Machine Learning : E0270 2014
Assignment 2
Due Date : March 26th

1 ν-SVC with libsvm

Download the libsvm library `http://www.csie.ntu.edu.tw/ cjlin/libsvm/`.

1. Run the ν-SVM classification algorithm for $\nu = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$ using the `Spambase.libsvm` dataset (libsvm version of dataset from the first assignment) provided. Plot the misclassification error (ratio of incorrectly classified isntances/total instances) on the test data after learning the model on the training data for different values of ν on x-axis. On the same graph plot the ratio of support vectors to the total number of train instances for each value of ν. What are your observations.

2. Fix $\nu = 0.1$. Randomly sample a $\{20, 40, 60, 80, 100\}$ percent subset of your training data. For each subset, run the ν-SVM algorithm and get the prediction accuracy (as a ratio between 0 and 1) on entire train set and the test set independantly. Plot both these values as a function of the percentage of training data used on the same graph. What are your observations.

2 Logistic Regression

Generative models for binary classification attempt to model the prior probability of a class $P(Y = 1)$ and the class conditional densities $P(X|Y)$, where $Y \in \{0, 1\}$ is the a random variable indicating the class label and X is an observed data instance (Recall the Naiave Bayes generative classification framework) . However, it is a well known fact that for certain types of class conditionals (belonging to the *exponential family*), the posterior can be written in the form of a logistic sigmoid of the form $P(Y = 1|X) = \frac{1}{1+e^{-\mathbf{w}^T X}}$, for an appropriately chosen $\mathbf{w}$. The IRLS algorithm for logistic regression attempts to learn these weight vectors.[Textbook Reference : Section 4.3.3 of Bishop].

1. Compute the posterior $P(Y = 1|X)$ with prior $P(Y = 1) = Bernoulli(p)$, $p \in \mathbb{R}^+$ and multinomial class conditionals $P(X|Y = k) = \Pi_{j=1}^{M} (\theta_{k,j})^{X_j}$ for $k \in \{0, 1\}$, where $X \in \{0, 1\}^M$ following the 1 in M representation with $\sum_{j=1}^{M} X_j = 1$ and θ_k is the parameter of multinomial distribution for class k. Show that this posterior can be written as a logistic sigmoid.

2. Consider the multiclass classification problem with K classes for real valued vectors $X \in \mathbb{R}^M$. For each class, $k = 1, \ldots, K$, assume we have a class conditional probability of $p(X|Y = k) = \mathcal{N}(\mu_k, \Sigma)$ and let the prior be a multinomial with $P(Y = k) = \pi_k$. Compute the posterior and show that it takes the form of a **softmax** function $P(Y = k|X) = \frac{e^{X^T \mathbf{w_k}}}{\sum_{l=1}^{K} e^{X^T \mathbf{w_l}}}$

3. Implement the IRLS algorithm for Logistic Regression for Binary Classification for the `Spambase` dataset from assignment 1. Compare your result with the best classification accuracy obtained with ν-SVC from the previous problem. [Textbook Reference : Section 4.3.3 of Bishop]. Start the algorithm by initializing w with the zero vector and run the algorithm for 500 iterations.

3 Linear and Ridge Regression

You are given a dataset $(\mathbf{X}, \mathbf{y}), \mathbf{X} \in \mathbb{R}^{N \times M}$(with N instances, each of dimension M) , $\mathbf{y} \in \mathbb{R}^m$(m-dimensional vector, with values corresponding the instances in$\mathbf{X}$).

1. Write a piece of MATLAB code `linear_least_squares_learner.m` implementing the linear least squares regression algorithm. The program takes as input $(\mathbf{X}, \mathbf{y})$ and produces as output a weight vector $\mathbf{w} \in \mathcal{R}^{\mathbf{M}}$ that is saved to a file representing the model. Write a piece of MATLAB code `linear_predictor.m` that reads the weight vector $\mathbf{w}$ (model) and predicts the corresponding value $\hat{y}$ for a new test instance $\mathbf{x}$. Run your code on the data set `synR1.mat` using the entire training data. Obtain the predicted y_n, by fitting the model to $\{\mathbf{x_n}\}_{n=1}^{N}$ and report the squared loss obtained (using the provided code `squared_error.m`).

2. The linear regression implemented in the above model can be looked at as an MLE estimate where we model $y_n = \mathbf{w^T x_n} + \epsilon_\mathbf{n}, \epsilon_\mathbf{n} \sim N(0, \sigma^2)$ for some σ. Report the log likelihood obtained based on your predictions on the test data from the previous experiment. Compute the MLE estimate of σ and report this value.

3. **Ridge Regression:** Recall ridge regression adds a regulizer term $\frac{\lambda}{2}||\mathbf{w}||_2^2$ to the least squares regression implemented in the previous question. Write a piece of MATLAB code `linear_ridge_learner.m` that takes $(\mathbf{X}, \mathbf{y})$ and an additional parameter λ to implement Ridge regression on the same dataset `synR1.mat`. Experiment with parameter $0.01, 0.1, 1, 10, 100$ and find the best value by 5 fold cross validation (the folds are readily available in `synR1_fold.mat`. Report the squared error with the best value of λ.

4 Kernel Ridge Regression

Kernel Ridge Regression extends ridge regression to be able to fit a wider class of functions of arbitrarily high degree using the kernel trick.

1. How would you extend the ridge regression to the kernel ridge regression using the transformation $\Phi : \mathbb{R}^M \to S$, where S is a vector space

equipped with a scalar product? (Hint : Argue that $(\lambda \mathbf{I} + \mathbf{X^T X})^{-1}\mathbf{X^T} = \mathbf{X^T}(\lambda \mathbf{I} + \mathbf{X X^T})^{-1}$ and proceed)

2. Implement the kernel ridge regression by writing a piece of MATLAB code `kernel_ridge_learner.m` by adapting the previous piece of code for ridge regression. Run your program with linear, polynomial degree-2, polynomial degree-3, RBF width-1, RBF width-4 kernels experimenting with values $0.01, 0.1, 1, 10, 100$ for λ using 5 fold cross validation using folds in `synR1_fold.mat` (report the average error with each fold). Report the error on the test data in `synR1.mat` with the best λ and find the value of λ that gives the best error ?

5 VC- Dimension

Recall that VC dimension is defined for a set $\mathcal{X}$ and a set of functions $\mathcal{F}$ from $\mathcal{X}$ to $\{+1, -1\}$.

1. **Axis parallel rectangles**

   Let $\mathcal{X} = \mathbb{R}^d$, $\mathcal{F} = \{f_R : R \in$ set of axis parallel rectangles in $\mathbb{R}^d\}$, where $f_R(x) = 1$ if $x \in R$ and -1 otherwise.

   What is the VC dimension of this pair of $\mathcal{X}, \mathcal{F}$?

2. **Halfspaces**

   Let $\mathcal{X} = \mathbb{R}^d$, $\mathcal{F} = \{f_w : x \mapsto \text{sign}(w^\top x + b), w \in \mathbb{R}^d, b \in \mathbb{R}\}$,
   Via Radon's theorem one can show that the VC dimension of this pair of $(\mathcal{X}, \mathcal{F})$ is upper bounded by $d+1$. Show a matching lower bound, i.e. give $d+1$ points that are shattered by this function class

3. **Convex sets**

   Let $\mathcal{X} = \mathbb{R}^2$, $\mathcal{F} = \{f_C : C \in$ set of all convex polygons in $\mathbb{R}^2\}$, where $f_C(x) = 1$ if $x \in C$ and -1 otherwise.

   Is the VC dimension of this pair of $\mathcal{X}, \mathcal{F}$, finite? Is it learnable?

6 Continuous Risk Minimizers

You have now seen various continuous risk minimization procedures used for binary classification $\mathcal{Y} = \{+1, -1\}$. These include logistic regression, least squares regression and SVM.

Assume that the instance space $\mathcal{X}$ is a singleton, say $\{0\}$. Hence the entire distribution is characterised by a single real value of $P(Y = 1|X = x) = \eta$. Here the learning procedure for all the above mentioned algorithm reduces to finding the bias b, that minimizes a corresponding loss term.

Assming you knew η, derive the expectation minimizer (minimizer of $E_Y \ell(b, Y)$ over b) for the following four losses corresponding to least squares, logistic regression, SVM and a variant of SVM, in terms of η.

1. $\ell(b, Y) = (1 - bY)^2$

2. $\ell(b, Y) = \log(1 + \exp(-Yb))$

3. $\ell(b, Y) = \max(0, 1 - bY)$
4. $\ell(b, Y) = (\max(0, 1 - bY))^2$

Notice that all three losses depend only on bY which can be interpreted as the margin. Notice also that for the last three losses if bY is large enough $\ell(b, Y) = 0$ (or close to zero). This can be interpreted as instances that are classified correctly with large margin incur zero loss.

`Note : Datasets for the assignment are available at http://drona.csa.iisc.ernet.in/ e0270/Jan-2014/Assignments/Assignment2-Files/Files_data_codes.zip`