Linear Discriminants

CSci 5525: Machine Learning

Instructor: Nicholas Johnson

September 17, 2020

Announcements

- HW1 will be posted next Tuesday (9/22)
- Project proposals due next Thursday (9/24)
    - Submit via Canvas

Problem

Suppose you work at a fruit company and you want to design a system which can determine whether a piece of fruit is good or bad. Let's say you have data from the past month which consists of the mass and label such as 'good' or 'bad' for each piece of fruit. For example:

Mass (g)	Label
70.2	Good
93.2	Good
40.9	Bad
82.3	Good
68.1	Bad
87.6	Bad
96.8	Good

How would you design the system?

Classification

- Dataset: $\mathcal{D} = \{(\text{Mass}_i, \text{Good}/\text{Bad}_i)\}_{i=1}^n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$
- $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$, $y \in \mathcal{Y}$ (discrete set)
- Mostly focus on binary classification $\mathcal{Y} = \{0, 1\}$
- Goal: find prediction function $f : \mathcal{X} \to \mathcal{Y}$

Linear Classification

- In this lecture we consider linear predictors $\hat{f}$ parameterized by weight vector $\mathbf{w} \in \mathbb{R}^p$

$$\hat{y} = \hat{f}(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x})$$

- Natural loss function is 0-1 loss:

$$\ell(y, \hat{y}) = \mathbb{1}[y \neq \hat{y}]$$

ERM for Linear Classification

- Given data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ the ERM problem is

$$\text{argmin}_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[y_i \neq \hat{y}]$$

- Question: Is it always possible to minimize empirical risk down to 0?

Linearly Separable

- Question: Is it always possible to minimize empirical risk down to 0?

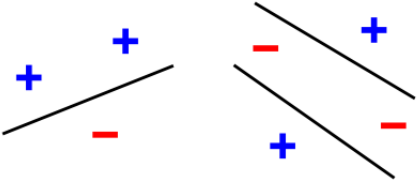


Figure : Illustration of linear separability from Wikipedia.

Feature Transformation/Representation

- Enrich linear regression/classification by transforming features $\mathbf{x}$ into $\phi(\mathbf{x})$
- Predict with transformed features: $\hat{f}(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x})$
- Examples:

$$x \in \mathbb{R}, \phi(x) = \ln(1 + x)$$
$$\mathbf{x} \in \mathbb{R}^p, \phi(\mathbf{x}) = (1, x(1), \ldots, x(p), x(1)^2, \ldots, x(p)^2, x(1)x(2), \ldots, x(p-1)x(p))$$
$$x \in \mathbb{R}, \phi(x) = (1, \sin(x), \cos(x), \sin(2x), \cos(2x), \ldots)$$

- Feature transformation could turn a linearly inseparable dataset into a linearly separable one

XOR Example

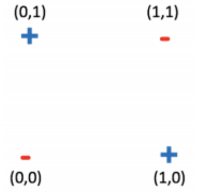


Figure : XOR dataset is not linearly separable.

- Consider the following feature transformation

$$\phi(\mathbf{x}) = (1, x_1, x_2, x_1 x_2)$$

XOR Example

- Using the previous feature transformation, we can learn the following predictor

$$\hat{f}(\mathbf{x}) = -1 + 2x_1 + 2x_2 - 3.5x_1x_2$$

- Predictor is linear in $\phi(\mathbf{x})$ and perfectly classifies XOR dataset

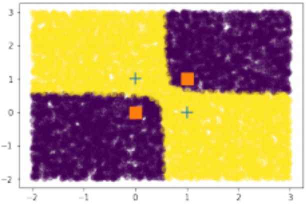


Figure : Nonlinear decision boundary of linear mapping $\hat{f}$.

Hardness of ERM

- ERM optimization problem:

$$\text{argmin}_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[y_i \neq \text{sign}(\mathbf{w}^\top \mathbf{x}_i)]$$

- This problem is NP-Hard (think about why)

Hardness of ERM

- To obtain efficient algorithms, replace 0-1 loss with other *surrogate loss* function (that is convex)
- Hinge Loss:

$$L(f, \mathbf{x}, y) = \max(0, 1 - yf(\mathbf{x})) = \begin{cases} 1 - yf(\mathbf{x}) & \text{if } yf(\mathbf{x}) < 1, \\ 0 & \text{otherwise.} \end{cases}$$

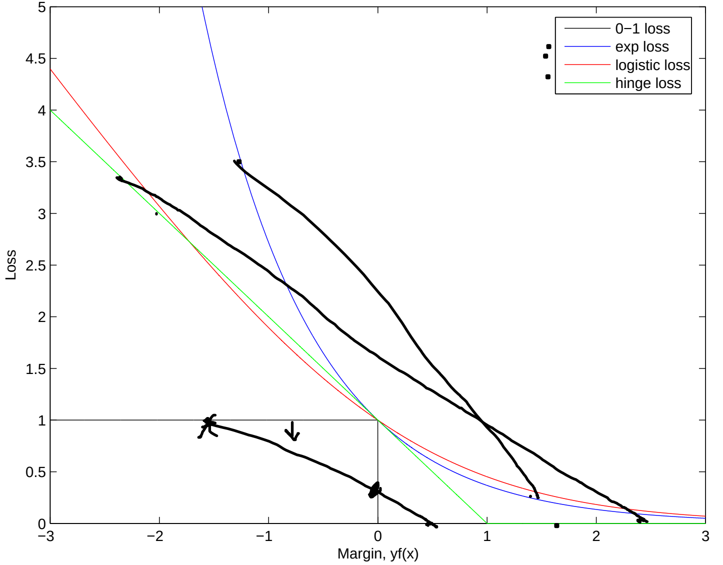
- Exponential Loss:

$$L(f, \mathbf{x}, y) = \exp(-yf(\mathbf{x}))$$

- Logistic Loss:

$$L(f, \mathbf{x}, y) = \log(1 + \exp(-yf(\mathbf{x})))$$

Loss Functions



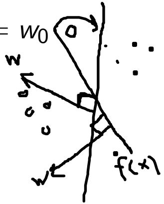
Discriminant Functions

- One of the simplest representation for a 2-class problem

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

- Class assignment based on $\text{sign}(f(\mathbf{x}))$
  - If $f(\mathbf{x}) \geq 0$, $\text{sign}(f(\mathbf{x})) = +1$, then $\mathbf{x} \in C_1$, otherwise $\mathbf{x} \in C_2$
- $\mathbf{w}$ is orthogonal to the decision boundary $f(\mathbf{x}) = w_0$
- With $\tilde{\mathbf{w}} = [\mathbf{w}, w_0]$ and $\tilde{\mathbf{x}} = (\mathbf{x}, 1)$, we have

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



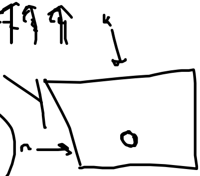
- At times, we will ignore the offset term w_0 w.l.o.g. (without loss of generality)

Least Squares for Classification

D = # features

- Consider a training dataset $\{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N}$ for a K-class problem
  - $\mathbf{y}_n$ encodes the class membership, say $\mathbf{y}_n^T = (0, 1, 0, 0)$
  - Y : $N \times K$ matrix with rows $\mathbf{y}_n^T$
  - X : $N \times D$ matrix with rows $\mathbf{x}_n^T$
  - W : $D \times K$ matrix with columns $\mathbf{w}_k$
  - Goal:

$$\mathbf{w}_k^T \mathbf{x}_n = \mathbf{x}_n^T \mathbf{w}_k = X_{n,:} \mathbf{w}_k \approx Y_{nk}$$



- The sum-of-squares error to be minimized over W is

$$E(W) = \frac{1}{2} \sum_{k=1}^{K} \sum_{n=1}^{N} \| Y_{nk} - X_{n,:} \mathbf{w}_k \|^2 = \frac{1}{2} \operatorname{Tr} \{ (Y - XW)^T (Y - XW) \}$$

$\operatorname{Tr}(\| Y - XW \|^2)$

Least Squares for Classification (Contd.)

- The sum-of-squares error to be minimized over W is

$$\underset{W}{\operatorname{argmin}}\ E(W) = \frac{1}{2}\sum_{k=1}^{K}\sum_{n=1}^{N}\|Y_{nk} - X_{n,:}\mathbf{w}_k\|^2 = \frac{1}{2}\operatorname{Tr}\left\{(Y - XW)^T(Y - XW)\right\}$$

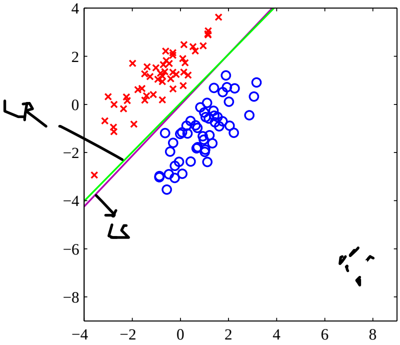
- The problem has a closed form solution

$$W^* = \underbrace{(X^TX)^{-1}X^T}_{X^\dagger}Y = X^\dagger Y$$

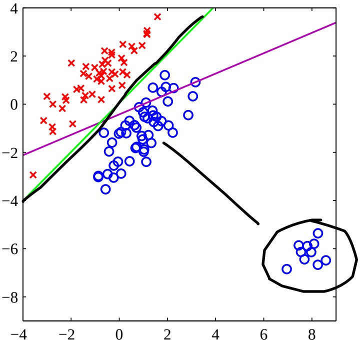
- Solving each problem separately: $\mathbf{w}_k = X^\dagger \mathbf{y}_k$
- The discriminant function has the following form

$$f(\mathbf{x}) = W^T\mathbf{x} = Y^T(X^\dagger)^T\mathbf{x}$$

Least Squares is Noise Sensitive

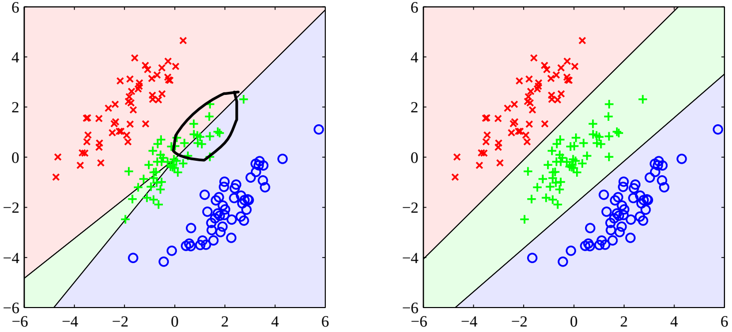


Least Squares vs Logistic



Least Squares with noise

Least Squares for Multiclass Problems



Least Squares Logistic Regression

Classification by Projection

- Classify after dimensionality reduction
  - Project D dimensional data $\mathbf{x}$ to 1 dimensions: $\mathbf{w}^T\mathbf{x}$
  - Make sure class separation is maximized
- If $\mathbf{m}_1, \mathbf{m}_2$ are the means of the two classes

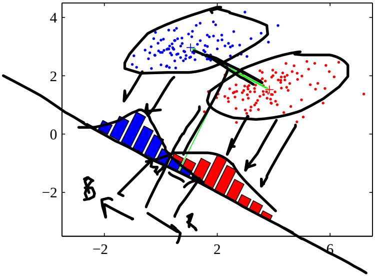
$$\max_{\|\mathbf{w}\|^2=1} \mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1)$$

- Performing the optimization (using 'Langrange multipliers')

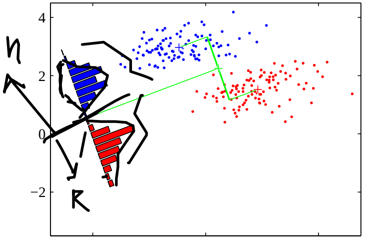
$$\mathbf{w} \propto (\mathbf{m}_2 - \mathbf{m}_1)$$

- May be problematic if data has non-diagonal covariance

Classification by Projection (Contd.)



Classification by Projection



Fisher's Linear Discriminant

Fisher's Linear Discriminant

- Desirable to have low within class variance

$$\sigma_k^2 = \sum_{\mathbf{x}_n \in C_k} \| w^T(\mathbf{x}_n - \mathbf{m}_k)\|^2$$

- Between-class and within-class covariance matrices

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

$$S_w = \sum_{\mathbf{x}_n \in C_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{\mathbf{x}_n \in C_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T$$

- Fisher's criterion: Ratio of between-class and within-class variance

$$J(\mathbf{w}) = \frac{\|\mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1)\|^2}{\sigma_1^2 + \sigma_2^2} = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_w \mathbf{w}}$$

Large

small

Fisher's Linear Discriminant (Contd.)

- Fisher's criterion is

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_w \mathbf{w}}$$

- A 'direct calculation' gives

$$\mathbf{w} \propto S_w^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$$

- A linear discriminant can be constructed using **w**
  - Construct the projected version of the data $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$
  - Choose a threshold w_0 to form linear discriminant $f(\mathbf{x}) \geq w_0$
- Extension to multiclass: Project to $(K-1)$ dimensions
- Need to train a classifier in the low dimensional representation