COMP 642 – Machine Learning Jan 19, 2023

Lecture 4: Linear Classifiers II

Lecturer: Anshumali Shrivastava
Scribe by: Nicolas Osa (nao5), Rhett Johnson (raj5), Todd Engelder (te12), Nasim Taheri (nt6)

Disclaimer: These lecture notes are intended to develop the thought process and intuition in machine learning. The materials are not thoroughly reviewed and can contain errors.

Linear Models Review

The data defines the problem, loss function selection is dependent on the domain (what is being solved), and model selection is dependent on what is believed to be a sufficient way to solve the problem. An algorithm is defined once a loss function and model are defined.

- A dataset consists of n number of features (x_i) and labels (y_i):

$$Data = \{x_i, y_i\}_{i=1}^{n}$$

- Linear functions are of the form:

$$F_\omega(x) = \omega^T x$$

- Least squares loss function for linear regression:

$$L(\omega) = \frac{1}{n}\sum_{i=1}^{n}(\omega^T x_i - y_i)^2$$

- Closed-form solution of linear regression when the loss function used is least squares:

$$\omega = (x^T x)^{-1} x^T y$$

This method was popular due to its mathematical properties and closed-form solution; however, the inverse in the equation becomes problematic and expensive when the dataset is large.

Regression vs. Classification

Regression: y_i is either a real number (e.g. price of a house) or a real vector (e.g. price of a stock, amount of investment in stock next year, etc.).

Classification: y_i is a class or category such as good or bad, negative or positive sentiment, apple or orange.

Loss Function

Linear regression and classification both make use of the linear function outlined above, however they are approached differently because the loss function for linear regression cannot be used in the same manner for linear classification.

Example:

The use of mathematical objects and optimizations requires that classes be assigned numbers, and so the classes of cats, cows, and dogs are mapped to numbers for use in the linear function as so:

Class	Assigned Value
Cats	0
Cows	1
Dogs	2

If the true value is Cats (0) then the loss function $L(\omega) = \frac{1}{n}\sum_{i=1}^{n}(\omega^T x_i - y_i)^2$ applied to each potential outcome has the associated loss given in the table below.

Class	Assigned Value	Loss
Cats	0	0
Cows	1	1
Dogs	2	4

This would indicate that incorrectly predicting cows is closer to the true value of cats than incorrectly predicting dogs, however, they should have the same error because both are equally wrong.

Binary Classification

In binary classification, there are only two possible classes and the classes need to be assigned numbers to work with mathematical objects and optimizations.

$$y_i = \begin{cases} -1 \\ 1 \end{cases} \quad \text{and } f(x_i) = \begin{cases} -1 \\ 1 \end{cases}$$

Note: 0 could be used in place of -1 depending on preference.

Linear classifier:

$$F_\omega(x) = sign(\omega^T x)$$

Loss function:

$$L = \frac{1}{n}\sum_{i=1}^{n} I_{y_i(\omega^T x_i)<0}$$

Regularization

Regularization is a technique in regression analysis that introduces constraints to the model to temper complexity and avoid overfitting. This is accomplished by decreasing the parameters and shrinking the model. Examples of regularization include K-means, Neural networks, and Random Forest.

There are several commonly used techniques of regularization. For example, the Residual Sum of Squares minimizes the following loss function:

$$Residual\ Sum\ of\ Squares\ (RSS) = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

Ridge Regression supplements the RSS loss function with a shrinkage quantity, which is likewise minimized. The estimates produced by this method are known as the L2 norm.

$$Ridge\ Regression = RSS + \lambda\sum_{j=1}^{p}{\beta_j}^2$$

Lasso is similar to Ridge Regression except that it only penalizes high coefficients. The coefficient estimates produced by this method are known as the L1 norm.

$$Lasso = RSS + \lambda\sum_{j=1}^{p}|\beta_j|$$

Support Vector Machines

Support Vector Machines (SVM) are a type of supervised learning model that has been widely applied to classification problems, but they can also be applied to regression (SVR). One reason that SVMs, which are simple algorithms, have been broadly adopted is that they achieve high accuracy with less computational power. During the lecture we talked about them in the context of linear binary classification; however, they can also perform non-linear classification by mapping their inputs into higher dimensional feature spaces using a kernel trick (discussed in the next lecture). Returning to the linear binary classification problem, the SVM predicts input data as either positive or negative instances based on their position with respect to a linear decision boundary or hyperplane in higher dimensional spaces (see figure 1). The linear function used in the SVM is the following:

$$f_\omega(x) = \omega^T x + \omega_o$$

Where x is the feature data and ω are the model parameters. The function $f_\omega(x)$ represents the orthogonal distance from the decision boundary to a given data point. And the SVM output is the following:

$$y_i = sign(\omega^T x + \omega_o) = \begin{cases} 0/-1, & \omega^T x + \omega_o < 0 \\ 1, & \omega^T x + \omega_o > 0 \end{cases}$$

Notice that the sign function only yields two values (-1,1) and doesn't give a sense of confidence in the classification like the output of logistical regression. Instead, the distance of the data from the hyperplane provides a measure of classification confidence. The closer the training data is to the decision boundary the less confidence we have in its classification as well as the future classification of new data.

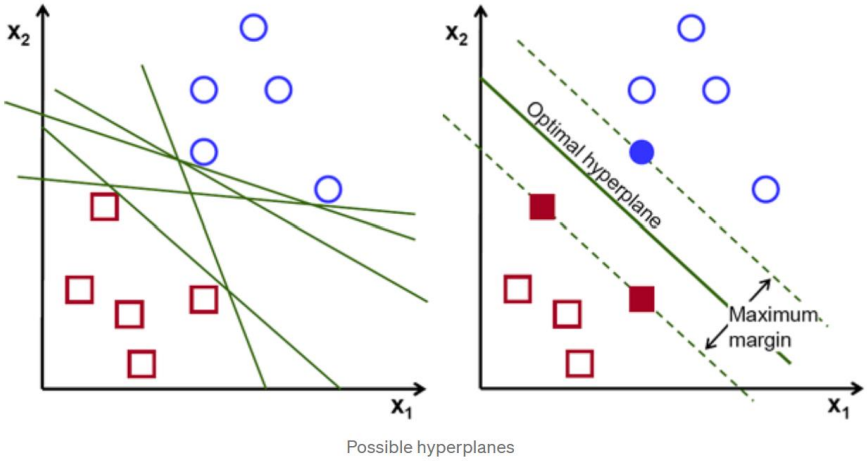

Figure 1. A two-dimensional example of binary classification with SVM.

Therefore, we want to maximize the minimum margin or orthogonal distance between the data and the decision boundary to minimize the chance of misclassification (i.e. Hard Margin SVM). Maximizing the margin also provides an additional constraint for selecting the optimal decision boundary. Notice in the left plot of figure 1 that several decision boundaries (green lines) can properly classify the data. This minimum distance is set by the closest data points on either side of the decision boundary (shown as filled symbols in the rightmost plot of figure 1). These data points are referred to as support vectors. Modifying or deleting these support vectors on a smaller dataset would significantly impact the location and orientation of the optimal decision boundary/hyperplane.

Support vector machine optimization minimizes a combination of regularized hinge loss and maximal margin width:

$$L(\omega) \;=\; min(\; C \sum_{i=1}^{n} max[0, 1 - y_i(\omega^T x)] \;+ \lambda \|\omega\|\, {}^{2}_{2})$$

The second term in the minimization attempts to maximize the margin and is derived by first defining the margins as follows:

$$\omega^T x \;+\; \omega_o \;\geq\; M \qquad when\;\; y_i \;=\; 1$$
$$\omega^T x \;+\; \omega_o \;\leq\; M \qquad when\;\; y_i \;=\; -1$$

The orthogonal distance from the decision line to a given data point is the following:

$$d_\omega(x) \;=\; \frac{\omega^T x + \omega_o}{\|\omega\|_2}$$

The constant M is generally set to equal 1. So, the margin width ends up being 2 times the distance from the decision line to the margin boundary:

$$d_\omega(x) \;=\; \frac{2}{\|\omega\|_2}$$

What's interesting is that the minimization of the L2 norm of ω ends up maximizing the margin width. However, maximizing the margin alone (i.e. Hard Margin SVM) can lead to overfitting and doesn't work for datasets that are less linearly separable. This leads us to the soft-margin SVM and the hinge-loss term in equation 3. In the soft-margin scenario, we are allowed to have some degree of misclassification, but we want to minimize this behavior using hinge loss. Figure 2 below graphically shows the relationship between the orthogonal distance of data from the decision boundary and the corresponding cost. Looking closer at the hinge loss function:

$$max[0, 1 - y_i(\omega^T x)]$$

This function chooses the max between 0 and the second term. If the real outcome (y_i) and the model prediction have the same sign and the data point is located at or further than a distance of one from the decision boundary, then the cost is zero (i.e. second term is ≤ 0). As the data points fall between the margin and the decision boundary the cost linearly increases to 1 (this scenario is shown in figure 2). This cost continues to linearly increase as the data are misclassified and located progressively away from the margin. However, this behavior could lead to outliers having a significant impact on the decision boundary and margin. As such, the hinge loss term is modulated by a parameter C and acts as a regularization. A smaller value of C imposes less penalty on misclassification and allows for larger margins.

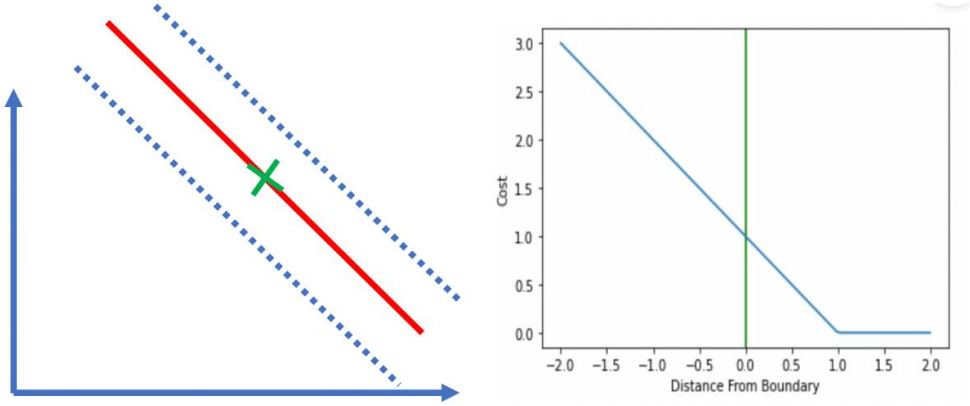

Figure 2: A graphical example of the hinge loss function.

References:

Regularization in Machine Learning
https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a

Support Vector Machines:
https://en.wikipedia.org/wiki/Support_vector_machine

Support Vector Machine — Introduction to Machine Learning Algorithms:
https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Understanding Hinge Loss and the SVM Cost Function:
https://programmathically.com/understanding-hinge-loss-and-the-svm-cost-function/