Instructional Objective

- What is classification?
- How is it different from regression?
- What is Bayes Rule for classification?
- Linear discrimination analysis (LDA)
- Logistic regression (LR)
  - What are LDA and LR?
  - How are they used for classification?
  - How to learn classifiers from given data?

An Overview of Classification

Some examples:

- A person arrives at an emergency room with a set of symptoms that could be infected by Nipah or influenza. Which one is it?
- Is a received email in your mailbox spam or not ?
- Given a set of sequenced DNA, can we determine whether various mutations are associated with different phenotypes?
- How will tomorrow's weather be: sunny, rainy or cloudy ?

All of these problems are not regression problems. They are classification problems.

The Running Example

Figure: Iris versicolour, iris setosa and iris verginica

The Running Example

Figure: Iris versicolour, iris setosa and iris verginica

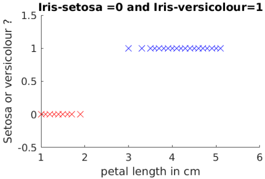
- Iris flower data set
- Introduced by the British statistician and biologist Ronald Fisher in his 1936 paper

The Set-up

It begins just like regression: suppose we have observations

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$$

Feature vector : $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ and
Class lable: $y_i \in \mathcal{Y} = \{C_1, C_2, \ldots, C_K\}$ (For binary classification $\mathcal{Y} = \{0, 1\}$)



Classifier $g(\cdot) : \mathcal{X} \to \mathcal{Y}$ **can be defined as** $g(\mathbf{x}) = 1$ **if** $\mathbf{x}_i$ **¿ 2.5.**

The same constraints apply:

- We want a classifier that predicts test data, not just the training data.

How do we measure quality?

Let the classifier g make predictions $\hat{y}$ based on $\mathcal{D}$
Our loss function is now a $K \times K$ matrix L with

Predicted class lables

True class lables

	C_1	C_2	...	C_k
C_1	0	$l(C_1, C_2)$		$l(C_1, C_k)$
C_2	$l(C_2, C_1)$	0		$l(C_2, C_k)$
...			...	
C_k	$l(C_k, C_1)$	$l(C_k, C_2)$		0

- $\ell(c, c')$ is the price paid for classifying an observation belonging to class $y = c$ as $\hat{y} = c'$.

Expected prediction error

$$\begin{aligned} \text{Risk} \quad R(g) = EPE \quad &= \quad \mathbb{E}_{(X,Y)}[\ell_g(Y, \hat{Y})] \text{ (where } \hat{Y} = g(X)) \\ &= \quad \mathbb{E}_X \sum_{k=1}^{K} \ell_g(C_k, \hat{Y}) \mathbb{P}(Y = C_k | X) \end{aligned}$$

This can be minimized point wise over X, to produce

$$g^*(\mathbf{x}) = \underset{\hat{y} \in \mathcal{Y}}{\text{argmin}} \sum_{k=1}^{K} \ell_g(C_k, \hat{y}) \mathbb{P}(Y = C_k | X = \mathbf{x})$$

(This is the Bayes' classifier. Also, $R(g^*)$ is the Bayes' limit)

Best classifier

If we make specific choices for ℓ, we can find g^* exactly

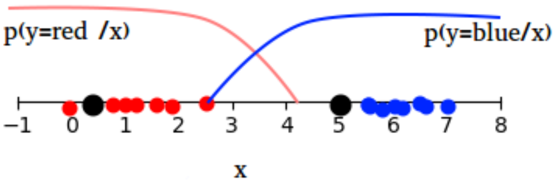
As Y takes only a few values, zero-one prediction risk is natural

$$\begin{aligned} \ell_g(Y, \hat{Y}) &= \mathbf{1}_{Y \neq \hat{Y}}(Y, \hat{Y}) \\ \Rightarrow R(g) &= \mathbb{E}[\ell_g(Y, \hat{Y})] = \mathbb{P}(g(X) \neq Y), \end{aligned} \tag{1}$$

Under this loss, we have

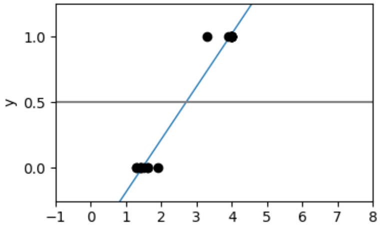
$$g^*(\mathbf{x}) = \underset{\hat{y} \in \mathcal{Y}}{\operatorname{argmin}} \left[1 - \mathbb{P}(Y = \hat{y} | X = \mathbf{x})\right] = \underset{\hat{y} \in \mathcal{Y}}{\arg\max}\, \mathbb{P}(Y = \hat{y} | X = \mathbf{x})$$

Best classifier



Does linear regression work?

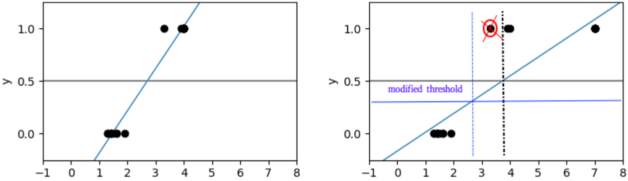
Suppose $Y = \{0, 1\}$
Let $\hat{f}$ be any estimate of linear regression problem



Classifier is : $\hat{g}(X) = \mathbf{1}_{(\hat{f}(X)>1/2)}$

Can we consider classification as regression?

Let $\hat{f}$ be any estimate of linear regression problem



Classifier is for the first sample : $\hat{g}(X) = \mathbf{1}(\hat{f}(X) > 1/2)$
Classifier is for the second sample : $\hat{g}(X) = \mathbf{1}(\hat{f}(X) > 1/3)$

Cons of linear regression:

- Y is continuous, with normally distributed error.
- $\hat{f} > 1$ and $\hat{f} < 0$

(When $\hat{f}$ estimates probability $\mathbb{P}(Y = 1|X)$?)

Bayes' rule and class densities

Suppose $Y = \{0, 1\}$, using Bayes' theorem

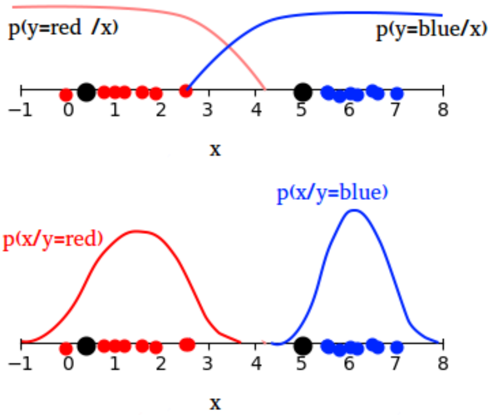
$$f^*(X) = \mathbb{P}(Y = 1|X) = \frac{p(X|Y = 1)\mathbb{P}(Y = 1)}{\sum_{y \in \{0,1\}} p(X|Y = y)\mathbb{P}(Y = y)} = \frac{p_1(X)\pi_1}{p_1(X)\pi_1 + p_0(X)\pi_0}$$

- $p_y(X) = \mathbb{P}(X|Y = y)$ is the class densities i.e., the likelihood of the covariates given the class labels
- $\pi_y = \mathbb{P}(Y = y)$ is the prior

The Bayes' rule can be rewritten

$$g^*(X) = \begin{cases} 1 & \text{if } \frac{p_1(X)}{p_0(X)} > \frac{\pi_0}{\pi_1} \text{ or } \frac{\mathbb{P}(Y=1|X)}{1-\mathbb{P}(Y=1|X)} > 1 \\ 0 & \text{otherwise} \end{cases}$$

Bayes' rule and class densities



P(y=red)=0.5 P(y=blue) =0.5

How to find a classifier

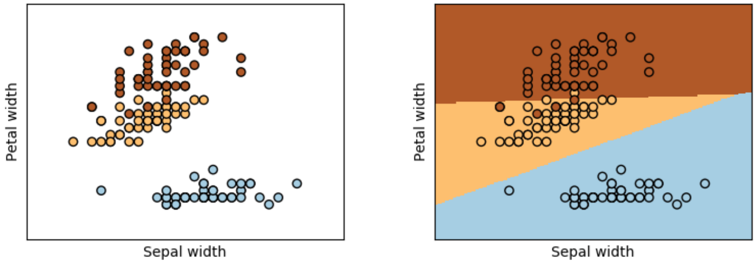
All of these prior expressions for g^* give rise to classifiers

- Density estimation: Estimate $\hat{\pi}_y$ and p_y from $\mathcal{D}$
- Regression: Find an estimate of $f^*(X)$ and plug it in to the Bayes' rule
- Empirical risk minimization: Choose a set of classifiers Γ and find $\hat{g} \in \Gamma$ that minimizes some estimate of $R(g)$

  (This can be quite challenging as, unlike in regression, the training error is nonconvex)

Linear classifiers

Linear classifier



The boundaries between these regions are known as decision boundaries These decision boundaries are sets of points at

which $\hat{g}$ is indifferent between two (or more) classes

A linear classifier is a $\hat{g}$ that produces linear decision boundaries

Bayes' rule-ian approach

The decision theory for classification indicates we need to know the posterior probabilities: $\mathbb{P}(Y = y|X)$ for doing optimal classification

Suppose that

- $p_y(X) = p(X|Y = y)$ is the class densities i.e., the likelihood of the covariates given the class labels
- $\pi_y = \mathbb{P}(Y = y)$ is the prior

Then

$$\mathbb{P}(Y = y|X) = \frac{p_y(X)\pi_y}{\sum_{y \in \mathcal{Y}} p_y(X)\pi_y} \propto p_y(X)\pi_y$$

Conclusion: Having the class densities almost gives us the Bayes' rule as the training proportions can usually be used to estimate π_y

Bayes' rule-ian approach: Summary

There are many techniques based on this idea

- Linear discriminant analysis

  (Estimates p_y assuming multivariate Gaussianity)
- General nonparametric density estimators
- Naive Bayes (Factors p_u assuming conditional independence)

Discriminant analysis

Suppose that

$$p_y(X) \propto |\Sigma_y|^{\frac{-1}{2}} e^{\frac{-(X-\mu_y)^\top \Sigma_y^{-1}(X-\mu_y)}{2}}$$

Let's assume that $\Sigma_y \equiv \Sigma$.

Then the log-odds between two classes y, y' is:

$$\begin{aligned}
\log\left(\frac{\mathbb{P}(Y=y|X)}{\mathbb{P}(Y=y'|X)}\right) &= \log \frac{p_y(X)}{p_{y'}(X)} + \log \frac{\pi_y}{\pi_{y'}} \\
&= \log \frac{\pi_y}{\pi_{y'}} - \frac{(\mu_y + \mu_{y'})^\top \Sigma^{-1}(\mu_y - \mu_{y'})}{2} \\
&\quad + X^\top \Sigma^{-1}(\mu_y - \mu_{y'})
\end{aligned}$$

This is linear in X, and hence has a linear decision boundary

Linear discriminant analysis

$$\hat{g}(\mathbf{x}) = \begin{cases} y & \text{if } \log\left(\frac{\mathbb{P}(Y=y|X=\mathbf{x})}{\mathbb{P}(Y=y'|X=\mathbf{x})}\right) > 0 \\ y' & \text{otherwise} \end{cases}$$

$$\hat{g}(\mathbf{x}) = \begin{cases} y & \text{if } \delta_y(\mathbf{x}) > \delta_{y'}(\mathbf{x}) \\ y' & \text{otherwise} \end{cases}$$

The linear discriminant function is :

$$\delta_y(\mathbf{x}) = \log \pi_y + \mathbf{x}^\top \Sigma^{-1} \mu_y - \frac{\mu_y^\top \Sigma^{-1} \mu_y}{2}$$

For mutli-class data

$$\hat{g}(\mathbf{x}) = \operatorname*{argmin}_y \delta_y(X)$$

(This is just minimum Euclidean distance, weighted by the covariance matrix and prior probabilities)

Linear/regularized discriminant analysis

Now, we must estimate μ_y and Σ. If we...

- use the intuitive estimators $\hat{\mu}_y = \overline{X}_y$ and

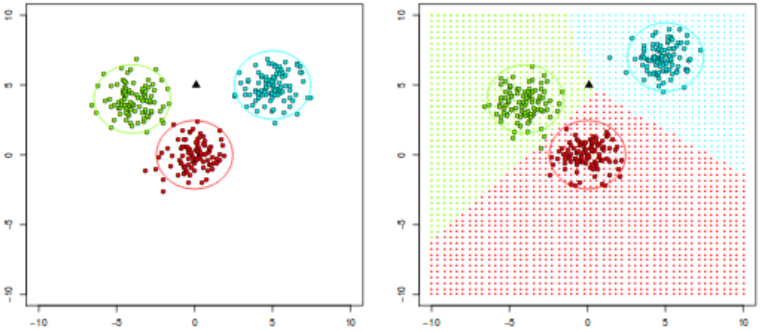
$$\hat{\Sigma} = \frac{1}{n-K} \sum_{y=0}^{K-1} \sum_{i:y_i=y} (\mathbf{x}_i - \hat{\mu}_y)(\mathbf{x}_i - \hat{\mu}_y)^\top$$

  the we have produced linear discriminant analysis (LDA)
- regularize these 'plug-in' estimates, we can form regularized discriminant analysis (Friedman (1989)). This could be (for $\lambda \in [0,1]$):

$$\hat{\Sigma}_\lambda = \lambda\hat{\Sigma} + (1-\lambda)\hat{\sigma}^2 I$$

LDA INTUITION

How would you classify a point with this data?



We can just classify an observation to the closest mean ($\overline{X}_y$)

What do we mean by close? (Need to define distance)

LDA intuition

Intuitively, assigning observations to the nearest $\overline{X}_g$ (but ignoring the covariance) would amount to

$$
\begin{aligned}
\tilde{g}(\mathbf{x}) &= \underset{y}{\operatorname{argmin}} \, ||\mathbf{x} - \mu_y||_2^2 \\
&= \underset{y}{\operatorname{argmin}} \, \mathbf{x}^\top \mathbf{x} - 2\mathbf{x}^\top \mu_y + \mu_y^\top \mu_y \\
&= \underset{g}{\operatorname{argmin}} \, -\mathbf{x}^\top \mu_y + \frac{1}{2}\mu_y^\top \mu_y
\end{aligned}
$$

compare this to:

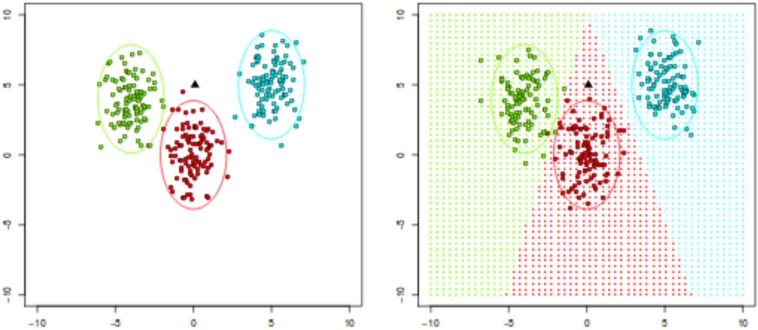
$$
\hat{g} = \underset{g}{\operatorname{argmin}} \, \underbrace{\mathbf{x}^\top \hat{\Sigma}_\lambda^{-1} \mu_y - \frac{1}{2}\mu_y^\top \hat{\Sigma}_\lambda^{-1} \mu_y}_{\textit{likelihood}} + \underbrace{\log(\hat{\pi}_y)}_{\textit{prior}}
$$

The difference is we weigh the distance by $\hat{\Sigma}_\lambda^{-1}$ and weigh the class assignment by fraction of observations in each class.

(Note: this generalization of Euclidean distance is called Mahalanobis distance)

Intuition

What if the data looked like this?



Performance of LDA

The quality of the classifier produced by LDA depends on two things:

- The sample size n

  (This determines how accurate the $\hat{\pi}_y$, $\hat{\mu}_y$, and $\hat{\Sigma}$ are)

- How wrong the LDA assumptions are

  (That is: $X|Y = g$ is a Gaussian with mean μ_y and variance Σ)

LDA: under correct assumptions

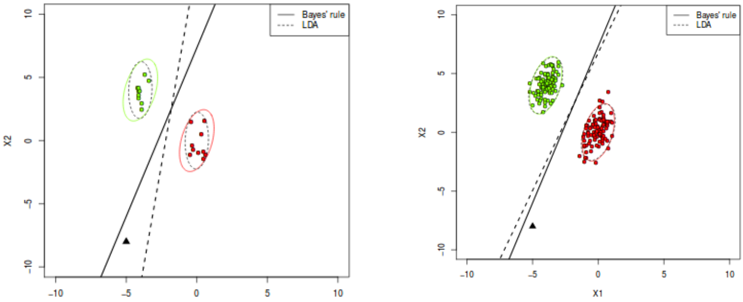


Figure: For $n = 20$ and $n = 200$

LDA: UNDER INCORRECT ASSUMPTIONS

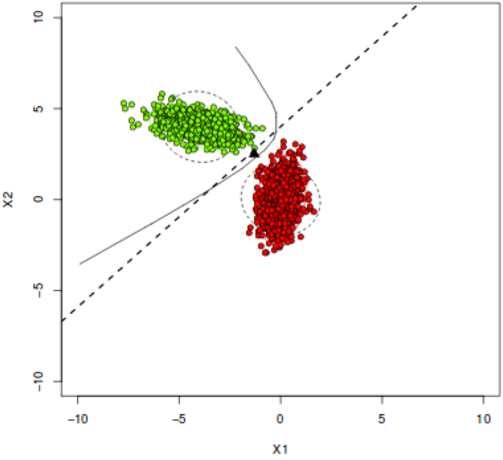


FIGURE: For $n = 200$

Bayes' rule-ian approach: Summary

There are many techniques based on this idea

- Linear discriminant analysis

  (Estimates p_y assuming multivariate Gaussianity)
- Naive Bayes (Factors p_y assuming conditional independence)
- General nonparametric density estimators

Naive Bayes

When to use

- Moderate or large training set available
- Features that describe instances are conditionally independent given class label

Successful applications:

- Diagnosis
- Classifying text documents

Naive Bayes assumption:

$$p_y(\mathbf{x}) = \Pi_{d=1}^{D} p(\mathbf{x}_{id}/Y = y)$$

Training: Estimate $p(\mathbf{x}_{id}/Y = y)$

Classify_New_Instance($\mathbf{x}_t$)

$$\hat{g}(\mathbf{x}_t)_{NB} = \arg\max_{y\in} P(Y = y) \prod_{d=1}^{D} p(\mathbf{x}_{td}|Y = y)$$

Bayes' rule-ian approach: Summary

There are many techniques based on this idea

- Linear discriminant analysis

  (Estimates p_y assuming multivariate Gaussianity)
- Naive Bayes (Factors p_y assuming conditional independence)
- General nonparametric density estimators

Nonparametric Methods

- What if parametric form of distribution is wrong?
- For example, most real-world entities have multimodal distributions whereas all classical parametric densities are unimodal.
- We will examine nonparametric procedures that can be used with arbitrary distributions and without the assumption that the underlying form of the densities are known.
  - Histograms.
  - Kernel Density Estimation / Parzen Windows.
  - k-Nearest Neighbor Density Estimation.

Kernel Density Estimation

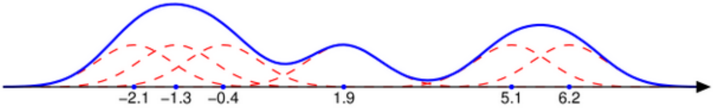
If $\{\mathbf{x}_1, x2, ..., xn\}$ is an independent and identically-distributed sample of a random variable, then the kernel density approximation of its probability density function is

$$p_h(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} K(\frac{\mathbf{x}-\mathbf{x}_i}{h}),$$

where K is some kernel and h is a smoothing parameter called the bandwidth.
Quite often K is taken to be a standard Gaussian function with mean zero and variance 1. Thus the variance is controlled indirectly through the parameter h:

$$K(\frac{\mathbf{x}-\mathbf{x}_i}{h}) = \frac{1}{2\sqrt{\pi}} e^{-\frac{(\mathbf{x}-\mathbf{x}_i)^2}{2h^2}}.$$

Kernel Density Estimation



Parzen Window and K-NN

Estimate the probability density to be

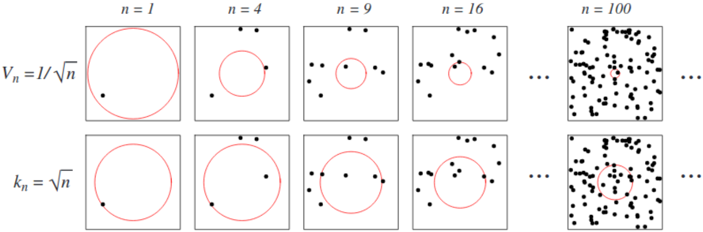
$$\hat{p}_n(\mathbf{x}) = \frac{K_n}{nV_n},$$

where,

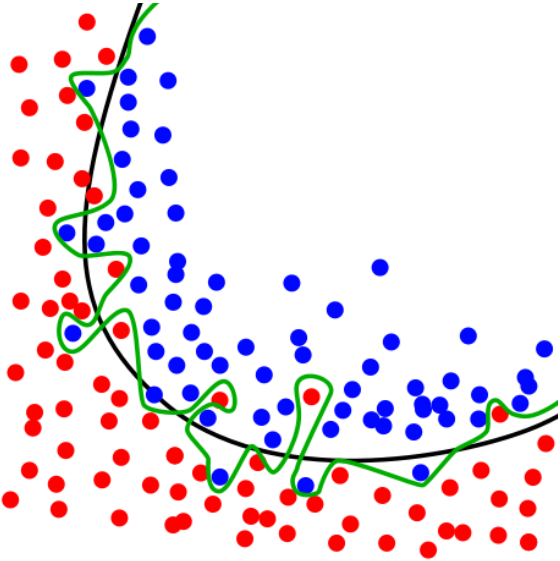
- K_n: number of sample in window.
- V_n: size of window

- K-NN : $K_n = \sqrt{n}$
- Parzenwindow : $V_n = \frac{1}{\sqrt{n}}$

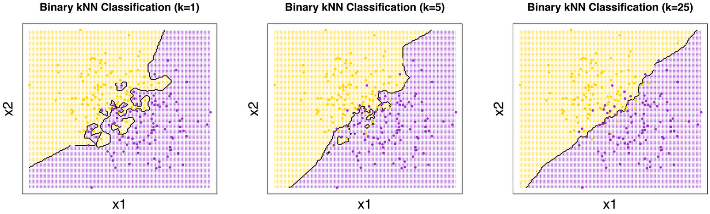
Parzen Window and K-NN



What should be K ?



What should be K ?



How to find a classifier

All of these prior expressions for g^* give rise to classifiers

- Density estimation: Estimate $\hat{\pi}_y$ and p_y from $\mathcal{D}$
- Regression: Find an estimate of $f^*(X)$ and plug it in to the Bayes' rule
- Empirical risk minimization: Choose a set of classifiers Γ and find $\hat{g} \in \Gamma$ that minimizes some estimate of $R(g)$

  (This can be quite challenging as, unlike in regression, the training error is nonconvex)

Recall: Bayes' rule and class densities

Suppose $Y = \{0, 1\}$, using Bayes' theorem

$$f^*(X) = \mathbb{P}(Y = 1|X) = \frac{p(X|Y = 1)\mathbb{P}(Y = 1)}{\sum_{y\in\{0,1\}} p(X|Y = y)\mathbb{P}(Y = y)}$$

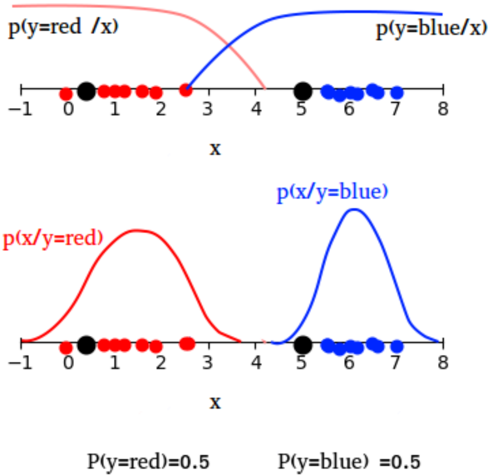
$$= \frac{p_1(X)\pi_1}{p_1(X)\pi_1 + p_0(X)\pi_0}$$

- $p_y(X) = \mathbb{P}(X|Y = y)$ is the class densities i.e., the likelihood of the covariates given the class labels
- $\pi_y = \mathbb{P}(Y = y)$ is the prior

The Bayes' rule can be rewritten

$$g^*(X) = \begin{cases} 1 & \text{if } \frac{p_1(X)}{p_0(X)} > \frac{\pi_0}{\pi_1} \text{ or } \frac{\mathbb{P}(Y=1|X)}{1-\mathbb{P}(Y=1|X)} > 1 \\ 0 & \text{otherwise} \end{cases}$$

Recall: Bayes' rule and class densities

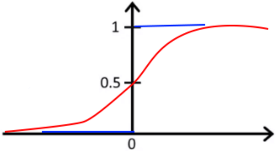


Linear classifier: Logistic Regression

Suppose $Y = \{0, 1\}$ and want

$$f(X) = \mathbb{P}(Y = 1|X)$$

$$1 \geq \hat{f}_{\mathbf{w}}(\mathbf{x}) \geq 0$$



Sigmoid function $h(z) = \frac{1}{1+\exp\{-z\}}$

Let $\hat{f}_{\mathbf{w}}(\mathbf{x}) = h(\mathbf{w}_0 + \mathbf{w}^\top \mathbf{x})$, the posterior probabilities are

$$\mathbb{P}(Y = 1|X = \mathbf{x}) = f_{\mathbf{w}}(\mathbf{x}) = \frac{\exp\{\mathbf{w}_0 + \mathbf{w}^\top \mathbf{x}\}}{1 + \exp\{\mathbf{w}_0 + \mathbf{w}^\top \mathbf{x}\}}$$

$$\mathbb{P}(Y = 0|X = \mathbf{x}) = 1 - f_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1 + \exp\{\mathbf{w}_0 + \mathbf{w}^\top \mathbf{x}\}}$$

Linear classifier: Logistic Regression

Bayes' rule-ian approach

$$g^*(\mathbf{x}) = \begin{cases} 1 & \text{if } \frac{\mathbb{P}(Y=1|X=\mathbf{x})}{\mathbb{P}(Y=0|X=\mathbf{x})} > 1 \\ 0 & \text{otherwise} \end{cases}$$

The log odds ratio (i.e.: logit) transformation forms a linear decision boundary

$$\log\left(\frac{\mathbb{P}(Y=1|X=\mathbf{x})}{\mathbb{P}(Y=0|X=\mathbf{x})}\right) = \mathbf{w}_0 + \mathbf{w}^\top\mathbf{x} > 0$$

The decision boundary is the hyperplane $\{\mathbf{x} : \mathbf{w}_0 + \mathbf{w}^\top\mathbf{x} = 0\}$

Maximum Likelihood Estimation

Assume,

$$\mathbb{P}(Y = 1 | X = \mathbf{x}) = f_{\mathbf{w}}(\mathbf{x})$$
$$\mathbb{P}(Y = 0 | X = \mathbf{x}) = 1 - f_{\mathbf{w}}(\mathbf{x})$$

write this more compactly as

$$\mathbb{P}(Y = y | X = \mathbf{x}) = f_{\mathbf{w}}(\mathbf{x})^y (1 - f_{\mathbf{w}}(\mathbf{x}))^{(1-y)}$$

Then the likelihood (assuming data independence) is

$$\mathbb{P}(Y | X, \mathcal{D}) \sim \Pi_{i=1}^{n} f_{\mathbf{w}}(\mathbf{x}_i)^{y_i} (1 - f_{\mathbf{w}}(\mathbf{x}_i))^{(1-y_i)}$$

And the negative log likelihood is

$$L(\mathbf{w}) = \sum_{i=1}^{N} [y_i \log(f_{\mathbf{w}}(\mathbf{x}_i)) + (1 - y_i) \log(1 - f_{\mathbf{w}}(\mathbf{x}_i))]$$

Logistic regression loss function

Using $Y = \{-1, 1\}$,

$$\mathbb{P}(Y = 1|X = \mathbf{x}) = f_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1 + \exp\{-(\mathbf{w}_0 + \mathbf{w}^\top \mathbf{x})\}}$$

$$\mathbb{P}(Y = -1|X = \mathbf{x}) = 1 - f_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1 + \exp\{+(\mathbf{w}_0 + \mathbf{w}^\top \mathbf{x})\}}$$

write this more compactly as

$$\mathbb{P}(Y = y|X = \mathbf{x}) = \frac{1}{1 + \exp\{-y(\mathbf{w}_0 + \mathbf{w}^\top \mathbf{x})\}}$$

Loss function: the negative log likelihood

$$L(\mathbf{w}) = \sum_{i=1}^{n} log(1 + \exp\{-y_i(\mathbf{w}_0 + \mathbf{w}^\top \mathbf{x}_i)\})$$

Regularized Logistic regression

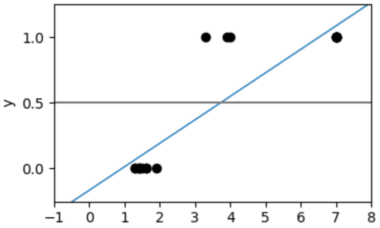
Using $Y = \{-1, 1\}$, Loss function: the negative log likelihood

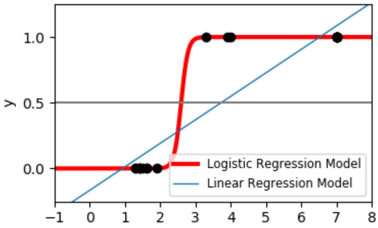
$$L(\mathbf{w}) = \sum_{i=1}^{n} log(1 + \exp\{-y_i(\mathbf{w}_0 + \mathbf{w}^\top \mathbf{x}_i)\})$$

with regularization

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \sum_{i=1}^{n} log(1 + \exp\{-y_i(\mathbf{w}_0 + \mathbf{w}^\top \mathbf{x}_i)\}) + \frac{\|\mathbf{w}\|_*}{C}$$

Logistic Regression vs Linear Regression





Logistic Regression for K classes

Likewise, multi class logistic follows (for $y = 1, \dots, K-1$):

$$\log \frac{\mathbb{P}(Y = y | X = \mathbf{x})}{\mathbb{P}(Y = 0 | X = \mathbf{x})} = \mathbf{w}_{y,0} + \mathbf{w}_y{}^\top \mathbf{x}$$

(The choice of base class K is arbitrary)

The posterior probabilities are

$$\mathbb{P}(Y = y | X = \mathbf{x}) = f_{\mathbf{w}_y}(\mathbf{x}) = \frac{\exp\{\mathbf{w}_{y,0} + \mathbf{w}_y^\top \mathbf{x}\}}{1 + \sum_{k=1}^{K-1} \exp\{\mathbf{w}_{k,0} + \mathbf{w}_k^\top \mathbf{x}\}}$$

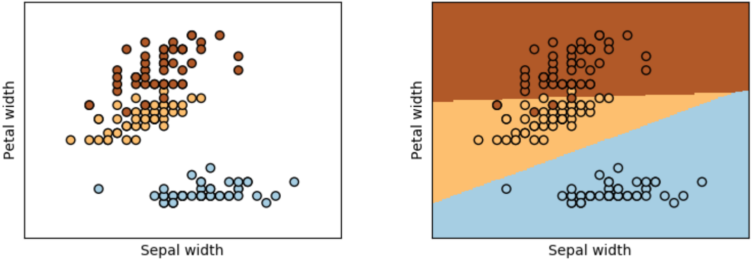
$$\mathbb{P}(Y = 0 | X = \mathbf{x}) = f_{\mathbf{w}_0}(\mathbf{x}) = 1 - \sum_{k=1}^{K-1} f_{\mathbf{w}_k}(\mathbf{x})$$

$$= \frac{1}{1 + \sum_{k=1}^{K-1} \exp\{\mathbf{w}_{k,0} + \mathbf{w}_k^\top \mathbf{x}\}}$$

Logistic Regression on Iris data set

Classifier

$$g^*(\mathbf{x}) = \underset{y \in \{0,1,\ldots,K-1\}}{\arg\max} f_{\mathbf{w}_y}(\mathbf{x})$$



Logistic Regression Vs LDA

The log posterior odds via the Gaussian likelihood (LDA) for class y versus 0 are

$$\log \frac{\mathbb{P}(Y = y|X = \mathbf{x})}{\mathbb{P}(Y = 0|X = \mathbf{x})} = \log \frac{\pi_y}{\pi_0} - (\mu_y + \mu_0)^\top \Sigma^{-1} (\mu_y - \mu_0)/2 + \mathbf{x}^\top \Sigma^{-1} (\mu_y - \mu_0)$$

$$= \alpha_{y,0} + {\alpha_y}^\top \mathbf{x}$$

Likewise, multi class logistic follows (for $y = 1, \ldots, K-1$):

$$\log \frac{\mathbb{P}(Y = y|X = \mathbf{x})}{\mathbb{P}(Y = 0|X = \mathbf{x})} = \mathbf{w}_{y,0} + {\mathbf{w}_y}^\top \mathbf{x}$$

(The choice of base class 0 is arbitrary)

THEY BOTH SPECIFY THE LOG-ODDS AS LINEAR MODELS!

Logistic Regression versus LDA

We can write the joint distribution of Y and X as

$$\mathbb{P}(X, Y) = \mathbb{P}(Y|X)\mathbb{P}(X)$$

The previous slide shows that $\mathbb{P}(Y|X)$ is the same for both methods:

- Logistic regression leaves $\mathbb{P}(X)$ arbitrary, and implicitly estimates it with the empirical measure

  (This could be interpreted as a frequentist approach, where we are maximizing the likelihood only and using the improper uniform prior)

- LDA models

$$\mathbb{P}(X, Y = y) = \mathbb{P}(X|Y = y)\mathbb{P}(Y = y) = N(X; \mu_y, \Sigma)\pi_y$$

Logistic regression versus LDA

Some remarks:

- Forming logistic regression requires fewer assumptions
- If some entries in X are qualitative, then the modeling assumptions behind LDA are suspect
- In practice, the two methods tend to give very similar results

Logistic regression versus LDA on Iris

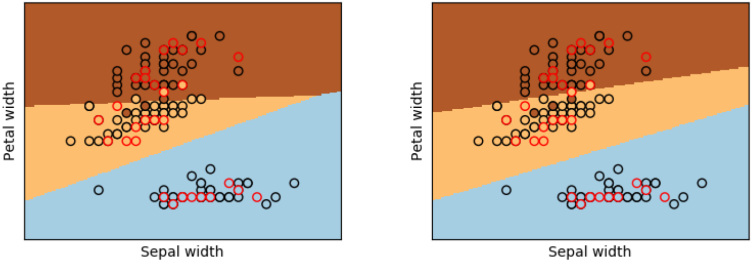


Figure: Decision Boundary by LR and LDA. Test accuracy for both the methods is 93.33%.

Take away Message

- For classification, $\mathcal{Y}$ is set as discrete value
- Best classifier by Bayes Rule
- Logistic regression estimate posterior probability without any other assumption
- LDA assumes multivariate Gaussianity for class density

- Question ?
- sahely@iitpkd.ac.in
- Thanks you !