Pattern Recognition and Machine Learning: Homework 3, Zhengzuo Liu

Problem 1

Answer:

LDA:

$$l(\hat{y}, y_i) = \frac{1}{2}(\hat{y} - y_i)^2$$

$$y_i \in \{\frac{N}{N_1}, -\frac{N}{N_2}\}$$

Logistic Regression:

$$l(\hat{y}, y_i) = \frac{1}{2}(\sigma(\hat{y}) - y_i)^2 = \frac{1}{2}(\frac{1}{1+e^{-\hat{y}}} - y_i)^2$$

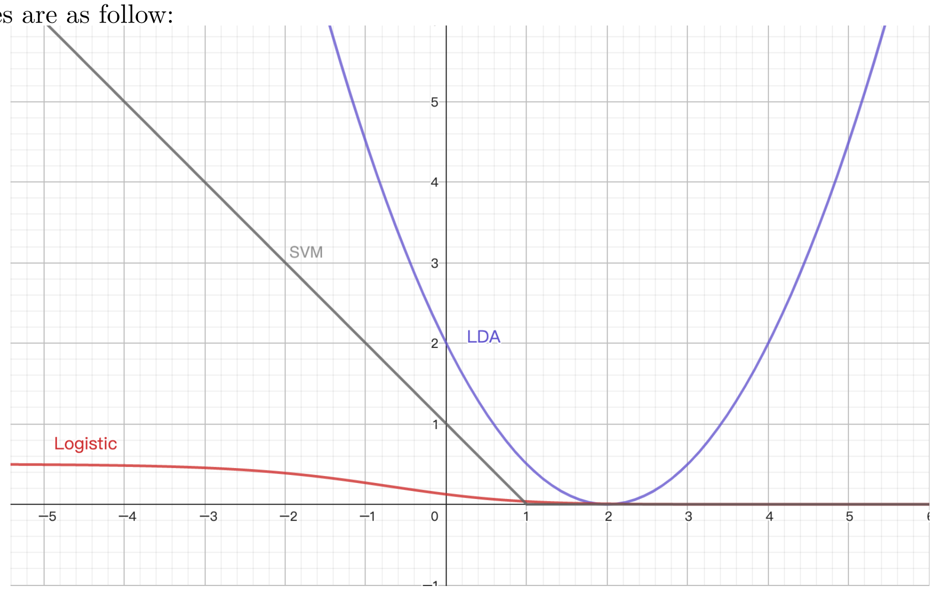
$$y_i \in \{0, 1\}$$

SVM:

$$l(\hat{y}, y_i) = max(0, 1 - y_i\hat{y})$$

$$y_i \in \{-1, 1\}$$

The curves are as follow:



It is seen that the sensitivity to false classification of each model is ranked as: LDA > SVM > Logistic Regression.

Problem 2

Answer:

First, convert the original function to an unconstrained Lagrangian function

$$L(\boldsymbol{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{N} \alpha_i \left(y_i \left(\boldsymbol{w}\cdot\boldsymbol{x_i} + b\right) - 1\right)$$

As the problem is a convex optimization and satisfies KKT condition, the original problem is equal to

$$\min_{\boldsymbol{w},b}\max_{\alpha_i\geq 0} L(\boldsymbol{w},b,\boldsymbol{\alpha}) = \max_{\alpha_i\geq 0}\min_{\boldsymbol{w},b} L(\boldsymbol{w},b,\boldsymbol{\alpha})$$

Let the partial derivatives of $L(\boldsymbol{w},b,\boldsymbol{\alpha})$ to $\boldsymbol{w}$ and b equal to 0, get

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x_i}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

Substitute in, get

$$L(\boldsymbol{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \left(\boldsymbol{x_i}\cdot\boldsymbol{x_j}\right) - \sum_{i=1}^{N}\alpha_i y_i \left(\left(\sum_{j=1}^{N}\alpha_j y_j \boldsymbol{x_j}\right)\cdot\boldsymbol{x_i} + b\right) + \sum_{i=1}^{N}\alpha_i$$

i.e.

$$\min_{\boldsymbol{w},b} L(\boldsymbol{w},b,\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \left(\boldsymbol{x_i}\cdot\boldsymbol{x_j}\right) + \sum_{i=1}^{N}\alpha_i$$

Then the original problem is equivalent to

$$\max_{\boldsymbol{\alpha}} \; -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \left(\boldsymbol{x_i}\cdot\boldsymbol{x_j}\right) + \sum_{i=1}^{N}\alpha_i$$

$$s.t. \quad \sum_{i=1}^{N}\alpha_i y_i = 0$$

$$\alpha_i \geq 0,\; i = 1,2,...,N$$

In hard margin SVM, $\alpha_i > 0$ is equivalent to that x_i is a supporting vector (only supporting vectors affect the objective function and they do), which is equivalent to the following equation:

$$\boldsymbol{w}^{*T}\boldsymbol{x}_i + b = y_i$$

Problem 3

Answer:

In this literature review, we will explore different regularization techniques and sophisticated forms of kernel functions that can be used to improve the performance of SVMs.

Regularization Techniques:

1. L1 Regularization: L1 regularization is also known as Lasso regularization. Lasso Regression (Least Absolute Shrinkage and Selection Operator) adds "absolute value of magnitude" of coefficient as penalty term to the loss function. The L1 regularization technique can be used to select the most relevant features in the dataset and reduce overfitting.

2. L2 Regularization: L2 regularization is also known as Ridge regularization. Ridge regression adds "squared magnitude" of coefficient as penalty term to the loss function. L2 regularization can be used to prevent overfitting and improve the generalization performance of SVMs.

Elastic Net Regularization: Elastic Net regularization is a combination of L1 and L2 regularization. It can be used to overcome the limitations of both L1 and L2 regularization techniques and improve the performance of SVMs on high-dimensional datasets.

Sophisticated Forms of Kernel Functions:

1. Gaussian Kernel: The Gaussian kernel is a popular kernel function that can be used to handle non-linearly separable data. It is a radial basis function that is based on the distance between the input data points. The Gaussian kernel is widely used in applications such as image recognition, text classification, and bioinformatics.

2. Laplacian Kernel: The Laplacian kernel is a kernel function that is based on the Laplace distribution. It can be used to handle non-linearly separable data and is particularly effective for image recognition tasks.

3. Polynomial Kernel: The Polynomial kernel is a kernel function that is based on the polynomial function. It can be used to handle non-linearly separable data and is particularly effective for tasks such as speech recognition and natural language processing.

4. Sigmoid Kernel: The Sigmoid kernel is a kernel function that is based on the sigmoid function. It can be used to handle non-linearly separable data and is particularly effective for tasks such as face recognition and pattern recognition.

Problem 4

Answer:

Kernel functions: linear, poly, rbf and sigmoid. C values: 0.1, 1, 10 and 100.

The result is as follows:

C=	linear	poly	rbf	sigmoid
0.1	0.98	0.98	0.88	0.47
1	0.98	0.98	0.97	0.47
10	0.98	0.98	0.98	0.4
100	0.98	0.98	0.98	0.29

Take kernel = linear, C = 1 as an example. The supporting vectors are:

[250 252 253 255 256 257 260 269 270 280 281 287 288 292 295 303 309 312 314 328 329 330 331 334 335 336 337 344 346 348 352 356 361 364 370 376 383 384 386 391 399 402 407 412 414 420 425 428 429 434 437 442 444 447 449 451 452 454 456 464 465 475 477 478 485 486 487 494 497 1 6 8 11 12 15 16 17 20 22 41 43 59 61 63 64 67 70 72 73 79 80 82 87 91 98 101 106 109 110 116 122 125 128 142 143 145 147 148 152 158 159 160 161 163 166 169 171 173 175 178 181 183 184 195 196 201 202 214 218 219 225 232 235 239 248]