Fundamentals of Machine Learning:

Linear Models for Classification: two Geometric Perspectives

Prof. Andrew D. Bagdanov (andrew.bagdanov AT unifi.it)

UNIVERSITÀ DEGLI STUDI FIRENZE

DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Outline

Introduction

Linear Discriminant Functions

Least Squares for Classification

Fisher's Linear Discriminant

Concluding Remarks

Introduction

Classification and decision surfaces

- The goal of classification is to take an input vector $\mathbf{x}$ and assign it to one of K classes.
- We denote these classes as $\mathcal{C}_k$ for $k \in \{1, \ldots, K\}$.
- The easiest setting is single label classification where every $\mathbf{x}$ belongs to exactly one class.
- Thus the input space is divided up into decision regions whose boundaries are called decision boundaries or decision surfaces.
- We will first consider linear models where these decision surfaces are linear functions of input $\mathbf{x}$.
- Data whose classes can be exactly separated with linear decision surfaces are called linearly separable.

Lecture objectives

At the end of this lecture you will:

- Understand the geometry of linear discriminant functions and how to interpret them.
- Understand how to apply least squares to estimate the parameters of linear discriminant models.
- Understand the limitations of least squares for fitting classification models.
- Understand how Fisher's Linear Discriminant addresses some of the shortcomings of least squares by finding a ”better” direction for discrimination.

Linear Discriminant Functions

Discriminant functions for two classes

- The simplest representation for a discriminant is a linear function:

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$$

- Again, $\mathbf{w}$ is the weight vector and w_0 is a scalar bias.
- For classification, the negative bias is sometimes called a threshold.
- The decision rule:

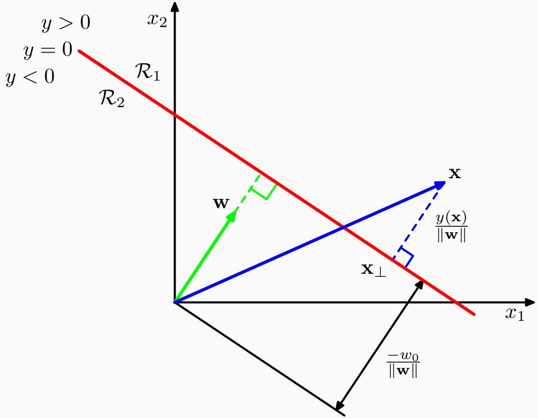
$$\text{class}(\mathbf{x}) = \begin{cases} \mathcal{C}_1 & \text{if } \mathbf{w}^T\mathbf{x} + w_0 \geq 0 \\ \mathcal{C}_2 & \text{if } \mathbf{w}^T\mathbf{x} + w_0 < 0 \end{cases}$$

- The normal distance from the origin to the decision surface is:

$$\frac{\mathbf{w}^T\mathbf{x}}{||\mathbf{w}||} = -\frac{w_0}{||\mathbf{w}||}$$

The geometry of linear discriminants

- This is the graphic I think of when I want to remember how linear discriminant geometry works:



Rolling bias into the basis

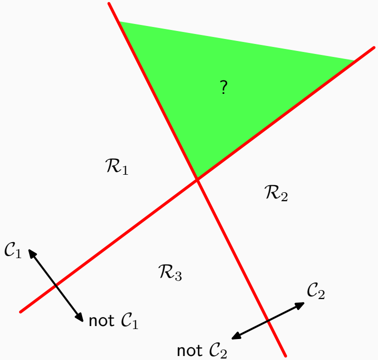
- As with regression, it is sometimes convenient to use a compact notation.
- So, we introduce a dummy dimension $x = 1$ and define:

$$\begin{aligned} \tilde{\mathbf{w}} &= (w_0, \mathbf{w})^T \\ \tilde{\mathbf{x}} &= (x_0 = 1, \mathbf{x})^T \\ \text{so that } y(\mathbf{x}) &= \tilde{\mathbf{w}}^T \tilde{\mathbf{x}} \end{aligned}$$

- So the decision surfaces in this space are D dimensional hyperplanes passing through the origin of the augmented $D + 1$ dimensional space.

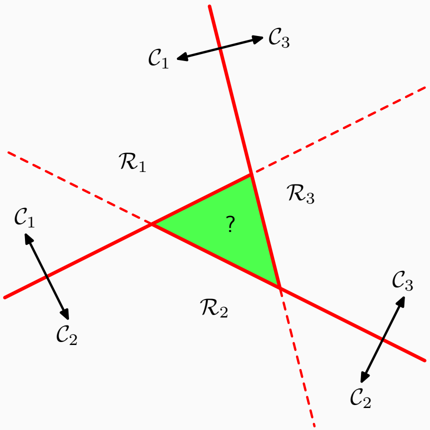
Multiple classes

- OK, but two-class problems are really boring. What if we have more?
- We could use $K-1$ classifiers, each solving a two-class problem separating a class $\mathcal{C}_k$ from points not in that class.
- This is known as the one-versus-rest classifier:



Multiple classes

- Back to the drawing board... Use $K(K-1)/2$ binary discriminant functions.
- One for every pair of classes:



Multiple classes

- We can avoid all of these hassles if we use a single K-class discriminant employing k linear functions:

$$y_k(\mathbf{x}) = \mathbf{w}_k^T\mathbf{x} + w_{k0}$$

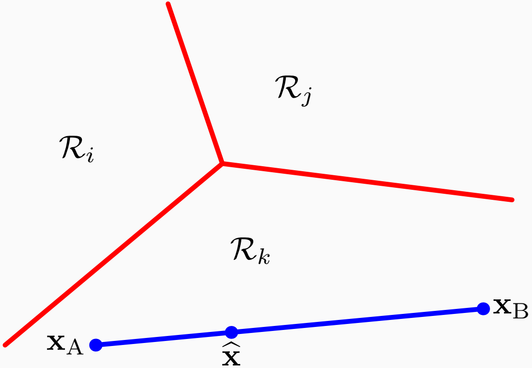
- Which we can also pack together in a single matrix multiplication:

$$y(\mathbf{x}) = \widetilde{\mathbf{W}}^T\tilde{\mathbf{x}}$$

- We now just assign $\mathbf{x}$ to class $\mathcal{C}_k$ if $y_k(\mathbf{x}) \geq y_j(\mathbf{x})$ for all $j \neq k$.

Multiple class decision boundaries

- Since we are taking a max over linear functions, we have singly-connected, convex decision regions:



Least Squares for Classification

A compact linear model

- For regression, linear models with a least squares error measure led to a simple closed form solution.
- So, let's see if we can pull off the same trick here.
- Each class is described by its own linear model:

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0}$$

- Or, even better:

$$y(\mathbf{x}) = \widetilde{\mathbf{W}}^T \tilde{\mathbf{x}}$$

- It is useful to think about what the columns of $\widetilde{\mathbf{W}}$ are.
- Also, what should our targets be for classification?

An analytical solution

- We solve for $\widetilde{\mathbf{W}}$ by minimizing a sum-of-squares error.
- Consider a training set $\{\mathbf{x}_n, \mathbf{t}_n\}$ for $n \in \{1, \ldots, N\}$.
- And define a matrix $\mathbf{T}$ whose n^{th} row is $\mathbf{t}_n^T$.
- Together with a corresponding matrix $\widetilde{\mathbf{X}}$ whose n^{th} row is $\tilde{\mathbf{x}}_n^T$.
- We can then write the error function as:

$$E(\widetilde{\mathbf{W}}) = \frac{1}{2}\mathrm{Tr}\left\{(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - T)^T(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - T)\right\}$$

- Setting the gradient wrt $\widetilde{\mathbf{W}}$ to zero and solving, we arrive at:

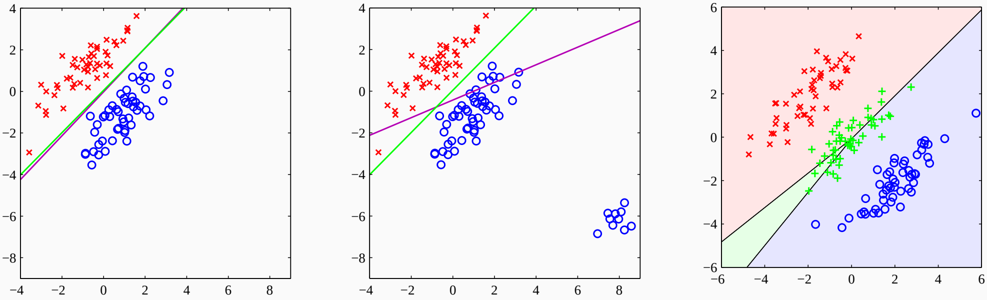
$$\begin{aligned} \widetilde{\mathbf{W}} &= (\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^T\mathbf{T} \\ &= \widetilde{\mathbf{X}}^\dagger\mathbf{T} \end{aligned}$$

An analytical solution

- So we have a nice analytic form for a K-class classifier from data:

$$y(\mathbf{x}) = \widetilde{\mathbf{W}}^T \tilde{\mathbf{x}} = \mathbf{T}^T (\widetilde{\mathbf{X}}^{\dagger})^T \mathbf{x}$$

- However, it is not very good classifier:



Fisher’s Linear Discriminant

Linear classification as dimensionality reduction

- A way to think about linear classification with discriminants is as a reduction of dimensionality to one dimension.
- Let's look at some motivating examples of this...

Separating the means

- Our first strategy might be to calculate the means of each class in feature space:

$$\mathbf{m}_1 = \frac{1}{N_1}\sum_{n=1}^{N_1}\mathbf{x}_{1,n}, \quad \mathbf{m}_1 = \frac{1}{N_1}\sum_{n=1}^{N_2}\mathbf{x}_{2,n}$$

- And then to compute $\mathbf{w}$ so that projecting these two points onto it maximizes the distance between the projections:

$$\mathbf{w}^T\mathbf{m}_2 - \mathbf{w}^T\mathbf{m}_1 = \mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1)$$

- Does anyone see a problem with maximizing this expression?

Separating the means

- Our first strategy might be to calculate the means of each class in feature space:

$$\mathbf{m}_1 = \frac{1}{N_1}\sum_{n=1}^{N_1}\mathbf{x}_{1,n}, \quad \mathbf{m}_1 = \frac{1}{N_1}\sum_{n=1}^{N_2}\mathbf{x}_{2,n}$$

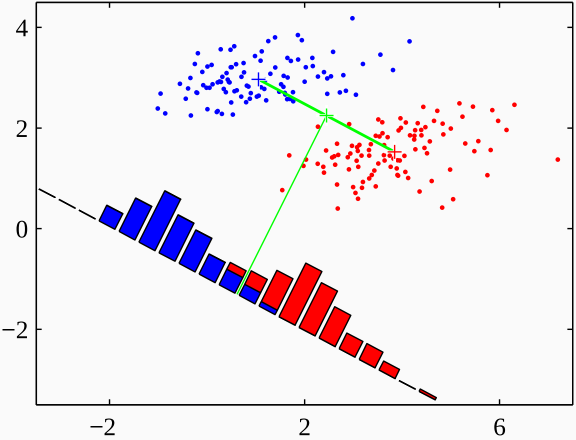
- And then to compute $\mathbf{w}$ so that projecting these two points onto it maximizes the distance between the projections:

$$\mathbf{w}^T\mathbf{m}_2 - \mathbf{w}^T\mathbf{m}_1 = \mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1)$$

- Does anyone see a problem with maximizing this expression?
- We can impose the constraint that $||\mathbf{w}||_2 = 1$ in the optimization (using a Lagrange multiplier).
- The result is, unsurprisingly: $\mathbf{w} \propto (\mathbf{m}_2 - \mathbf{m}_1)$

Separating the means

- Is it any good? What goes wrong?



Accounting for anisotropy

- Both class distributions have strongly non-diagonal covariance matrices.
- Fisher's insight was to maximize inter-class variance, while simultaneously minimizing intra-class variance *in the projected, 1-dimensional space*.
- The within-class variance – also called compactness – is:

$$s_k^2 = \sum_{n=1}^{N_k} (\mathbf{w}^T \mathbf{x}_{k,n} - \mathbf{w}^T \mathbf{m}_k)^2$$

- The Fisher Criterion is then the ratio of between-class to within-class variance:

$$J(\mathbf{w}) = \frac{(\mathbf{w}^T \mathbf{m}_2 - \mathbf{w}^T \mathbf{m}_1)^2}{s_1^2 + s_2^2}$$

Generalizing

- We can make the between and within more explicit by writing:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_\mathrm{B} \mathbf{w}}{\mathbf{w}^T \mathbf{S}_\mathrm{W} \mathbf{w}} \text{ (see Eqs 4.27 and 4.28 in Bishop)}$$

- Where here $\mathbf{S}_\mathrm{B}$ is the between-class covariance, and $\mathbf{S}_\mathrm{W}$ is the within-class.
- After differentiating and massaging terms, we find:

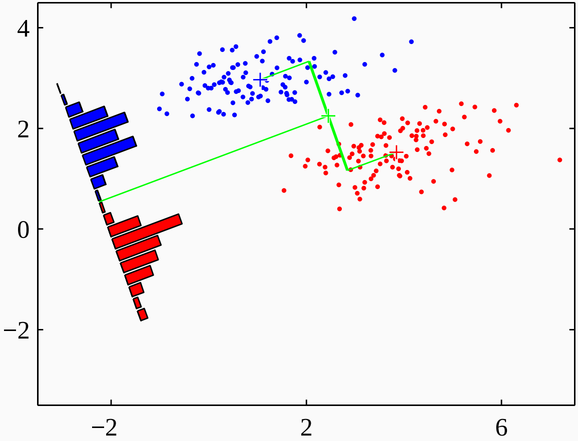
$$(\mathbf{w}^T \mathbf{S}_\mathrm{B} \mathbf{w}) \mathbf{S}_\mathrm{W} \mathbf{w} = (\mathbf{w}^T \mathbf{S}_\mathrm{W} \mathbf{w}) \mathbf{S}_\mathrm{B} \mathbf{w}$$

- We don't care (for now) about the scale of $\mathbf{w}$, so after dropping scalar factors and multiplying my $\mathbf{S}_\mathrm{W}^{-1}$:

$$\mathbf{w} \propto \mathbf{S}_\mathrm{W}^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$$

Fisher's Linear Discriminant for two classes

- Which works much better:



Relation to least squares

- The least squares approach to linear discriminant-based classification is based on learning linear functions that are as close as possible to the set of target values.
- The Fisher Criterion, instead, tries to maximize class separation in the discriminant space.
- For the two-class problem, we can show that the Fisher discriminant is really a special case of good old least squares.
- Instead of using a target of 1 for the one-hot target probabilities, we will use:

$$t_n = \begin{cases} \frac{N}{N_1} & \text{if } \mathbf{x}_n \in \mathcal{C}_1 \\ -\frac{N}{N_2} & \text{if } \mathbf{x}_n \in \mathcal{C}_2 \end{cases}$$

- This is an estimate of the reciprocal prior for each class.

Relation to least squares

- Our error is still least squares:

$$E = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}^T\mathbf{x}_n + w_0 - t_n)^2$$

- Setting gradients to zero with respect to w_0 and $\mathbf{w}$:

$$\begin{aligned}\sum_{n=1}^{N}(\mathbf{w}^T\mathbf{x}_n + w_0 - t_n) &= 0 \\ \sum_{n=1}^{N}(\mathbf{w}^T\mathbf{x}_n + w_0 - t_n)\mathbf{x}_n &= 0\end{aligned}$$

Relation to least squares

- Solving, we find (using liberally the expression for the new targets):

$$w_0 = -\frac{\mathbf{w}^T(N_1\mathbf{m}_1 + N_2\mathbf{m}_2)}{N}$$

- And for the direction:

$$\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$$

Concluding Remarks

Linear discriminants

- In this introduction we looked at linear discriminants for supervised classification.
- These approaches are roughly equivalent to the geometric approaches to linear regression.
- Actually, they are more like the maximum likelihood solutions to regression, but we have yet to add a probabilistic interpretation.
- It is important to get a feel for the geometry of linear discriminants.
    - We project inputs onto the model weights $\mathbf{w}$.
    - This weight vector $\mathbf{w}$ is perpendicular to the discriminant surface.
    - A linear discriminant implicitly reduces to a one-dimensional threshold problem.

Least squares kinda sucks as a classifier

- Least squares classification has significant drawbacks – most notably its sensitivity to outliers.
- As we will see, we are implicitly making assumptions about the shape of the class distributions when using least squares.
- Nonetheless, we still retain the analytic closed-form solution offered by least squares.

Fisher's Linear Discriminant

- If we rotate the space so that we maximize certain properties of the class projections, we can do better.
- This is what Fisher's Criterion does: maximize between-class variance, while minimizing within-class variance.
- Doing this significantly improves linear classification performance.
- But, still retains the nice analytic properties of least squares.

The way forward

- Following our development for regression, in the next lecture we will take a look a probabilistic models for classification.
- We will look at classification from generative and discriminative perspectives.
- And we will develop a fully Bayesian approach to classification.
- These models will retain the linear properties we have come to know and love, and will admit analytic solutions given some key assumptions.

The Perceptron

- There is another important geometric approach to linear discriminant classification.
- The Perceptron Algorithm is an important precursor to modern Artificial Neural Networks (ANNs).
- Indeed, the Perceptron – i.e. a linear discriminant function – is the fundamental building block of modern neural network models.
- We will delay our discussion of the Perceptron, however, until we are about to dive into modern Deep Learning models.

Reading and Homework Assignments

Reading Assignment:

- Bishop: Chapter 4 (4.1.1 – 4.1.6)