Weekly report of lessons

Name: Ravi Pratap Singh
Roll No: 20CS60R60
The week: 04 Nov, 05 Nov ,06 Nov
The topics covered: Linear Discrimination,Discriminant functions,Two-Category Case,Perceptron Criterion,Gradient Descent algorithm,Stringent criteria for linear separability,Batch relaxation algo with margin,Suport Vector Machine, Optimization problem, SVM testing, Soft margin, kernel function, Logistic discrimination
Summary topic wise:

- Linear discriminant analysis is used as a tool for classification, dimension reduction, and data visualization
- A discriminant function that is a linear combination of the components of x can be written as $g_i(x) = w^T x + w_{i0}$ .A quadratic function can be written as $g(x) = x^T W_i x + w_i^T x + w_{i0}$. Generalized Linear Discriminant Functions g(x) can be written as $g_i(x) = w_{i0} + \sum_{i=1}^{d} w_i x_i$.
- A two-category classifier implements the following decision rule: Decide w_1 if $g(\mathbf{x})>0$ and w_2 if $g(\mathbf{x})<0$. Thus, **x** is assigned to w_1 if the inner product $\mathbf{w}^T\mathbf{x}$ exceeds the threshold $-w_0$ and to w_2 otherwise.
- The *Perceptron criterion function* is given by $J_p(a) = \sum_{y \in Y} (-a^T y)$ .where y(a) is the set of samples misclassified by **a**. Because $a^T y \le 0$ if **y** is misclassified . Geometrically, $J_p(\mathbf{a})$ is proportional to the sum of the distances from the misclassified samples to the decision boundary.
- In gradient descent we start with some arbitrarily chosen weight vector **a**(1) and compute the gradient vector *J*(**a**(1)). The next value **a**(2) is obtained by moving some distance from **a**(1) in the direction of steepest descent, i.e., along the negative of the gradient. $W(i) = W^{(i-1)} - \eta(i)\ \nabla J(W)$
- LinearSVM is a linearly scalable routine meaning that it creates an SVM model in a CPU time which scales linearly with the size of the training data set.
- Batch relaxation with margin performs update on W by considering samples one by one in every iteration.
- SVM uses vapnik's principle of never solving more complex problem as first step before actual problem.SVM is sufficient to compute class boundaries, and also compute boundaries separating those x having low P(x)
- There is a general method for solving optimization problems with constraints (the method of Lagrange multipliers). Define the Lagrangian function: $L_p = \frac{1}{2}\|w\|^2 - \sum_{t=1}^{N} \alpha^t [r^t(w^T x^t + w_0) - 1]$ .Taking the derivative with respect to w , $w = \sum_{i=1}^{n} t^{(i)} \alpha_i x^{(i)}$.
  Applying quadratic optimization technique give $O(n^3)$ time and $O(n^2)$ space. Most of α^t will be zero.
- Check only signs of discriminant function and only support vectors deceive class boundaries.
- Soft margin allows SVM to make a certain number of mistakes and keep margin as wide as possible so that other points can still be classified correctly. we would aim to minimize the following objective:
  $L = \frac{1}{2}\|w\| + C\sum_i \xi_i + \sum_i \lambda_i (y_i(w \cdot x_i + b) - 1 + \xi_i)$. Where ξ_i is linear penalty. The optimal values can be calculated by differentiating the equation with respect w and b and equating it to 0 .
  We can use any Kernel function(eg. quadratic) in place of dot product that has the capability of measuring similarity in higher dimensions ,without increasing the computational costs much.
- The function of the kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions. These functions can be different types. For example linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.
- Logistic discrimination of two classes assume log likelihood ratio is linear and uses weights of logit function .The logit function is the natural log of the odds that Y equals one of the categories. It regresses posterior directly from labelled data.

Any novel idea of yours out of the lessons : With Linear SVC model we cannot get the probability estimates which we get from the ROC curve.But we can make it work with a technique called **probability calibration**. `sklearn` package has a module `sklearn.calibration` with `CalibratedClassifierCV` class which realizes two methods of calibrations: `'isotonic'` or `'sigmoid'` .
Difficulty level of the Quiz : Fair
Was the time given to you for solving the quiz appropriate? : Yes , Enough time was given .
Did the quiz questions enhance your understanding of the topics covered : Yes