Likelihood and Naïve Bayes

Predicting amount of rainfall

https://esan108.com/%E0%B8%9E%E0%B8%A3%E0%B8%B0%E0%B9%82%E0%B8%84%E0%B8%81%E0%B8%B4%E0%B8%99%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B8%AB%E0%B8%A1%E0%B8%B2%E0%B8%A2%E0%B8%96%E0%B8%B6%E0%B8%87.html

(Linear) Regression

- $h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5$
- θs are the parameter (or weights)

Assume x_0 is always 1

- We can rewrite

$$h_\theta(x) = \Sigma_{i=0}^{n} \theta_i x_i = \theta^T \mathbf{x}$$

- Notation: vectors are bolded
- Notation: vectors are column vectors

LMS regression with gradient descent

$$\frac{\partial J}{\partial \theta_j} = -\Sigma_{i=1}^{m}(y_i - \theta^T \mathbf{x}_i)x_i^{(j)}$$

$$\theta_j \Leftarrow \theta_j + r\Sigma_{i=1}^{m}(y_i - \theta^T \mathbf{x}_i)x_i^{(j)}$$

Interpretation?

Logistic Regression

- Pass $\theta^T \mathbf{x}$ through the logistic function

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Logistic Regression update rule

$$\theta_j \Leftarrow \theta_j - r\Sigma_{i=1}^{m}(y_i - h_\theta(x_i))x_i^{(j)}$$

Update rule for linear regression

$$\theta_j \Leftarrow \theta_j - r\Sigma_{i=1}^{m}(y_i - \theta^T\mathbf{x}_i)x_i^{(j)}$$

Overview

- Probabilistic view of linear regression
  - Solution for logistic regression
- Homework 1 notes
  - Overfitting vs Underfitting (Bias – variance trade-off)
- Bayes decision models
  - Parameter estimation
    - MLE, MAP
  - Naïve Bayes

Distribution parameter estimation

- $P(head) = \theta$, θ = #heads/#tosses
- HHTTH

- $L(\theta) = P(X; \theta) = P(HHTTH; \theta)$
- Maximum Likelihood Estimate (MLE)
    - Likelihood - Probability of encountering the data X given the parameters θ

Linear Regression Revisit

- $h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5$
- θs are the parameter (or weights)
- We can rewrite

$$h_\theta(x) = \Sigma_{i=0}^{n} \theta_i x_i = \theta^T \mathbf{x}$$

- Notation: vectors are bolded
- Notation: vectors are column vectors

Probabilistic Interpretation of linear regression

- Real world data is our model plus some error term
  - Noise in the data
  - Something that we do not model (features we are missing)

- Let's assume the error is normally distributed with mean zero and variance σ^2
  - Why Gaussian?
  - Why saying mean is zero is a valid assumption?

$$y_i = \theta^T \mathbf{x}_i + \epsilon_i$$

Probabilistic view of Linear regression

- Find θ
- Maximize Likelihood of seeing x and y in training

- From our assumption we know that

$$y_i = \theta^T \mathbf{x}_i + \epsilon_i$$

error

$$p(y_i | \mathbf{x}_i; \theta) = \frac{1}{\sqrt{2\pi\sigma}} exp(-\frac{\boxed{(y_i - \theta^T \mathbf{x}_i)}^2}{2\sigma^2})$$

Error term is normally distributed with mean 0 and variance σ^2

Maximizing Likelihood

What is the assumption here?
Is it accurate?

- Max $L(\theta) = \Pi_{i=1}^{m} p(y_i|\mathbf{x}_i;\theta)$
  We use the log likelihood instead log(L(θ)) = l(θ)

From our previous lecture

Min $J(\theta) = \frac{1}{m}\Sigma_{i=1}^{m}(y_i - \theta^T\mathbf{x_i})^2$

Mean square error solution and MLE solution

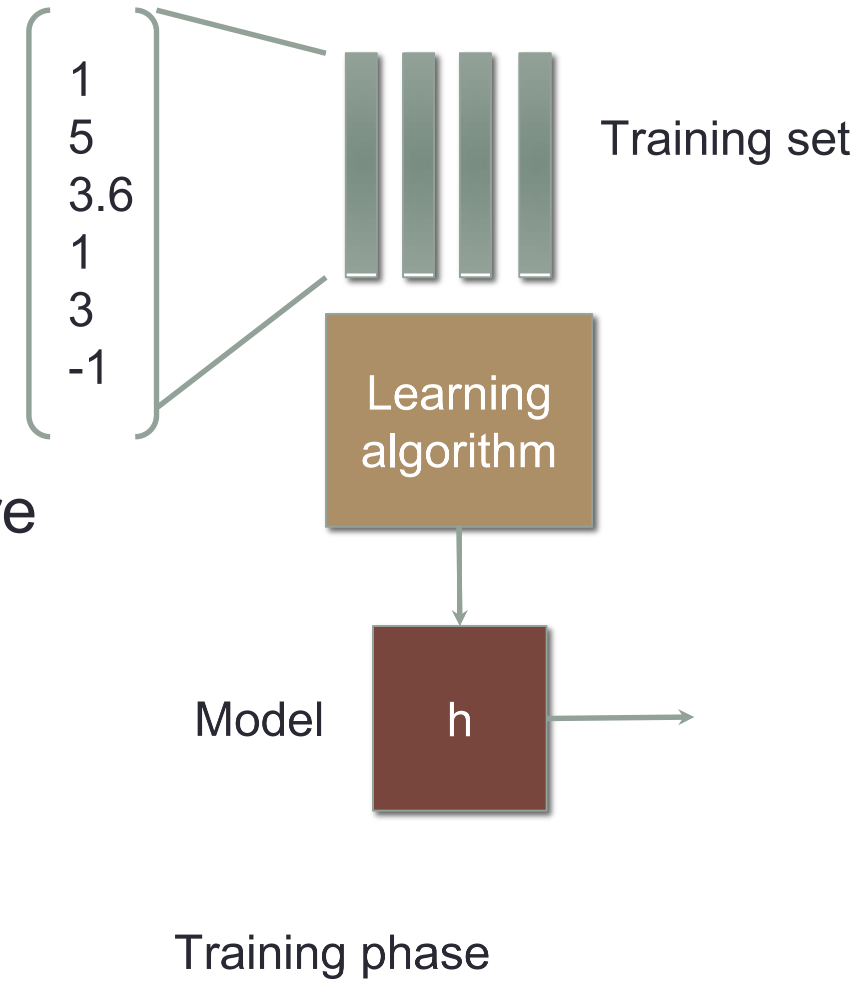
- Turns out MLE and MSE gets to the same solution
  - This justifies our choice of MSE as the Loss for linear regression
  - This does not mean MSE is the best Loss for regression, but you can at least justify it under this probabilistic reasoning and assumptions
- Note how our choice of variance σ^2 falls out of the maximization, so this derivation is true regardless of which assumption for variance is.
- Note that MLE derivation assumes that the error is normally distributed! This is a key assumption for linear regression.
  - Error is normally distributed is not that same as y is normally distributed.

Flood or no flood

- What would be the output?

- y = 0 if not flooded
- y = 1 if flooded

- Anything in between is a score for how likely it is to flood

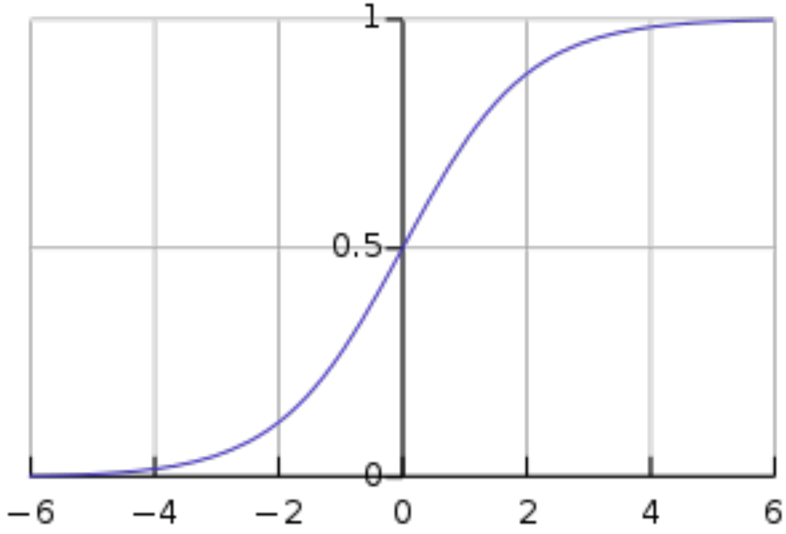


Can we use regression?

- Yes
- $h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5$
- But
- What does it mean when h is higher than 1?
- Can h be negative? What does it mean to have a negative flood value?

Logistic function

- Let's force h to be between 0 and 1 somehow
- Introducing the logistic function (sigmoid function)



$$f(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x}$$

https://en.wikipedia.org/wiki/Logistic_function

Logistic Regression

- Pass $\theta^T \mathbf{x}$ through the logistic function

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Loss function?

- MSE error no longer a good candidate

- Let’s turn to use probabilistic argument for logistic regression

Logistic Function derivative

The derivative has a nice property by design.
This is also why many algorithm we'll learn later in class also uses the logistic function

$$
\begin{aligned}
g'(z) &= \frac{d}{dz}\frac{1}{1+e^{-z}} \\
&= \frac{1}{(1+e^{-z})^2}\left(e^{-z}\right) \\
&= \frac{1}{(1+e^{-z})}\cdot\left(1-\frac{1}{(1+e^{-z})}\right) \\
&= g(z)(1-g(z)).
\end{aligned}
$$

Probabilistic view of Logistic Regression

- Let's assume, we'll classify as 1 with probability according to the output of

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$\begin{aligned} P(y=1 \mid x; \theta) &= h_\theta(x) \\ P(y=0 \mid x; \theta) &= 1 - h_\theta(x) \end{aligned}$$

or

$$p(y \mid x; \theta) = (h_\theta(x))^y \, (1 - h_\theta(x))^{1-y}$$

Maximizing log likelihood

$$p(y \mid x; \theta) = (h_\theta(x))^y \, (1 - h_\theta(x))^{1-y}$$

$$\begin{aligned}
g'(z) &= \frac{d}{dz}\frac{1}{1+e^{-z}} \\
&= \frac{1}{(1+e^{-z})^2}\left(e^{-z}\right) \\
&= \frac{1}{(1+e^{-z})} \cdot \left(1 - \frac{1}{(1+e^{-z})}\right) \\
&= g(z)(1-g(z)).
\end{aligned}$$

Maximizing log likelihood

$$g'(z) = \frac{d}{dz}\frac{1}{1+e^{-z}} = \frac{1}{(1+e^{-z})^2}\left(e^{-z}\right) = \frac{1}{(1+e^{-z})}\cdot\left(1-\frac{1}{(1+e^{-z})}\right) = g(z)(1-g(z)).$$

$$p(y \mid x;\theta) = (h_\theta(x))^y \, (1-h_\theta(x))^{1-y}$$

$$\theta_j \Leftarrow \theta_j + r\Sigma_{i=1}^{m}(y_i - h_\theta(x_i))x_i^{(j)}$$

Logistic Regression update rule

$$\theta_j \Leftarrow \theta_j + r\Sigma_{i=1}^{m}(y_i - h_\theta(x_i))x_i^{(j)}$$

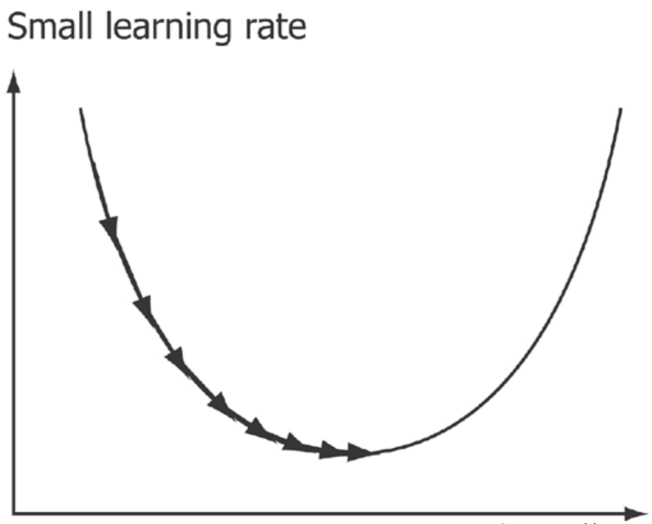
Update rule for linear regression

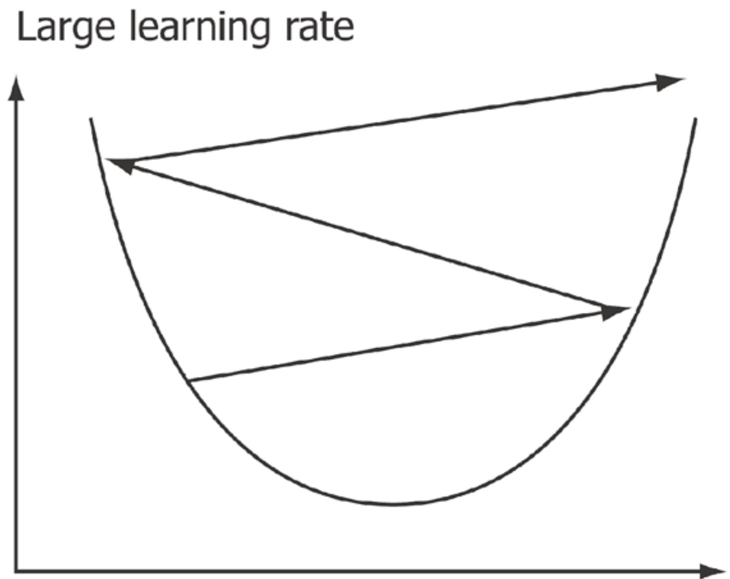
$$\theta_j \Leftarrow \theta_j + r\Sigma_{i=1}^{m}(y_i - \theta^T\mathbf{x}_i)x_i^{(j)}$$

Loose Ends from HW

- How to select r? (The learning rate)

$$\theta_j \Leftarrow \theta_j + r\Sigma_{i=1}^{m}(y_i - \theta^T \mathbf{x}_i)x_i^{(j)}$$





https://www.packtpub.com/books/content/big-data

r too small and the model converges slowly

r too large and the model diverges

Learning rate issues

- Typically, r is normalized with the amount of training examples in a mini-batch. (Divide by m)

$$\theta_j \Leftarrow \theta_j + r\Sigma_{i=1}^{m}(y_i - \theta^T \mathbf{x}_i)x_i^{(j)}$$

- Typical values are 0.1-0.001
- Usually have a decay over time

Scaling the input data

- We use age, passenger class, gender, and embark as our input.
- Age has a lot more variance (0.42 – 80) than the other data.
- This makes parameter initialization hard, and makes the learning rate selection hard.
- $h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$

Scaling the input data

- Scale all input data to be in the same range
- Using statistics from training data
  - Scale to [-1,1]
  - Scale to [0, 1]
  - Scale to standard normal
- Don't forget to apply the same scaling to the test data

Feature selection

- Most likely you will get better results with just two features.
- This is the importance of feature selection.
- Knowing what good features to select is not trivial
- Approaches for feature selection (or for not having to do feature selection)
  - Cross validation
  - Random forest
  - Boosting

Feature engineering

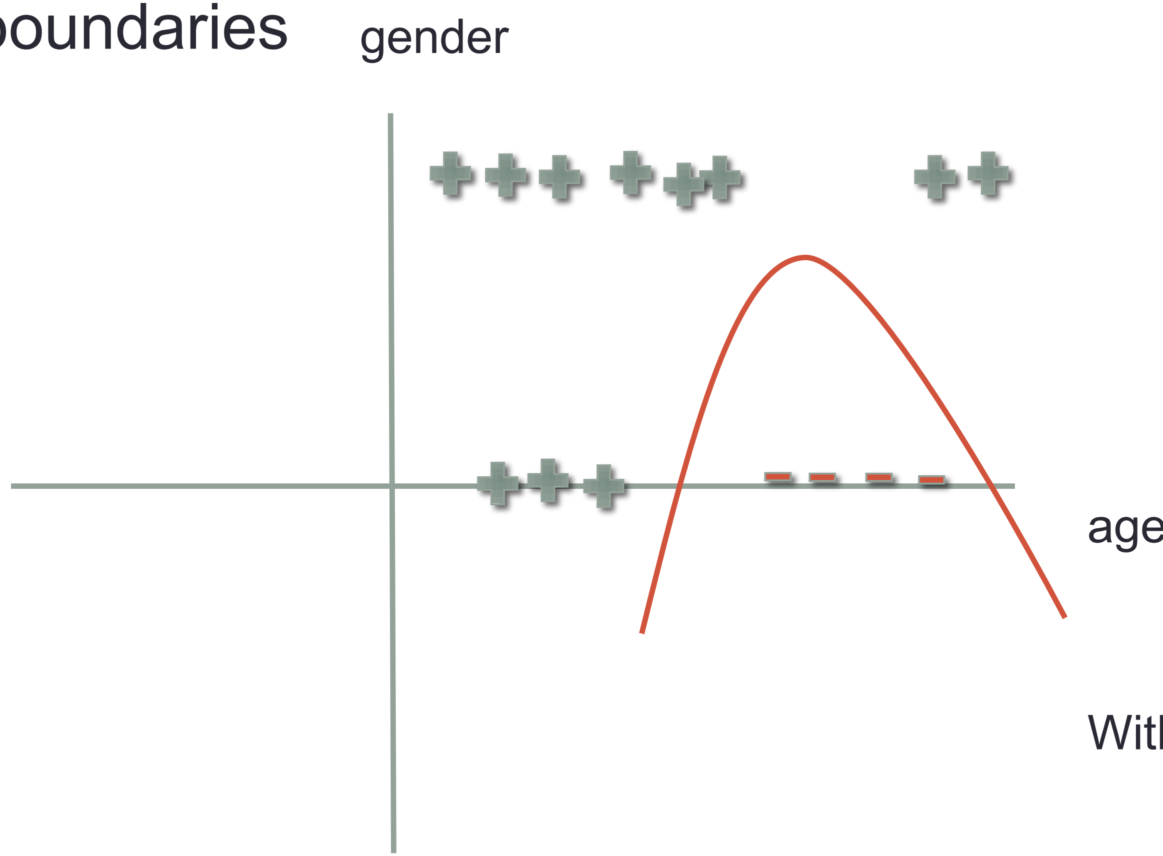
- Logistic regression is a linear classification

Feature engineering

- Logistic regression is a linear classification

Feature engineering

- Add non-linear features to get non-linear decision boundaries



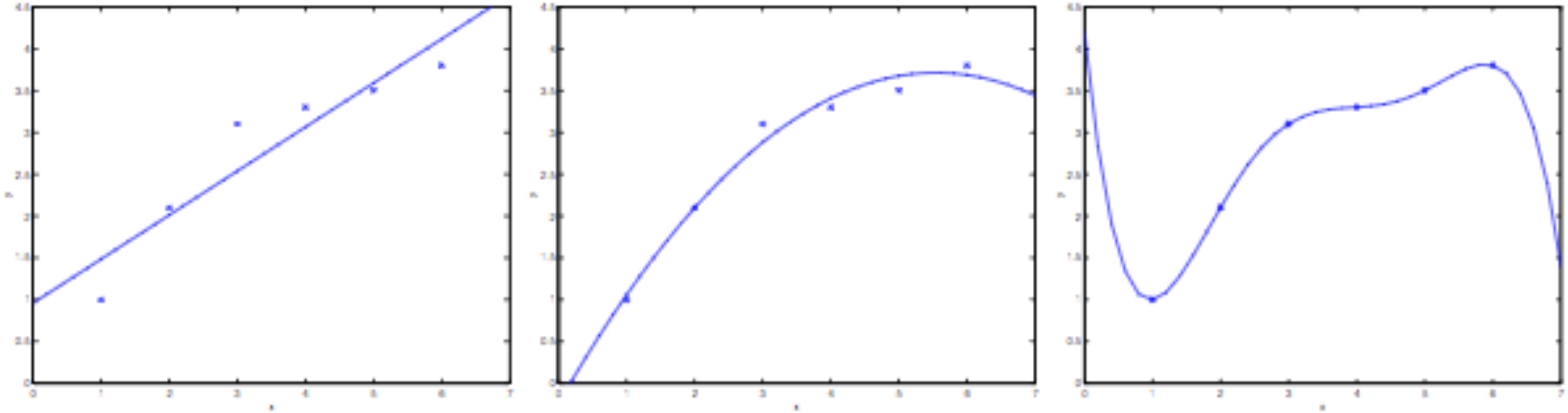
h(x) > 0.5

Decision boundary

With non-linear features

This is also a form of feature selection (more specifically feature engineering)

Overfitting Underfitting



Adding more non-linear features makes the line more curvy

(Adding more features also means more model parameters)

The curve can go directly to the outliers with enough parameters.

We call this effect overfitting

For the opposite case, having not enough parameters to model the data is called underfitting

Bias-Variance trade-off

- We will formulate overfitting and underfitting mathematically
- Using regression model

Regression with Gaussian noise

- $y = h(\mathbf{x}) + \varepsilon$
  - Where ε is normally distributed with mean zero and variance σ^2
  - The training data D = $\{(\mathbf{x_1},y_1), (\mathbf{x_3},y_3), (\mathbf{x_3},y_3)\ldots\}$ is drawn from some distribution $P(\mathbf{x},y)$ governing our universe!
  - Assume $(\mathbf{x_i},y_i)$ is iid
- Given D we can train a regressor $h_D(x)$
- We calculate the expected error (squared error) on new $(\mathbf{x},y)$ data with the regressor
  - $E_{(\mathbf{x},y)}[(h_D(\mathbf{x}) - y)^2] = \int_x \int_y (h_D(\mathbf{x}) - y)^2 \Pr(\mathbf{x}, y) \partial y \partial \mathbf{x}$
- But D is actually random too!

Regression with Gaussian noise

- We calculate the expected error (squared error) on new ($\mathbf{x}$,y) data with the regressor
  - $E_{(\mathbf{x},y)}[(h_D(\mathbf{x}) - y)^2] = \int_x \int_y (h_D(\mathbf{x}) - y)^2 \Pr(\mathbf{x}, y) \partial y \partial \mathbf{x}$
- Consider parallel worlds, we can receive different training data D which yields different regression $h_D(x)$
- The expectation of error over all possible new test data point ($\mathbf{x}$,y) and different possible training data D is

$$E_{\substack{(\mathbf{x},y)\sim P \\ D\sim P^n}} \left[(h_D(\mathbf{x}) - y)^2 \right] = \int_D \int_{\mathbf{x}} \int_y (h_D(\mathbf{x}) - y)^2 \mathrm{P}(\mathbf{x}, y) \mathrm{P}(D) \partial \mathbf{x} \partial y \partial D$$

http://www.cs.cornell.edu/courses/cs4780/2017sp/lectures/lecturenote11.html

Regression with Gaussian noise

- This expression tells the expected quality of our model with random training data and a random test data

$$E_{\substack{(\mathbf{x},y)\sim P\\ D\sim P^n}}\left[(h_D(\mathbf{x})-y)^2\right]=\int_D\int_{\mathbf{x}}\int_y (h_D(\mathbf{x})-y)^2\mathrm{P}(\mathbf{x},y)\mathrm{P}(D)\partial\mathbf{x}\partial y\partial D$$

$$\underbrace{E_{\mathbf{x},y,D}\left[(h_D(\mathbf{x})-y)^2\right]}_{\text{Expected Test Error}}=\underbrace{E_{\mathbf{x},D}\left[\left(h_D(\mathbf{x})-\bar{h}(\mathbf{x})\right)^2\right]}_{\text{Variance}}+\underbrace{E_{\mathbf{x},y}\left[(\bar{y}(\mathbf{x})-y)^2\right]}_{\text{Noise}}+\underbrace{E_{\mathbf{x}}\left[\left(\bar{h}(\mathbf{x})-\bar{y}(\mathbf{x})\right)^2\right]}_{\text{Bias}^2}$$

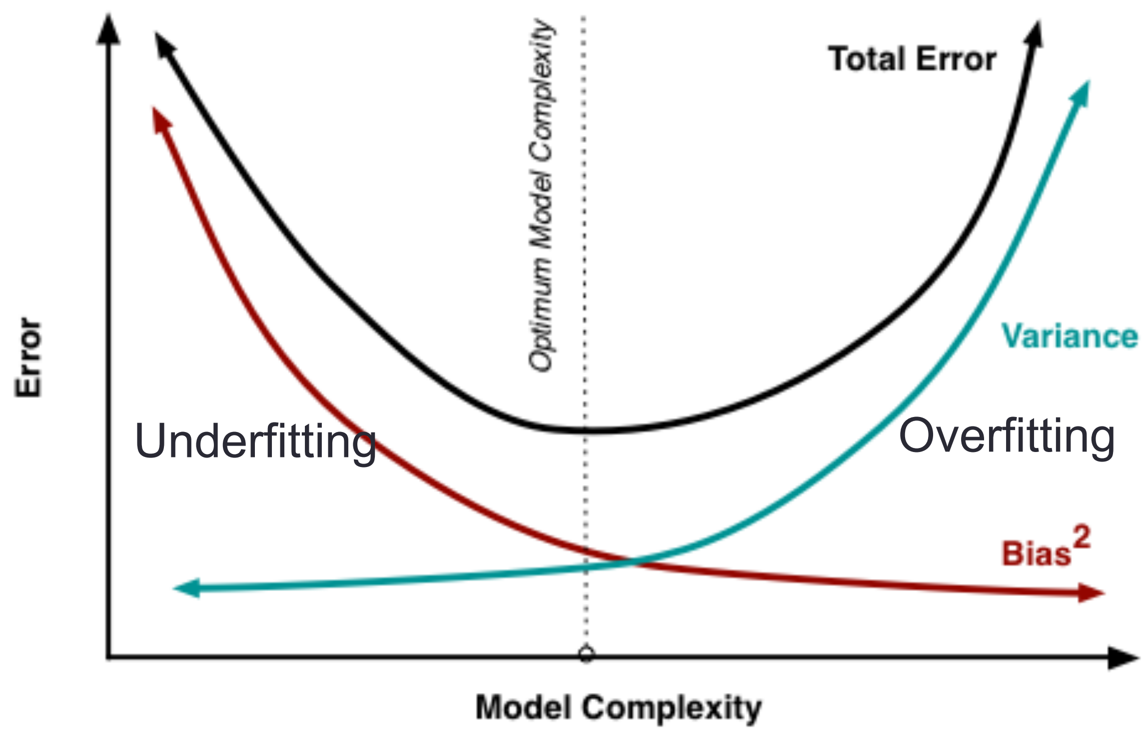
Variance, Bias, and noise

$$\underbrace{E_{\mathbf{x},y,D}\left[\left(h_D(\mathbf{x})-y\right)^2\right]}_{\text{Expected Test Error}} = \underbrace{E_{\mathbf{x},D}\left[\left(h_D(\mathbf{x})-\bar{h}(\mathbf{x})\right)^2\right]}_{\text{Variance}} + \underbrace{E_{\mathbf{x},y}\left[\left(\bar{y}(\mathbf{x})-y\right)^2\right]}_{\text{Noise}} + \underbrace{E_{\mathbf{x}}\left[\left(\bar{h}(\mathbf{x})-\bar{y}(\mathbf{x})\right)^2\right]}_{\text{Bias}^2}$$

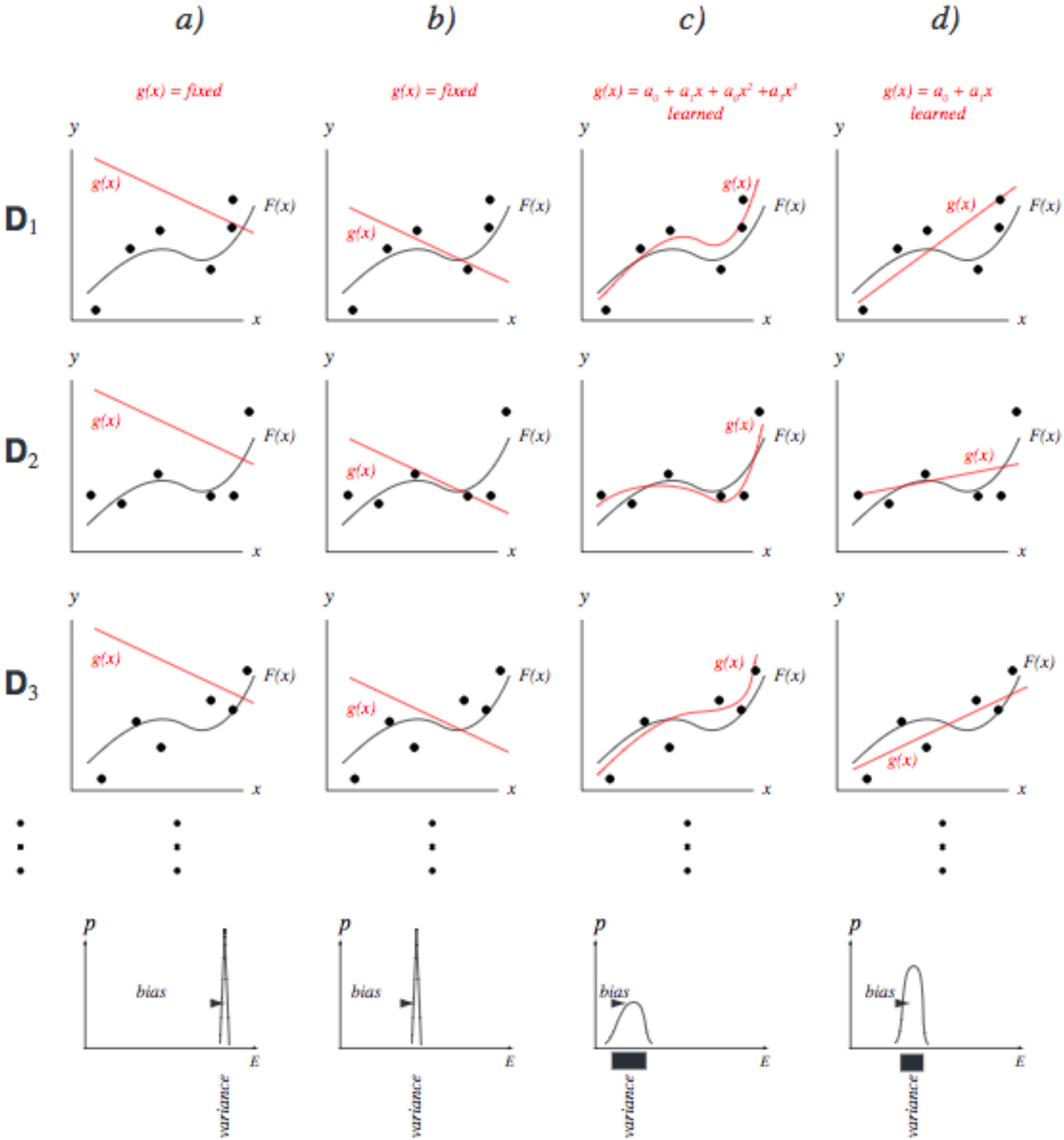
- Variance: how your classifier changes if the training data changes. Measures generalizability.
- Bias: The model's inherent error. If you have infinite training data, you will have the average classifier $\bar{h}$ and still left with this error.
  - For example, even with infinite training data, a linear classifier will still have errors if the distribution is non-linear.
- Noise: data-intrinsic noise. Noise from measurement, noise from feature extraction, etc. Regardless of your model this remains.

Bias-Variance Underfitting-Overfitting

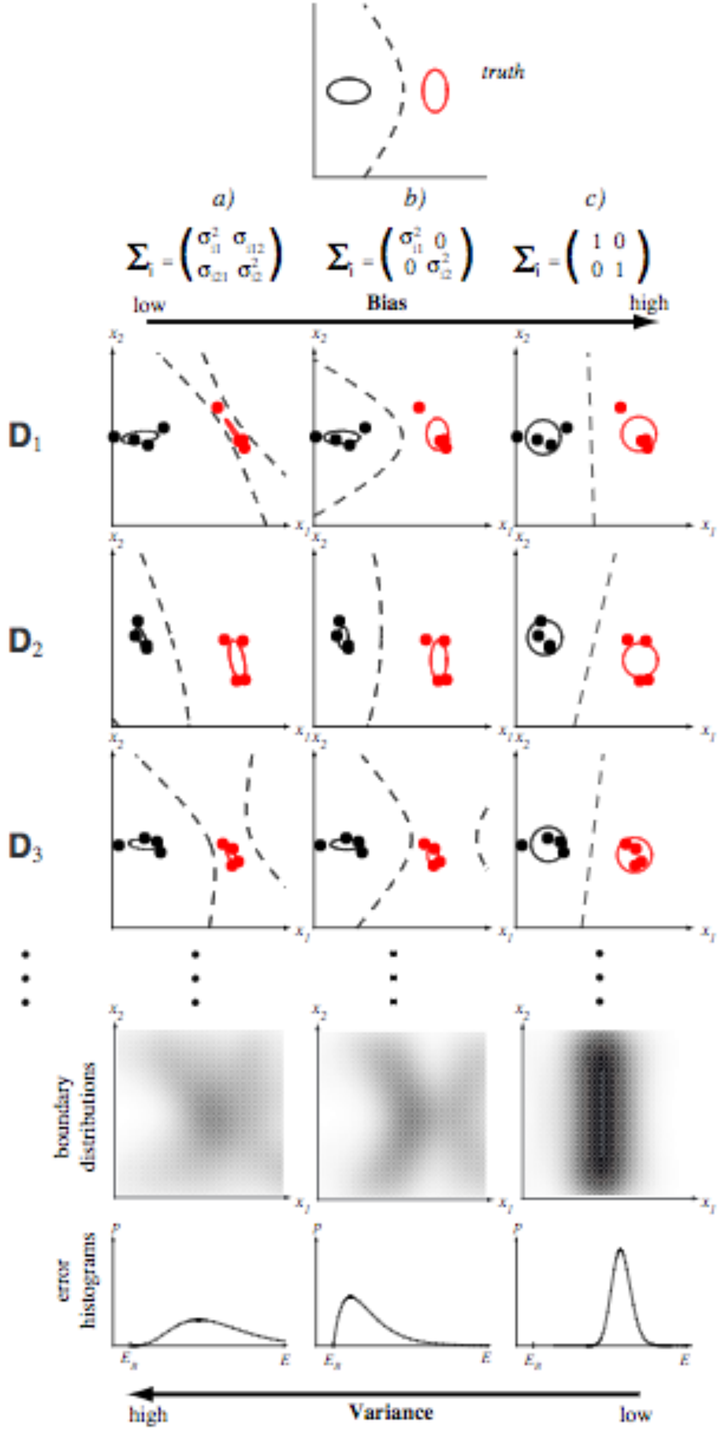
- Usually if you try to reduce the bias of your model, the variance will increase, and vice versa.
- Called the bias-variance trade-off



http://www.cs.cornell.edu/courses/cs4780/2017sp/lectures/lecturenote11.html



Duda et al. Pattern Classification

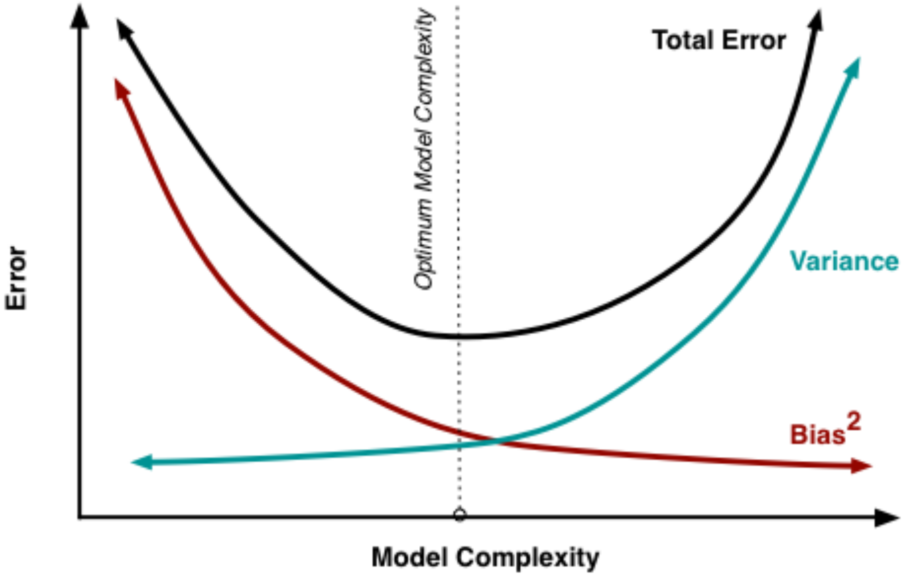


Duda et al. Pattern Classification

When to stop the update?

- Consider the updates of Logistic regression as trying to reduce the bias of the model
  - As we keep updating, the model overfits more to the training data
- We want to stop when the error on the validation set increases*
  - More on this later
- Validation test: a separate set that is used to measure overfitting

  Training set
  Validation set
  Test set



More tricks?

- http://ahmedbesbes.com/how-to-score-08134-in-titanic-kaggle-challenge.html
- Feature Engineering/selection
- Parameter tuning
- Try different models

449	▲ 62...	Kaustubh Kulkarni 2	0.81340	6	6h
450	new	AshishDoshi	0.81340	1	5h
451	new	SouravKarwa	0.81340	2	31m
452	▲ 18...	Ahmed Besbes	0.81340	15	now

Your Best Entry ↑

Your submission scored 0.81340, which is not an improvement of your best score. Keep trying!

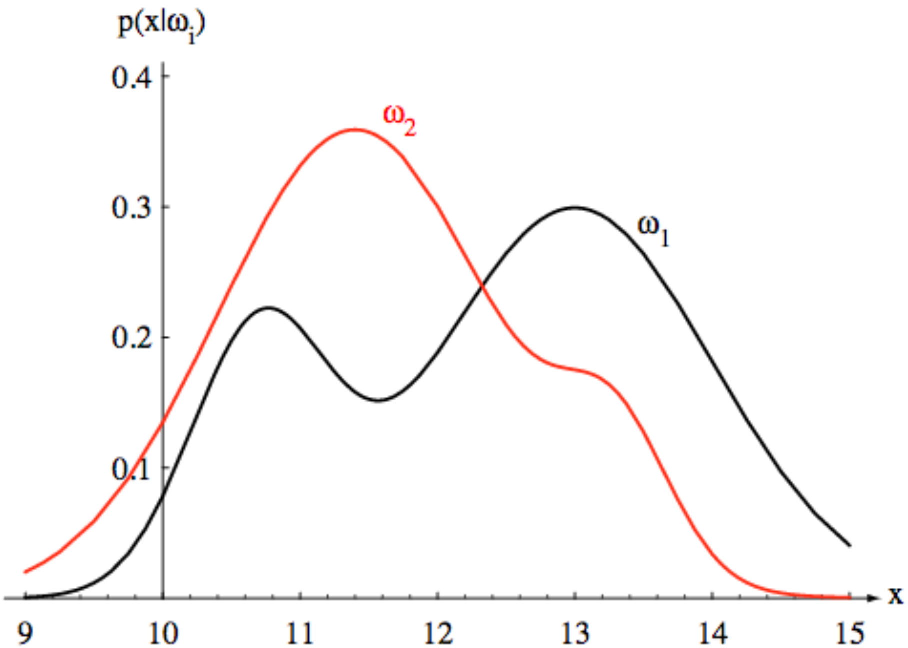
453	▼ 7	Clement Sengelen	0.80861	11	2mo

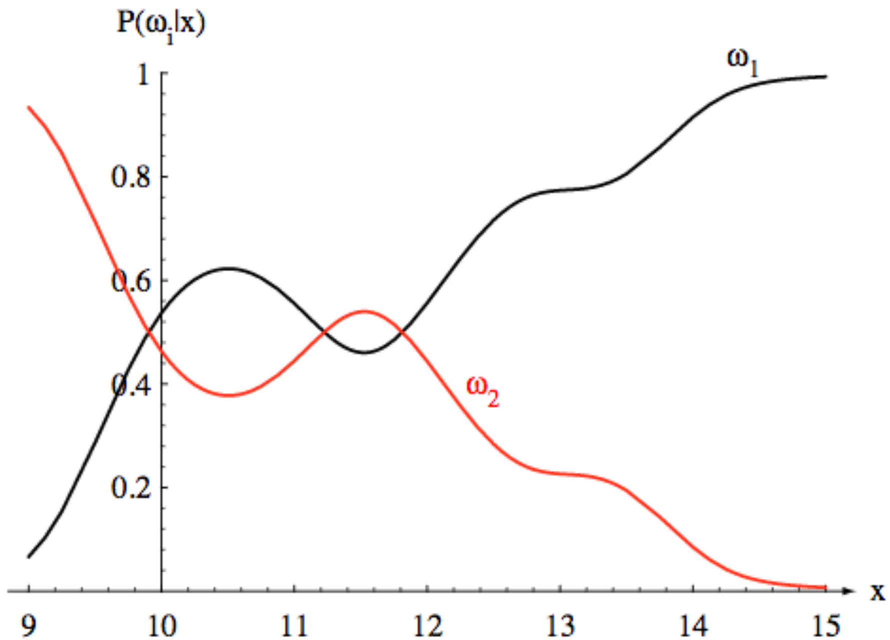
The Bayes Lecture

- Bayes Decision Rule
- Naïve Bayes

A simple decision rule

- If we can know either p(x|w) or p(w|x) we can make a classification guess

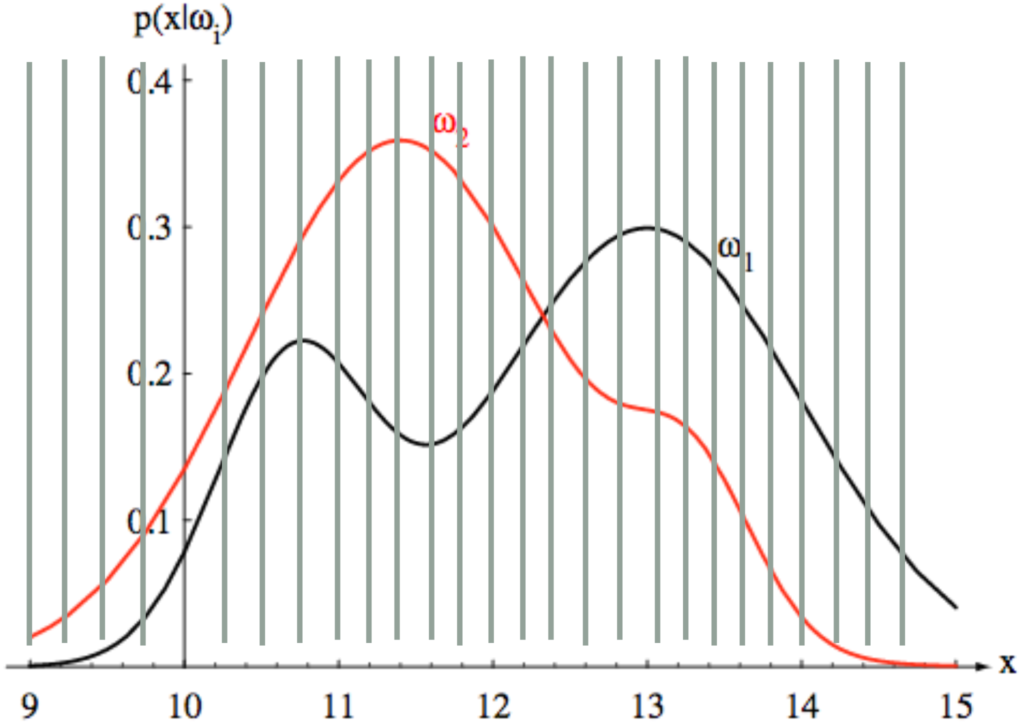




Goal: Find p(x|w) or p(w|x)

Duda et al. Pattern Classification

A simple way to estimate p(x|w)



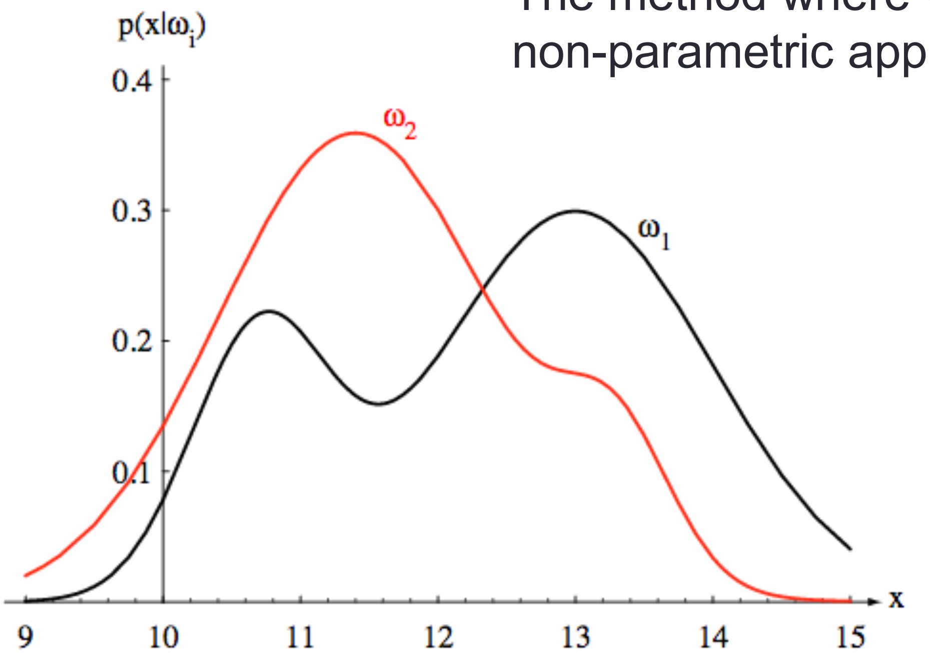
Make a histogram!

What happens if there is no data in a bin?

The parametric approach

- We assume p(x|w) or p(w|x) follow some distributions with parameter θ

The method where we find the histogram is a non-parametric approach



$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$$

Goal: Find θ so that we can estimate p(x|w) or p(w|x)

Maximum Likelihood Estimate (MLE)

$$p(w_i|x) = \frac{p(x|w_i)p(w_i)}{p(x)}$$

$$\text{Posterior} = \frac{\text{likelihood} * \text{prior}}{\text{evidence}}$$

- Maximizing the likelihood (probability of data given model parameters)

$p(\mathbf{x}|\theta) = L(\theta)$ <- This assumes the data is fixed

- Usually done on log likelihood

- Take the partial derivative wrt to θ and solve for the θ that maximizes the likelihood

MLE of binomial trials

- A coin with bias is tossed N times. k times are heads. Find θ, the probability of the coin landing head.

MLE of Gaussian

- Observe x_i estimate the mean and the variance. Assume the data is normally distributed.

Maximum Likelihood Estimate (MLE)

$$p(w_i|x) = \frac{p(x|w_i)p(w_i)}{p(x)}$$

$$\text{Posterior} = \frac{\text{likelihood} * \text{prior}}{\text{evidence}}$$

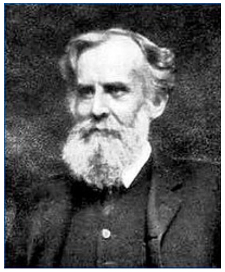
- Maximizing the likelihood (probability of data given model parameters)

$p(\mathbf{x}|\theta) = L(\theta)$ <- This assumes the data is fixed

- Usually done on log likelihood

- Take the partial derivative wrt to θ and solve for the θ that maximizes the likelihood

Frequentist vs Bayesian view



- Frequentist
  - Probability is “frequency of occurrence”
  - Data is from a random procedure that draw from unknown but fixed phenomenon.
    - Distribution parameter is a constant

- Bayesian
  - Probability is “degree of uncertainty”
  - Data is fixed and you want to infer about the unknown phenomenon.
    - Distribution parameter is a distribution
    - Prior knowledge about the phenomenon can change the inference results.

https://en.wikipedia.org/wiki/Frequentist_probability#/media/File:John_Venn.jpg
https://en.wikipedia.org/wiki/Thomas_Bayes#/media/File:Thomas_Bayes.gif

Maximum A Posteriori (MAP) Estimate

MLE

- Maximizing the likelihood (probability of data given model parameters)

$$\underset{\theta}{\operatorname{argmax}}\ p(\mathbf{x}|\theta)$$

$p(\mathbf{x}|\theta)$
$= L(\theta)$

- Usually done on log likelihood

- Take the partial derivative wrt to θ and solve for the θ that maximizes the likelihood

MAP

- Maximizing the posterior (model parameters given data)

$$\underset{\theta}{\operatorname{argmax}}\ p(\theta|\mathbf{x})$$

- But we don't know $p(\theta|\mathbf{x})$

- Use Bayes rule

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})}$$

- Taking the argmax for θ we can ignore $p(\mathbf{x})$

- $\underset{\theta}{\operatorname{argmax}}\ p(\mathbf{x}|\theta)\ p(\theta)$

MAP on Gaussian

- We know x is Gaussian with unknown mean μ that we need to estimate and known variance σ^2
- Assume the prior of μ is $N(\mu_0, \sigma_0^2)$

- MAP estimate of μ is

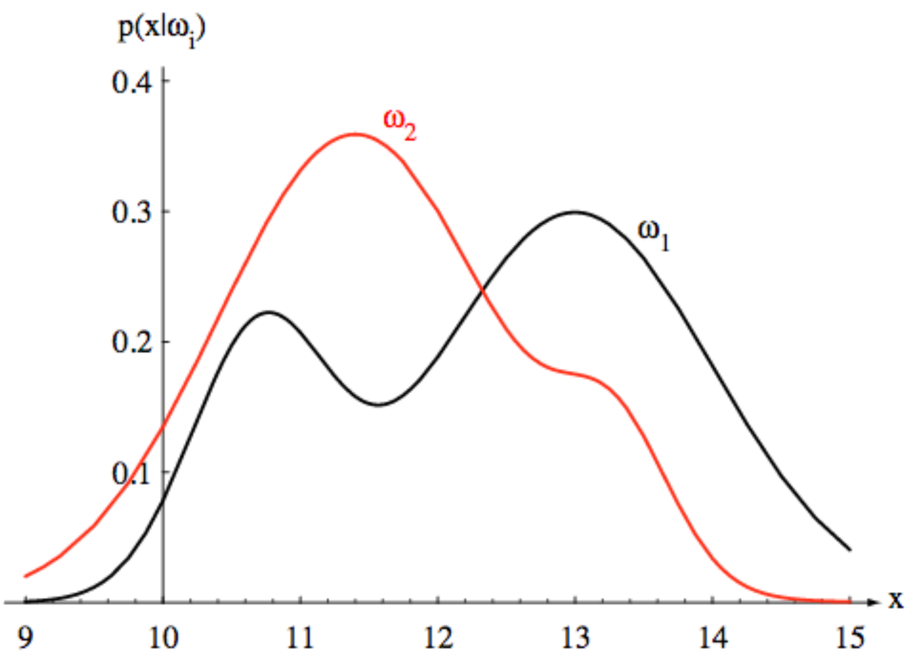
$$\mu_n = (\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2})[\frac{1}{n}\Sigma_{i=1}^n x_i] + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2}\mu_0$$

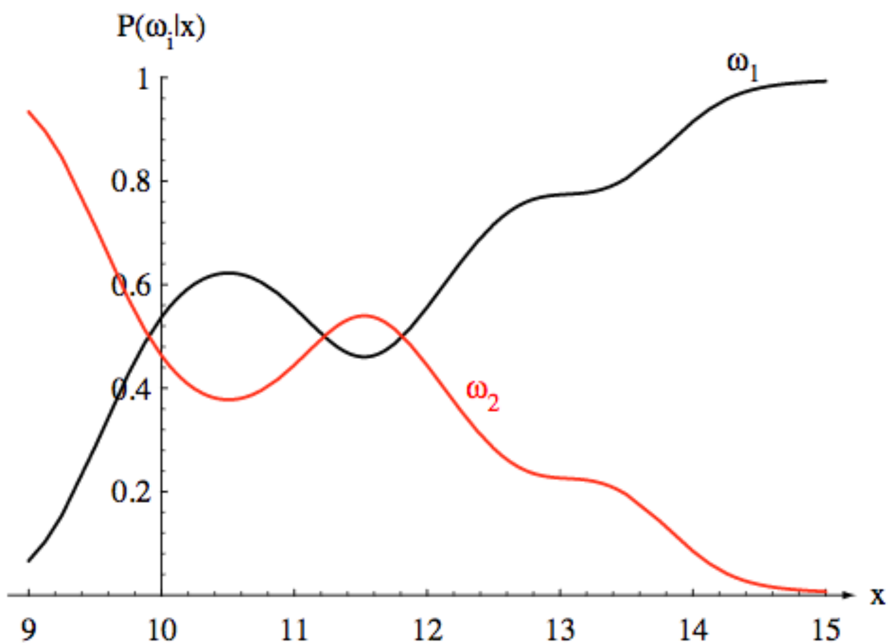
Notes of MAP estimate

- Usually harder to estimate than MLE
- If we use an uninformative prior for θ
  - MAP estimate = MLE
- Given infinite data
  - MAP estimate converges to MLE
- MAP is useful when you have less data, so you need additional knowledge about the domain
  - MAP estimate tends to converges to faster than MLE even with an arbitrary distribution
  - Can help prevent overfitting
- Useful for model adaptation (MAP adaptation)
  - Learn MLE on larger dataset, use this as your prior distribution
  - Learn MAP estimate on your dataset

A simple decision rule

- If we can know either p(x|w) or p(w|x) we can make a classification guess





Goal: Find p(x|w) or p(w|x) by finding the parameter of the distribution

Likelihood ratio test

- If $P(w_1|x) > P(w_2|x)$, that x is more likely to be class w_1
- Again we know $P(x|w_1)$ is more intuitive and easier to calculate than $P(w_1|x)$

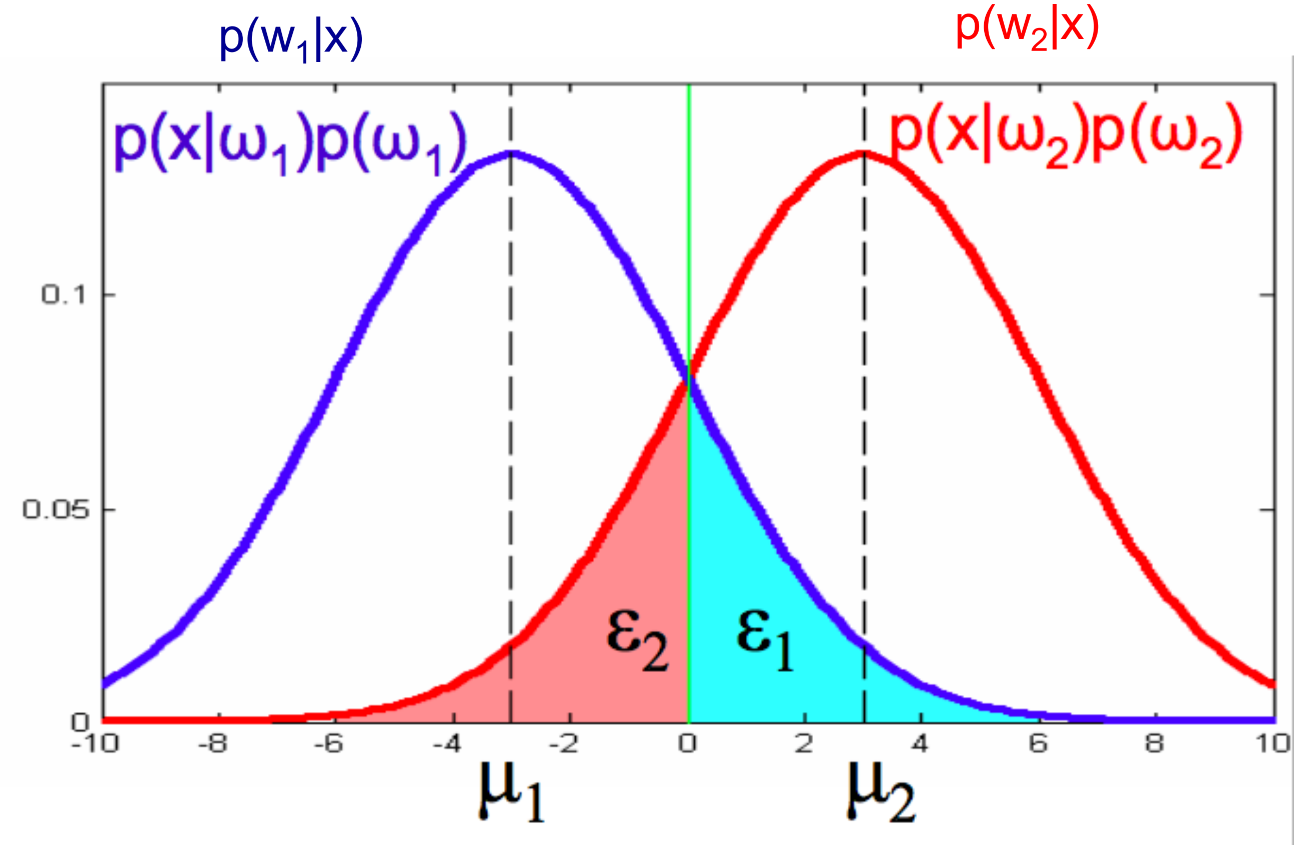
- Our classifier becomes
- $$P(x|w_1)P(w_1) \quad ? \quad P(x|w_2)P(w_2)$$
- $$\boxed{\frac{P(x|w_1)}{P(x|w_2)}} \quad ? \quad \boxed{\frac{P(w_2)}{P(w_1)}}$$

Ratio of priors

Likelihood ratio

Notes on likelihood ratio test (LRT)

- LRT minimizes the classification error (all errors are equally bad)

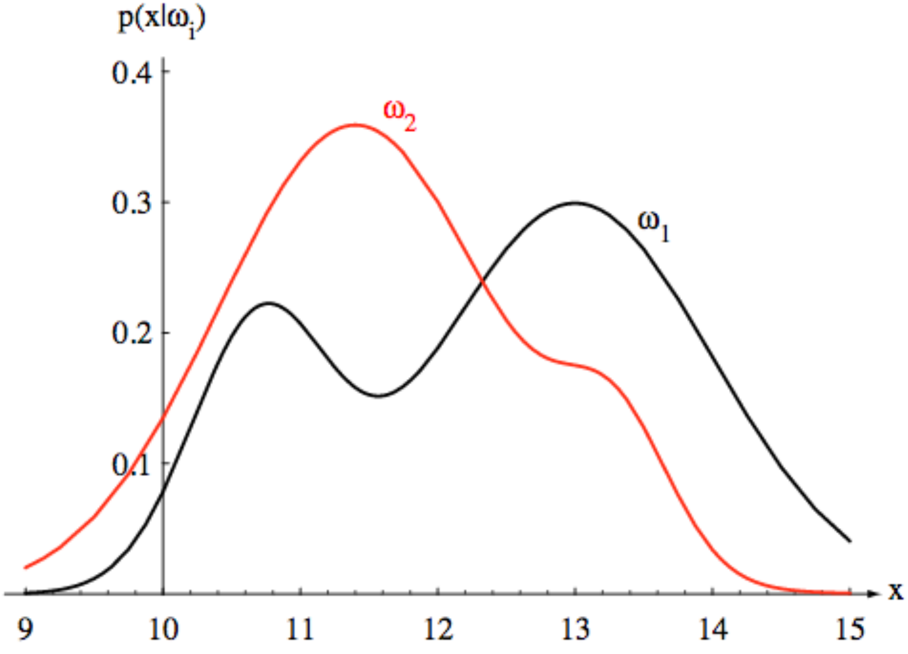


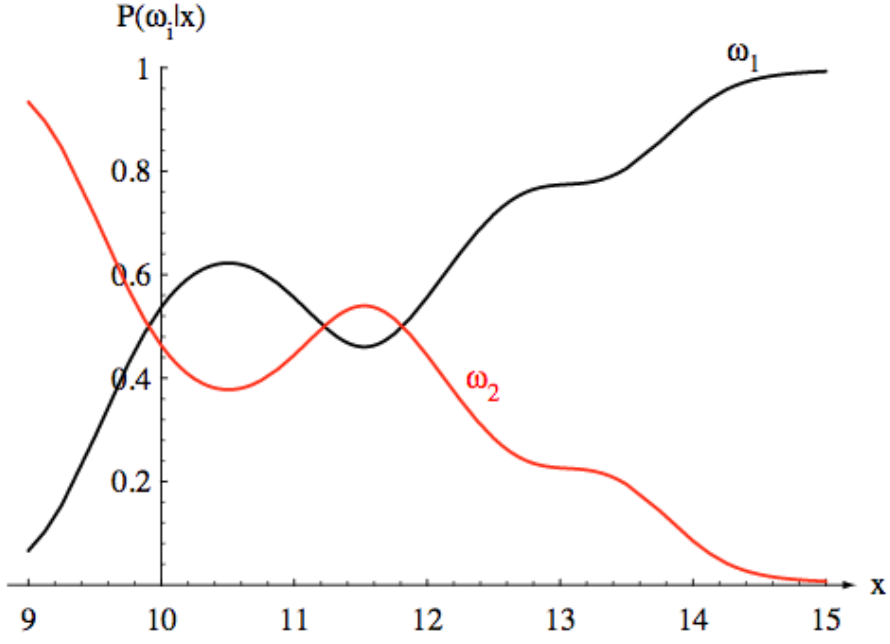
Notes on LRT

- If $P(w_1|x) > P(w_2|x)$, x is more likely to be class w_1
  - Also known as MAP decision rule
  - The classifier is sometimes called the Bayes classifier
- If we do not want to treat all error equally, we can assign different loss to each error, and minimize the expected loss. This is called Bayes loss/risk classifier
- $$\frac{P(x|w_1)}{P(x|w_2)} \quad ? \quad \frac{P(w_2)(L_{1|2} - L_{2|2})}{P(w_1)(L_{2|1} - L_{1|1})}$$
- When we treat errors equally, we refer to the zero-one loss
- $L_{1|2} = 1,\ L_{2|2} = 0,\ L_{2|1} = 1,\ L_{1|1} = 0$

Notes on LRT

- If we treat the priors as equal, we get the maximum likelihood criterion
- $$\frac{P(x|w_1)}{P(x|w_2)} \quad ? \quad 1$$





Naïve Bayes

- Below is the LRT or the Bayes classifier

$$P(x|w_1)P(w_1) \quad ? \quad P(x|w_2)P(w_2)$$

- What about Naïve Bayes?

- Here x is a vector with m features $[x_1, x_2, \ldots x_m]$
- $P(x|w_i)$ is m+1 dimensional
  - Sometimes to hard to model, not enough data, overfit, *curse of dimensionality*, etc.
- Assumes $x_1, x_2, \ldots x_m$ independent given w_i (conditional independence)
  - What does this mean?

Modeling distributions

Wind in the morning $X \in$ {Calm, Windy}

PM2.5 level in the afternoon $Y \in$ {Low, Med, High}

$\arg\max P(Y \mid X) = \arg\max P(X \mid Y)\, P(Y)$

Day	X	Y
1	W	M
2	C	M
3	W	M
4	W	H
5	C	L
6	W	L
7	C	H
8	W	L

Modeling distributions

Wind in the morning $X \in \{\text{Calm, Windy}\}$

PM2.5 level in the afternoon $Y \in \{\text{Low, Med, High}\}$

argmax P(Y | X)

Day	X	Y
1	W	M
2	C	M
3	W	M
4	W	H
5	C	L
6	W	L
7	C	H
8	W	L

P(X, Y)	L	M	H
C	1/8	1/8	1/8
W	2/8	2/8	1/8

Joint distribution

P(Y | X)	L	M	H
C	1/3	1/3	1/3
W	2/5	2/5	1/5

Conditional distribution

Modeling distributions

Wind in the morning $X \in \{\text{Calm, Windy}\}$

PM2.5 level in the afternoon $Y \in \{\text{Low, Med, High}\}$

argmax $P(Y \mid X)$

Day	X	Y
1	W	M
2	C	M
3	W	M
4	W	H
5	C	L
6	W	L
7	C	H
8	W	L

Joint distribution

P(X, Y)	L	M	H
C			
W			

Total data
8

count(X,Y)	L	M	H
C	1	1	1
W	2	2	1

$$P(X, Y) = \frac{\text{Count}(X, Y)}{\text{Total count}}$$

is the Maximum Likelihood Estimate (MLE) of P(X,Y)

Modeling distributions

Wind in the morning $X \in$ {Calm, Windy}

PM2.5 level in the afternoon $Y \in$ {Low, Med, High}

argmax P(Y | X)

Day	X	Y
1	W	M
2	C	M
3	W	M
4	W	H
5	C	L
6	W	L
7	C	H
8	W	L

P(Y | X)	L	M	H
C			
W			

Conditional distribution

Total data
8

count(X,Y)	L	M	H	Total
C	1	1	1	3
W	2	2	1	5

$$P(Y \mid X) = \frac{\text{Count}(X, Y)}{\text{Total count}(X)}$$

is the Maximum Likelihood Estimate (MLE) of P(Y|X)

Curse of dimensionality

Wind in the morning $X \in \{Calm, Windy\}$

PM2.5 level in the afternoon $Y \in \{Low, Med, High\}$

PM2.5 level in the evening $Z \in \{Low, Med, High\}$

$\text{argmax}\ P(Z \mid Y, X) = \text{argmax}\ P(Y,X \mid Z)\ P(Z)$

Day	X	Y	Z
1	W	L	M
2	C	M	M
3	W	H	M
4	W	M	H
5	C	M	L
6	W	M	L
7	C	L	H
8	W	H	L

count(Z,Y,X)	Z=L	Z=M	Z=H
X=W,Y=L	0	1	0
X=W,Y=M	1	0	1
X=W,Y=H	1	1	0
X=C,Y=L	0	0	1
X=C,Y=M	1	1	0
X=C,Y=H	0	0	0

Naïve Bayes

- $P(\mathbf{x}|w_i)P(w_i) = P(w_i) \prod_j P(x_j|w_i)$

- This assumption simplifies the calculation

Simplifying assumptions

Wind in the morning X ∈ {Calm, Windy}

PM2.5 level in the afternoon Y ∈ {Low, Med, High}

PM2.5 level in the evening Z ∈ {Low, Med, High}

$$\text{argmax } P(Z \mid Y, X) = \text{argmax } P(Y,X \mid Z)\, P(Z)$$
$$= \text{argmax } P(Y|Z)P(X|Z)P(Z)$$

Day	X	Y	Z
1	W	L	M
2	C	M	M
3	W	H	M
4	W	M	H
5	C	M	L
6	W	M	L
7	C	L	H
8	W	H	L

P(Y | Z)	Y = L	M	H
Z = L			
M			
H			

P(X | Z)	X = W	C
Z = L		
M		
H		

Dealing with zero probs

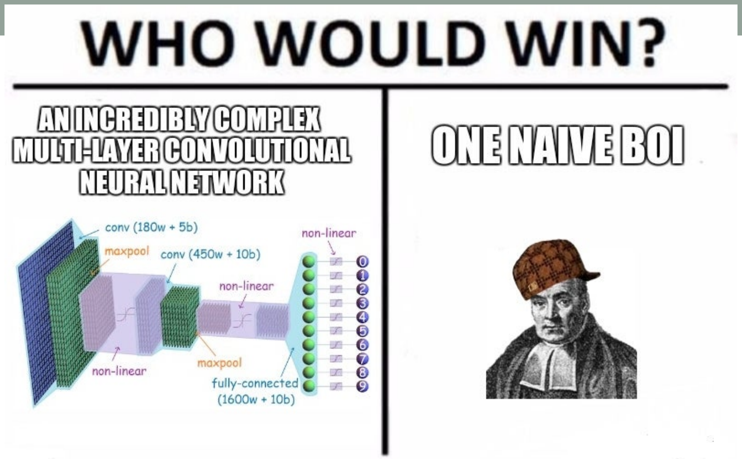
1. Use a very small value instead of zero (flooring)
2. Smooth the values using counts from other observations (smoothing)
3. Use priors (MAP adaptation)

Day	X	Y	Z
1	W	L	M
2	C	M	M
3	W	H	M
4	W	M	H
5	C	M	L
6	W	M	L
7	C	L	H
8	W	H	L

P(Y | Z)	Y = L	M	H
Z = L	0		
M			
H			

P(X | Z)	X = W	C
Z = L		
M		
H		

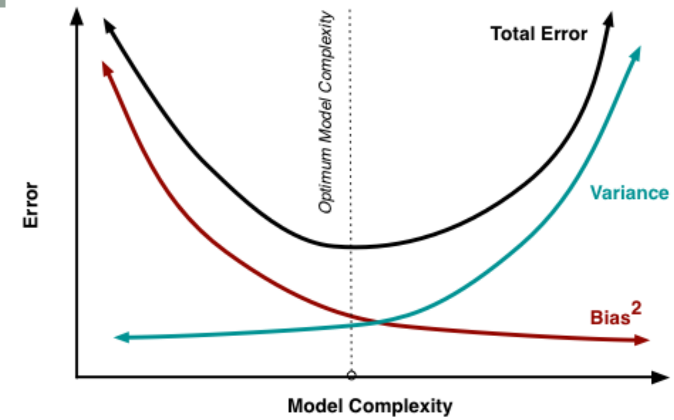
Naïve Bayes Notes



- $P(\mathbf{x}|w_i)P(w_i) = P(w_i) \prod_j P(x_j|w_i)$
- Note that we do not say anything about what kind of distribution $P(x_j|w_i)$ is.
  - In the homework you will play with this
    - Clean data
    - Estimate $P(x_j|w_i)$ using MLE, parametric and non-parametric version
    - Do prediction
    - Understand more about metrics
- Naïve Bayes can handle missing data
- Naïve Bayes is fast and quite good in practice
    - https://www.reddit.com/r/datascience/comments/hmhg9v/why_is_naive_bayes_so_popular_for_nlp/

Next homework

Summary



- Probabilistic view of linear regression
- Bias-Variance trade-off
  - Overfitting and underfitting
- MLE vs MAP estimate
  - How to use the prior
- LRT (Bayes Classifier)
  - Naïve Bayes

$$\boxed{\frac{P(x|w_1)}{P(x|w_2)}} \quad ? \quad \boxed{\frac{P(w_2)}{P(w_1)}}$$

Likelihood ratio

Ratio of priors