Machine Learning 6.867 - Pset 2

October 27, 2015

1 Logistic Regression

1.1 Implementation

We implemented L_2-regularized logistic regression using gradient descent. The objective function to be minimized over is:

$$\sum_{i=1}^{n} \log(1 + e^{-y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)}+w_0)}) + \lambda \mathbf{w}^T\mathbf{w} \tag{1}$$

We used both our implementation of gradient descent and the `MATLAB` function `fminunc`. Convergence criterion is reached within reasonable iterations in both implementations.

1.2 Testing in data with $\lambda = 0$

We run the logistic regression on the four data sets, setting the regularization parameter $\lambda = 0$. The estimated coefficients are listed in Table 1.

Table 1: Estimated logistic regression coefficients and accuracy, $\lambda = 0$

Data	w_0	w_1	w_2	Train accuracy	Valid accuracy	Test accuracy
stdev1	-66.3378	95.2461	101.1527	1.0000	1.0000	0.9975
stdev2	-0.0466	0.7636	1.1148	0.9075	0.9200	0.9250
stdev4	-0.0093	0.2363	0.2034	0.7400	0.7525	0.7825
nonsep	0.0006	-0.0247	-0.0237	0.5150	0.4925	0.4975

The decision boundaries at various thresholds are plotted in Figure 3. We observe the following phenomenon:

1. As data become more linearly non-separable, the accuracy is lower, and the decision boundary has higher variance with respect to data (for example, as σ increases, the decision boundary and accuracy are more different between training and validation.)

2. As data become more linearly non-separable, the estimated logistic function is also less steep, reflected in the wider bands in the plots and lower norm of $\mathbf{w}$. This is reasonable because the classifier is not as certain about how to classify the data points in the mix zone.

3. In the totally non-separable case, logistic regression fails to classify, barely reaching the 50% baseline.

1.3 Testing in data with positive λ

Similarly, we run logistic regression with other values of λ. With higher values of λ, the decision boundary is flatter, the training accuracy lower, and the validation accuracy typically increases (or stays the same) and

then decreases, as shown in Figure 2. We use the cross-validation technique to select best value of λ using the validation set accuracy. In particular, for the four datasets, we choose $\lambda = 0$, $\lambda = 0$, $\lambda = 100$, and $\lambda = 1000$ respectively. The corresponding accuracy in test sets are 0.9975, 0.9250, 0.7850, and 0.5025. Compared to not regularizing, the test set accuracies are slightly higher. The accuracy for the non-separable data set remains relatively constant regardless of how we regularize.

2 Support Vector Machine

Support Vector Machines are a popular classification method to construct linear or nonlinear decision boundaries by solving a convex optimization problem. There are two common forms of the optimization problem considered for SVM, which we refer to as the primal and dual. In this paper, we only consider the dual form, because it is computationally more tractable for many problems, and this method has the ability to generalize to different choices of kernel. The dual form of SVM for a general kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is as follows:

$$\begin{aligned} \max_{\alpha \in \mathbb{R}^n} \quad & \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y^{(i)} y^{(j)} k(x^{(i)}, x^{(j)}) \\ \text{s.t.} \quad & 0 \le \alpha_i \le C, \qquad\qquad i = 1, \ldots, n, \\ & \sum_{i=1}^{n} \alpha_i y^{(i)} = 0. \end{aligned} \tag{2}$$

2.1 Implementation

First, we implemented the dual form of the SVM with a linear kernel, where k is the usual dot product $k(x, z) = \langle x, z \rangle$ for all $x, z \in \mathcal{X}$. In MATLAB, we created a function with inputs: data $X \in \mathbb{R}^{n \times p}$, labels $Y \in \{-1, 1\}$, and cost parameter $C \in \mathbb{R}^+$. Within the function, we use the quadratic solver `quadprog` to solve the SVM dual problem (2) with these parameters to find the optimal α's. Since `quadprog` requires that the problem fit into a certain functional form, we reformulate the problem (2) as follows:

$$\begin{aligned} -\min_{\alpha \in \mathbb{R}^n} \quad & \tfrac{1}{2} \alpha^T H \alpha - \sum_{i=1}^{n} \alpha_i \\ \text{s.t.} \quad & 0 \le \alpha_i \le C, \qquad i = 1, \ldots, n, \\ & \sum_{i=1}^{n} \alpha_i y^{(i)} = 0, \end{aligned} \tag{3}$$

where: $H \in \mathbb{R}^{n \times n}$ is a matrix with $(i, j)^{th}$ entry $H_{ij} = y^{(i)} y^{(j)} k(x^{(i)}, x^{(j)})$. Given the optimal solution $\alpha \in \mathbb{R}^n$ for the SVM problem with a linear kernel, the chosen linear decision boundary $\theta^T x + \theta_0 = 0$ is given by:

$$\theta = \sum_{i=1}^{n} \alpha_i y^{(i)} x^{(i)} \tag{4}$$

$$\theta_0 = \frac{1}{\mathcal{M}} \Big(\sum_{j \in \mathcal{M}} \Big(y^{(j)} - \sum_{i \in \mathcal{S}} \alpha_i y^{(i)} (x^{(j)})^T x^{(i)} \Big) \Big) \tag{5}$$

where $\mathcal{M} = \{i : 0 < \alpha_i < C\}$ and $\mathcal{S} = \{i : \alpha_i > 0\}$. The output of our linear SVM function is $[\theta, \theta_0]$. We tested our function on the 2D example $X = \{(1, 2), (2, 2), (0, 0), (-2, 3)\}$, $Y = \{1, 1, -1, -1\}$. For this problem, the objective function generated for minimization problem (3) is:

$$\frac{1}{2} \alpha^T H \alpha - \sum_{i=1}^{4} \alpha_i, \tag{6}$$

where:

$$H = \begin{bmatrix} 5 & 6 & 0 & -4 \\ 6 & 8 & 0 & -2 \\ 0 & 0 & 0 & 0 \\ -4 & -2 & 0 & 13 \end{bmatrix}$$

The constraints are:

$$0 \leq \alpha_i \leq C, \quad i = 1, \ldots, 4, \tag{7}$$

$$\alpha_1 + \alpha_2 - \alpha_3 - \alpha_4 = 0. \tag{8}$$

2.2 Performance on datasets

We tested our linear SVM function on the same 2D datasets from the previous section, with parameter $C = 1$. The estimated coefficients are listed in Table 2.

Table 2: Estimated SVM coefficients and accuracy, $C = 1$

Data	w_0	w_1	w_2	Train accuracy	Valid accuracy	Test accuracy
stdev1	1.2333	1.2409	-0.4204	1.0000	1.0000	0.9975
stdev2	0.4573	0.7552	-0.1410	0.9050	0.9200	0.9175
stdev4	0.1998	0.1820	-0.0470	0.7450	0.7650	0.7800
nonsep	-0.2194	-0.2118	-0.4280	0.6975	0.6950	0.7000

The decision boundaries at various thresholds are plotted in Figure TODO **??**. We observe the following phenomenon:

1. As data become more linearly non-separable, the accuracy is lower, and the decision boundary has higher variance with respect to data (for example, as σ increases, the decision boundary and accuracy are more different between training and validation.)

2. As data become more linearly non-separable, the estimated logistic function is also less steep, reflected in the wider bands in the plots and lower norm of $\mathbf{w}$. This is reasonable because the classifier is not as certain about how to classify the data points in the mix zone.

3. In the totally non-separable case, SVM achieves close to 70% accuracy, which is higher than logistic regression. These two methods yield different results because they use different loss functions. SVM uses a hinge loss function which is piecewise linear, while logistic regression uses a logistic loss function which is nonlinear convex.

2.3 Kernel SVM

We extended our SVM implementation in MATLAB to operate with more general kernels, taking the kernel function or kernel matrix as input.

Questions:

(a) As C increases, the geometric margin $1/\|\mathbf{w}\|$ decreases. If the data is not linearly separable, then the geometric margin $1/\|\mathbf{w}\|$ decreases strictly monotonically as C increases. However, if the data is linearly separable, then this does not always happen as we increase C. In this case, once the margin is sufficiently small such that all points are correctly classified, then it will not decrease further even as C approaches infinity.

(b) As C decreases, the number of support vectors decreases. This is because the larger penalty on misclassified points leads to a decision boundary with fewer misclassifications on the training data. However, the number of the support vectors is bounded below by two as C approaches infinity, because there will always be at least one support vector on each side of the decision boundary.

(c) Maximizing the geometric margin $1/\|\mathbf{w}\|$ on the training data is not an appropriate criterion for selecting C because this leads to a classifier which is overfit to the training set. To obtain a classifier which generalizes well on test data, we should use out-of-sample data to select an appropriate value for C. To do this, we can train the SVM model with $C = \{0.01, 0.1, 1, 10, 100\}$, and then select the value for C which yields the classifier with the highest accuracy on the validation set.

3 Titanic Data

3.1 Logistic Regression

We use Logistic Regression classifier on the Titanic data to make predictions on survivor results. Before running the regression, we scale the features in two ways: 1) standardizing using mean and standard deviation by $(X_j^{(i)} - \mu(X_j))/\sigma(X_j)$; 2) scale so each dimension is within the $[0, 1]$ range using min and max: $(X_j^{(i)} - \min X_j)/(\max X_j - \min X_j)$. We find the scaling constants in the training sets only, and use the same constants in the validation and testing sets. Note comparing the two scaling methods is simply for exploration purposes and we are not choosing one or the other in the model selection process.

With no regularization, we obtain a testing set accuracy of 77.78% in both cases of scaling methods. To find the best λ, we use the cross validation technique. The validation set accuracy with respect to λ is presented in Figure 4. We therefore choose $\lambda = 10$ in standard-deviation-based scaling, and $\lambda = 0.1$ in range-based scaling. on the validation accuracy. The test set accuracy is then 75.13% and 76.19%, respectively. Unfortunately neither is not as good as the non-regularized logistic regression. Furthermore, the way how one scales the features can also impact the accuracy.

3.2 SVM

TODO

3.3 Comparison

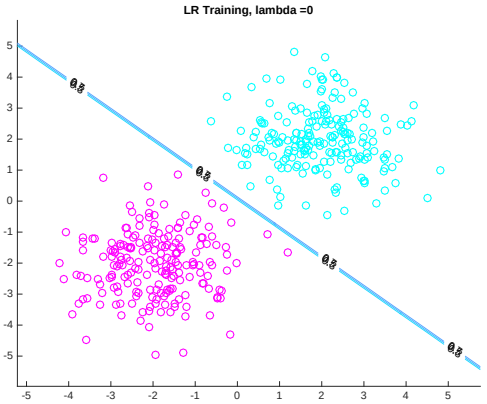
TODO compare LR and SVM

The estimated coefficients for logistic regression and SVM are presented in Table 3. Since we do not have standard error and confidence interval information, we can only rely on the magnitude of the estimator (even if they are not statistically significant).

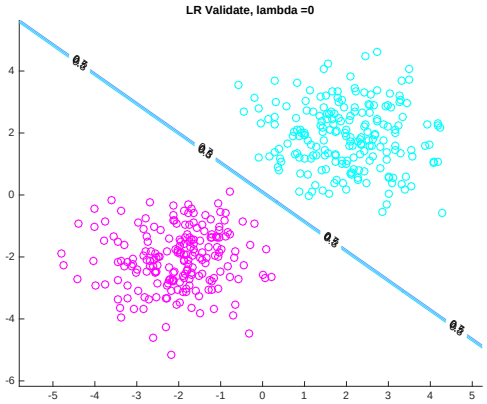
We observe that being woman, higher class and higher fare are associated with higher likelihood of survival, whereas being 3rd class is strongly associated with death.

Table 3: Estimated logistic regression coefficients in Titanic data, $\lambda = 10$

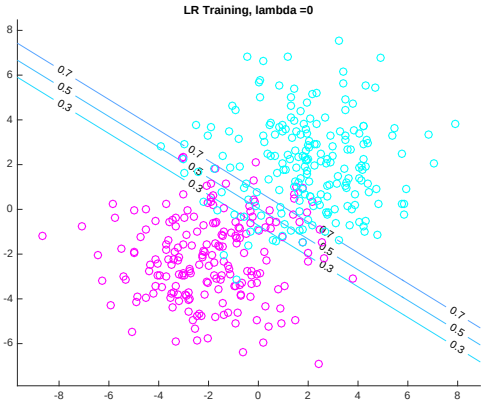
Coefficient	Description	Logistic estimator
w_0	Constant	-0.6289
w_1	Passenger class 1	0.1680
w_2	Passenger class 2	0.2013
w_3	Passenger class 3	-0.3245
w_4	Sex	0.7795
w_5	Age	-0.1417
w_6	Num siblings/Spouses aboard	-0.0477
w_7	Num parents/children aboard	0.1422
w_8	Pasenger fare	0.1533
w_9	Port of embarkation = Southampton	-0.0955
w_{10}	Port of embarkation = Cherbourg	0.1090
w_{11}	Port of embarkation = Queenstown	0.0393



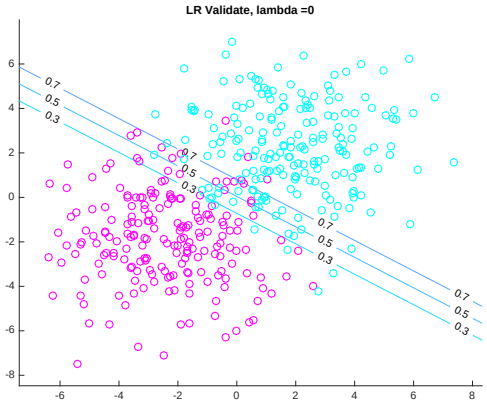
(a) Data with $\sigma = 1$, training



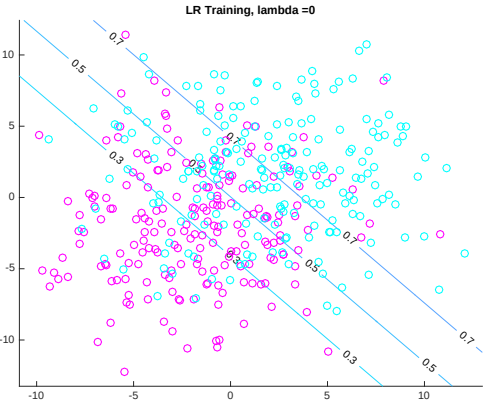
(b) Data with $\sigma = 1$, validation



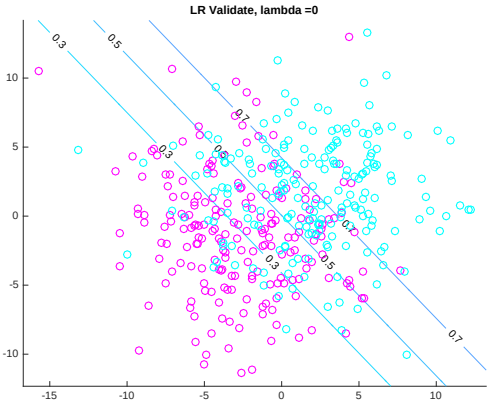
(c) Data with $\sigma = 2$, training



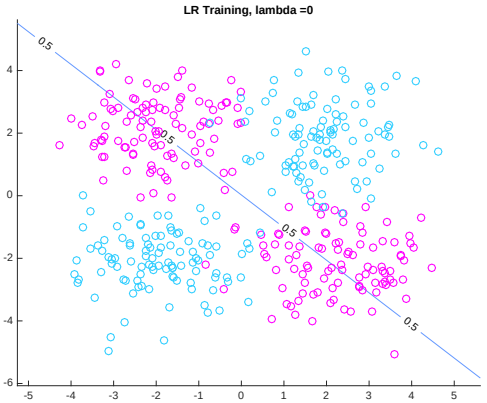
(d) Data with $\sigma = 2$, validation



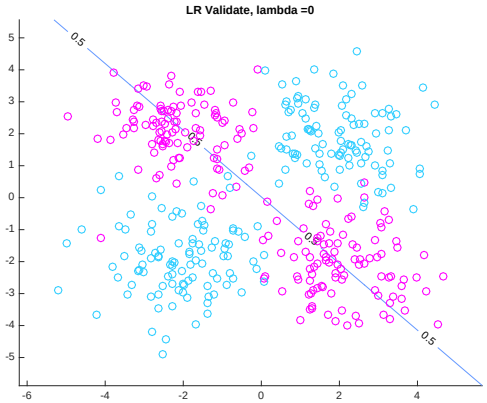
(e) Data with $\sigma = 4$, training



(f) Data with $\sigma = 4$, validation



(g) Non-seperable data, training



(h) Non-seperable data, validation

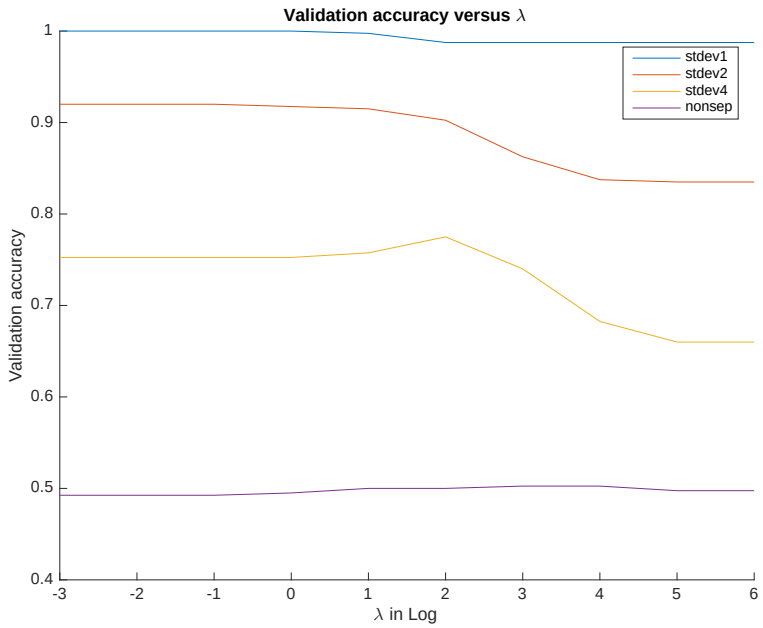
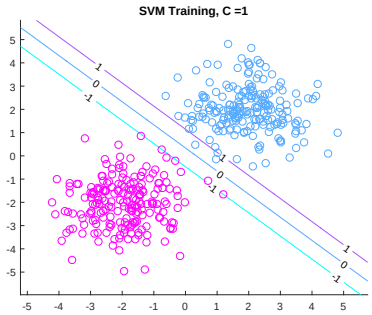
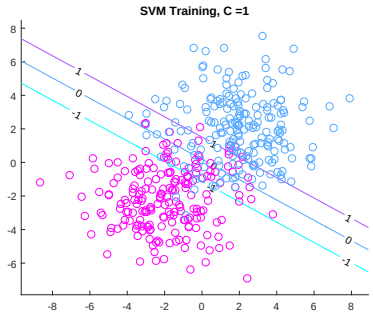


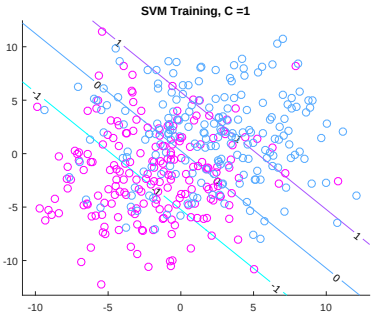
Figure 2: Cross validation error with respect to λ



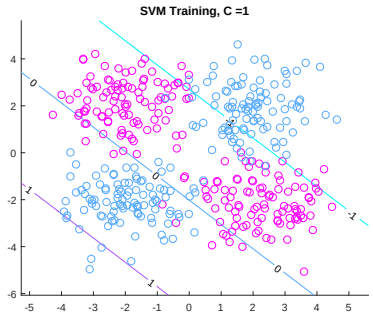
(a) Data with $\sigma = 1$, training



(b) Data with $\sigma = 2$, training



(c) Data with $\sigma = 4$, training



(d) Non-seperable data, training

Figure 3: Plots of decision boundaries from SVM with various data sets.

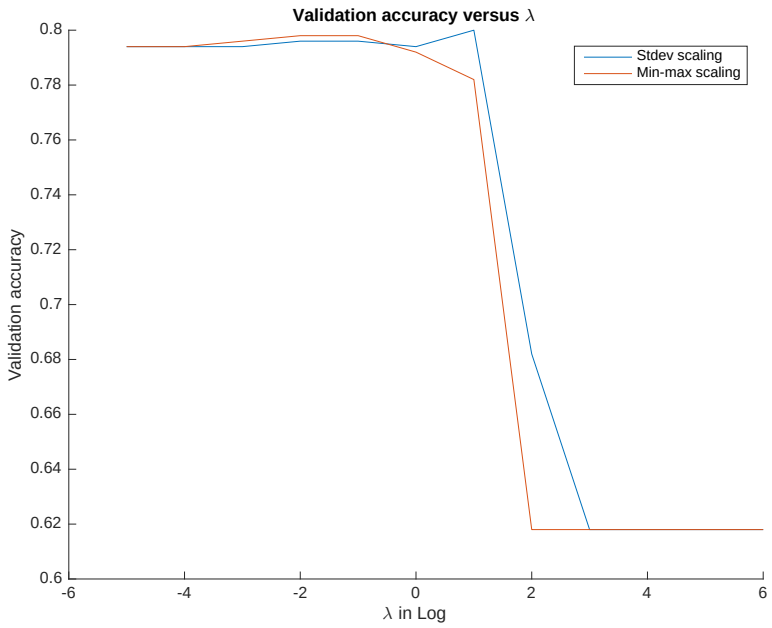


Figure 4: Tinanic Data, Cross validation error in logistic regression with respect to λ, under two different scaling methods.