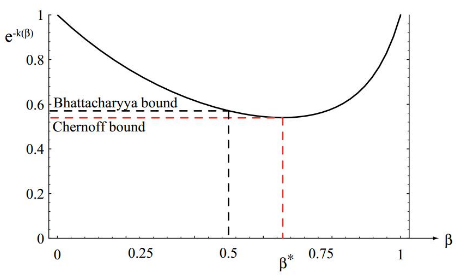
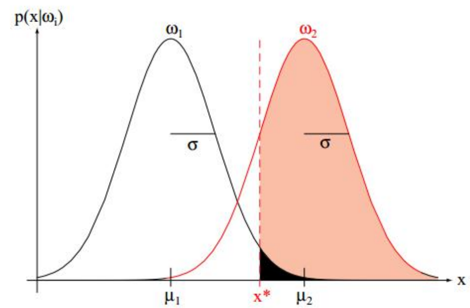
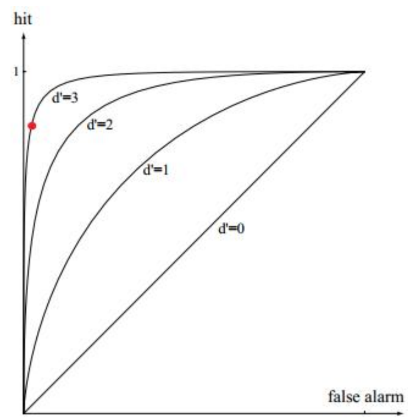
Lecture 5: Bayesian Estimation and Naive Bayes Classification

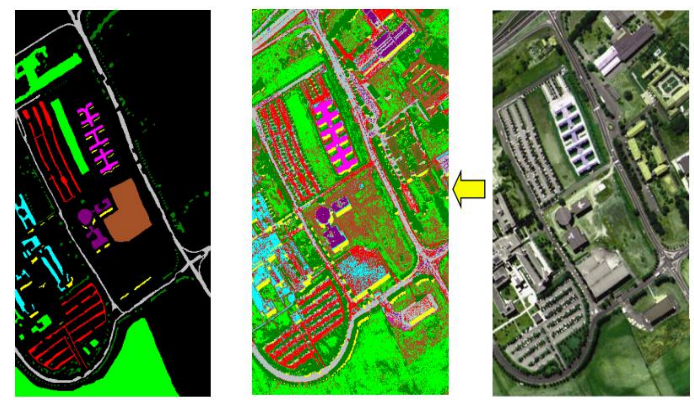
Dr. Chengjiang Long
Computer Vision Researcher at Kitware Inc.
Adjunct Professor at RPI.
Email: **longc3@rpi.edu**

Recap Previous Lecture









Outline

- Bayesian Estimation
- Naive Bayes Classifier
- Data Split and Cross-Validation
- Overfitting

Outline

- **Bayesian Estimation**
- Naive Bayes Classifier
- Data Split and Cross-Validation
- Overfitting

Bayesian Estimation

- In MLE θ is assumed fixed
- In BE θ is a random variable
- Suppose we have some idea of the range where the parameters θ should be
  - Shouldn't we utilize this prior knowledge in hope that it will lead to better parameter estimation？

Bayesian Estimation

- Let θ be a random variable with prior distribution $P(\theta)$.
- This is the key difference between ML and Bayesian parameter estimation.
- This allows us to use a prior to express the uncertainty present before seeing the data.
- Frequentist approach does not account for uncertainty in θ (see bootstrap for more on this, however)

Motivation

- As in MLE, suppose p(x|θ) is completely specified if θ is given.
- But now θ is a random variable with prior p(θ).
  - Unlike MLE case, p(x|θ) is a conditional density
- After we observe the data D, using Bayes rule we can compute the posterior p(θ|D)

Motivation

- Recall that for the MAP classifier we find the class ω_i that maximizes the posterior $p(\omega|D)$
- By analogy, a reasonable estimate of θ is the one that maximizes the posterior $p(\theta|D)$
- But θ is not our final goal, our final goal is the unknown $p(x)$
- **Therefore a better thing to do is to maximize $p(x|D)$, this is as close as we can come to the unknown $p(x)$!**

Parameter Distribution

- Assumptions:
  - $p(x)$ is unknown, but has known parametric form
  - Parameter vector θ is unknown
  - **$p(x|\theta)$ is completely known**
  - **Prior density $p(\theta)$ is known**
- Observation of samples provides posterior density $p(\theta|D)$
  - Hopefully peaked around true value of θ
- Treat each class separately and drop subscripts

Parameter Distribution

- Converted problem of learning probability density function to learning parameter vector
- Goal: compute p(x|D) as best possible estimate of p(x)

$$p(\mathrm{x} \mid \mathrm{D}) = \int \mathrm{p}(\mathrm{x}, \theta \mid D) d\theta$$

$$= \int \mathrm{p}(\mathrm{x} \mid \theta, D) p(\theta \mid D) d\theta = \int \mathrm{p}(\mathrm{x} \mid \theta) p(\theta \mid D) d\theta$$

p(x) is completely known given θ,
independent of samples in D

Links class-conditional density p(x|D) to posterior density p(θ|D)

The Univariate Case: $p(\mu|\mathcal{D})$

- Goal: Estimate θ using the a posteriori density $P(\theta \mid D)$
- The univariate case: $p(\mu \mid D)$
  - μ is the only unknown parameter

$$p(x \mid \mu) \sim N(\mu, \sigma^2)$$

$$p(\mu) \sim N(\mu_0, \sigma_0^2)$$

  - μ_0 and σ_0 are known.
  - μ_0 is best guess for μ, σ_0 is uncertainty of guess.

The Univariate Case: $p(\mu|\mathcal{D})$

$$p(\mu \mid \mathrm{D}) = \frac{p(\mathrm{D} \mid \mu)p(\mu)}{\int p(\mathrm{D} \mid \mu)p(\mu)d\mu}$$

$$= \alpha \prod_{k=1}^{k=n} p(x_k \mid \mu)p(\mu)$$

- α depends on D, not μ
- The above equation shows how training samples affect our idea about the true value of μ

The Univariate Case: $p(\mu|\mathcal{D})$

$$p(\mu \mid \mathrm{D}) = \frac{p(\mathrm{D} \mid \mu)\, p(\mu)}{\int p(\mathrm{D} \mid \mu)\, p(\mu)\, d\mu}$$

$$= \alpha \prod_{k=1}^{k=n} p(x_k \mid \mu)\, p(\mu)$$

$$p(\mu|\mathcal{D}) = \alpha \prod_{k=1}^{n} \overbrace{\frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x_k-\mu}{\sigma}\right)^2\right]}^{p(x_k|\mu)} \overbrace{\frac{1}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_0}{\sigma_0}\right)^2\right]}^{p(\mu)}$$

$$= \alpha' \exp\left[-\frac{1}{2}\left(\sum_{k=1}^{n}\left(\frac{\mu-x_k}{\sigma}\right)^2 + \left(\frac{\mu-\mu_0}{\sigma_0}\right)^2\right)\right]$$

$$= \alpha'' \exp\left[-\frac{1}{2}\left[\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}\right)\mu^2 - 2\left(\frac{1}{\sigma^2}\boxed{\sum_{k=1}^{n} x_k} + \frac{\mu_0}{\sigma_0^2}\right)\mu\right]\right]$$

= n x empirical mean

The Univariate Case: $p(\mu|\mathcal{D})$

$$p(\mu \mid \mathrm{D}) = \frac{p(\mathrm{D} \mid \mu)\, p(\mu)}{\int p(\mathrm{D} \mid \mu)\, p(\mu)\, d\mu} \qquad (1)$$

$$= \alpha \prod_{k=1}^{k=n} p(x_k \mid \mu)\, p(\mu)$$

- Reproducing density (remains Gaussian)

$$p(\mu|\mathrm{D}) \sim N(\mu_n, \sigma_n^2) \qquad (2)$$

- (1) and (2) yield:

$$\mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2} \right) \hat{\mu}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} \mu_0$$

$$and \;\; \sigma_n^2 = \frac{\sigma_0^2 \sigma^2}{n\sigma_0^2 + \sigma^2}$$

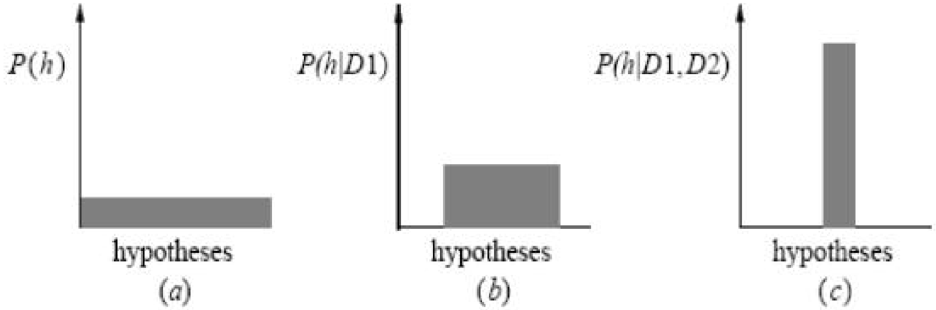
Empirical (sample) mean

The Univariate Case: $p(\mu|\mathcal{D})$

$$\mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2} \right) \hat{\mu}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} \mu_0$$

$$and \;\; \sigma_n^2 = \frac{\sigma_0^2 \sigma^2}{n\sigma_0^2 + \sigma^2}$$

- μ is linear combination of empirical and prior information
- Each additional observation decreases uncertainty about μ



The Univariate Case: $p(x|\mathcal{D})$

- The univariate case
  - p(μ | D) computed
  - p(x | D) remains to be computed*
  - $p(x \mid \mathrm{D}) = \int p(x \mid \mu) p(\mu \mid \mathrm{D}) d\mu$ is Gaussian
  - It provides: $p(x \mid \mathrm{D}) \sim N(\mu_n, \sigma^2 + \sigma_n^2)$

$$
\begin{aligned}
p(x|\mathcal{D}) &= \int p(x|\mu)p(\mu|\mathcal{D})\, d\mu \\
&= \int \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_n}{\sigma_n}\right)^2\right] d\mu \\
&= \frac{1}{2\pi\sigma\sigma_n} \exp\left[-\frac{1}{2}\frac{(x-\mu_n)^2}{\boxed{\sigma^2+\sigma_n^2}}\right] f(\sigma, \sigma_n)
\end{aligned}
$$

where

$$
f(\sigma, \sigma_n) = \int \exp\left[-\frac{1}{2}\frac{\sigma^2+\sigma_n^2}{\sigma^2\sigma_n^2}\left(\mu - \frac{\sigma_n^2 x + \sigma^2\mu_n}{\sigma^2+\sigma_n^2}\right)^2\right] d\mu
$$

The Univariate Case: $p(x|\mathcal{D})$

- Using Bayes formula, we obtain the Bayesian classification rule:

$$\underset{\omega_j}{Max}\left[p(\omega_j \mid x, \mathrm{D})\right] \equiv \underset{\omega_j}{Max}\left[p(x \mid \omega_j, \mathrm{D}_j)\, p(\omega_j)\right]$$

$$p(x \mid \mathrm{D}) \sim N(\mu_n, \sigma^2 + \sigma_n^2)$$

- We have:

✓ Replaced mean with conditional mean

✓ Increased variance to account for additional uncertainty in x due to inexact knowledge of mean

The Multivariate Case

$$p(\mathbf{x}|\boldsymbol{\mu}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \qquad\qquad p(\boldsymbol{\mu}) \sim N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

$$\left\{\begin{aligned} p(\boldsymbol{\mu}|\mathcal{D}) &= \alpha \prod_{k=1}^{n} p(\mathbf{x}_k|\boldsymbol{\mu}) p(\boldsymbol{\mu}) \\ &= \alpha' \exp \left[-\frac{1}{2} \left(\boldsymbol{\mu}^t (n\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}_0^{-1}) \boldsymbol{\mu} - 2\boldsymbol{\mu}^t \left(\boldsymbol{\Sigma}^{-1} \sum_{k=1}^{n} \mathbf{x}_k + \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0 \right) \right) \right] \\ p(\boldsymbol{\mu}|\mathcal{D}) &= \alpha'' \exp \left[-\frac{1}{2} (\boldsymbol{\mu} - \boldsymbol{\mu}_n)^t \boldsymbol{\Sigma}_n^{-1} (\boldsymbol{\mu} - \boldsymbol{\mu}_n) \right] \end{aligned}\right.$$

$$\Downarrow$$

$$\left\{\begin{aligned} \boldsymbol{\mu}_n &= \boldsymbol{\Sigma}_0 \left(\boldsymbol{\Sigma}_0 + \frac{1}{n} \boldsymbol{\Sigma} \right)^{-1} \hat{\boldsymbol{\mu}}_n + \frac{1}{n} \boldsymbol{\Sigma} \left(\boldsymbol{\Sigma}_0 + \frac{1}{n} \boldsymbol{\Sigma} \right)^{-1} \boldsymbol{\mu}_0 \\ \boldsymbol{\Sigma}_n &= \boldsymbol{\Sigma}_0 \left(\boldsymbol{\Sigma}_0 + \frac{1}{n} \boldsymbol{\Sigma} \right)^{-1} \frac{1}{n} \boldsymbol{\Sigma} \end{aligned}\right.$$

Bayesian Parameter Estimation: General Theory

- $p(x \mid D)$ computation can be applied to any situation in which the unknown density can be parameterized.
- The basic assumptions are:
  - ✓ The form of $p(x \mid \theta)$ assumed known, but the value of θ is not known exactly
  - ✓ Our knowledge about θ is assumed to be contained in a known prior density
  - ✓ The rest of our knowledge θ is contained in a set D of n random variables $x_1, x_2, \ldots, x_n$ that follows $p(x)$

Bayesian Parameter Estimation: General Theory

- The basic problem is: **"Compute the posterior density** $p(\theta \mid D)$**" then "Derive** $p(x \mid D)$ **"**

$$p(x \mid D) = \int \underbrace{p(x \mid \theta)}_{\text{known}} \underbrace{p(\theta \mid D)}_{\text{unknown}} d\theta$$

- Using Bayes formula, we have:

$$p(\theta \mid \mathrm{D}) = \frac{p(\mathrm{D} \mid \theta) p(\theta)}{\int p(\mathrm{D} \mid \theta) p(\theta) d\theta}$$

- And by the independence assumption:

$$p(\mathrm{D} \mid \theta) = \prod_{k=1}^{k=n} p(x_k \mid \theta)$$

Recursive Bayes Learning

- Assume that training samples become available one by one

$$p(\mathrm{D}^{n} \mid \theta) = p(x_n \mid \theta)\, p(\mathrm{D}^{n-1} \mid \theta)$$

- Due to independence, result is independent of order:

$$p(\mathrm{D} \mid \theta) = \prod_{k=1}^{k=n} p(x_k \mid \theta)$$

Bayesian Estimation vs. MLE

- BE: p(x|D) can be thought of as the weighted average of the proposed model for all possible values of θ

$$p(x \mid D) = \int \underbrace{p(x \mid \theta)}_{\substack{\text{proposed model} \\ \text{with certain } \theta}} \overbrace{p(\theta \mid D)}^{\substack{\text{support } \theta \text{ receives} \\ \text{from the data}}} d\theta$$

- Contrast this with the MLE solution which always gives us a single model:

$$p(x|\hat{\theta})$$

- When we have many possible solutions, taking their sum averaged by their probabilities seems better than pick just one solution.

Bayesian Estimation vs. MLE

- In practice, it may be hard to do integration analytically and we may have to resort to numerical methods
- The MLE solution requires differentiation, instead of integration, to get

$$p(x|\hat{\theta})$$

- Differentiation is easy and can always be done analytically

When do Maximum-Likelihood and Bayes Methods Differ?

- Equivalent asymptotically (for infinite training data)
  - For reasonable prior distributions
  - When prior p(θ) is uninformative and p(θ |D) is peaked

- MLE computationally cheaper, simpler solutions
- BE uses more information (more general model)

Outline

- Bayesian Estimation
- **Naive Bayes Classifier**
- Data Split and Cross-Validation
- Overfitting

Bayes Classifier

- Traing data

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

$$\overbrace{\qquad\qquad\qquad\qquad}^{X} \quad \overbrace{\qquad}^{Y}$$

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

- **Learning** = estimating P(X|Y), P(Y)
- **Classification** = using Bayes rule to Classification

using Bayes rule to calculate $P(Y \mid X^{new})$

Bayes Classifier

- Traing data

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

$$\overbrace{\text{Sky Temp Humid Wind Water Forecst}}^{X}\ \overbrace{\text{EnjoySpt}}^{Y}$$

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

- Learning = estimating P(X|Y), P(Y)
- Classification = using Bayes rule to Classification

using Bayes rule to calculate $P(Y \mid X^{new})$

How shall we represent P(X|Y), P(Y)?
How many parameters must we estimate?

Bayes Classifier

- Traing data

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

$$\overbrace{\hphantom{Sky\ Temp\ Humid\ Wind\ Water\ Forecst}}^{X}\ \overbrace{\hphantom{EnjoySpt}}^{Y}$$

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

- **Learning** = estimating P(X|Y), P(Y)
- **Classification** = using Bayes rule to Classification

using Bayes rule to calculate $P(Y \mid X^{new})$

Full joint P(X1,X2,...,Xn|Y) usually impractical.

Conditional Independence

Definition: X is conditionally independent of Y given Z, if the probability distribution governing X is independent of the value of Y, given the value of Z

$$(\forall i, j, k) P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

Which we often write

$$P(X|Y, Z) = P(X|Z)$$

E.g.,

$$P(Thunder|Rain, Lightning) = P(Thunder|Lightning)$$

Conditional Indepence

- Naïve Bayes uses assumption that the Xi are conditionally independent, given Y.
- Given this assumption, then:

P(X1,X2|Y) = P(X1|X2, Y) P(X2|Y)
= P(X1|Y) P(X2|Y)

In general

$$P(X_1 \ldots X_n|Y) = \prod_i P(X_i|Y)$$

How many parameters needed to describe P(X|Y)? P(Y)?
- Without conditional independent assumption?
- With conditional independent assumption?

Naïve Bayes in a Nutshell

Bayes rule:

$$P(Y = y_k | X_1 \dots X_n) = \frac{P(Y = y_k) P(X_1 \dots X_n | Y = y_k)}{\sum_j P(Y = y_j) P(X_1 \dots X_n | Y = y_j)}$$

Assuming conditional independence among Xi's:

$$P(Y = y_k | X_1 \dots X_n) = \frac{P(Y = y_k) \prod_i P(X_i | Y = y_k)}{\sum_j P(Y = y_j) \prod_i P(X_i | Y = y_j)}$$

So, classification rule for Xnew =(X1, …, Xn) is:

$$Y^{new} \leftarrow \arg\max_{y_k} \; P(Y = y_k) \prod_i P(X_i^{new} | Y = y_k)$$

Naïve Bayes Classifier (not BE)

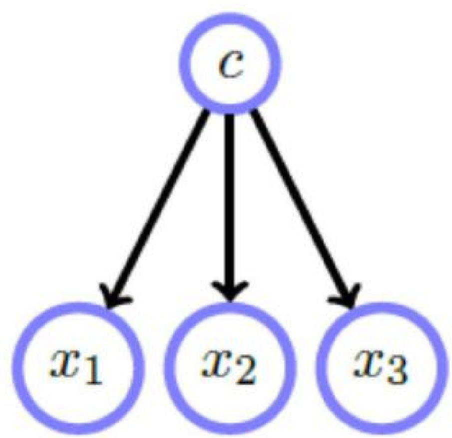
- Simple classifier that applies Bayes' rule with strong (naive) independence assumptions
- A.k.a. the 'independent feature model"

$$p(\omega_i|x_1, x_2,\ldots) = \alpha\, p(x_1|\omega_i)\, p(x_2|\omega_i)\ldots\, p(\omega_i)$$

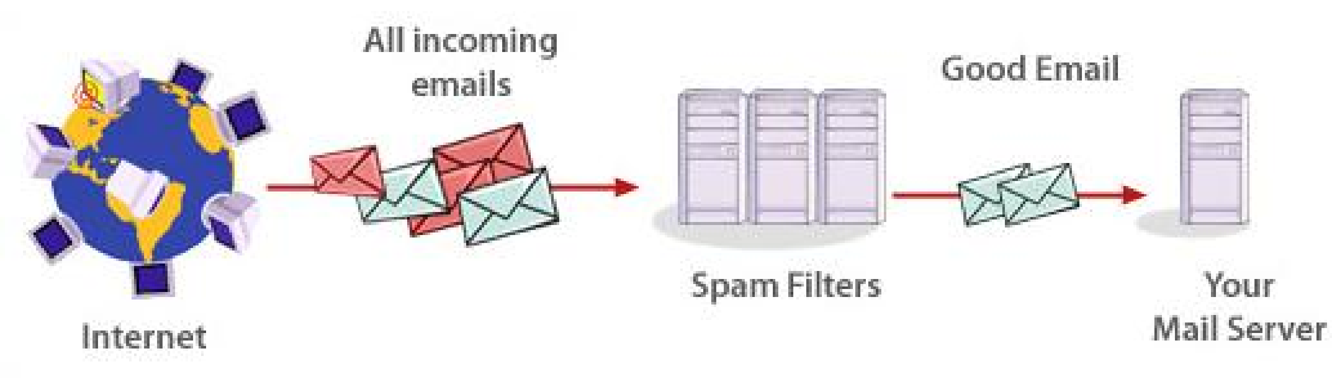
- Often performs reasonably well despite simplicity

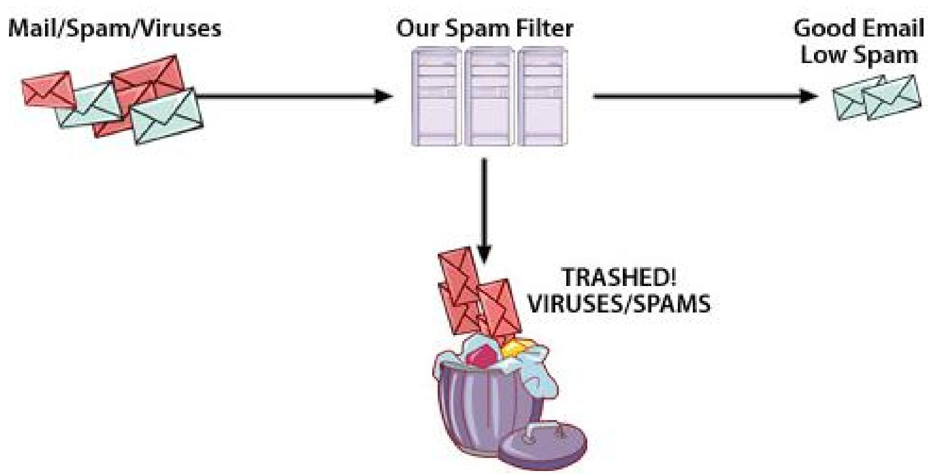
Naïve Bayes Classifier

- NB is known to produce posteriors closer to extremes (0 or 1) than true posteriors
  - Why？
- NB performs well when only small amounts of training data are available
  - Why？



Example: Email Classification





Example: Email Classification

- **Training data**: a corpus of email messages, each message annotated as spam or no spam.
- **Task**: classify new email messages as spam/no spam.
- To use a naive Bayes classifier for this task, we have to first find an attribute representation of the data.
- Treat each text position as an attribute, with as its value the word at this position.
- e.g., email starts: "get rich".
- The naive Bayes classifier is then:

$$\begin{aligned} v_{\mathrm{NB}} &= \underset{v_j \in \{\mathrm{spam,nospam}\}}{\arg\max} P(v_j) \prod_i P(a_i | v_j) \\ &= \underset{v_j \in \{\mathrm{spam,nospam}\}}{\arg\max} P(v_j) P(a_1 = get | v_j) P(a_2 = rich | v_j) \end{aligned}$$

Example: Email Classification

- Using naive Bayes means we assume that **words are independent of each other**. Clearly incorrect, but doesn't hurt lot for our task.

- The classifier $P(a_i = w_k | v_j)$ uses i.e., the probability that the i–th word in the email is the k–word in our vocabulary, given the email has been classified as v_j.

Example: Email Classification

- Simplify by assuming that **position is irrelevant**: estimate $P(w_k|v_j)$, i.e., the probability that word w_k occurs in the email, given class v_j .

- Create a **vocabulary**: make a list of all words in the training corpus, discard words with very high or very low frequency.

Example: Email Classification

- **Training**: estimate priors:

  $P(v_j) = \frac{n}{N}$

- Estimate likelihoods using the m–estimate:

  $P(w_k|v_j) = \frac{n_k+1}{n+|Vocabulary|}$

  N: total number of words in all emails
  n: number of words in emails with class v_j
  n_k: number of times word w_k occurs in emails with class v_j
  $|Vocabulary|$: size of the vocabulary

- **Testing**: to classify a new email, assign it the class with the highest posterior probability. Ignore unknown words.

Outline

- Bayesian Estimation
- Naive Bayes Classifier
- **Data Split and Cross-Validation**
- Overfitting

Training/Test Split

- Randomly split dataset into two parts:
  - Training data
  - Test data
- Use training data to optimize parameters
- Evaluate error using test data

Training/Test Split

- How many points in each set？
- Very hard question
  - Too few points in training set, learned classifier is bad
  - Too few points in test set, classifier evaluation is insufficient
- Cross–validation
- Leave–one–out cross–validation

Cross-Validation

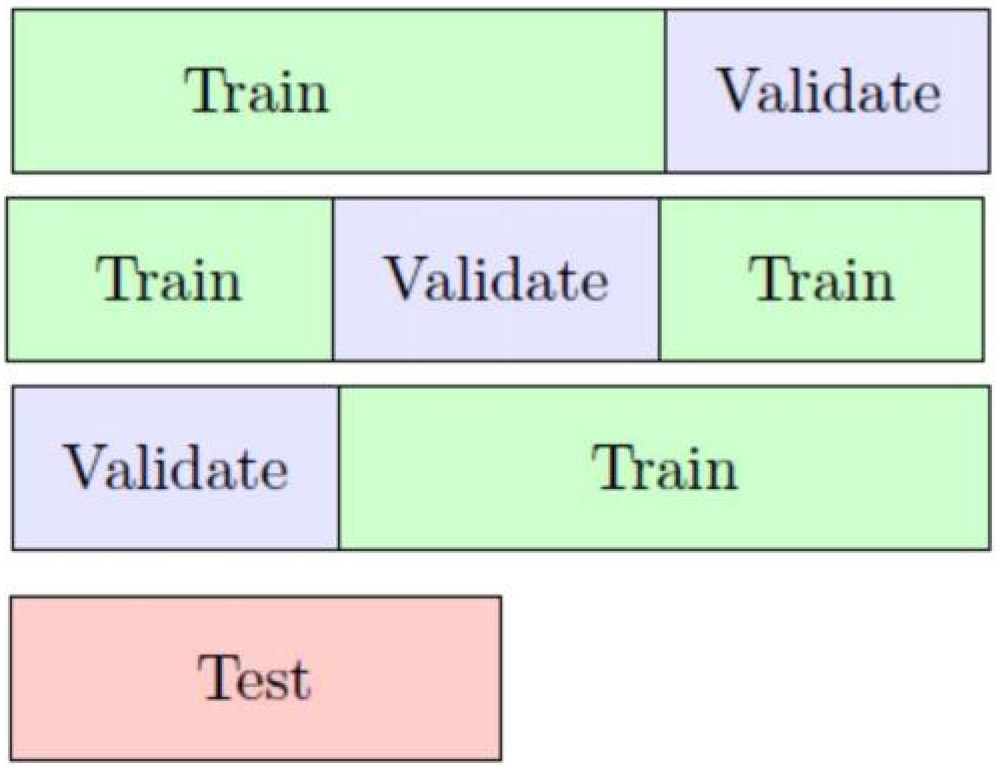
- In practice
- Available data => training and validation
- Train on the training data
- Test on the validation data
- k–fold cross validation:
  - Data randomly separated into k groups
  - Each time k
  - 1 groups used for training and one as testing

Cross Validation and Test Accuracy

- If we select parameters so that CV is highest:
  - Does CV represent future test accuracy?
  - Slightly different
- If we have enough parameters, we can achieve 100% CV as well
  - e.g. more parameters than # of training data
- But test accuracy may be different
- So split available data with class labels, into:
  - training
  - validation
  - testing

Cross Validation and Test Accuracy

- Using CV on training + validation
- Classify test data with the best parameters from CV



Outline

- Bayesian Estimation
- Naive Bayes Classifier
- Data Split and Cross-Validation
- **Overfitting**

Overfitting

- Prediction error: probability of test pattern not in class with max posterior (true)
- Training error: probability of test pattern not in class with max posterior (estimated)
- Classifier optimized w.r.t. training error
  - Training error: optimistically biased estimate of prediction error

Overfitting

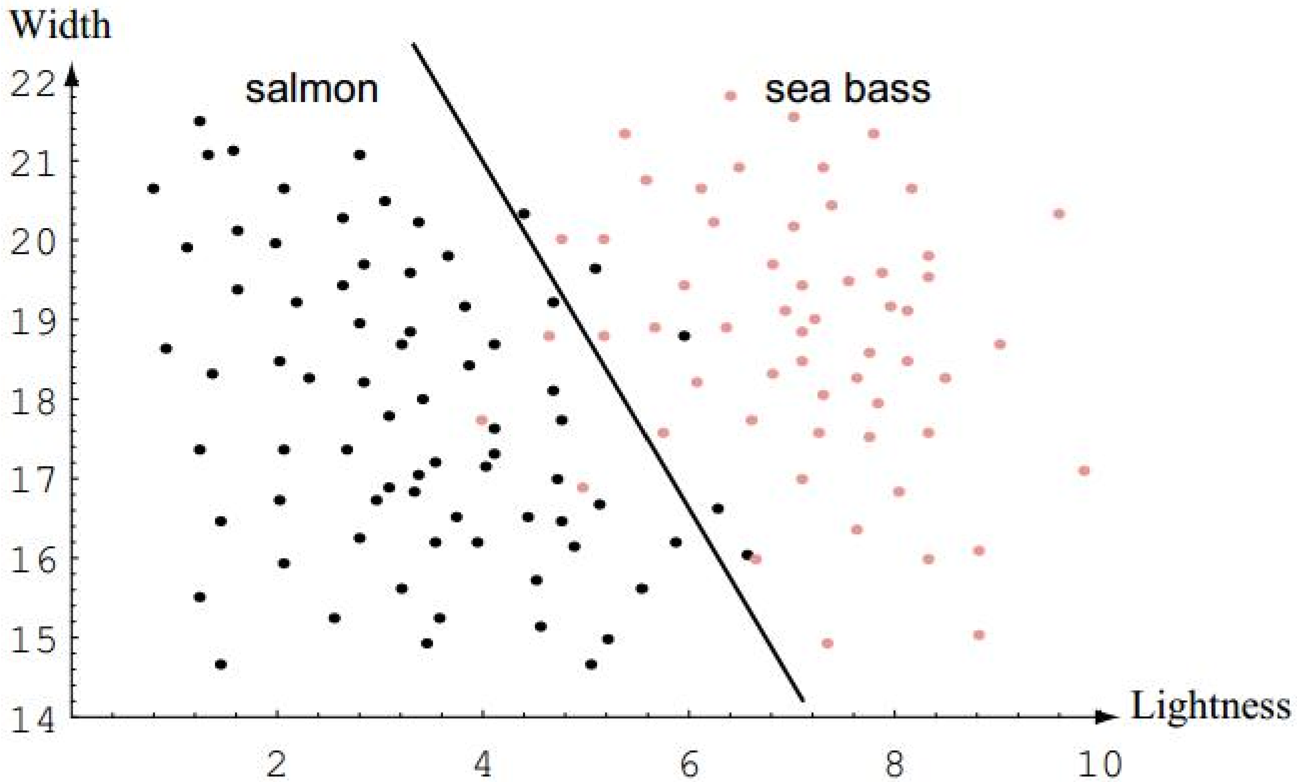
- Overfitting: a learning algorithm overfits the training data if it outputs a solution w when another solution w' exists such that:

$$\text{error}_{\text{train}}(w) < \text{error}_{\text{train}}(w')$$

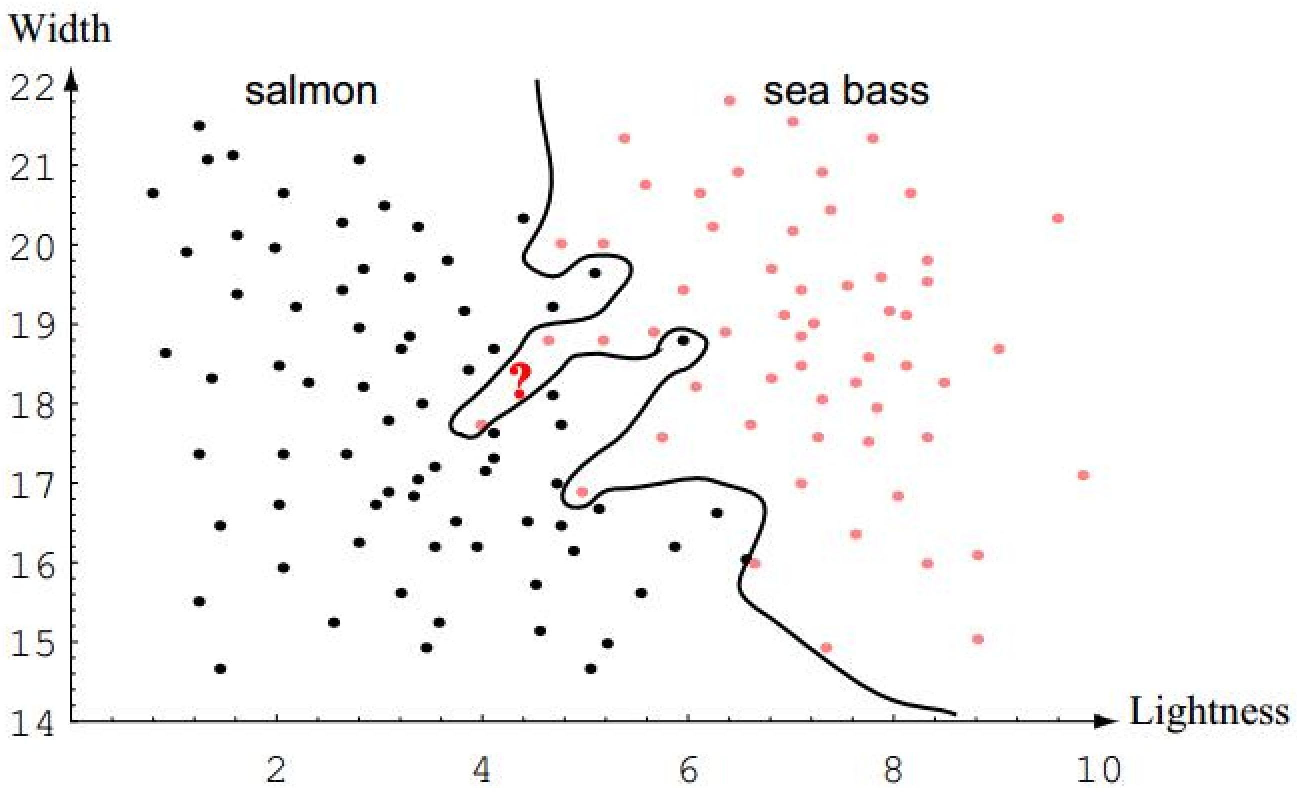
AND

$$\text{error}_{\text{true}}(w') < \text{error}_{\text{true}}(w)$$

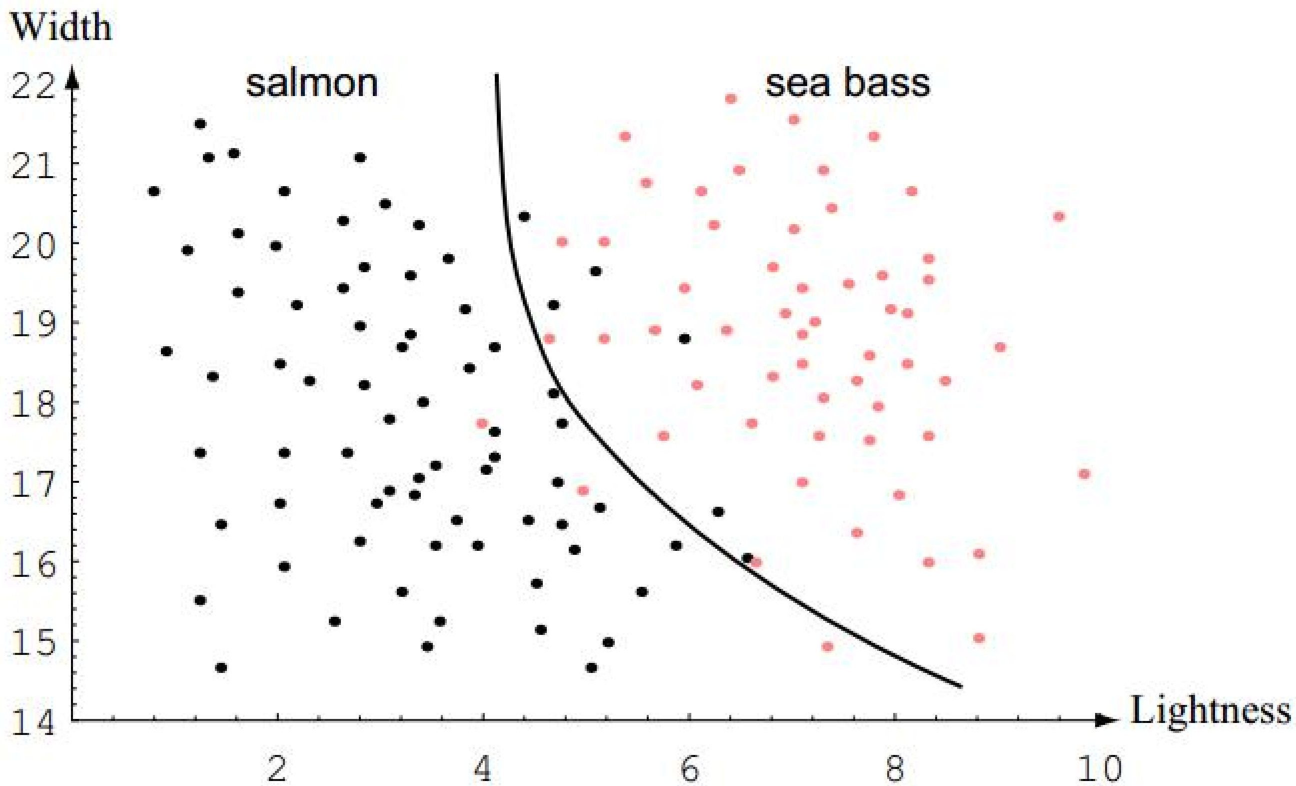
Example: Fish Classifier



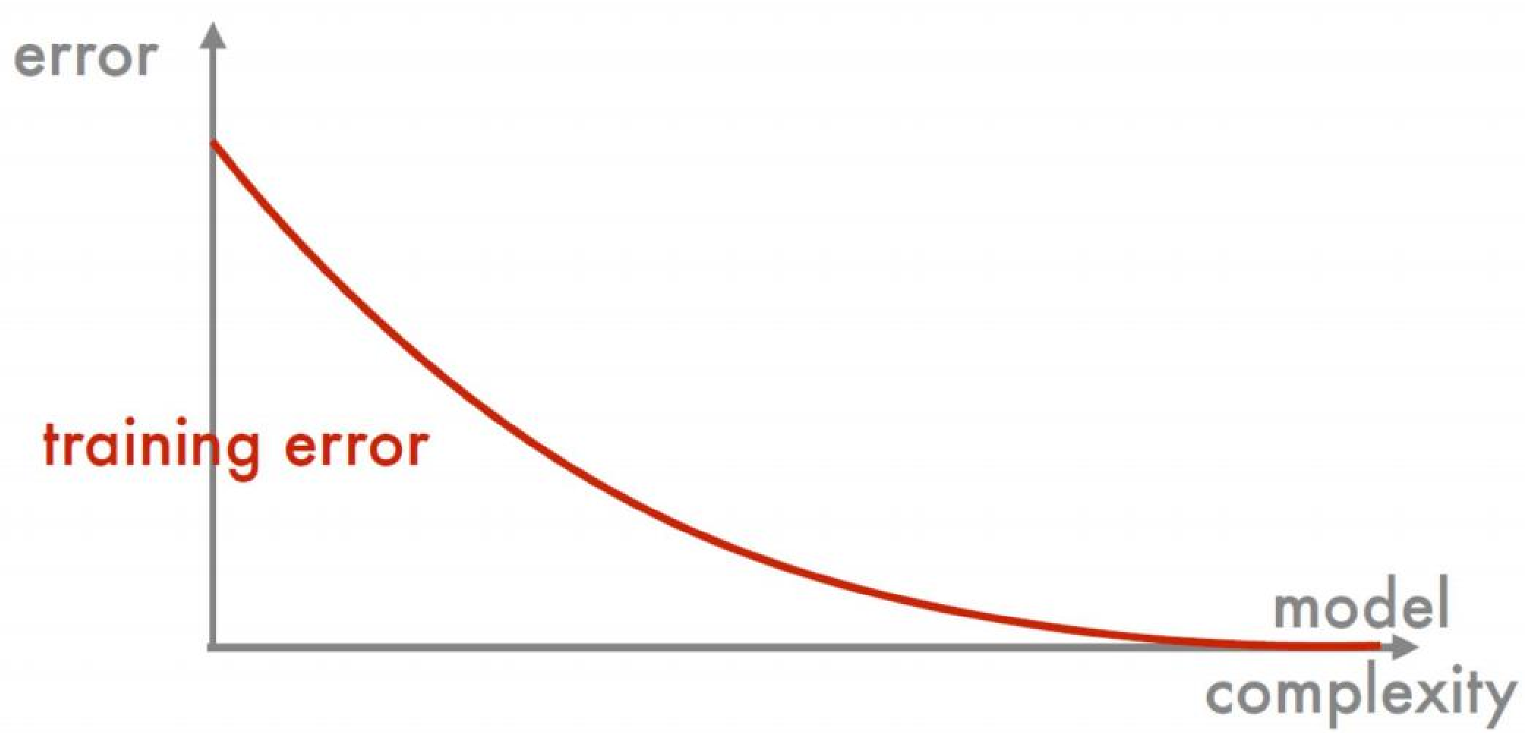
Minimum Training Error



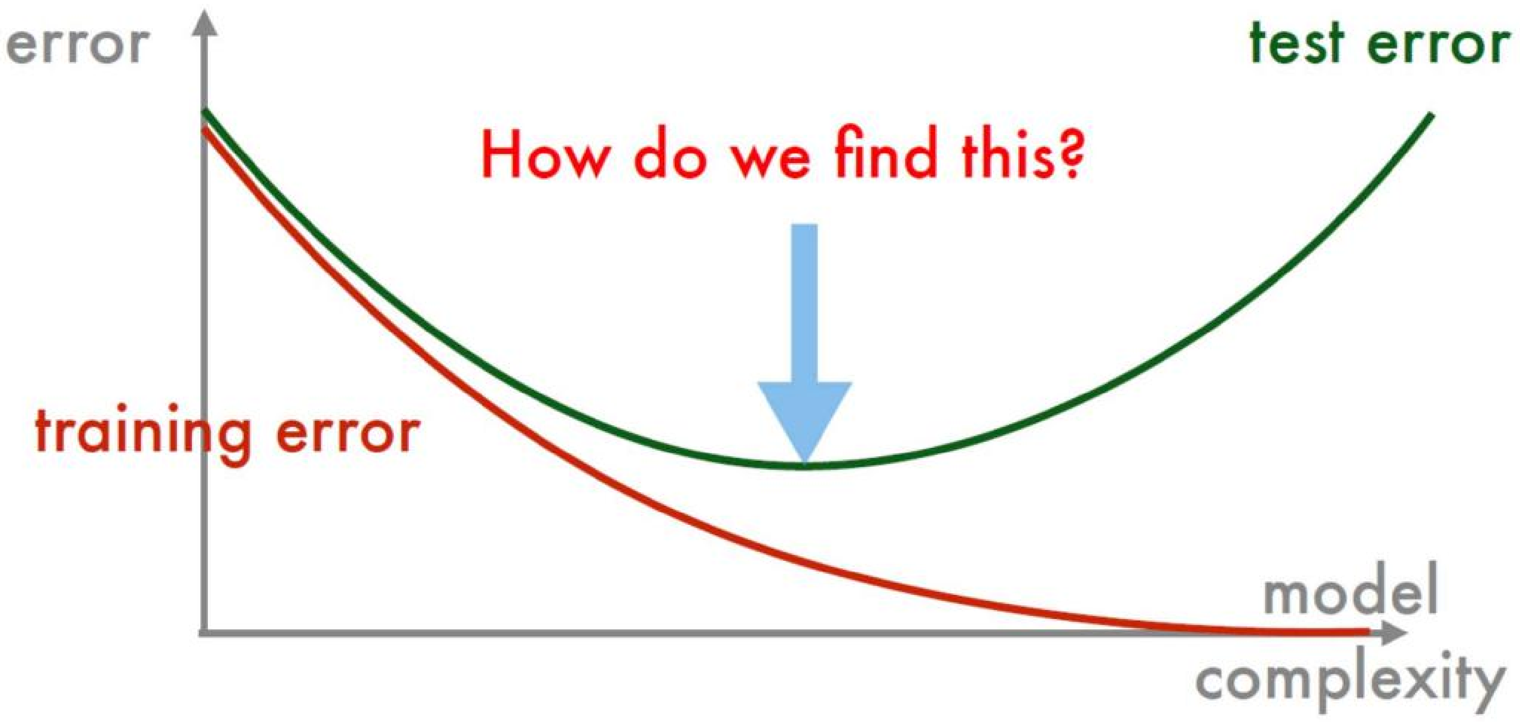
Final Decision Boundary



Typical Behavior

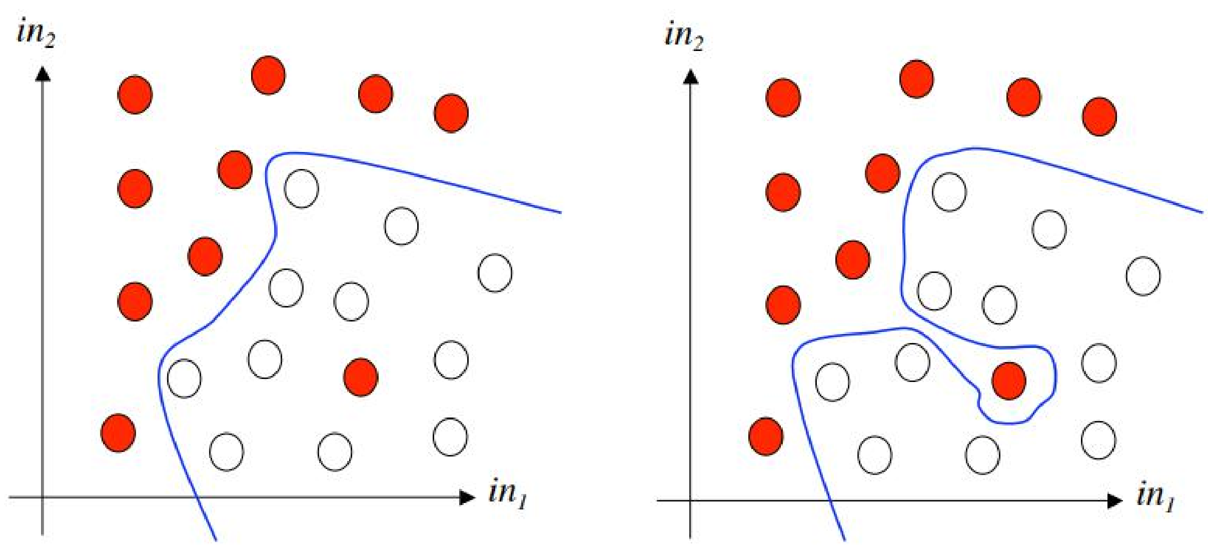


Typical Behavior



Typical Behavior

- The aim is to get the network to generalize to classify new inputs appropriately.
- If the training data is known to contain noise, we don't necessarily want the training data to be classified totally accurately, because that is likely to reduce the generalization ability.



Q & A