Fundamentals of Machine Learning: Part 2: Linear Classifiers

Binary Classification Problem

Dataset. Consider a collection of features $\mathbf{X} = \{X_1, \ldots, X_d\}$, such that $dom(X_i) \subseteq \mathbb{R}$ for all $i = 1 \ldots d$. These are our *independent variables*.

Consider also an additional variable Y, such that $dom(Y) = \{0, 1\}$ or $dom(Y) = \{-1, +1\}$. This is our *binary dependent variable*.

Let $X = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$ be a collection of *data points*, such that $(\forall j \in 1 \ldots n)(\mathbf{x_j} \in \mathbb{R}^d)$. Let $\mathbf{y} = \{y_1, \ldots, y_n\}$ such that $(\forall j \in 1 \ldots n)(y_j \in dom(Y))$. We write X as

$$\mathbf{X} = \begin{pmatrix} X_1 & X_2 & \ldots & X_d \\ \hline x_{11} & x_{12} & \ldots & x_{1d} \\ x_{21} & x_{22} & \ldots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{nd} \end{pmatrix}$$

We also write $\mathbf{x_i} = (x_{i1}, \ldots, x_{id})$.

The *binary classification problem* can be specified as follows:

> Build a function $f : \mathbb{R}^d \longrightarrow dom(Y)$ that predicts the binary label of a data point $\backsim \in \mathbb{R}^d$.

Dependent Variable. In classification scenarios, the dependent variable Y is typically considered to be *categorical*. Many classification methods, in order to allow for the use of mathematical functions to represent classification decisions, assume that Y takes *numeric values*. For *binary classification problems*, some methods take advantage of treating values of Y as 0 and 1, while other methods (primarily those structured around *separating planes*) take advantage of treating values of Y as -1 and $+1$. In what follows, we will treat levels of the dependent variable Y (i.e., the class labels) as whatever values that suit the best the method we are studying.

If $dom(Y) = \{v_1, v_2\}$, we sometimes use abbreviations $\mathbf{X}_{v_1}$ and X_{v_2} to represent all data points belonging to classes v_1 and v_2 respectively.

LDA: Linear Discriminant Analysis

Separation of classes. For binary classification problems where the data points reside in the $\mathbb{R}^d$ space (or a subspace of thereof), we often refer to solving the classification problem as in terms of *separating* the classes. Often, a mathematical (geometrical construct) like a plane, a hyperspace, or multidimensional surface are used as actual *separators* - with data points on one side of it classified into one (e.g., positive) class, and data points on the other side classified into the other (e.g., negative) class.

Idea. Consider our d-dimensional space $\mathbb{R}^d$. If we draw some line $L(\mathbf{w}) : w_0 + w_1x_1 + \ldots w_dx_d = 0$ through this space, and *project* the data points $\mathbf{X}$ on $L(\mathbf{w})$, then we can reduce the problem of separating data points in d-dimensional space to the problem of *finding the line equation* $L(\mathbf{w})$ *that* ***best separates*** *the projections of the data points from* $\mathbf{X}$ *along a 1-dimensional space.* (note, here $\mathbf{w} = (w_0, w_1, \ldots, w_d)$)

Figures 1 and 2 demonstrate this idea. Figure 1 shows how the labeled data points from a dataset $\mathbf{X}$ projected onto *some* line $L(\mathbf{w})$. Figure 2 shows just the one-dimensional picture - the projections of points from $\mathbf{X}$ onto the line $L(\mathbf{w})$.

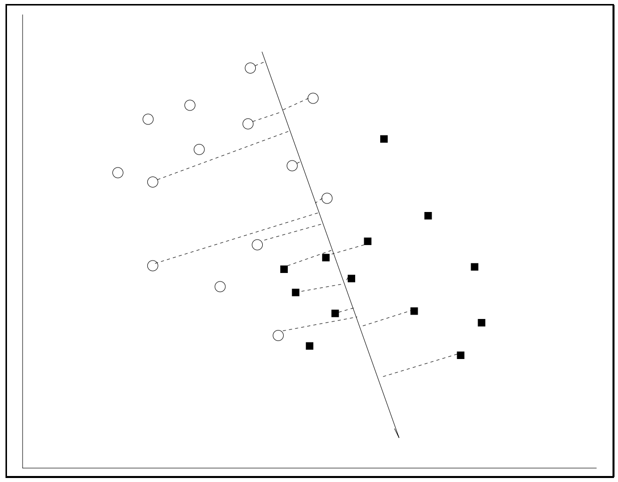

Figure 1: Linear Discriminant Analysis: projections of data points onto a single line.

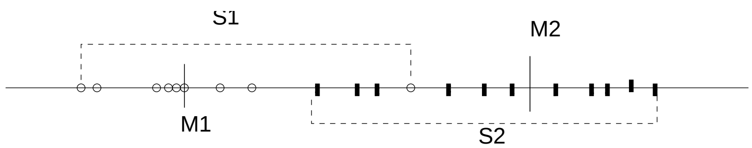

Figure 2: Linear Discriminant Analysis: separation of classes along a single line.

Goal. Following the idea expressed above, we want to *find the set of values* $\mathbf{w} = (w_0, w_1, \ldots, w_d)$ *such that the line* $L(\mathbf{w})$ *is the* ***best separating line*** *for the training data* $\langle \mathbf{X}, Y \rangle$.

Model Shape. From the above, it is clear that we are looking for a model that is essentially an equation of a line in a d-dimensional space. The model **is** our line equation

$$L(\mathbf{w}) : w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_d x_d = 0$$

Cost function/Optimization Criterion. What is the right mathematical criterion that matches the intuition of "best separation" for the two classes along a line based on projections of the data points on this line?

We start our derivation by constructing the set $A_{\mathbf{w}}$ of projections of points from $\mathbf{X}$ onto $L(\mathbf{w})$.

Specifically, without loss of generality, let us assume that $\mathbf{w}$ is a *unit vector*, i.e.,

$$\mathbf{w}^T \mathbf{w} = 1$$

The projection of a data point (d-dimensional vector) $\mathbf{x} = (x_1, \ldots, x_d)$ onto $\mathbf{w}$ is

$$\frac{\mathbf{w}^T \mathbf{x}}{\mathbf{w}^T \mathbf{w}} \mathbf{w} = (\mathbf{w}^{\mathbf{T}} \mathbf{x}) \mathbf{w}$$

The value $a = \mathbf{w}^T \mathbf{x}$ is the numeric offset of the projection of $\mathbf{x}$ onto $\mathbf{w}$. The set $A_{\mathbf{w}}$ is then defined as

$$A = \{\mathbf{w}^T \mathbf{x} | \mathbf{x} \in \mathbf{X}\} = \{a_1, \ldots, a_n\}$$

Let us split A into $A_0 = \{a_i | y_i = 0\}$ and $A_1 = \{a_i | y_i = 1\}$.
We can find the *means* of the sets A_0 and A_1:

$$m_0 = \frac{1}{|A_0|} \sum_{a_i \in A_0} a_i = \frac{1}{|A_0|} \sum_{\mathbf{x_i} \in \mathbf{X_0}} \mathbf{w}^T \mathbf{x} = \frac{1}{|\mathbf{X_0}|} \mathbf{w^T} \sum_{mathbfx_i \in \mathbf{X_0}} \mathbf{x_i} = \mathbf{w}^T \mu_0,$$

where μ_0 is the centroid of class 0 (set $\mathbf{X_0}$).
Similarly, if we set $\mu_1 = \frac{1}{|\mathbf{X_1}|} \sum_{mathbfx_i \in \mathbf{X_1}} \mathbf{x_i}$, then the mean point for class 1 along the line $\mathbf{w}$ is found as:

$$m_1 = \mathbf{w}^T \mu_1$$

Attempt 1 at cost function. What if we set

$$f(w) = |m_0 - m_1| = \mathbf{w}^T (\mu_0 - \mu_1) \longrightarrow \max$$

as our cost function? This is a good idea but it needs to be upgraded, because it is possible that the means of the two classes are far apart, but the points are widely distributed (have a high variance). So, the real cost function needs to take variance into account.

Attempt 2 at cost function. Let

$$s_0^2 = \sum_{a_i \in A_0} (a_i - m_0)^2$$

$$s_1^2 = \sum_{a_i \in A_1} (a_i - m_1)^2$$

We call s_0 and s_1 the *scatter* of classes 0 and 1 respectively. Scatter is the *total squared deviation* of all data points in the class.

We want $|m_0 - m_1|$ to be large, while we also want s_0 and s_1 to be small. One cost function that achieves this effect is the ***Fisher Linear Discriminant Analysis (LDA) objective***):

$$J(\mathbf{w}) = \frac{(m_0 - m_1)^2}{s_0^2 + s_1^2} \longrightarrow \max$$

We call the vector $\mathbf{w}$ that optimizes $J(\mathbf{w})$ the *optimial linear discriminant*.

Closed form solution? Let us try to optimize $J(\mathbf{w})$.
First, let's consider the $(m_0 - m_1)^2$ term.

$$(m_0 - m_1)^2 = (\mathbf{w}^T (\mu_0 - \mu_1))^2 = \mathbf{w}^T ((\mu_0 - \mu_1)(\mu_0 - \mu_1)^T) \mathbf{w}$$

In this expression, $(\mu_0 - \mu_1)(\mu_0 - \mu_1)$ is a $d \times d$ rank-one matrix.
Setting $\mathbf{B} = \mu_0 - \mu_1)(\mu_0 - \mu_1)$, we arrive to

$$(m_0 - m_1)^2 = \mathbf{w}^T \mathbf{B} \mathbf{w}$$

.

Next, let us figure out the scatter:

$$s_0^2 = \sum_{a_i \in \mathbf{A_0}} (a_i - m_0)^2 = \sum_{\mathbf{x_i} \in \mathbf{X_0}} (\mathbf{w}^T \mathbf{x_i} - \mathbf{w}^T \mu_0)^2 = \sum_{\mathbf{x_i} \in \mathbf{X_0}} (\mathbf{w}^T (\mathbf{x_i} - \mu_0))^2 = \mathbf{w}^T \left(\sum_{\mathbf{x_i} \in \mathbf{X_0}} (\mathbf{x_i} - \mu_0)(\mathbf{x_i} - \mu_0)^T \right) \mathbf{w}$$

Here, $\sum_{\mathbf{x_i} \in \mathbf{X_0}} (\mathbf{x_i} - \mu_0)(\mathbf{x_i} - \mu_0)^T$ is also a $d \times d$ rank-one matrix we call *scatter matrix for class 0*. Let's set $\mathbf{S_0} = \sum_{\mathbf{x_i} \in \mathbf{X_0}} (\mathbf{x_i} - \mu_0)(\mathbf{x_i} - \mu_0)^T$.

Similarly, we can set $\mathbf{S_1} = \sum_{\mathbf{x_i} \in \mathbf{X_1}} (\mathbf{x_i} - \mu_1)(\mathbf{x_i} - \mu_1)^T$. Setting

$$\mathbf{S} = \mathbf{S_0} + \mathbf{S_1},$$

we get

$$s_0^2 + s_1^2 = \mathbf{w}^T S_0 \mathbf{w} + \mathbf{w}^T S_1 \mathbf{w} = \mathbf{w}^T (\mathbf{S_0} + \mathbf{S_1}) \mathbf{w} = \mathbf{w}^T \mathbf{S} \mathbf{w}$$

Thus,

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{B} \mathbf{w}}{\mathbf{w}^T \mathbf{S} \mathbf{w}}.$$

Optimization. Now, let's set

$$\frac{d}{d\mathbf{w}} J(\mathbf{w}) = 0$$

$$\frac{d}{d\mathbf{w}} J(\mathbf{w}) = \frac{2\mathbf{B}\mathbf{w}(\mathbf{w}^T \mathbf{S} \mathbf{w}) - 2\mathbf{S}\mathbf{w}(\mathbf{w}^T \mathbf{B} \mathbf{w})}{(\mathbf{w}^T \mathbf{S} \mathbf{w})^2} = 0$$

From here:

$$\mathbf{B}\mathbf{w}(\mathbf{w}^T \mathbf{S} \mathbf{w}) - \mathbf{S}\mathbf{w}(\mathbf{w}^T \mathbf{B} \mathbf{w}) = 0$$

or

$$\mathbf{B}\mathbf{w}(\mathbf{w}^T \mathbf{S} \mathbf{w}) = \mathbf{S}\mathbf{w}(\mathbf{w}^T \mathbf{B} \mathbf{w})$$

Following this:

$$\mathbf{B}\mathbf{w} = \mathbf{S}\mathbf{w} \frac{\mathbf{w}^T \mathbf{B} \mathbf{w}}{\mathbf{w}^T \mathbf{S} \mathbf{w}}$$

$$\mathbf{B}\mathbf{w} = J(\mathbf{w}) \mathbf{S}\mathbf{w} = \lambda \mathbf{S}\mathbf{w},$$

where $\lambda = J(\mathbf{w})$.

This leads to:

$$(\mathbf{S^{-1}B})\mathbf{w} = \lambda \mathbf{w},$$

that is:

> *The set of parameters $\mathbf{w}$ that optimizes the Fisher LDA objective is the eigenvector of the matrix $\mathbf{S^{-1}B}$ that corresponds to the largest eigenvalue λ of this matrix.*

For a two-class problem with a non-singular matrix $\mathbf{S}$ (i.e., S^{-1} exists), the solution can be obtained as

$$\hat{\mathbf{w}} = \mathbf{S}^{-1}(\mu_0 - \mu_1)$$

with $\mathbf{w}$ being $\hat{\mathbf{w}}$ normalized.