# 6.867: Homework 2

## 1. Logistic regression

In this section, we explore logistic regression with L1 and L2 regularization. We use gradient descent to compare the resulting weight vectors under different regularizers and regularization parameters, and we evaluate the effect of these choices in the context of multiple data sets.

### 1.1. L2 regularization

We first consider L2 regularization, in which the objective function to minimize is

$$E_{LR}(w, w_0) = \mathrm{NLL}(w, w_0) + \lambda ||w||_2^2$$

where

$$\mathrm{NLL}(w, w_0) = \sum_i \log(1 + \exp(-y^{(i)}(wx^{(i)} + w_0)))$$

and in the case of L2 regularization,

$$||w|| = |w|_2 = \sqrt{w_1^2 + ... + w_n^2}$$

Gradient descent was run with this objective function on the training dataset `data1_train.csv` with $\lambda = 0$. We observed how the weight vector changed as a function of the number of iterations of gradient descent across various initial guesses, step sizes, and convergence criterion. There were three key findings that we were able to make. First, we saw that the ultimate convergence weight was heavily dependent on the initial guess. This actually makes a good deal of sense because of the fact that there are infinitely many ways to perfectly separate a dataset that's linearly separable. The second observation was that $w_0$ and $w_1$ always decreased as number of iterations increased whereas $w_2$ also increased. Finally, we saw that the convergence weight values were all within the same order of magnitude as the initial guess. This makes intuitive sense since we know from class that in the case of linearly separable data, the optimization aims to make the weights as large as possible.

When $\lambda = 1$, we see drastically different behavior in how the weight vector changes throughout the course of the gradient descent. The primary difference is the fact that $w_0, w_1, and w_2$ all converge to much smaller values. For example, when we set the initial guesses to be 100 for all 3 components of the weight vector, the converged weights were two orders of magnitude less than the initial guesses. The reason for this phenomenon is due to the regularization penalty incurred when we set $\lambda = 1$

### 1.2. L1 regularization

In the case of L1 regularization, the objective function is the same except that $||w_2^2||$ becomes replaced with $||w_1||$, whereby $||w_1||$ is defined by:

$$|w| = |w|_1 = \sum_{i=1}^{n} |w_i|$$

We can evaluate the different regularization techniques under different values of $\lambda$ in the context of the weight vectors, the decision boundary, and the classification error rate in each of the training data sets. Before we dive into details, we make the obvious observation that when $\lambda = 0$, the choice of regularizer ($L_1$ vs. $L_2$) makes no difference on anything since the entire term is just 0.

#### 1.2.1. Weight vector

In terms of the weight vector, we discovered that L2 regularization with $\lambda = 1$ resulted in weights with slightly lower magnitude compared with L1 regularization. This makes intuitive sense given the fact that $w_2 > 1$ in both cases meaning that it dominates the expression and squaring results in a larger penalty for cases where $n > 0$.

#### 1.2.2. Decision boundary

The decision boundary was able to perfectly separate the data for all values of $\lambda$ as well as for both L1 and L2 regularization. This is most likely due to the fact that our dataset is linearly separable.

#### 1.2.3. Classification error rate

The classification error rate is always 0 for all values of $\lambda$ as well as for both L1 and L2 regularization. This is due to the fact that our dataset is linearly separable.

| Data | Best regularizer | Best $\lambda$ | Test performance |
|---|---|---|---|
| 1 | both | all | 1.0 |
| 2 | both | all | 0.805 |
| 3 | L2 | 1 | 0.97 |
| 4 | L1 | 1 | 0.5 |

*Table 1.* Optimal regularizer and $\lambda$ for datasets

### 1.3. Optimization

By using the training and validation data sets, we can identify the best regularizer and value for $\lambda$ for each of the four data sets. These results are presented in Table 1 (above). Graphs of the decision boundaries for the optimal regularizers and $\lambda$s are shown below.

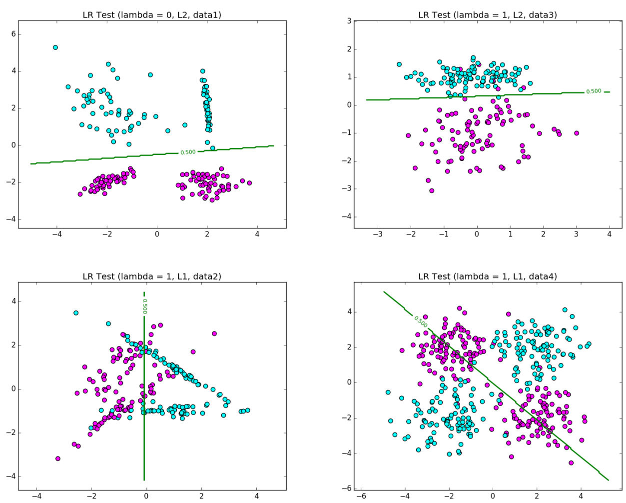


*Figure 1.* decision boundaries for test data

## 2. Support Vector Machine

In this section, we explore various versions of the dual form of support vector machines, first with slack variables and then with generalized kernel functions.

### 2.1. Dual form with slack variables

We here implement a dual form of linear SVMs with slack variables. More specifically, we solve the following optimization problem with respect to $\alpha$:

$$\max_{\alpha} -\frac{1}{2}|\sum_i \alpha_i y^{(i)} x^{(i)}|^2 + \sum_i \alpha_i$$

$$\text{s.t.} \sum_i \alpha_i y^{(i)} = 0$$

$$0 \leq \alpha_i \leq C, 1 \leq i \leq n$$

Written another way, this maximization problem can be framed as a minimization problem:

$$\min_{\alpha} \frac{1}{2} x^T P x + q^T x$$

$$\text{s.t.} Gx \leq h$$

$$Ax = b$$

where $b = 0$ and

$$x = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}, q = \begin{bmatrix} -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix}, A^T = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix}, G = \begin{bmatrix} I \\ -I \end{bmatrix},$$

where $I$ and $-I$ are the identity and negative identity matrix respectively. Furthermore,

$$P = \begin{bmatrix} x_0^2 y_0^2 & x_0 y_0 x_1 y_1 & \dots & x_0 y_0 x_{n-1} y_{n-1} \\ x_1 y_1 x_0 y_0 & x_1^2 y_1^2 & \dots & x_1 y_1 x_{n-1} y_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-1} y_{n-1} x_0 y_0 & x_{n-1} y_{n-1} x_1 y_1 & \dots & x_{n-1}^2 y_{n-1}^2 \end{bmatrix}$$

$$h^T = \begin{bmatrix} C & \dots & C & 0 & \dots & 0 \end{bmatrix}$$

In the context of the four-point 2D problem, we seek to solve the following optimization:

$$\min_{\alpha} \frac{1}{2} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}^T \begin{bmatrix} 16 & 24 & 0 & 24 \\ 24 & 36 & 0 & 36 \\ 0 & 0 & 0 & 0 \\ 24 & 36 & 0 & 36 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}^T \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

$$\text{s.t.} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} \leq \begin{bmatrix} C \\ C \\ C \\ C \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \\ -1 \\ -2 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = 0$$

Solving for $x$ when $C = 1$, we get that $\begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix}^T = \begin{bmatrix} 0.1875 & 0 & 0.3750 & 0 \end{bmatrix}^T$. This indicates that the first and third samples are support vectors because $0 \leq \alpha_i \leq C$.

Our implementation of the dual form of SVMs with slack variables was run on each of the four 2D datasets provided. With $C = 1$, the decision boundaries and classification error rates were determined. This information is illustrated and summarized in Figure 1 and Table 2, respectively. As can be seen from the classification error rates in training and validation, depending on the values of the trained parameters and the spread of the datasets, the classification error could be greater for the training data (as in dataset 2), for the validation data (as in dataset 4), or equal (as in dataset 1).

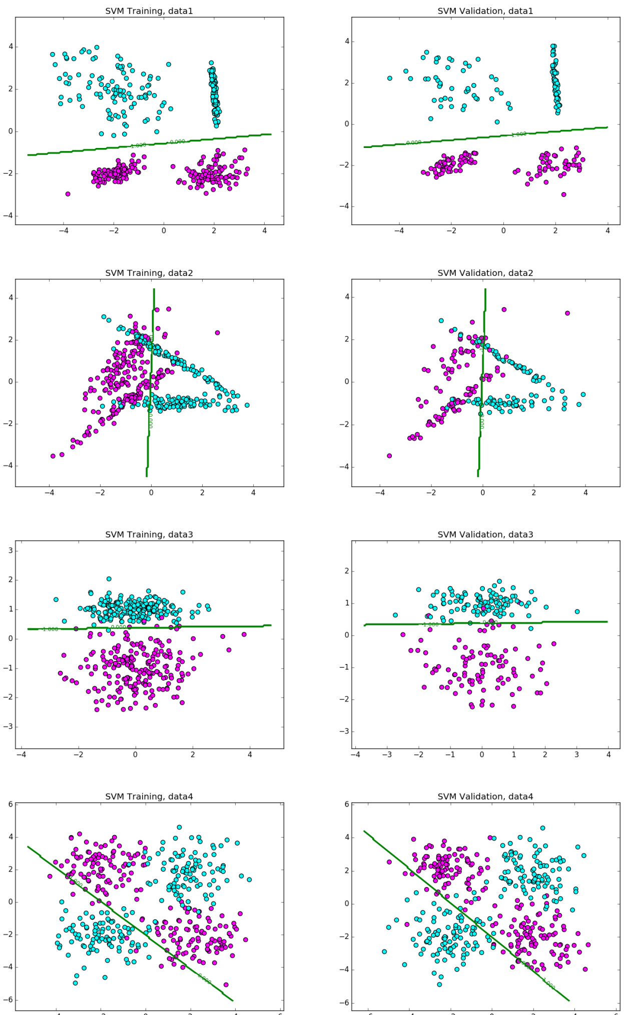


*Figure 2.* Training and validation decision boundary

| Dataset | Training error rate | Validation error rate |
|---|---|---|
| 1 | 0.0 | 0.0 |
| 2 | 0.1775 | 0.09 |
| 3 | 0.02 | 0.015 |
| 4 | 0.3 | 0.305 |

*Table 2.* Training and validation error rates

### 2.2. Dual form with kernels

## 3. Pegasos

## 4. Handwritten digit recognition