Bayesian Learning

Logistic

- Group project proposal due date is changed to **Feb 19**
- **Assignment 1** has been released Today, which due **Friday Feb 05, 11:59 pm**

More on Data Leakage

Data Leakage

- Training data leakage
  - Oversampling before splits
    - Training data may overlap with testing data
  - Prepare features on the entire data instead of just training data
    - Create vocab/preprocessing scaler from train+test data
  - Group leakage
    - A patient has 2 CT scans, 1 in train, 1 in test.
- **Feature leakage**
  - Some form of the label “slip” into the features
  - **This same information is not available during inference**

Feature Leakage Example I

- T: Detect lung cancer from CT scans
- E: collected from hospital I
- Performs well on unseen data from I
- Performs poorly on new data from hospital II

Date
Doctor note
Medical record
Scanner type
CT scan Image

Feature Leakage Example I

- **T**: Detect lung cancer from CT scans
- **E**: collected from hospital I
- Performs well on unseen data from I
- Performs poorly on new data from hospital II

Date
Doctor note
Medical record
Scanner type
CT scan Image

At hospital I, when doctors suspect that a patient has lung cancer, they send that patient to a higher-quality scanner

Feature Leakage Example II

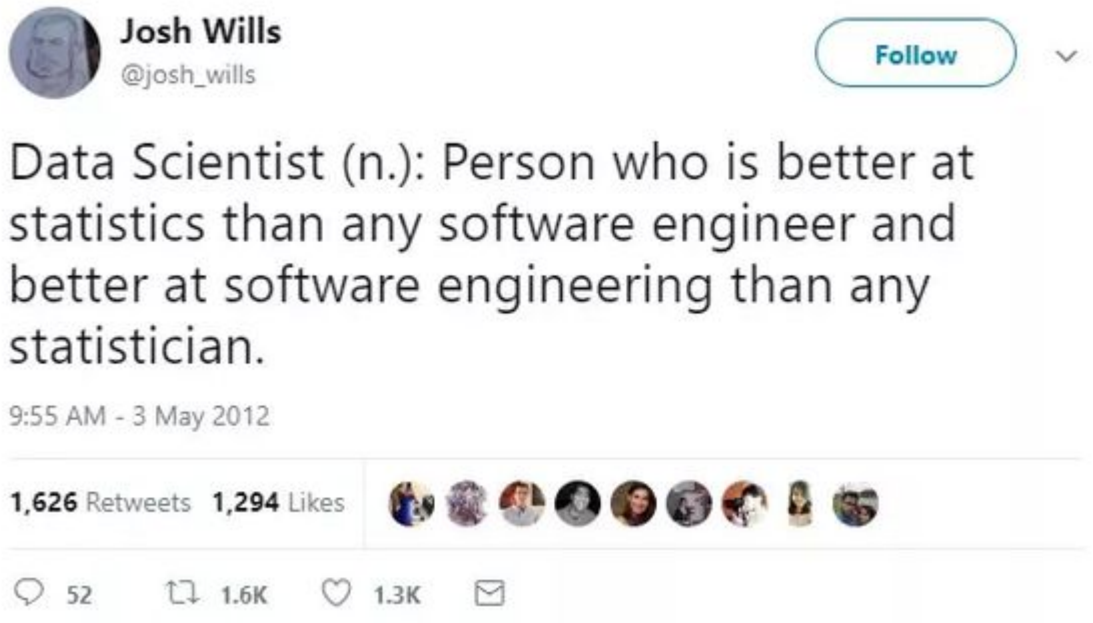
- **T**:Predict how many views an article will get
- **E**: historical data on the site
- Where might data leakage come from？

Date
Title
Article
Author
Language

How to avoid leakage?

- Check for duplication between train and valid/test splits
- Use only train splits for feature engineering (model training for sure)
- Train model on subset of features
  - If performance very high on subset, either high quality features or leakage!
- Monitor model performance as more features are added
  - If sudden increase, either high quality features or leakage!
- Check the correlation between feature and label
- Keep asking yourself during model development: can we use this information when the model is deployed for inference?

Pre Study


Josh Wills
@josh_wills

Data Scientist (n.): Person who is better at statistics than any software engineer and better at software engineering than any statistician.

9:55 AM - 3 May 2012

1,626 Retweets 1,294 Likes

52 1.6K 1.3K

Basics of Probability

- $P(A)$: probability that A happens
- $P(A|B)$: probability that A happens, given that B happens (conditional probability)
- Some rules:
  - Complement: $P(A^C) = 1 - P(A)$
  - Disjunction: $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
  - Conjunction: $P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$
  - If A and B are independent, $P(A \cap B) = P(A)P(B)$
  - Total probability: $P(B) = \sum_{i=1}^{k} P(A|B_i)P(B_i)$

Conditional Probability

- If: the patient has symptom of toothache
- Then: conclude cavity with probability P
- where P is the following conditional probability

$$P(\text{cavity}|\text{toothache})$$

- To compute p(cavity|toothache), we can compute

$$p(\text{cavity} \wedge \text{toothache}) / p(\text{toothache})$$

Density Estimation

- Density Estimation task
    - To construct an estimate of an unobservable underling probability density function, based on some observed data
- Data
    - Data sample x drawn i.i.d (independent identically distributed) from set **X** according to some distribution d, $x_i, \ldots, x_m \in \mathbf{X}$
- Problem
    - To find a distribution p out of a set P that best estimates the true distribution d

Argmax/Argmin

argmax stands for the argument of the maximum, that is to say, the set of points of the given argument for which the given function attains its maximum value.

$$\underset{x}{\operatorname{argmax}}\ f(x) = \{x \mid \forall y : f(y) \leq f(x)\}$$

$$\underset{x}{\operatorname{argmax}}\ (-|x|) = \{0\}$$

Maximum-Likelihood Estimation (MLE)

- Likelihood: probability of observing sample under distribution d, which, given the independence assumption is

$$Pr[x_1, \ldots, x_m] = \prod_{i=1}^{m} p(x_i)$$

- MLE Principle: select a distribution maximizing the sample probability

$$p_* = argmax_{p \in \mathcal{P}} \prod_{i=1}^{m} p(x_i)$$ Likelihood

$$p_* = argmax_{p \in \mathcal{P}} \sum_{i=1}^{m} log p(x_i)$$ Log-Likelihood

Maximum-Likelihood Estimation (MLE)

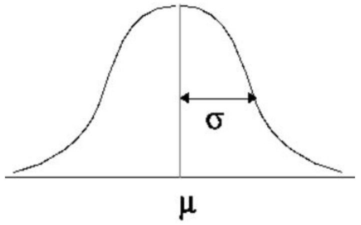
- Given training data D, MLE is to find the best hypothesis h that maximizes the likelihood of the training data

$$h_{ML} = argmax_{h \in (H)} P(D|h)$$

Example: Gaussian Distribution

- Task: find the most likely Gaussian distribution, given sequence of m real-valued observations: 3.23, 1.23, 0.55, 1.23, ….
- Normal distribution

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



- Log-likelihood: $l(p) = -\frac{1}{2} m log(2\pi\sigma^2) - \sum_1^m \frac{(x_i-\mu)^2)}{2\sigma^2}$
- Solution (estimate mean and stand dev):

$$\frac{\partial l(p)}{\partial \mu} \Leftrightarrow \mu = \frac{1}{m}\sum x_i \qquad \frac{\partial l(p)}{\partial \sigma^2} \Leftrightarrow \sigma^2 = \frac{1}{m}\sum (x_i - \mu)^2$$

Bayes Theorem

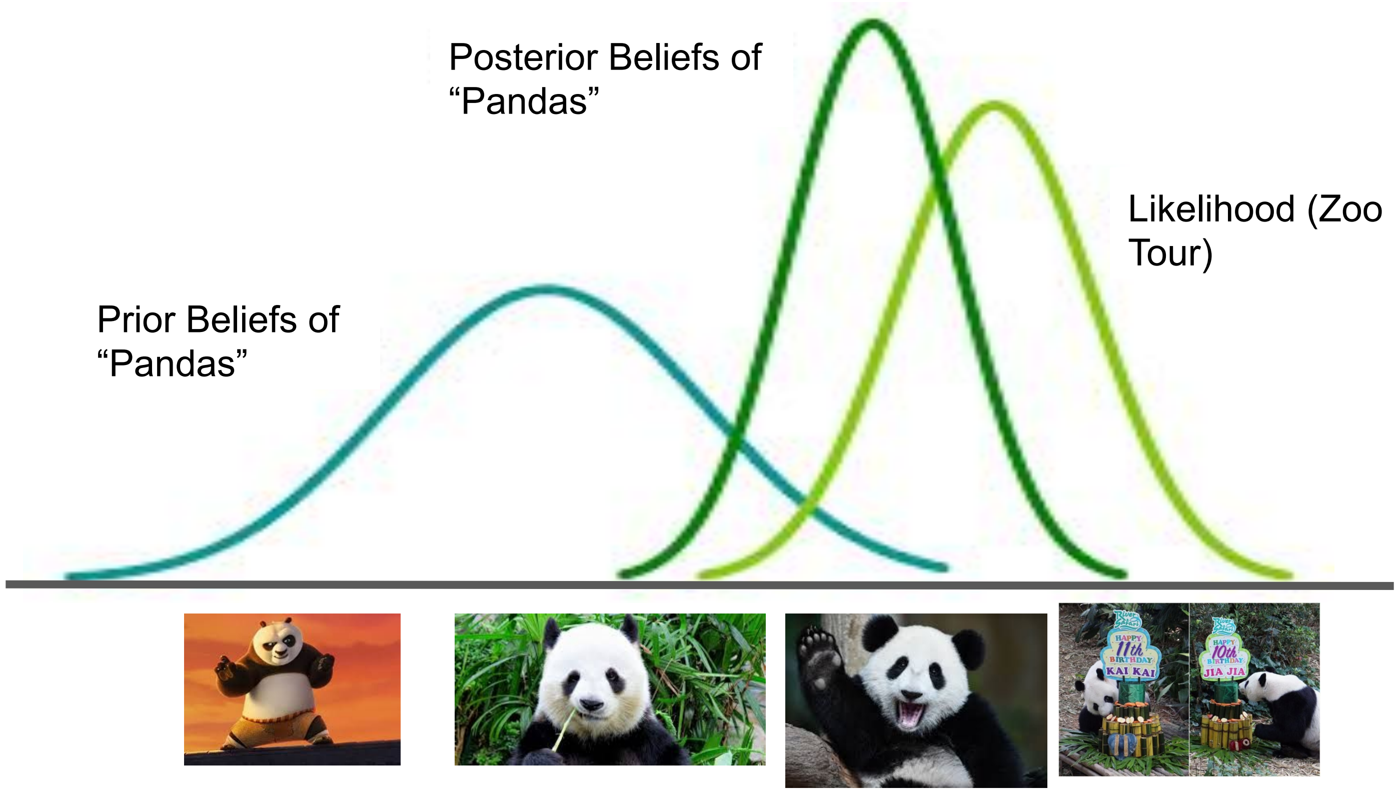
Bayes' Rule

- By definition of conditional probability:

$$\boxed{p(h|D)} = \frac{p(h,D)}{p(D)} = \frac{\boxed{p(D|h)}\boxed{p(h)}}{p(D)}$$

  - P(h): prior probability of hypothesis h
  - P(h|D): posterior probability of h given evidence D
  - P(D|h): likelihood of D given h
  - P(D): prior probability of evidence D

Reverse Probability

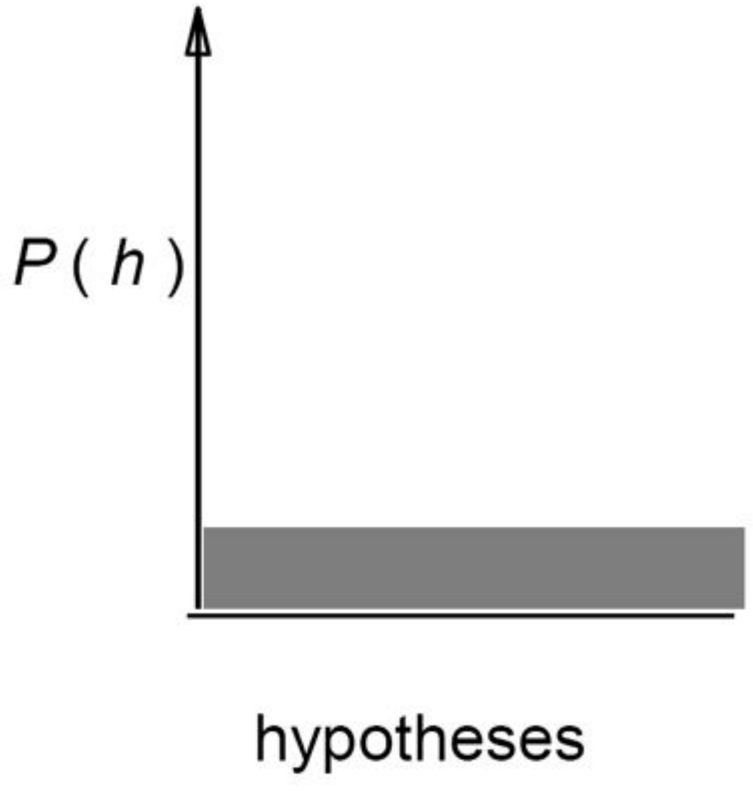

Posterior Beliefs of "Pandas"
Likelihood (Zoo Tour)
Prior Beliefs of "Pandas"
HAPPY 11th BIRTHDAY KAI KAI
HAPPY 10th BIRTHDAY JIA JIA

Example: BT5153 Student Model

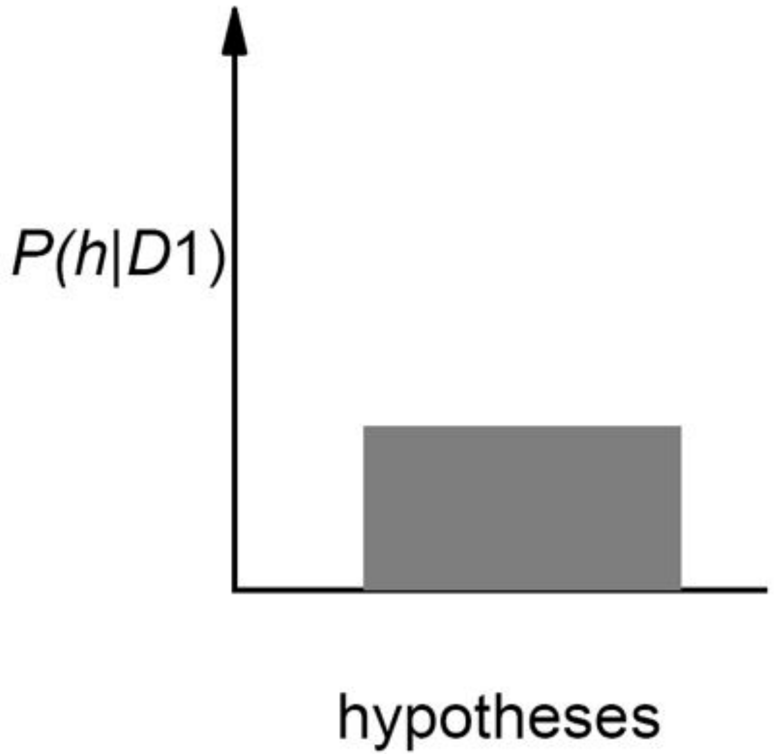
- Given:
  - The faculty knows that taking BT5153 causes that you stay in library **80%** of the time
  - Prior probability of any MSBA student taking BT5153 is **1/100**
  - Prior probability of any MSBA student staying in library is 1/10
- If a MSBA student stay in the library, what is the probability he/she took the BT5153?
  - D(Evidence) - Stay in Library   h(hypothesis) - Taking BT5153

$$p(h|D) =$$

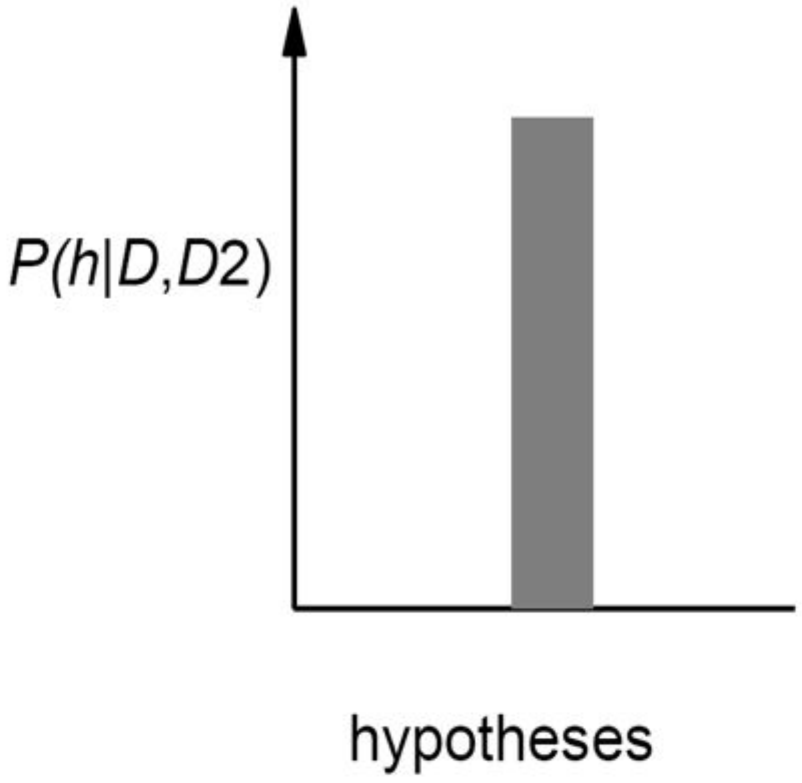
Evolution of Posterior Probabilities



Evolution of Posterior Probabilities



Evolution of Posterior Probabilities



Maximum A Posteriori

Find the most probable hypothesis given the training data (Maximum A Posteriori hypothesis H_{map})

$$h_{\text{MAP}} = \arg\max_{h \in \mathcal{H}} P(h|\mathcal{D})$$

$$= \arg\max_{h \in \mathcal{H}} \frac{P(\mathcal{D}|h)P(h)}{P(\mathcal{D})}$$

$$h_{\text{MAP}} = \arg\max_{h \in \mathcal{H}} P(\mathcal{D}|h)\boxed{P(h)}$$

Prior encodes the knowledge /preference

MAP vs MLE

- MLE: Finding a hypothesis h that maximizes the likelihood of the training data

$$h_{ML} = argmax_{h \in (H)} P(D|h)$$

- MAP: Finding a hypothesis h that maximizes the posterior probability given the training data

$$h_{MAP} = argmax_{h \in \mathcal{H}} P(h|D)$$

- When will MLE and MAP give the same results?

Classification Using Bayes Rule

Given multiple attribute values $\mathbf{d} = [d_1, d_2, \ldots, d_n]$, what is the most probable value of the target variable?

features

$$h_{MAP} = \underset{h_i \in \mathbb{H}}{\operatorname{argmax}}\, p(h_i | d_1, d_2, \cdots, d_n)$$

$$= \underset{h_i \in \mathbb{H}}{\operatorname{argmax}} \frac{p(h_i) p(d_1, d_2, \cdots, d_n | h_i)}{p(d_1, d_2, \cdots, d_n)}$$

$$= \underset{h_i \in \mathbb{H}}{\operatorname{argmax}}\, p(h_i) p(d_1, d_2, \cdots, d_n | h_i)$$

Problem: too much data needed to estimate $p(d_1,d_2,\ldots,d_n|h_i)$ when n is large

Curse of Dimensionality

Curse of Dimensionality

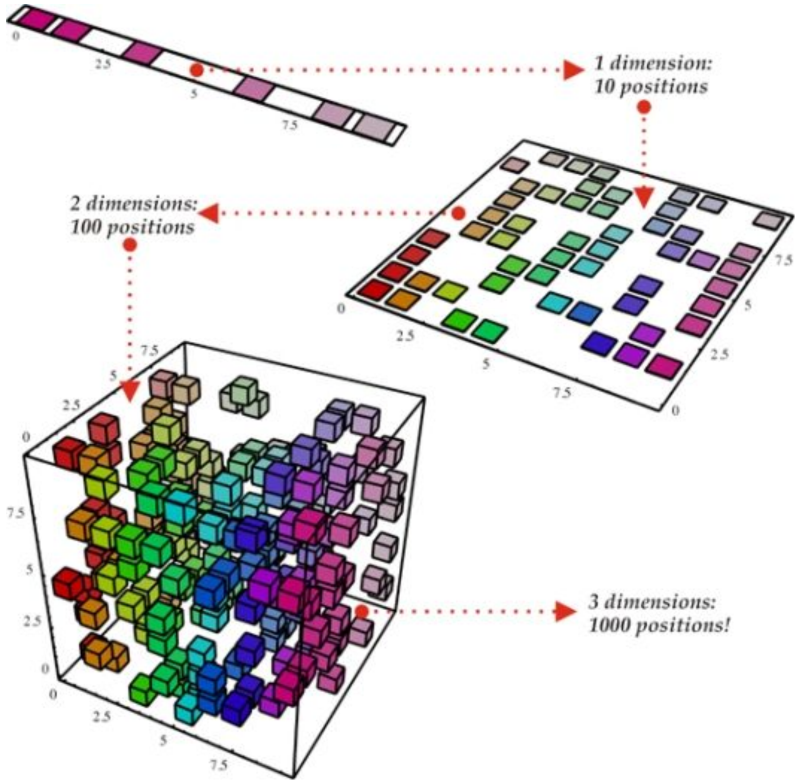
- What is the usual way to compute $p(d_1, d_2, \ldots, d_n | \hbar_i)$?

- Chain rule:

$$p(d_1, d_2, \ldots, d_n | \hbar_i) = p(d_1 | \hbar_i) \times p(d_2 | \hbar_i, d_1) \times p(d_3 | \hbar_i, d_1, d_2) \times \ldots \times p(d_n | \hbar_i, d_1, d_2, \ldots, d_{n-1})$$

Curse of Dimensionality

When the dimensionality of problem increases, the volume of the space becomes so large, you need exponentially more points to sample the space at the same density. This phenomenon is sometimes referred as the curse of dimensionality.



https://haifengl.wordpress.com/2016/02/29/there-is-no-big-data-in-machine-learning/

Naive Bayes Classifier

Naïve Bayes Classifier

- Hard to estimate $p(\mathbf{d}|h_i)$ for high dimensional data $\mathbf{d}$
- Conditional Independence assumption
  - All attributes are **conditionally independent**
  - assumption often *violated in practice*
  - even then, it usually works well
- Successful application: classification of text documents, Diagnosis

Conditional Independence

- $p(d_1, d_2, \ldots, d_n | \hbar_i) = p(d_1 | \hbar_i) \times p(d_2 | \hbar_i, d_1) \times p(d_3 | \hbar_i, d_1, d_2) \times \ldots \times p(d_n | \hbar_i, d_1, d_2, \ldots, d_{n-1})$

- **Naïve Bayes** (conditionally independence) assumption : attributes are **independent**, given the class
  - $p(d_2 | \hbar_i, d_1) = p(d_2 | \hbar_i)$
  - $p(d_3 | \hbar_i, d_1, d_2) = p(d_3 | \hbar_i)$

    $\ldots,$
  - $p(d_n | \hbar_i, d_1, d_2, \ldots, d_{n-1}) = p(d_n | \hbar_i)$
  - $\boldsymbol{p(d_1, d_2, \ldots, d_n | \hbar_i) = p(d_1 | \hbar_i) * p(d_2 | \hbar_i) \ldots p(d_n | \hbar_i)}$

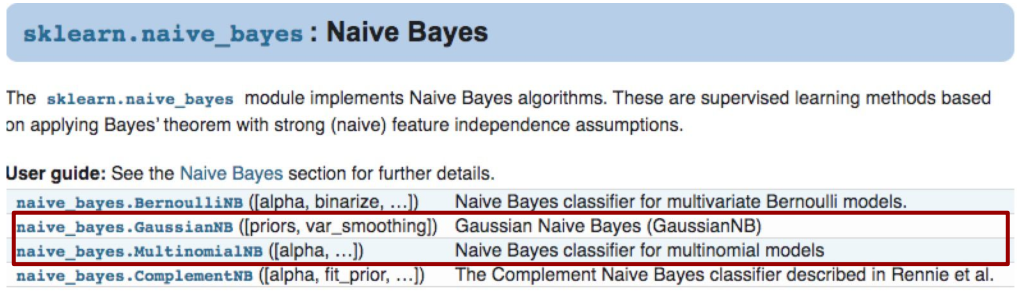
Naïve Bayes Classifier

Based on Bayes' rule + assumption of conditional independence

$$\begin{aligned} h_{NB} &= \underset{h_i \in \mathbb{H}}{\operatorname{argmax}}\, p(h_i | d_1, d_2, \cdots, d_n) \\ &= \underset{h_i \in \mathbb{H}}{\operatorname{argmax}}\, \frac{p(h_i) p(d_1, d_2, \cdots, d_n | h_i)}{p(d_1, d_2, \cdots, d_n)} \\ &= \underset{h_i \in \mathbb{H}}{\operatorname{argmax}}\, p(h_i) p(d_1, d_2, \cdots, d_n | h_i) \\ &= \underset{h_i \in \mathbb{H}}{\operatorname{argmax}}\, p(h_i) \prod_{j=1}^{n} p(d_j | h_i) \end{aligned}$$

Learning a Naïve Bayes Classifier

- We need to estimate $p(h_i)$, $p(d_1|h_i)$, $p(d_2|h_i)$,…,$p(d_n|h_i)$ from data.
- Then $h_{NB} = \underset{h_i \in \mathbb{H}}{\operatorname{argmax}}\, p(h_i) \prod_{j=1}^{n} p(d_j|h_i)$
- How to estimate?
  - Simplest: standard estimate from statistics
    - estimate probability from sample proportion
    - If **d(feature)** is continuous, Gaussian Distribution
    - If **d(feature)** is discrete, Multinomial Distribution



sklearn.naive_bayes : Naive Bayes

The sklearn.naive_bayes module implements Naive Bayes algorithms. These are supervised learning methods based on applying Bayes' theorem with strong (naive) feature independence assumptions.

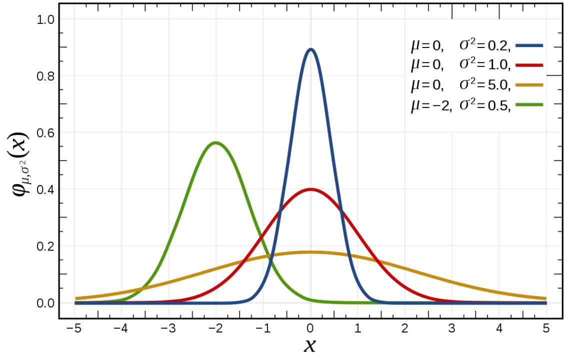
User guide: See the Naive Bayes section for further details.

naive_bayes.BernoulliNB ([alpha, binarize, …])	Naive Bayes classifier for multivariate Bernoulli models.
naive_bayes.GaussianNB ([priors, var_smoothing])	Gaussian Naive Bayes (GaussianNB)
naive_bayes.MultinomialNB ([alpha, …])	Naive Bayes classifier for multinomial models
naive_bayes.ComplementNB ([alpha, fit_prior, …])	The Complement Naive Bayes classifier described in Rennie et al.

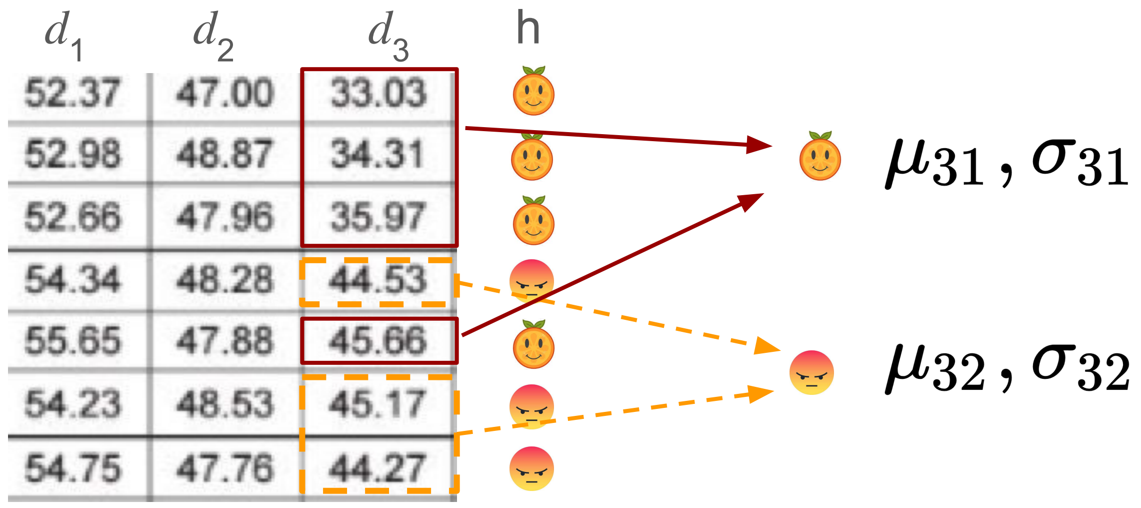
Gaussian NB

- Estimate $p(d_j|h_i)$

$$p(d_j|h_i) = \mathcal{N}(d_j|\mu_{ji}, \sigma^2_{ji})$$



Gaussian Distribution



d_1	d_2	d_3	h
52.37	47.00	33.03	🍊
52.98	48.87	34.31	🍊
52.66	47.96	35.97	🍊
54.34	48.28	44.53	😠
55.65	47.88	45.66	🍊
54.23	48.53	45.17	😠
54.75	47.76	44.27	😠

🍊 μ_{31}, σ_{31}

😠 μ_{32}, σ_{32}

Multinomial NB

- Estimate $p(d_j|h_i)$
  - Simply count the frequencies in the data

- **Parameter Estimation:**

$$p(d_j = v|h_i) = \frac{\sum_{k=1}^{n} \delta(d_{kj}, v)\delta(y_k, h_i)}{\sum_{k=1}^{n} \delta(y_k, h_i)}$$

Count Samples

1 the label is yk
2 the j-th feature is v

Count Samples

1 the label is yk

$$\delta(y_k, h_i) = \begin{cases} 1, & \text{if } y_k = h_i \\ 0, & \text{otherwise} \end{cases}$$

$$\delta(d_{kj}, v) = \begin{cases} 1, & \text{if } d_{kj} = v \\ 0, & \text{otherwise} \end{cases}$$

Example: Go to Library or Not

Day	Outlook	Temperature	Humidity	Wind	Go to Library
Day1	Sunny	Hot	High	Weak	Yes
Day2	Sunny	Hot	High	Strong	Yes
Day3	Overcast	Hot	High	Weak	No
Day4	Rain	Mild	High	Weak	No
Day5	Rain	Cool	Normal	Weak	No
Day6	Rain	Cool	Normal	Strong	Yes
Day7	Overcast	Cool	Normal	Strong	No
Day8	Sunny	Mild	High	Weak	Yes
Day9	Rain	Cool	Normal	Weak	No

Example: Go to Library or Not

- Based on the history data in the table, predict whether you will go to library when Outlook == sunny, Temp==cool, Hum==High, Wind==Strong

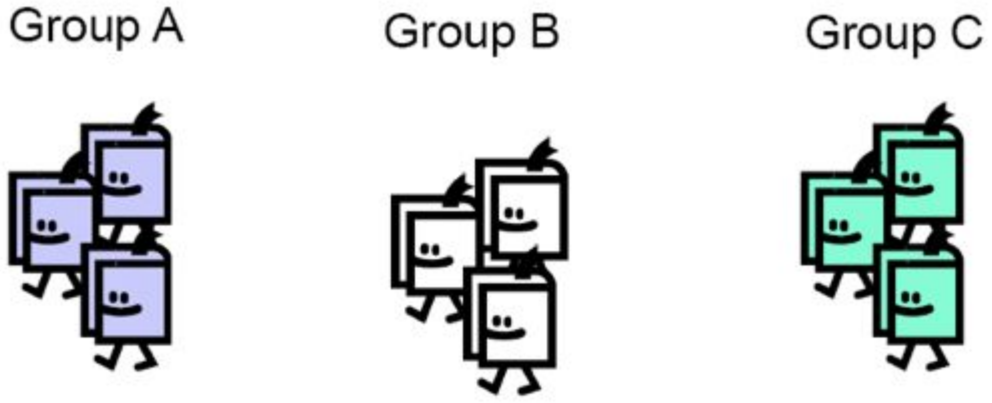
Comments on Naïve Bayes Learner

- One of the most practical learning methods, along with decision trees, neural networks, etc.
- Requires :
  - Moderate or large training data set
  - Attributes that describe instances should be conditionally independent given the classification.
- Successful applications include diagnosis and text classification.
- It may not estimate probabilities accurately when independence is violated, it still picks correct category

Text Classification using Naive Bayes

Text Classification

- Given text of newsgroup article, guess which newsgroup it is taken from.
- Naïve Bayes turns out to work well on this application.
- Key issue : how do we represent examples? what are the attributes?



Text Classification

- Class h_j: Binary classification (+/−) or multiple classes possible H (j = 1,2, …, k)

- How about attributes?

Example

- 1000 training documents that someone has 700 classified as “dislikes” (h_0) and 300 classified as “likes” (h_1).
- Suppose document 1 is “**This is a very interesting document**”

$$h_{NB} = \max_{h_j \in \{\text{like,dislike}\}} p(h_j) \times p(d_1 = \text{this}|h_j) \times$$
$$p(d_2 = \text{is}|h_j) \cdots \times p(d_6 = \text{document}|h_j)$$

$p(like)$=300/1000=0.3

$p(dislike)$=1−$p(like)$=0.7

- How to estimate $p(d_i|h_j)$?

Parameter Estimation

- Learning by Maximum Likelihood Estimate
  - Simply count the frequencies in the data

$$p(d_i = w|h_j) = \frac{count(w,h_j)}{\sum_{d \in \mathcal{V}} count(d_i,h_j)}$$

The count of the specific word di=w in the mega-doc

The count of total words in the mega-doc

  - Create a mega-document for class hj by concatenating all the docs in this class
  - Compute the frequency of the word w in the mega-document

New Word Problem

- What if some words do not exit a certain category: h

$$p(d_i = newword|h) = 0$$

- The predicted likelihood will be zero

$$p(\mathbf{d}|h_i) = p(d_1|h_i)*p(d_2|h_i)...p(d_n|h_i) = p(d_1|h_i)*p(d_2|h_i)...*0*p(d_n|h_i) = 0$$

How to Solve it?

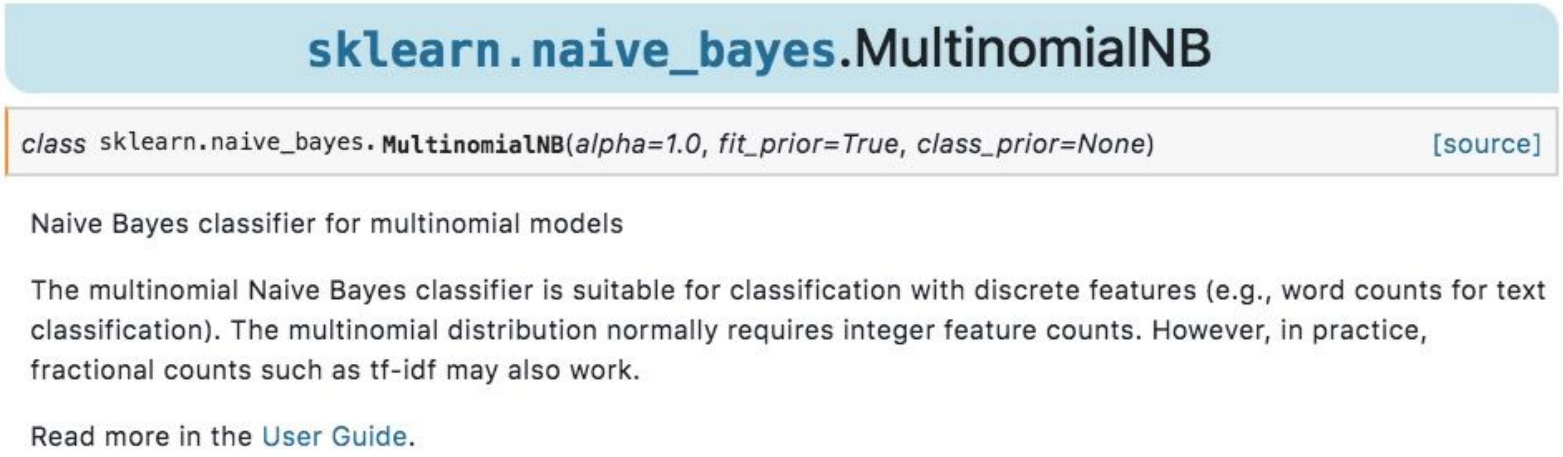
Additive Smoothing

smoothing parameter

$$p(d_i = w|h_j) = \frac{count(w,h_j)+\alpha}{\sum_{d\in\mathbb{V}} count(d_i,h_j)+\alpha V}$$

Vocab. Size

- A weighted estimation of
  - Relative frequency: $\frac{count(w,h_j)}{\sum_{d\in\mathcal{V}} count(d_i,h_j)}$
  - Uniform probability: $\frac{1}{V}$



sklearn.naive_bayes.MultinomialNB

class `sklearn.naive_bayes.`**`MultinomialNB`**(*alpha=1.0*, *fit_prior=True*, *class_prior=None*) [source]

Naive Bayes classifier for multinomial models

The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work.

Read more in the User Guide.

Bayesian Machine Learning

Model Parameters as Hypothesis

- Bayesian theory
    - Evidence will be data that you have: D={x1, x2, …, xn). Then, the **likelihood** is given:

$$p(D|\theta)$$

    - Specify a prior:

$$p(\theta)$$

    - Compute the **posterior**:

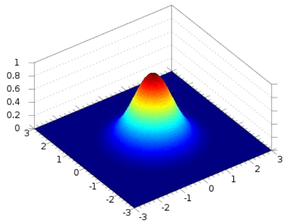
$$p(\theta|D)$$

Prior Beliefs

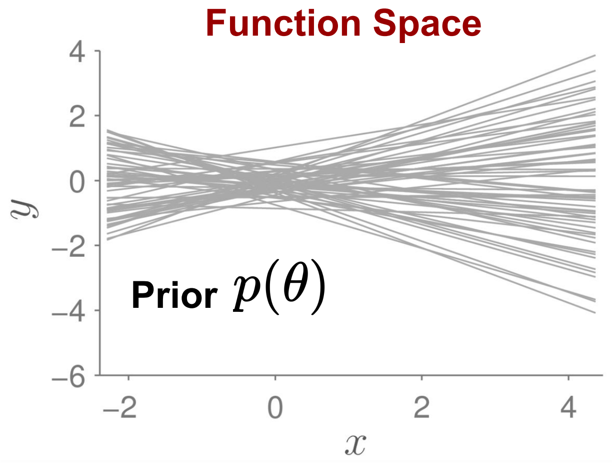
For linear regression, prior beliefs about models are represented by a distribution over the parameters, specifying which models we think are plausible before observing any data.

Linear Regression Model: $y = \mathbf{x}^T\theta = \theta_0 + \theta_1 x_0$

$$\theta \sim \mathcal{N}(\mathbf{0}, 0.4^2\mathbf{I})$$



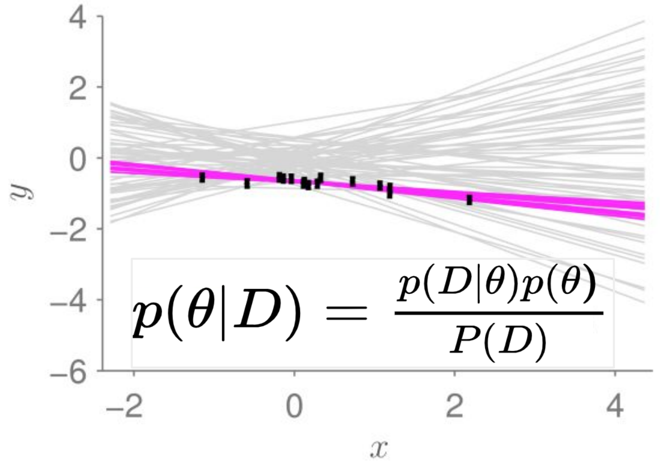
Source: wikipedia



Posterior Beliefs

After observing data D, we can formulate posterior beliefs about model parameters. Based on bayes rules:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{P(D)}$$



Posterior Beliefs

Easy

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{P(D)}$$

Usually very complicated and intractable

$$p(D) = \int_\theta p(D|\theta)p(\theta)d\theta$$

How to address this **normalisation factor issue**?

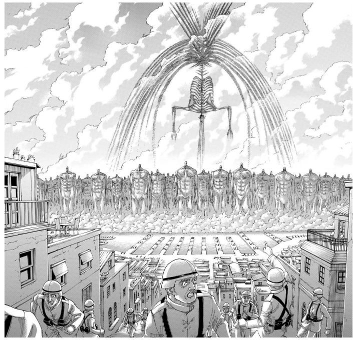
- Conjugate Prior
- Markov Chain Monte Carlo
- Variational Inference

Frequentist Vs Bayesian

- Frequentist: model parameters are fixed numbers. There exists a perfect model. (such as MLE and MAP)

- Bayesian: No perfect model. Model parameters are random variables that have distributions. $p(\theta|D) = \frac{p(D|\theta)p(\theta)}{P(D)}$

Frequentist: Levi Ackerman is the "strongest" character.



Bayesian: Founding titan who can call infinite number of titans.

Bayesian Predictions

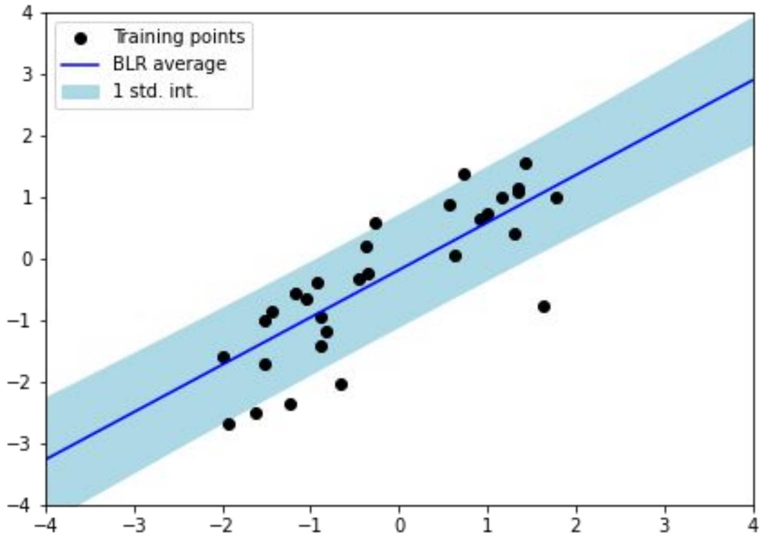
For prediction, bayesian models marginalize over the posterior. Model parameters are random variables that have distributions instead of fixed point values.

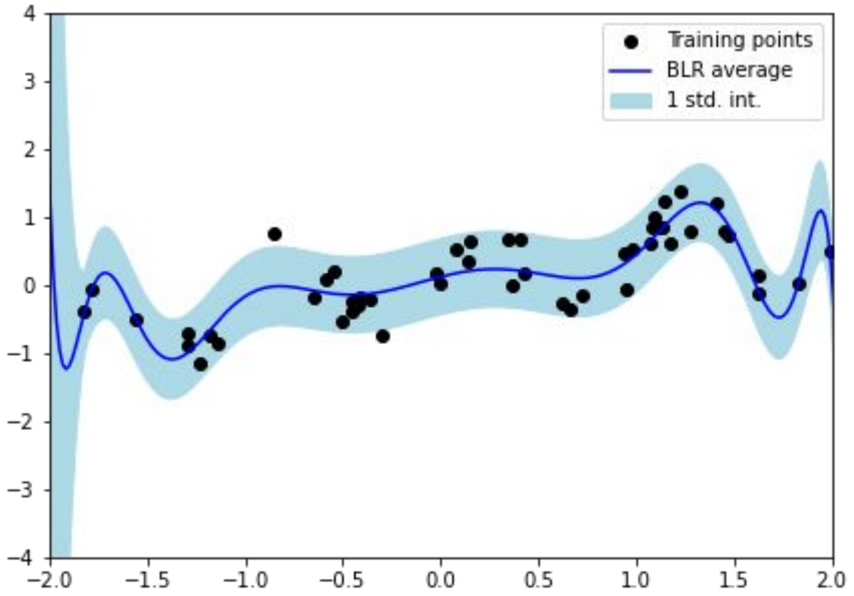
Given a new data sample, we can predict the value by ensembling all the possible models.

$$p(y_{test}|D, x_{test}) = \int_\theta p(\theta|D)p(y_{test}|x_{test}, \theta)d\theta$$

Similar to Ensemble

Bayesian Predictions





Bayes-Frequentist Debate

- To analyze subjective beliefs in a principled way: use Bayesian Methods
- To design methods with long run frequency guarantees: use Frequentist Methods
- In low-dimensional models with enough data, their performances should be similar.
- Bayesian methods often have poor behavior in high dimensional data.

Comparison of Frequentist and Bayesian:
https://arxiv.org/pdf/1411.5018.pdf