CS 536 : Computing Solutions

16:198:536

Linear Regression

Consider data generated in the following way:

- X_1 through X_{10} and X_{16} through X_{20} are i.i.d. standard normals.
- $X_{11} = X_1 + X_2 + N(\mu = 0, \sigma^2 = 0.1)$
- $X_{12} = X_3 + X_4 + N(\mu = 0, \sigma^2 = 0.1)$
- $X_{13} = X_4 + X_5 + N(\mu = 0, \sigma^2 = 0.1)$
- $X_{14} = 0.1X_7 + N(\mu = 0, \sigma^2 = 0.1)$
- $X_{15} = 2X_2 - 10 + N(\mu = 0, \sigma^2 = 0.1)$

The values Y are generated according to the following linear model:

$$Y = 10 + \sum_{i=1}^{10} (0.6)^i X_i. \tag{1}$$

Note, the variables X_{11} through X_{20} are technically irrelevant.

1) Generate a data set of size $m = 1000$. Solve the naive least squares regression model for the weights and bias that minimize the training error - how do they compare to the true weights and biases? What did your model conclude as the most significant and least significant features - was it able to prune anything? Simulate a large test set of data and estimate the 'true' error of your solved model.

2) Write a program to take a data set of size m and a parameter λ, and solve for the ridge regression model for that data. Write another program to take the solved model and estimate the true error by evaluating that model on a large test data set. For data sets of size $m = 1000$, plot estimated true error of the ridge regression model as a function of λ. What is the optimal λ to minimize testing error? What are the weights and biases ridge regression gives at this λ, and how do they compare to the true weights? What did your model conclude as the most significant and least significant features - was it able to prune anything? How does the optimal ridge regression model compare to the naive least squares model?

3) Write a program to take a data set of size m and a parameter λ, and solve for the Lasso regression model for that data. For a data set of size $m = 1000$, show that as λ increases, features are effectively eliminated from the model until all weights are set to zero.

4) For data sets of size $m = 1000$, plot estimated true error of the lasso regression model as a function of λ. What is the optimal λ to minimize testing error? What are the weights and biases lasso regression gives at this λ, and how do they compare to the true weights? What did your model conclude as the most significant and least signficiant features - was it able to prune anything? How does the optimal regression model compare tot he naive least squares model?

5) Consider using lasso as a means for feature selection: on a data set of size $m = 1000$, run lasso regression with the optimal regularization constant from the previous problems, and identify the set of relevant features; then run ridge regression to fit a model to only those features. *How can you determine a good ridge regression regularization constant to use here?* How does the resulting lasso-ridge combination model compare to the naive least squares model? What features does it conclude are significant or relatively insignificant? How do the testing errors of these two models compare?

Note: In the above problems, I suggest you generate lots of testing data to estimate the true error. In actual problems, data is limited and this would be infeasible. In these cases, you would instead set aside a fraction of data that you did not train on to evaluate the testing error, and use that to inform your choice of hyper parameters.

SVMs

We are going to try to write a program to generate Dual SVM solutions and use them to generate SVM classifiers. Recall the dual SVM:

$$\begin{aligned} \max_{\underline{\alpha}} \ & \left[\sum_{i=1}^{m} \alpha_i\right] - \frac{1}{2}\left[\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i y^i (\underline{x}^i . \underline{x}^j) y^j \alpha_j\right] \\ \text{(s.t.)} \ & \sum_{i=1}^{m} \alpha_i y^i = 0 \\ & \forall i : \alpha_i \geq 0. \end{aligned} \tag{2}$$

In the notes, I outline how you could take a pairwise coordinate-descent approach to solving this problem. One thing that makes this interesting is that it combines equality constraints and inequality constraints - this might suggest an alternate approach combining penalty and barrier methods. We can simplify our life somewhat with variable elimination. Note that

$$\alpha_1 y^1 = -\sum_{i=2}^{m} \alpha_i y^i, \tag{3}$$

Since $y^1 * y^1 = 1$, this allows us to solve for α_1 explicitly:

$$\alpha_1 = -\sum_{i=2}^{m} \alpha_i (y^i * y^1). \tag{4}$$

We are essentially taking the view that given values for $\alpha_2, \ldots, \alpha_m$, the value for α_1 is fixed by the equality constraint. In this case, we are effectively reducing the original problem from an m-variable problem to an $(m-1)$-variable problem. Note though, that we still need the 'fixed' α_1 to satisfy feasibility, i.e.,

$$-\sum_{i=2}^{m} \alpha_i (y^i * y^1) \geq 0. \tag{5}$$

This suggests the following restatement of the dual problem, substituting in this value for α_1:

$$\begin{aligned}
\max_{\alpha_2,\ldots,\alpha_m} \quad & \left[\left(-\sum_{i=2}^{m}\alpha_i(y^i * y^1)\right) + \sum_{i=2}^{m}\alpha_i\right] \\
& -\frac{1}{2}\left[\left(-\sum_{i=2}^{m}\alpha_i(y^i * y^1)\right)^2 (\underline{x}^1.\underline{x}^1) + 2\left(-\sum_{i=2}^{m}\alpha_i(y^i * y^1)\right) y^i \sum_{i=2}^{m}\alpha_i y^i(\underline{x}^i.\underline{x}^1) + \sum_{i=2}^{m}\sum_{j=2}^{m}\alpha_i y^i(\underline{x}^i.\underline{x}^j)y^j\alpha_j\right] \\
\text{(s.t.)} \quad & -\sum_{i=2}^{m}\alpha_i(y^i * y^1) \geq 0 \\
& \forall i = 2,\ldots,m : \alpha_i \geq 0.
\end{aligned} \tag{6}$$

We can easily apply barrier methods here: Let $\epsilon_t \to 0$ be a sequence of decreasing ϵ, and $F(\alpha_2,\ldots,\alpha_m)$ the objective function, a logarithmic barrier method would do the following:

- Start with an initial feasible choice of $\alpha_2,\ldots,\alpha_m$, and time $t = 0$.
- At time t, with the current values of $\alpha_2,\ldots,\alpha_m$ as a starting point, compute the (unconstrained) minimizer for
$$F(\alpha_2,\ldots,\alpha_m) - \epsilon_t\left[\sum_{i=2}^{m}\ln(\alpha_i) + \ln\left(-\sum_{i=2}^{m}\alpha_i(y^i * y^1)\right)\right]. \tag{7}$$
- Update $\alpha_2,\ldots,\alpha_m$ based on the minimizing solution, go to time $t+1$.
- As $t \to \infty$, the solution $\alpha_2,\ldots,\alpha_m$ should converge to the constrained minimizer / dual SVM solution.

1) Implement a barrier-method dual SVM solver. How can you (easily!) generate an initial feasible $\underline{\alpha}$ solution away from the boundaries of the constraint region? How can you ensure that you do not step outside the constraint region in any update step? How do you choose your ϵ_t? Be sure to return all α_i including α_1 in the final answer.

2) Use your SVM solver to compute the dual SVM solution for the XOR data using the kernel function $K(\underline{x},\underline{y}) = (1+\underline{x}.\underline{y})^2$. Solve the dual SVM by hand to check your work.

3) Given the solution your SVM solver returns, reconstruct the primal classifier and show that it correctly classifies the XOR data.