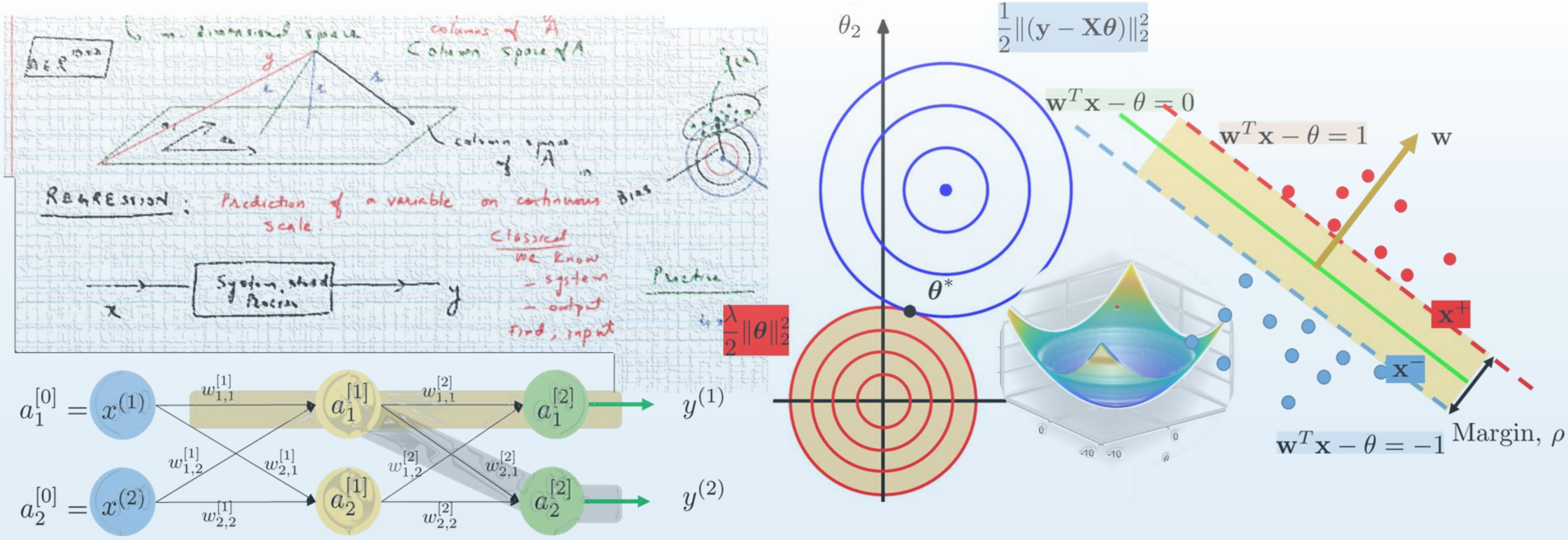

REGRESSION: Prediction of a variable on continuous scale.
System, model Process
x
y
Classical
Practice
θ_2
$\frac{1}{2}\|(\mathbf{y}-\mathbf{X}\boldsymbol{\theta})\|_2^2$
$\frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2$
$\boldsymbol{\theta}^*$
$\mathbf{w}^T\mathbf{x}-\theta=0$
$\mathbf{w}^T\mathbf{x}-\theta=1$
$\mathbf{w}^T\mathbf{x}-\theta=-1$
$\mathbf{w}$
$\mathbf{x}^+$
$\mathbf{x}^-$
Margin, ρ
$a_1^{[0]}=x^{(1)}$
$a_2^{[0]}=x^{(2)}$
$a_1^{[1]}$
$a_2^{[1]}$
$a_1^{[2]}$
$a_2^{[2]}$
$y^{(1)}$
$y^{(2)}$

Machine Learning

Bayesian Learning: MAP and ML Estimation, Naïve Bayes Classifier

School of Science and Engineering

https://www.zubairkhalid.org/ee514_2025.html

LUMS
A Not-for-Profit University

Why Probability Theory is Crucial for ML?

Probability – Significance:

At the core of AI’s ability to make **decisions, predict outcomes**, and **learn from data** lies a foundational pillar:

PROBABILITY

- By leveraging probability, ML systems gain the ability to
  - navigate uncertainty
  - make data-driven predictions, and
  - adapt effectively to ever-changing environments

Why Probability Theory is Crucial for ML?

Probability – Significance:

- **Handling Uncertainty**
  - Real-world data is noisy and incomplete
  - Probability theory provides a mathematical framework to reason about uncertainty
- **Foundation for Probabilistic Models**
  - Core of models like Bayesian networks, Hidden Markov Models, and Gaussian Mixture Models
  - Allows us to encode prior knowledge and update beliefs based on evidence
- **Bayesian Inference**
  - Key in machine learning for parameter estimation and model selection
  - Supports decision-making under uncertainty

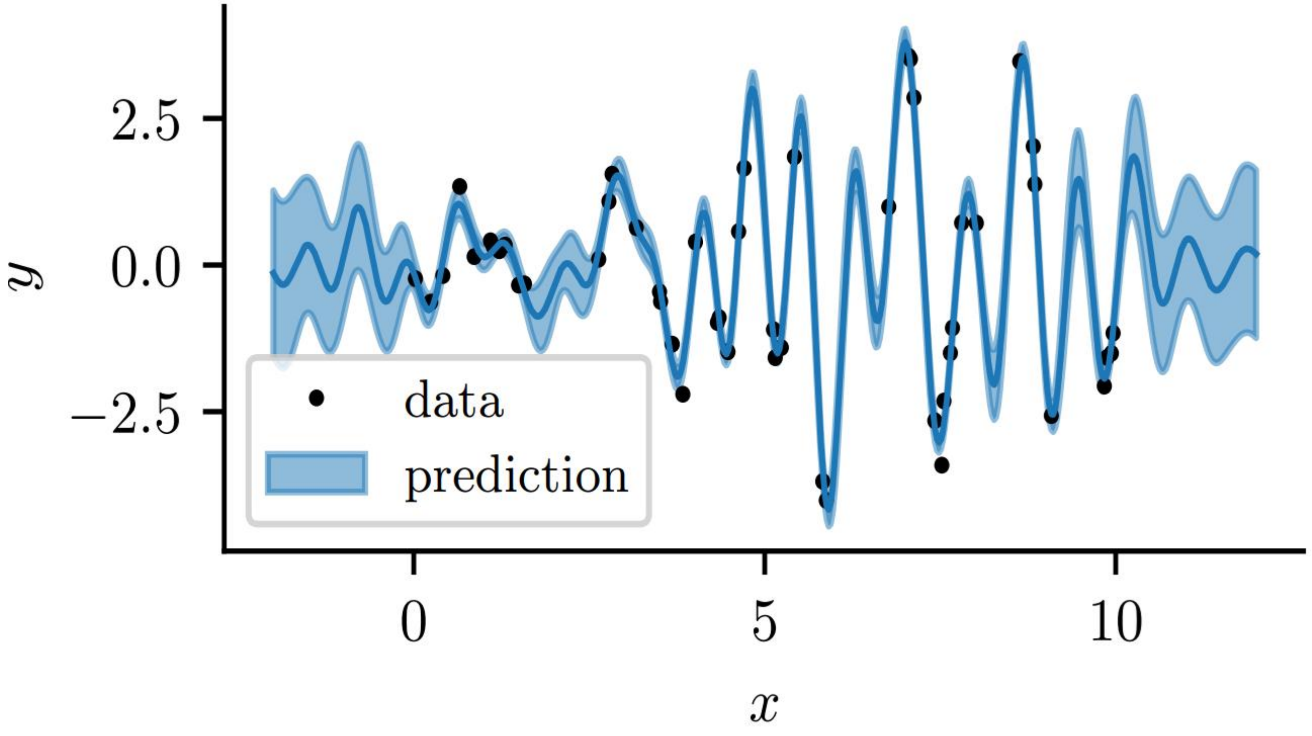
Probabilistic ML

Examples – Uncertainty Matters:

Given data of N observations $D = \{(x_i, y_i)\}_{i=1}^{N}$

Find the best fit model f that depends on the parameters θ: $y = f(x, \theta)$

- Uncertainties:
  - Measurement noise in the data
  - Uncertainty in the values of estimated parameters
  - Uncertainty in the structure of the model
    - E.g., polynomial fit or neural network

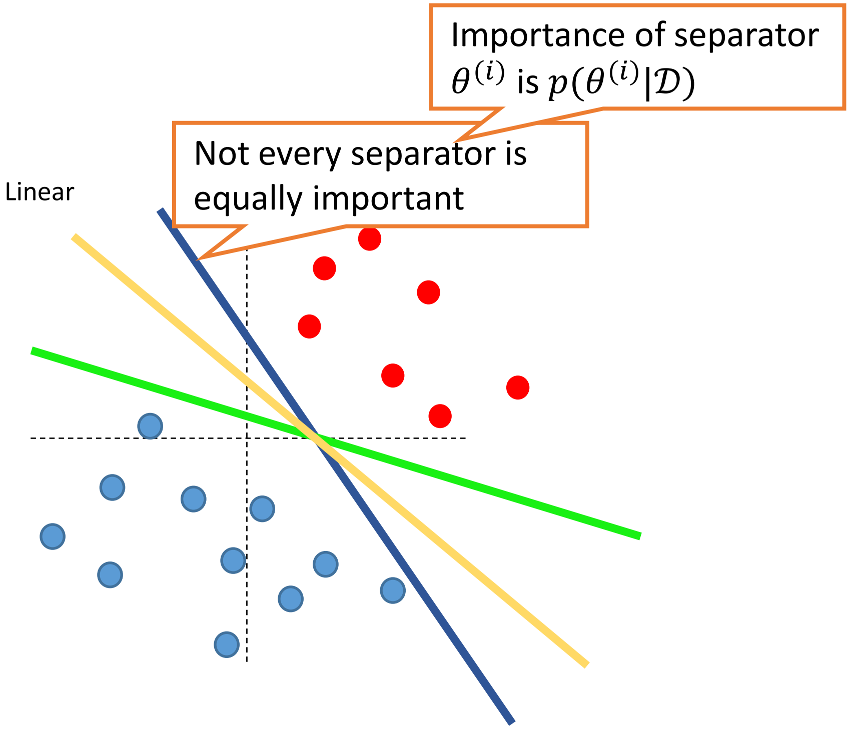


Probabilistic Machine Learning – Advanced Topics - Figure 18.23

Probabilistic ML

Uncertainty Types – Epistemic or Model Uncertainty – Example:

- Epistemic uncertainty is related to the model: both the structure and the parameters



Importance of separator $\theta^{(i)}$ is $p(\theta^{(i)}|\mathcal{D})$

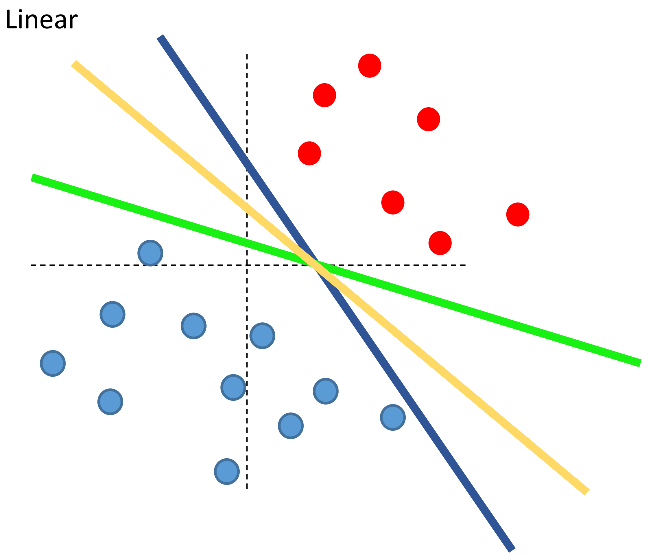
Not every separator is equally important

Linear separator but we have *uncertainty* about the weights

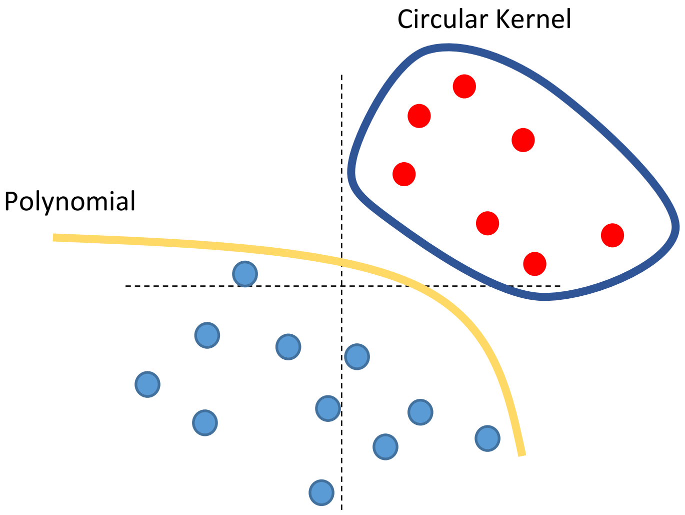
- Consider a linear classification model for 2-dim inputs
- Classifier weight will be a 2-dim vector $\theta = [\theta_1, \theta_2]$
- Its posterior will be some 2-dim distribution $p(\theta|\mathcal{D})$
- Sampling from this distribution will generate 2-dim vectors
- Each vector will correspond to a linear separator (left fig)
- Thus, the posterior in this case is equivalent to a "collection" or "ensemble" of weights, each representing a different linear separator

Probabilistic ML

Uncertainty Types – Epistemic or Model Uncertainty:



Linear separator but we have *uncertainty* about the weights



Uncertainty about the model structure as well

Probabilistic approach to formulate model uncertainty:

Model structure or parameter distribution conditioned on data, for example:

$$p(\theta|D)$$

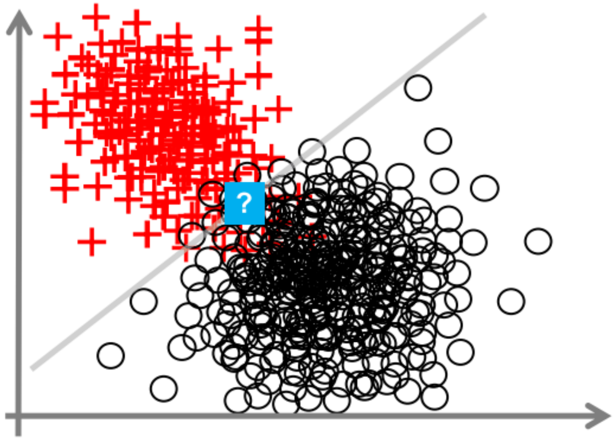
Also referred to as 'Posterior distribution' and is hard to compute, in general, but we will look at some methods to compute this.

Model uncertainty is usually reducible with the increase in the amount of data.

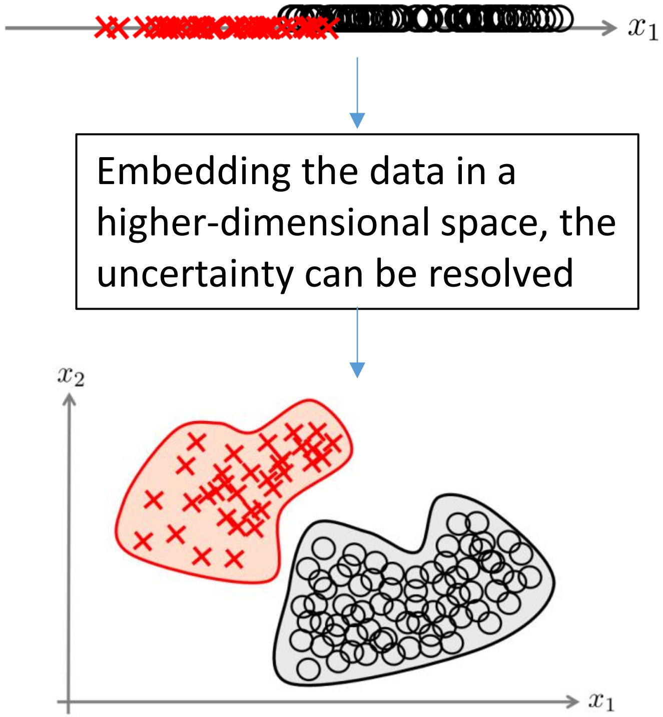
Probabilistic ML

Uncertainty Types – Aleatoric or Data Uncertainty:

- Aleatoric uncertainty is related to the data: noisy measurements, overlapping of classes, incorrect labelling, etc.



Aleatoric uncertainty: the prediction at the query point is uncertain



Embedding the data in a higher-dimensional space, the uncertainty can be resolved

Probabilistic approach to formulate data uncertainty:

The distribution of data being modeled conditioned on model parameters and other inputs, for example:

$$p(y|\theta, x)$$

Data uncertainty is mostly irreducible (even with infinite amount of data).

Sometimes reduced by adding more features or using more complex model

Probabilistic ML

Overview:

Review the foundations of machine learning from the probabilistic and Bayesian perspective

We will answer *fundamental* questions:

- How do we set up a probabilistic model for a given machine learning problem?
- How do we quantify uncertainty in the process of estimation and prediction of parameters?
- What are the estimation and inference algorithms to learn the parameters of the model?

Outline

- Bayesian Learning Framework
  - MAP Estimation
  - ML Estimation
- Linear Regression as Maximum Likelihood Estimation
- Naïve Bayes Classifier

Reference: Chapter 6 (Machine Learning by Tom Mitchell)

Bayesian Learning Framework

Overview:

- In machine learning, the idea of Bayesian Learning is to use **Bayes Theorem** to find the hypothesis function or parameters of the model.

Example: Test the fairness of the coin!

Frequentist Statistics:

- Conduct trials and observe heads to compute the probability P(H).
- Confidence of estimated P(H) increases with the number of trials.
- In frequentist statistics, we do not use prior (**valuable**) information to improve our Hypothesis. For example, we have information that the coins are not made biased.

Bayesian Learning:

- Assume that P(H)=0.5 (prior or beliefs or past experiences).
- Adjust the belief P(H) according to your observations from the trials.
- Better hypothesis by combining our beliefs and observations.

- Each training data point contributes to the estimated probability that a hypothesis is correct.
    - More **flexible** approach as compared to learning algorithms that eliminate a given hypothesis inconsistent with any single data point.

Bayesian Learning Framework

Bayes Rule:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Rewriting it using the ML notation:

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

- $P(\theta|D)$ is called the posterior
- $P(D|\theta)$ is called the likelihood
- $P(\theta)$ is called the prior
- $P(D)$ is called the evidence

Bayesian Learning Framework

Maximum Likelihood Estimation:

$$P(\theta|D) = \frac{P(D|\theta) \cdot P(\theta)}{P(D)} = \frac{P(D|\theta) \cdot P(\theta)}{\int_\theta P(D|\theta) \cdot P(\theta) d\theta}$$

Given a dataset D, find the parameters θ that maximize the likelihood of the data.

$$\theta_{\text{MLE}} = \arg\max_\theta P(D|\theta)$$

For example, given a linear regression problem setup, we set the likelihood as normal distribution and find the parameters θ that maximize the likelihood of the data.

Bayesian Learning Framework

Maximum A Posteriori Estimation:

$$P(\theta|D) = \frac{P(D|\theta) \cdot P(\theta)}{P(D)} = \frac{P(D|\theta) \cdot P(\theta)}{\int_\theta P(D|\theta) \cdot P(\theta) d\theta}$$

Given a dataset D, find the parameters θ that maximize the posterior of the data considering both the likelihood and the prior.

$$\theta_{\text{MAP}} = \arg\max_\theta P(\theta|D) = \arg\max_\theta P(D|\theta) \cdot P(\theta)$$

For example, given a linear regression problem, we assume prior over the parameters θ and find the parameters θ that maximize the posterior of the data.

Bayesian Learning Framework

Main Challenge in Bayesian Inference:

$$P(\theta|D) = \frac{P(D|\theta) \cdot P(\theta)}{P(D)} = \frac{P(D|\theta) \cdot P(\theta)}{\int_\theta P(D|\theta) \cdot P(\theta) d\theta}$$

Compute the evidence $P(D)$ is intractable in most cases. It involves integrating over all possible values of θ. Thus, computing the posterior $P(\theta|D)$ is intractable in most cases.

Bayesian Learning Framework

Overview:

Supervised Learning Formulation:

Data: $D = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)\} \subseteq \mathcal{X}^d \times \mathcal{Y}$

We call the set of possible functions or candidate models (linear model, neural network, decision tree, etc.) "the hypothesis class".

Denoted by $\mathcal{H}$.

For a given problem, we wish to select **best** hypothesis (machine) $h \in \mathcal{H}$.

- In Bayesian learning, the **best** hypothesis is the **most probable** hypothesis, given the data D and initial knowledge about the prior probabilities of the various hypotheses in H.

- We can use Bayes theorem to determine the probability of a hypothesis based on its prior probability, the observed data and the probabilities of observing various data given the hypothesis.

Bayesian Learning Framework

Maximum a Posterior (MAP) Hypothesis or Estimation:

- Find h that maximizes the distribution $P(h \mid \mathcal{D})$.

Using Bayes theorem, we can write this as

Likelihood function

$$P(h \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid h)\, P(h)}{P(\mathcal{D})}$$

Posterior

Prior

- The prior probability $P(h)$ is the probability that the hypothesis holds before looking at the training data. It refelcts our prior knowledge about candidate hypothesis h.
- $P(\mathcal{D})$ is the probability of the training data given no information about hypothesis, that is, independent of h.
- $P(\mathcal{D} \mid h)$, likelihood function, quantifies the probability of observing $\mathcal{D}$ given hypothesis h.
- $P(h \mid \mathcal{D})$, posterior probability, quantifies the influence of data on our prior probability or our confidence that h holds after observing the data.

Bayesian Learning Framework

Maximum a Posterior (MAP) Hypothesis or Estimation:

- Find h that maximizes the distribution $P(h \mid \mathcal{D})$.

- Maximizing posterior probability yields

$$h_{\text{MAP}} = \underset{h \in \mathcal{H}}{\text{maximize}}\, P(h \mid \mathcal{D}) = \underset{h \in \mathcal{H}}{\text{maximize}}\, \frac{P(\mathcal{D} \mid h)\, P(h)}{P(\mathcal{D})}$$

$$h_{\text{MAP}} = \underset{h \in \mathcal{H}}{\text{maximize}}\, P(\mathcal{D} \mid h)\, P(h)$$

Interpretation:

- We begin with prior distribution of hypothesis.

- Using candidate hypothesis, we determine probability data given hypothesis.

- Using these two, we update posterior probability distribution.

Bayesian Learning Framework

Maximum Likelihood (ML) Hypothesis or Estimation:

- If each hypthesis $h \in \mathcal{H}$ is equally probable, we can reformulate MAP hypothesis as by maximizing the probability of data given hypothesis. This is termed as maximum likelihood hypothesis given by

$$h_{\text{MAP}} = \underset{h \in \mathcal{H}}{\text{maximize}}\, P(\mathcal{D} \mid h)\, P(h) \longrightarrow h_{\text{ML}} = \underset{h \in \mathcal{H}}{\text{maximize}}\, P(\mathcal{D} \mid h)$$

Maximizing Likelihood function

Example:

- Predict the face side (head, H or tail, T) of the loaded coin.

- If x is our event, we want to learn P(x=H) or P(x=T)=1 – P(x=H).

- Data-set: outcomes of n events. (x_1=H, x_2=T, x_3=H, x_4=H,….)

- Intuitive prediction: count the number of heads and divide it by n. If this quantity is greater than 0.5, head is more probable.

- Let's apply ML estimation to this problem.

Bayesian Learning Framework

Maximum Likelihood (ML) Hypothesis or Estimation:

Example:

- We want to estimate $P(x = H) = 1 - P(x = T)$ and therefore hypothesis space can be parameteterized by a single variable θ such that $P(x = H) = \theta$, that is, $P(\mathcal{D} \mid h) = P(\mathcal{D} \mid \theta)$.
- Assuming independence between events, we have $P(\mathcal{D} \mid h) = \prod_{i=1}^{n} p(x_i \mid \theta)$
- We use log of the likelihood function due to notational convenience and since the product of probabilities can be very small:

$$\log P(\mathcal{D} \mid h) = \log \prod_{i=1}^{n} p(x_i \mid \theta) = \sum_{i=1}^{n} \log p(x_i \mid \theta)$$

- ML estimate is given by

$$h_{\mathrm{ML}} = \underset{h \in \mathcal{H}}{\text{maximize}}\, P(\mathcal{D} \mid h) \qquad \Rightarrow \theta_{\mathrm{ML}} = \underset{\theta}{\text{maximize}} \sum_{i=1}^{n} \log p(x_i \mid \theta)$$

The maximum likelihood estimation maximizes the log-likelihood.

Bayesian Learning Framework

Maximum Likelihood (ML) Hypothesis or Estimation:

Example:

- We can solve this analytically.
- If number of heads in the data is n_H.

$$\theta_{\text{ML}} = \underset{\theta}{\text{maximize}} \ \big(n_H \log \theta + (n - n_H) \log(1 - \theta)\big)$$

- Derivative with respect to θ yields

$$\frac{n_H}{\theta} - \frac{n - n_H}{1 - \theta} = 0$$

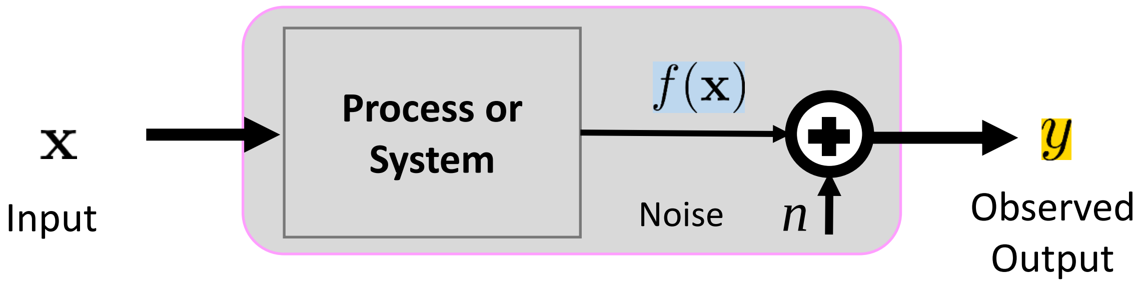
$$\theta_{\text{ML}} = \theta = \frac{n_H}{n}$$

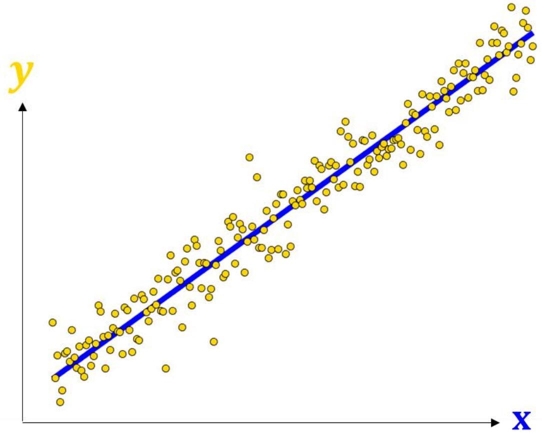
Outline

- Bayesian Learning Framework
    - MAP Estimation
    - ML Estimation
- Linear Regression as Maximum Likelihood Estimation
- Naïve Bayes Classifier

Linear Regression as ML Estimation

Regression:





$$y = f(\mathbf{x}) + n$$

- Assume noise is i.i.d. Gaussian distributed: $n \sim N(0, \sigma^2)$.
- $y_i = f(\mathbf{x_i}) + n_i$ is also Gaussian distributed: $y_i \sim N(f(\mathbf{x_i}), \sigma^2)$.

Linear Regression:

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

(Assuming bias term is included in the formulation)

- Hypothesis class $\mathcal{H}$: hypothesis functions of the form $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$.
- Problem is to find $\mathbf{w}$ given data $\mathcal{D}$. $\mathcal{D} = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)\} \subseteq \mathcal{X}^d \times \mathcal{Y}$

Linear Regression as ML Estimation

Maximum Likelihood (ML) Hypothesis or Estimation:

- We can define likelihood estimate as

$$h_{\mathrm{ML}} = \underset{h \in \mathcal{H}}{\text{maximize}}\, P(\mathcal{D} \mid h) \qquad \Rightarrow \qquad \mathbf{w}_{\mathrm{ML}} = \underset{\mathbf{w}}{\text{maximize}}\, P(\mathcal{D} \mid f(\mathbf{x}))$$

- Noting $y_i \sim N(f(\mathbf{x_i}), \sigma^2)$.

$$\mathbf{w}_{\mathrm{ML}} = \underset{\mathbf{w}}{\text{maximize}} \quad \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \ \exp\,(-\frac{(y_i - f(\mathbf{x}_i))^2}{2\sigma^2})$$

- Maximizes the log (natural, ln) of the function instead.

$$\mathbf{w}_{\mathrm{ML}} = \underset{\mathbf{w}}{\text{maximize}} \log\left(\prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \ \exp\,(-\frac{(y_i - f(\mathbf{x}_i))^2}{2\sigma^2})\right) = \underset{\mathbf{w}}{\text{maximize}} \quad \sum_{i=1}^{n} \log\left(\frac{1}{\sigma\sqrt{2\pi}} \ \exp\,(-\frac{(y_i - f(\mathbf{x}_i))^2}{2\sigma^2})\right)$$

$$= \underset{\mathbf{w}}{\text{maximize}} \quad \sum_{i=1}^{n} -\log(\sigma\sqrt{2\pi}) + \log\left(\exp\,(-\frac{(y_i - f(\mathbf{x}_i))^2}{2\sigma^2})\right) \qquad = \underset{\mathbf{w}}{\text{maximize}} \quad \sum_{i=1}^{n} (-\frac{(y_i - f(\mathbf{x}_i))^2}{2\sigma^2})$$

Linear Regression as ML Estimation

Maximum Likelihood (ML) Hypothesis or Estimation:

$$\mathbf{w}_{\mathrm{ML}} = \underset{\mathbf{w}}{\text{maximize}} \quad \sum_{i=1}^{n} \left(-\frac{(y_i - f(\mathbf{x}_i))^2}{2\sigma^2}\right)$$

$$= \underset{\mathbf{w}}{\text{minimize}} \quad \sum_{i=1}^{n} \left(y_i - f(\mathbf{x}_i)\right)^2$$

We have seen this before! *Squared-error.*

- For linear regression case: $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$

$$\mathbf{w}_{\mathrm{ML}} = \underset{\mathbf{w}}{\text{minimize}} \quad \sum_{i=1}^{n} \left(y_i - \mathbf{w}^T\mathbf{x_i}\right)^2$$

We have an analytical solution.

- We can compute variance as:

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{w}_{\mathrm{ML}}^T\mathbf{x})^2$$

Notes:

- *Maximizing ML estimate is equivalent to minimizing least-squared error.*
- *ML Solution is same as least-squared error solution.*
- *This is a probabilistic interpretation or Bayesian explanation of the least-squared error solution and why did we choose squared error for defining a loss function.*

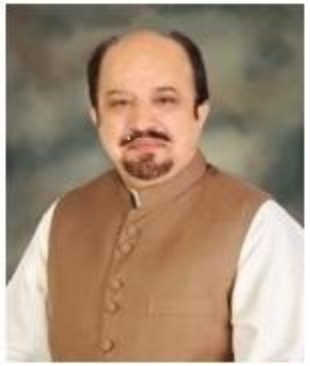
Outline

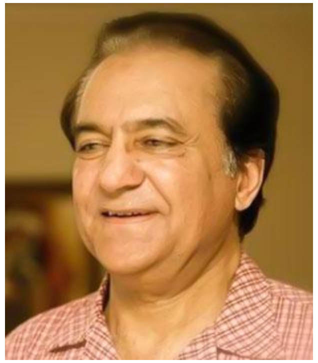
- Bayesian Learning Framework
    - MAP Estimation
    - ML Estimation
- Linear Regression as Maximum Likelihood Estimation
- Naïve Bayes Classifier

Naïve Bayes Classifier

Example:

- Given the name of a person, we want to predict the sex of a person.
- For example, we have a person say ‘Firdous’.
- Classifying ‘Firdous’ as female or male is equivalent to asking is it more probable that ‘Firdous’ is male or female.
- Mathematically, which one is greater $P(\text{male} \mid \text{Firdous})$ or $P(\text{female} \mid \text{Firdous})$
- Let’s apply Bayes theorem





Probability of being named Firdous given male

$$P(\text{male} \mid \text{Firdous}) = \frac{P(\text{Firdous} \mid \text{male})P(\text{male})}{P(\text{Firdous})}$$

Probability of being male

Probability of being named Firdous

Naïve Bayes Classifier

Example:

- We will look at the database of names vs gender.

$$P(\text{male} \mid \text{Firdous}) = \frac{P(\text{Firdous} \mid \text{male})P(\text{male})}{P(\text{Firdous})}$$

- $n = 48$, male count $= 28$, female count $= 20$

- Firdous: male count $=4$, female count $= 6$

$$P(\text{Firdous} \mid \text{male}) = \frac{4}{28} \qquad P(\text{male}) = \frac{28}{48}$$

$$P(\text{Firdous}) = \frac{10}{48}$$

$$P(\text{male} \mid \text{Firdous}) = 0.4$$

Name	Gender
Ahtesham	male
Iyad	male
Maleeha	female
Firdous	male
Shawal	male
Firdous	male
Ahmed	male
Zainab	female
Firdous	female
Ubaid	male
Badar	male
Firdous	female
Hassan	male
Kash	male
Hajira	female
Shehla	female
Firdous	female
Haram	female
Abdullah	male
Fahad	male
Bilal	male
Habeel	male
Farhan	male
Firdous	male
Anam	female
Firdous	female
Osama	male
Fatima	female
Mahnoor	female
Balaj	male
Razi	male
Zuhaib	male
Firdous	female
Shaharyar	male
Firdous	female
Ali	male
Mustansar	male
Sana	female
Anam	female
Marium	female
Khadija	female
Salaar	male
Faaiq	male
Hamza	male
Mahad	male
Ayesha	female
Firdous	male
Jawaria	female

Naïve Bayes Classifier

Example:

- Given Outlook, Temperature, Humidity and Wind Information, we want to carry out prediction for Play: Yes or No.

- Mathematically, which one is greater

$$P(\text{Play} = \text{Yes} \mid \text{Outlook},\ \text{Temp.},\ \text{Humidity},\ \text{Wind})$$

$$P(\text{Play} = \text{No} \mid \text{Outlook},\ \text{Temp.},\ \text{Humidity},\ \text{Wind})$$

- Predict for Sunny outlook, High humidity, Cool temperatue and Weak wind.

- Predict the most likely.

Day	Outlook	Temp.	Humidity	Wind	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Reference: Section 6.9.1 (Machine Learning by Tom Mitchell)

Naïve Bayes Classifier

Example:

$$P(\text{Play} = \text{Yes} \mid \text{Outlook} = \text{Sunny},\ \text{Temp} = \text{Cool},\ \text{Humidity} = \text{High},\ \text{Wind} = \text{Weak})$$

$$= \frac{P(\text{Outlook} = \text{Sunny},\ \text{Temp} = \text{Cool},\ \text{Humidity} = \text{High},\ \text{Wind} = \text{Strong} \mid \text{Play} = \text{Yes})\, P(\text{Play} = \text{Yes})}{P(\text{Outlook} = \text{Sunny},\ \text{Temp} = \text{Cool},\ \text{Humidity} = \text{High},\ \text{Wind} = \text{Strong})}$$

Naïve Assumption:

- Feature are mutually independent given the label!

$$P(\text{Outlook} = \text{Sunny},\ \text{Temp} = \text{Cool},\ \text{Humidity} = \text{High},\ \text{Wind} = \text{Strong} \mid \text{Play} = \text{Yes})$$

$$= P(\text{Outlook} = \text{Sunny} \mid \text{Play} = \text{Yes})\, P(\text{Temp} = \text{Cool} \mid \text{Play} = \text{Yes})\, P(\text{Humidity} = \text{High} \mid \text{Play} = \text{Yes})\, P(\text{Wind} = \text{Strong} \mid \text{Play} = \text{Yes})$$

Naïve Bayes Classifier

Example:

$$P(\text{Outlook} = \text{Sunny} \mid \text{Play} = \text{Yes}) = \frac{2}{9}$$

$$P(\text{Temp} = \text{Cool} \mid \text{Play} = \text{Yes}) = \frac{3}{9}$$

$$P(\text{Humidity} = \text{High} \mid \text{Play} = \text{Yes}) = \frac{3}{9}$$

$$P(\text{Wind} = \text{Strong} \mid \text{Play} = \text{Yes}) = \frac{3}{9}$$

$$P(\text{Play} = \text{Yes}) = \frac{9}{14}$$

$$P(\text{Outlook} = \text{Sunny} \mid \text{Play} = \text{No}) = \frac{3}{5}$$

$$P(\text{Temp} = \text{Cool} \mid \text{Play} = \text{No}) = \frac{1}{5}$$

$$P(\text{Humidity} = \text{High} \mid \text{Play} = \text{No}) = \frac{4}{5}$$

$$P(\text{Wind} = \text{Strong} \mid \text{Play} = \text{No}) = \frac{3}{5}$$

$$P(\text{Play} = \text{No}) = \frac{5}{14}$$

Day	Outlook	Temp.	Humidity	Wind	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Naïve Bayes Classifier

Example:

$$P(\text{Outlook} = \text{Sunny} \mid \text{Play} = \text{Yes})\, P(\text{Temp} = \text{Cool} \mid \text{Play} = \text{Yes})\, P(\text{Humidity} = \text{High} \mid \text{Play} = \text{Yes})\, P(\text{Wind} = \text{Strong} \mid \text{Play} = \text{Yes})$$

$$\times P(\text{Play} = \text{Yes}) \qquad = \frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14} \qquad = 0.0053$$

$$P(\text{Outlook} = \text{Sunny} \mid \text{Play} = \text{No})\, P(\text{Temp} = \text{Cool} \mid \text{Play} = \text{No})\, P(\text{Humidity} = \text{High} \mid \text{Play} = \text{No})\, P(\text{Wind} = \text{Strong} \mid \text{Play} = \text{No})$$

$$\times P(\text{Play} = \text{No}) \qquad = \frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} \times \frac{5}{14} \qquad = 0.0206$$

$$P(\text{Play} = \text{Yes} \mid \text{Outlook} = \text{Sunny},\ \text{Temp} = \text{Cool},\ \text{Humidity} = \text{High},\ \text{Wind} = \text{Strong}) \ = \frac{0.0053}{0.0053 + 0.0206} = 0.2046$$

$$P(\text{Play} = \text{No} \mid \text{Outlook} = \text{Sunny},\ \text{Temp} = \text{Cool},\ \text{Humidity} = \text{High},\ \text{Wind} = \text{Strong}) \ = \frac{0.0206}{53 + 0.0206} = 0.7954$$

Play = No *is more likely!*

Naïve Bayes Classifier

Generative Classifier:

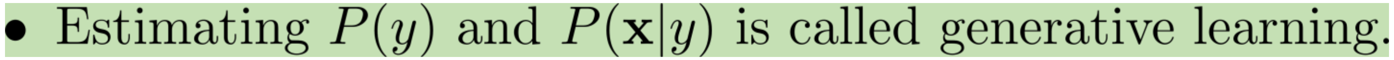
- Attempts to model class, that is, build a generative statistical model that informs us how a given class would generate input data.

- Ideally, we want to learn the joint distribution of the input **x** and output label y, that is, P(**x**,y).

- For a test-point, generative classifiers predict which class would have **most-likely** generated the given observation.

- Mathematically, prediction for input **x** is carried out by computing the conditional probability P(y|**x**) and selecting the most-likely label y.

- Using the Bayes rule, we can compute P(y|**x**) by computing P(y) and P(**x**|y).



- Estimating $P(y)$ and $P(\mathbf{x}|y)$ is called generative learning.

Naïve Bayes Classifier

Overview of Naïve Bayes Classifier:

- We have $D = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)\} \subseteq \mathcal{X}^d \times \mathcal{Y}$

  $\mathcal{Y} = \{1, 2, \ldots, M\}$ (M-class classification)

Key Idea:

- Estimate $P(y|\mathbf{x})$ from the data using the Bayes Theorem.

- Using Bayes theorem and MAP learning framework, we can write this as

$$h_{\text{MAP}}(\mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad P(y \mid \mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad \frac{P(\mathbf{x} \mid y)\, P(y)}{P(\mathbf{x})} = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad P(\mathbf{x} \mid y)\, P(y)$$

- Estimating $P(y)$ is easy. If y takes on discrete binary values, coin tossing or spam vs non-spam for example, we simply need to count how many times we observe each class outcome.

- Estimating $P(\mathbf{x}|y)$, however, is not easy, Why?

Naïve Bayes Classifier

Overview of Naïve Bayes Classifier:

Example:

- $M = 2$ and features $d = 6$. Assuming binary features/classification.

- We want to estimate

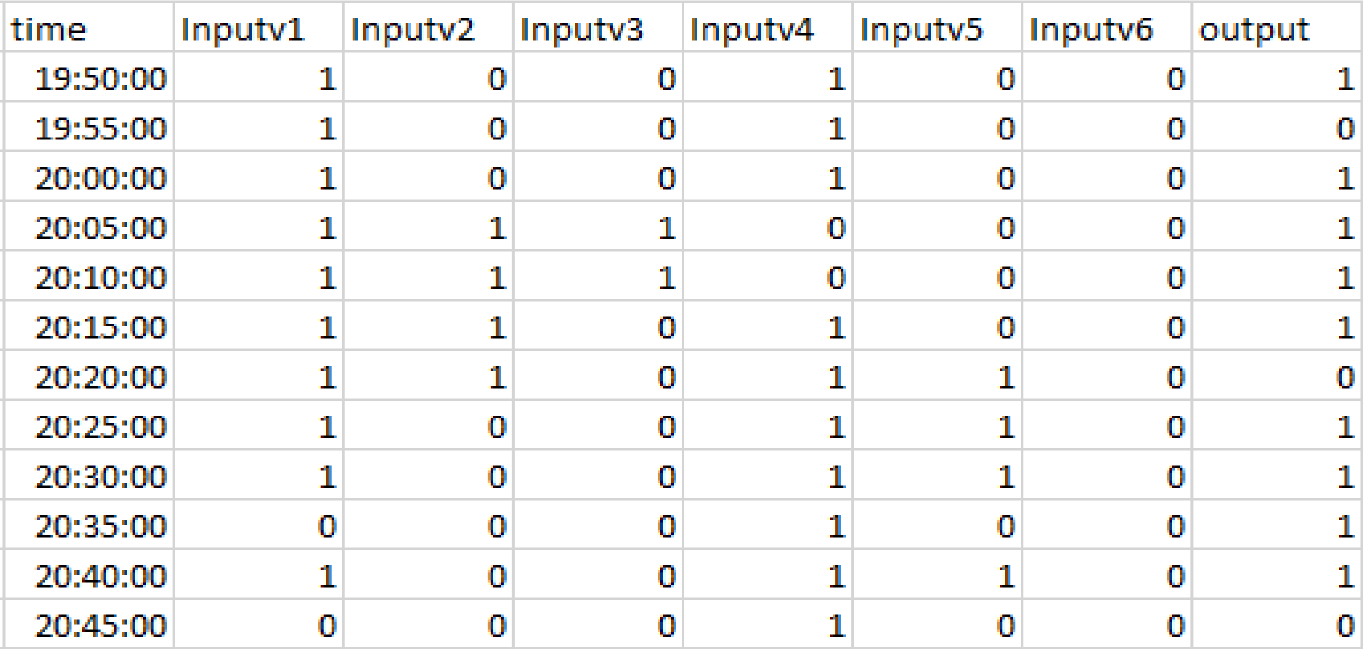
$$P(\mathbf{x} \mid y) = P(x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}, x^{(5)}, x^{(6)}|y)$$

- How many parameters do we need to fully estimate $P(\mathbf{x}|y)$?

- We need to represent all 2^6 outcomes or probabilities for each $y = 0, 1$.

- For d binary features, we need to represent all 2^d outcomes.

- Learning the values for the full conditional probability would require enormous amounts of data.



time	Inputv1	Inputv2	Inputv3	Inputv4	Inputv5	Inputv6	output
19:50:00	1	0	0	1	0	0	1
19:55:00	1	0	0	1	0	0	0
20:00:00	1	0	0	1	0	0	1
20:05:00	1	1	1	0	0	0	1
20:10:00	1	1	1	0	0	0	1
20:15:00	1	1	0	1	0	0	1
20:20:00	1	1	0	1	1	0	0
20:25:00	1	0	0	1	1	0	1
20:30:00	1	0	0	1	1	0	1
20:35:00	0	0	0	1	0	0	1
20:40:00	1	0	0	1	1	0	1
20:45:00	0	0	0	1	0	0	0

Naïve Bayes Classifier

Naïve Bayes Classifier:

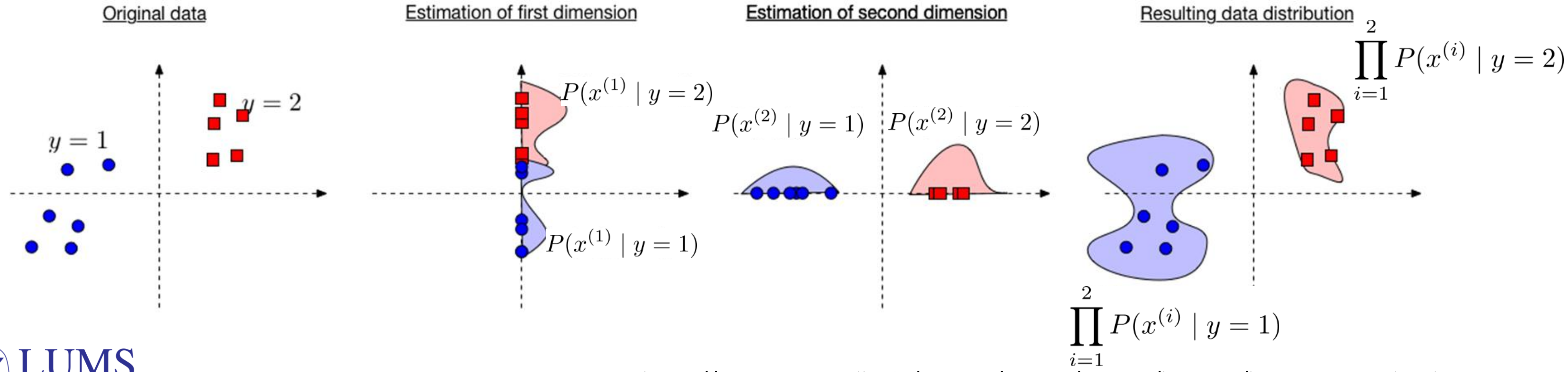
- To overcome this requirement of enormous data for the computation of conditional probability, we can make a 'naive Bayes' assumption.

Naïve Assumption:

- Features are mutually independent given the label!
- Consequence: $P(\mathbf{x} \mid y) = P(x^{(1)}, x^{(2)}, \ldots, x^{(d)}|y) = \prod_{i=1}^{d} P(x^{(i)} \mid y)$

- How many probabilities now?
  one for each feature/label.
  $2d$

Interpretation[1]:



Original data
$y = 1$
$y = 2$
Estimation of first dimension
$P(x^{(1)} \mid y = 2)$
$P(x^{(1)} \mid y = 1)$
Estimation of second dimension
$P(x^{(2)} \mid y = 1)$
$P(x^{(2)} \mid y = 2)$
Resulting data distribution
$\prod_{i=1}^{2} P(x^{(i)} \mid y = 2)$
$\prod_{i=1}^{2} P(x^{(i)} \mid y = 1)$

[1] Source: https://www.cs.cornell.edu/courses/cs4780/2018sp/lectures/lecturenote05.html

Naïve Bayes Classifier

Naïve Bayes Classifier:

- We can reformulate our hypothesis function, referred to as Naive Bayes (NB) Classifier, as

$$h_{\text{NB}}(\mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad P(y \mid \mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad \prod_{i=1}^{d} P(x^{(i)} \mid y)\, P(y)$$

- Maximizes the log (natural, ln) of the function instead.

$$\begin{aligned} h_{\text{NB}}(\mathbf{x}) &= \underset{y \in \mathcal{Y}}{\text{maximize}} \quad \sum_{i=1}^{d} \log\left(P(x^{(i)} \mid y)\, P(y)\right) \\ &= \underset{y \in \mathcal{Y}}{\text{maximize}} \quad \sum_{i=1}^{d} \log P(x^{(i)} \mid y) \,+\, \log P(y) \end{aligned}$$

- How many probabilities?

$2d + 1$

Naïve Bayes Classifier

Naïve Bayes Classifier - Training:

Assume each feature and label as a binary variable

- Hypothesis space: $2d+1$ different binomial distributions.
  - $P(x^{(i)} \mid y)$ and $P(y)$ for each $x^{(i)}$ and each $y = \{0,1\}$, $i = 1, 2, \ldots, d$.
  - Each probability can be parameteterized by a single variable θ.
- We treat learning of each of these as a separate MLE problem.

$$P(x^{(i)} = j \mid y = k) = \frac{\text{count}(x^{(i)} = j \text{ and } y = k)}{\text{count}(y = k)}, \quad j, k \in \{0, 1\}$$

$$P(y = k) = \frac{\text{count}(y = k)}{\text{count}(y = 0) + \text{count}(y = 1)} = \frac{\text{count}(y = k)}{n}, \quad k \in \{0, 1\}$$

- We compute these probabilities during training stage.
- As we saw earlier, these probability estimates maximizes the likelihood.

Naïve Bayes Classifier

Naïve Bayes Classifier - Prediction:

Assume each feature and label as a binary variable

- For a new test-point $\mathbf{x}_{\text{new}}$, we assign the label as

$$h_{\text{NB}}(\mathbf{x}_{\text{new}}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad P(y \mid \mathbf{x}_{\text{new}}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad \prod_{i=1}^{d} P(x_{\text{new}}^{(i)} \mid y)\, P(y)$$

We have a problem here!

- We have a product of probabilities. If any of the estimated probability is zero, the product would be zero.

Solution: Additive Smoothing or Laplace Smoothing

$$P(x^{(i)} = j \mid y = k) = \frac{\text{count}(x^{(i)} = j \text{ and } y = k) + \ell}{\text{count}(y = k) + \ell R}, \quad j, k \in \{0, 1\}$$

$$P(y = k) = \frac{\text{count}(y = k) + \ell}{n + \ell M}, \quad k \in \{0, 1\}$$

- Here $\ell > 0$. If $\ell = 1$, we refer to it as add-1 smoothing.
- R is the number of values $x^{(i)}$ can take. For binary case, $R = 2$.
- M is the number of classes. For binary case $M = 2$.

Naïve Bayes Classifier

Naïve Bayes Classifier - Extensions:

- We have $D = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)\} \subseteq \mathcal{X}^d \times \mathcal{Y}$

  $\mathcal{Y} = \{1, 2, \ldots, M\}$ (M-class classification)

- We assume that each feature $x^{(i)}$ takes L_i values, that is, $x^{(i)} \in \{1, 2, \ldots, L_i\}$.

How many probability tables do we have if we have d features and M labels?

- $dM + 1$: we have one probability table for each feature and each value of the label and one more table for the prior $P(y)$.
- The set of tables for a single feature (for all labels y) is referred to as a conditional probability table (CPT), and here we have d of those.

Incorporating model parameters in the formulation

- We considered a binary case and assumed that a single parameter characterizes probability model associated with each feature.

- In general, we can have parameters defining the probability model and we learn parameters of the probability model during the learning stage.

Naïve Bayes Classifier

Naïve Bayes Classifier – Extensions:

Gaussian Naïve Bayes – Continuous Features:

- In practice, some features are discrete (e.g., gender, marital status) and some are continuous (weight).
- The probability model or distribution for each $x^{(i)}$ can be parameterized differently.
- If $x^{(i)} \in \mathbf{R}$, what kind of distribution can we use for $P(x^{(i)}|y)$?
- For real-valued features, we often use a Gaussian distribution to **model probability density function**, that is,

$$p(x^{(i)} \mid y = k) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x^{(i)} - \mu)^2}{2\sigma^2}\right) \qquad p(x^{(i)} \mid y = k) \sim N(\mu, \sigma^2).$$

- For succinct representation, the depenence of μ and σ on feature index i and label index k is dropped. We can have different distributions or parameters for each i and each k. just like we have different probabilities for discrete features.

Naïve Bayes Classifier

Naïve Bayes Classifier – Extensions:

Gaussian Naïve Bayes – Training:

- We have $p(x^{(i)} \mid y = k) \sim N(\mu, \sigma^2)$, given data we want to learn μ and σ for each i and each k.

- Given i and k, we compute the μ and σ as sample mean and sample variance, where the sample corresponds to $x^{(i)}$ for which associated label $y = k$.

$$\mu = \frac{1}{\text{count}(y = k)} \sum_{j=1}^{n} \delta(y_j - k)\ x_j^{(i)}$$

$$\sigma^2 = \frac{1}{\text{count}(y = k)} \sum_{j=1}^{n} \delta(y_j - k)\ \left(x_j^{(i)} - \mu\right)^2$$

- For each label y, we need to estimate d means and d variances during training.

Naïve Bayes Classifier

Naïve Bayes Classifier - Summary:

- In Naïve Bayes, we compute the probabilities or parameters of the distribution defining probabilities and use these to carry out predictions.
- Naïve Bayes can handle missing values by ignoring the sample during probability computation, is robust to outliers and irrelevant features.
- Naïve Bayes algorithm is very easy to implement for applications involving textual information data (e.g., sentiment analysis, news article classification, spam filtering).
- Convergence is quicker relative to logistic regression that is discriminative in nature.
- It performs well even when the independence between features assumption does not hold.
- The resulting decision boundaries can be non-linear and/or piecewise.
- Disadvantage: It is not robust to redundant features. If the features have a strong relationship or correlation with each other, Naïve Bayes is not a good choice. Naïve Bayes has high bais and low variance and there are no regularization here to adjust the bias thing

NB Classifier – Text Classification

Text Classification Overview:

- Applications of text classification include
  - Sentiment analysis
  - Spam detection
  - Language Identification; to name a few.

Classification Problem:

Input: a document and a fixed set of classes (e.g., spam, non-spam)

Output: a predicted class for the document

Classification Methods:

- ***Hand-coded rules:*** Rules based on combinations of words or other features
  - e.g., spam: black-list-address OR ("dollars" AND "you have been selected")
  - Accuracy can be high if rules carefully refined by **expert**
  - But building and *maintaining* these rules is *expensive*

Ref: "Speech and Language Processing" by *Daniel Jurafsky & James H. Martin*
Source: https://web.stanford.edu/~jurafsky/slp3/

NB Classifier – Text Classification

Text Classification – Supervised Learning:

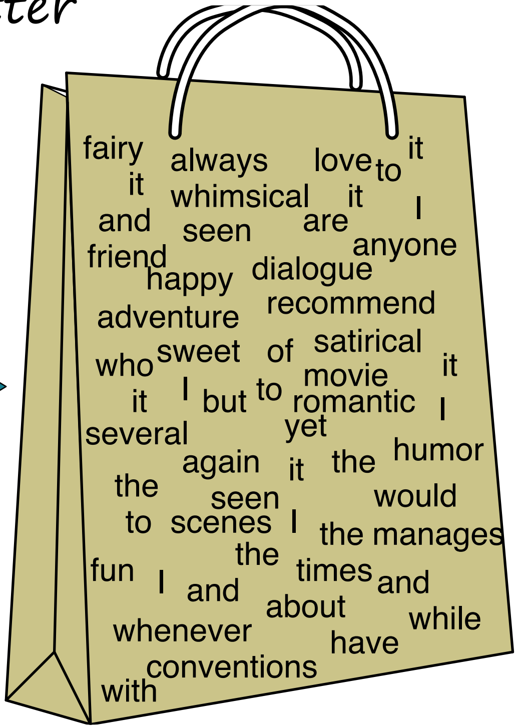
Input: a document and a fixed set of classes (e.g., spam, non-spam)
+ training data (n labeled documents)

Output: a predicted class for the document

Bag of Words – Representation of a document for classification:

Assumption: Position doesn't matter

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1

Ref: **"Speech and Language Processing"** by *Daniel Jurafsky & James H. Martin*
Source: https://web.stanford.edu/~jurafsky/slp3/

NB Classifier – Text Classification

Text Classification – Terminology and Preprocessing :

- Corpus: A collection of documents; data.

- Vocabulary, denoted by V, is the union of all the word types in all classes (not just one class).

Preprocessing documents:

- Clean the corpus: (e.g., Hello, hello or hello! should be considered the same)
  - Remove numbers, punctuation and excessive white spaces
  - Use lowercase representation

- Stop words concept: very frequent words (**a** or **the**)
  - Sort vocabulary with respect to frequency, call the top 5 or 20 words the stopword list and remove from all of the documents or from the vocabulary.

- In naïve Bayes, it's more common to **not** remove stop words and use all the words.

- After pre-processing, create a **mega document** for each class by concatenating all the documents of the class.

- Use bag of words on mega document to obtain a frequency table for each class.

Ref: **"Speech and Language Processing"** by *Daniel Jurafsky & James H. Martin*
Source: https://web.stanford.edu/~jurafsky/slp3/

NB Classifier – Spam Filtering

Example: Spam vs Non-Spam:

Category	Document
Spam	send us your password
Spam	review us
Spam	send us your account
Spam	send your password
Non-spam	password review
Non-spam	send us your review
?	review us now
?	review account

Issue 1:

‘**now**’ is not in the training data.

- unknown word or out of vocabulary word.

Solution:

remove out of vocabulary word from the test document.

Issue 2:

‘**account**’ is only available in one class

Solution:

Use add-1 smoothing. We will see this shortly.

- Vocabulary, V = {send, us, your, password, review, account}

NB Classifier – Spam Filtering

Naïve Bayes (NB) Classification:

- NB Classifier:

$$h_{\mathrm{NB}}(\mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad P(y \mid \mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad \prod_{i=1}^{d} P(x^{(i)} \mid y)\, P(y)$$

- $\mathbf{x}$ represents the test document for which we want to carry out prediction. Each feature represents a word in the document.
- d here represents the number of words in the test document.
- For $\mathbf{x} =$ "review us now", $d = 3$.
- For $\mathbf{x} =$ "review account", $d = 2$.

Ref: **"Speech and Language Processing"** by *Daniel Jurafsky & James H. Martin*
Source: https://web.stanford.edu/~jurafsky/slp3/

NB Classifier – Spam Filtering

Naïve Bayes (NB) Classification – Example:

Category	Document
Spam	send us your password
Spam	review us
Spam	send us your account
Spam	send your password
Non-spam	password review
Non-spam	send us your review
?	review us now
?	review account

Bag of Words

Vocabulary	Spam Count	Non-spam Count
send	3	1
us	3	1
your	3	1
password	2	1
review	1	2
account	1	0
	13	6

- For $\mathbf{x} =$ “review us now”, $d = 3$.

We compute $P(\text{Spam} \mid \mathbf{x})$ and $P(\text{Non} - \text{spam} \mid \mathbf{x})$

NB Classifier – Spam Filtering

Naïve Bayes (NB) Classification – Example:

- For $\mathbf{x} =$ "review us now".
- Ignore 'now': unknown word, out of vocabulary
- We compute $P(\mathbf{x} \mid \text{Spam})\, P(\text{Spam})$ and $P(\mathbf{x} \mid \text{Non} - \text{spam})\, P(\text{Non} - \text{spam})$

$$P(\mathbf{x} \mid \text{Spam})\, P(\text{Spam}) = P(\text{review} \mid \text{Spam}) P(\text{us} \mid \text{Spam})\, P(\text{Spam})$$

$$P(\text{review} \mid \text{Spam}) = \frac{1}{13} \qquad P(\text{us} \mid \text{Spam}) = \frac{3}{13} \qquad P(\text{Spam}) = \frac{4}{6}$$

$$P(\mathbf{x} \mid \text{Spam})\, P(\text{Spam}) = 0.012$$

$$P(\text{review} \mid \text{Non} - \text{spam}) = \frac{2}{6} \qquad P(\text{us} \mid \text{Non} - \text{spam}) = \frac{1}{6} \qquad P(\text{Non} - \text{spam}) = \frac{2}{6}$$

$$P(\mathbf{x} \mid \text{Non} - \text{spam})\, P(\text{Non} - \text{spam}) = 0.0185$$

Vocabulary	Spam Count	Non-spam Count
send	3	1
us	3	1
your	3	1
password	2	1
review	1	2
account	1	0
	13	6

Document is likely a non-spam.

NB Classifier – Spam Filtering

Naïve Bayes (NB) Classification – Example:

- For $\mathbf{x} =$“review account”.
- For ‘account’: non-spam count is zero. Consequently, $P(\text{account} \mid \text{Non} - \text{spam}) = 0$.

Solution: Add 1 smoothing

$$P(\text{Spam}) = \frac{4}{6} \qquad P(\text{Non} - \text{spam}) = \frac{2}{6}$$

$$P(\text{review} \mid \text{Spam}) = \frac{1+1}{13+6} = \frac{2}{19} \qquad P(\text{account} \mid \text{Spam}) = \frac{1+1}{13+6} = \frac{2}{19}$$

We have added numerator factor times the size of the vocabulary in the denominator.

$$P(\text{review} \mid \text{Non} - \text{spam}) = \frac{2+1}{6+6} = \frac{3}{12} \qquad P(\text{account} \mid \text{Non} - \text{spam}) = \frac{0+1}{6+6} = \frac{1}{12}$$

$$P(\mathbf{x} \mid \text{Spam})\, P(\text{Spam}) = 0.00738$$

$$P(\mathbf{x} \mid \text{Non} - \text{spam})\, P(\text{Non} - \text{spam}) = 0.00694$$

Document is likely a spam.

Vocabulary	Spam Count	Non-spam Count
send	3	1
us	3	1
your	3	1
password	2	1
review	1	2
account	1	0
	13	6