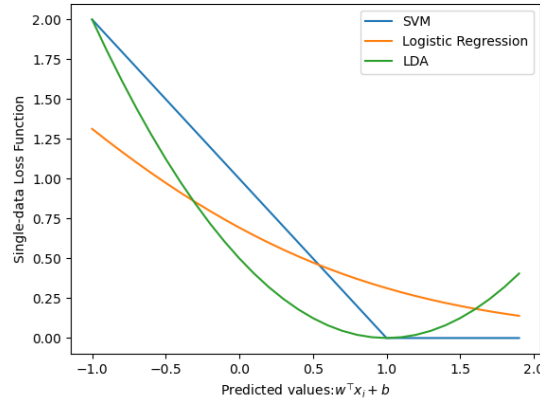


Figure 1: The single-data loss function for the three methods

In LDA, every sample in the dataset contributes to the loss function if it is wrongly classified, so LDA is somewhat prone to outliers.

In Logistic Regression, we punish those totally misclassified points severely, but don't pay much attention to those data that are most difficult to classify, so it is robust to outliers but may misclassify those ambiguous ones.

In SVM, only those points that are most difficult to discriminate (near the hyperplane) contribute to the total loss function, so it may be able to classify those samples near the hyperplane well and is prone to outliers.

Problem 2

The **hard-margin** problem is:

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i \cdot (\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad 0 \leq i \leq n. \end{aligned}$$

The Lagrangian function is:

$$\begin{aligned} L(\mathbf{w}, b, \alpha, \xi, \mu) &= \frac{1}{2} \|\mathbf{w}\|_2^2 + \sum_{i=1}^n \alpha_i [1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)] \\ \alpha_i &\geq 0, i = 1, \dots, n \end{aligned}$$

Take the partial derivatives of Lagrangian w.r.t \mathbf{w}, b and set to zero:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}} &= 0 \quad \mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \\ \frac{\partial L}{\partial b} &= 0 \quad \sum_{i=1}^n \alpha_i y_i = 0 \end{aligned}$$

Plug the above relations into the original Lagrangian function and we can get:

$$\begin{aligned} L(\mathbf{w}, b, \alpha, \xi, \mu) &= \frac{1}{2} \mathbf{w}^\top \mathbf{w} + \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \alpha_i y_i (\mathbf{w}^\top \mathbf{x}_i + b) \\ &= \frac{1}{2} \left(\sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \right)^\top \left(\sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \right) + \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \left(\alpha_i y_i \left(\sum_{j=1}^n \alpha_j y_j \mathbf{x}_j \right)^\top \mathbf{x}_i \right) \\ &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \end{aligned}$$

Therefore, the original optimal problem is equivalent to the dual problem:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y_i = 0, \alpha_i \geq 0, i = 1, \dots, n \end{aligned}$$

Denote the solution in the first problem as \mathbf{w}^* and solution in the second problem as α^* , and the solutions satisfy:

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

From the complementary slackness of KKT conditions, we have:

$$\alpha_i [1 - y_i (\mathbf{w}^\top x_i + b)] = 0$$

If $\alpha_i > 0$, then $y_i (\mathbf{w}^\top x_i + b) = 1$. Notice that $y_i = \pm 1$, so $\mathbf{w}^\top x_i + b = y_i$.

Problem 3

Search the paper and conduct a literature review for how to add different regularizations to SVM, and more sophisticated forms of kernel function.

Answer: Classical ¹ regularization theory formulates the regression and classification problem as a variational problem of finding the function f that minimizes the functional:

$$\min_{f \in H} H[f] = \frac{1}{l} \sum_{i=1}^l V(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_K^2 \quad (3)$$

where $\|f\|_K^2$ is a norm in a Reproducing Kernel Hilbert Space H defined by the positive definite function K , l is the number of data points, λ is the regularization parameter and $V(\cdot)$ is the loss function. Under rather general conditions the solution of the above equation is:

$$f(\mathbf{x}) = \sum_{i=1}^l c_i K(\mathbf{x}, \mathbf{x}_i) + b \quad (4)$$

For a classical (L_2) Regularization Networks (RN), the loss function is:

$$V(y_i, f(\mathbf{x}_i)) = (y_i - f(\mathbf{x}_i))^2 \quad (5)$$

For Support Vector Machine Regression (SVMR), the loss function is:

$$V(y_i, f(\mathbf{x}_i)) = |y_i - f(\mathbf{x}_i)|_\epsilon \quad (6)$$

For Support Vector Machine Classification (SVMC), the loss function is:

$$V(y_i, f(\mathbf{x}_i)) = |1 - y_i f(\mathbf{x}_i)|_+ \quad (7)$$

where $|\cdot|_\epsilon$ is Vapnik's epsilon-insensitive norm, $|x|_+ = x$ if x is positive and zero otherwise, and y_i is a real number in RN and SVMR, whereas it takes values -1, 1 in SVMC.

The different regularizations lie in the different norms $\|f\|_K^2$ in a Reproducing Kernel Hilbert Space H .

It can be proved that every Reproducing Kernel Hilbert Space H there corresponds to a unique positive definite function $K(x, y)$, called the reproducing kernel of H that has the following property:

$$f(\mathbf{x}) = \langle f(\mathbf{y}), K(\mathbf{x}, \mathbf{y}) \rangle_H \quad \forall f \in H \quad (8)$$

¹My answer refers to this comprehensive review paper: Evgeniou, T., Pontil, M. & Poggio, T. Regularization Networks and Support Vector Machines. Advances in Computational Mathematics 13, 1-50 (2000). <https://doi.org/10.1023/A:1018946025316>

the norm is defined by:

$$\|f\|_K^2 = fKf^\top \quad (9)$$

Some common and sophisticated forms of kernel function are listed in Tab.. The first four are radial kernels. The multiquadric and thin plate splines are positive semidefinite and thus require an extension of the simple RKHS. The last two kernels are one-dimensional: multidimensional kernels can be built by tensor products of one-dimensional ones. The functions B_n are piecewise polynomials of degree n .

Kernel function	Regularization Network
$K(\mathbf{x} - \mathbf{y}) = \exp(-\ \mathbf{x} - \mathbf{y}\ ^2)$	Gaussian RBF
$K(\mathbf{x} - \mathbf{y}) = (\ \mathbf{x} - \mathbf{y}\ ^2 + c^2)^{-1/2}$	Inverse multiquadric
$K(\mathbf{x} - \mathbf{y}) = (\ \mathbf{x} - \mathbf{y}\ ^2 + c^2)^{1/2}$	Multiquadric
$K(\mathbf{x} - \mathbf{y}) = \ \mathbf{x} - \mathbf{y}\ ^{2n+1}$	Thin plate splines
$K(\mathbf{x} - \mathbf{y}) = (1 + \mathbf{x} \cdot \mathbf{y})^d$	Polynomial of degree d
$K(x, y) = B_{2n+1}(x - y)$	B-splines
$K(x, y) = \frac{\sin((d+1/2)(x-y))}{\sin((x-y)/2)}$	Trigonometric polynomial of degree d

Problem 4

(a) Use the SVC classifier in sklearn to classify the train set and predict on the test set

Use Linear SVC to classify the samples on the training set, and the accuracy on the test set is 97%. Besides, I also try to write my own SVM by solving the dual problem using the quadratic program solver, but I didn't get proper results and I haven't found where my problems are. My own try to build a SVM is in the 'My own SVM' section of the jupyter-notebook 'face_svm.ipynb'. Any help will be greatly appreciated if you help me find my problems.

(b) Different kernel functions and penalty strength C

Kernel	Penalty Strength	Accuracy(%)
Linear	0.1	96
Poly	0.1	79
RBF	0.1	84
Sigmoid	0.1	86
Linear	1	96
Poly	1	83
RBF	1	97
Sigmoid	1	81
Linear	10	96
Poly	10	90
RBF	10	97
Sigmoid	10	76

Table 1: Different kernel functions and penalty strength for SVM

The best composition is the RBF kernel with a penalty strength of 1 or 10, which both have an accuracy of 97%.

(c) The supporting vectors

Use the `sklearn.SVC` 'support_' attribute to find the indexes for the support vectors in the training set and plot them in the second 'Sklearn SVM' section of the jupyter-notebook 'face_svm.ipynb'. These images are selected as support vectors maybe because they are difficult to classify. Some of them are not images of the front of the face and some of them may have low contrast.