DS 5220

Supervised Machine Learning and Learning Theory

Alina Oprea
Associate Professor, CCIS
Northeastern University
Guest lecture by Virgil Pavlu

September 18 2019

Outline

- Classification
- Linear classification
- Perceptron
  - Online and batch perceptron
- LDA
  - Generative models

Supervised learning

Problem Setting

- Set of possible instances $\mathcal{X}$
- Set of possible labels $\mathcal{Y}$
- Unknown target function $f : \mathcal{X} \rightarrow \mathcal{Y}$
- Set of function hypotheses $H = \{h \mid h : \mathcal{X} \rightarrow \mathcal{Y}\}$

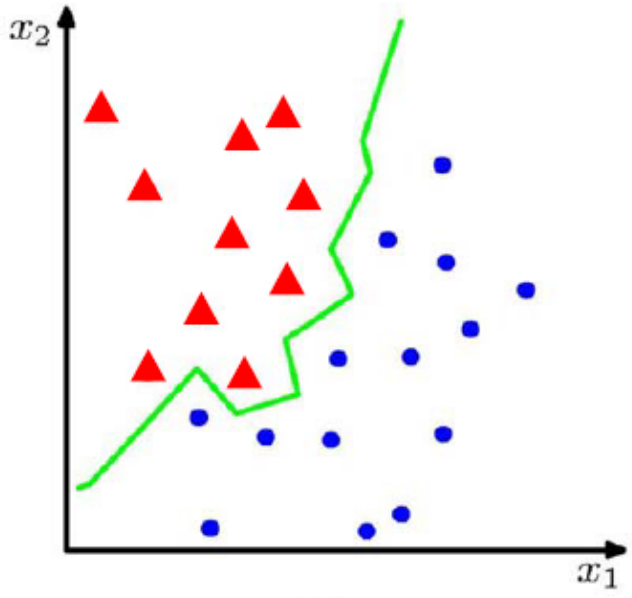
Input: Training examples of unknown target function f

$$\{x_i, y_i\}, \text{for } i = 1, \ldots, n$$

Output: Hypothesis $\hat{f} \in H$ that best approximates f

$$\hat{f}(x_i) \approx y_i$$

Classification



Binary or discrete

- Suppose we are given a training set of N observations

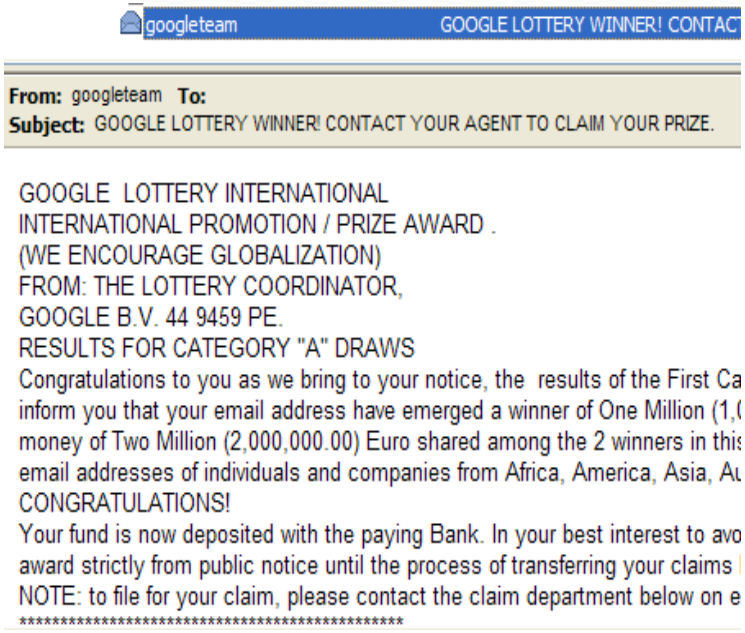
$$\{x_1, \dots, x_N\} \text{ and } \{y_1, \dots, y_N\}, x_i \in R^d, y_i \in \{-1, 1\}$$

- Classification problem is to estimate f(x) from this data such that

$$f(x_i) = y_i$$

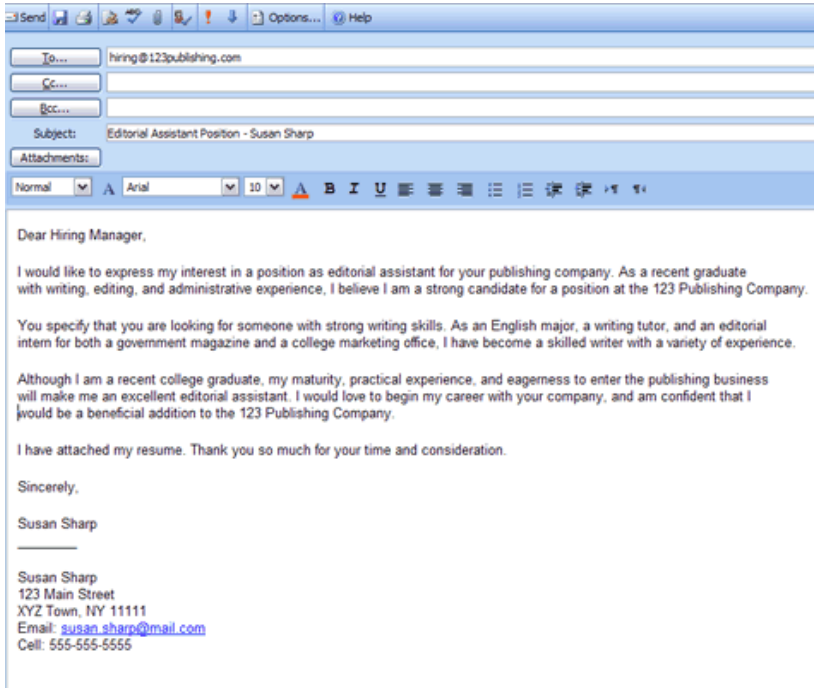
Example 1

Classifying spam email


googleteam GOOGLE LOTTERY WINNER! CONTACT

From: googleteam **To:**
Subject: GOOGLE LOTTERY WINNER! CONTACT YOUR AGENT TO CLAIM YOUR PRIZE.

GOOGLE LOTTERY INTERNATIONAL
INTERNATIONAL PROMOTION / PRIZE AWARD .
(WE ENCOURAGE GLOBALIZATION)
FROM: THE LOTTERY COORDINATOR,
GOOGLE B.V. 44 9459 PE.
RESULTS FOR CATEGORY "A" DRAWS
Congratulations to you as we bring to your notice, the results of the First Ca
inform you that your email address have emerged a winner of One Million (1,(
money of Two Million (2,000,000.00) Euro shared among the 2 winners in this
email addresses of individuals and companies from Africa, America, Asia, Au
CONGRATULATIONS!
Your fund is now deposited with the paying Bank. In your best interest to avo
award strictly from public notice until the process of transferring your claims
NOTE: to file for your claim, please contact the claim department below on e
****


Send Options... Help

To... hiring@123publishing.com
Cc...
Bcc...
Subject: Editorial Assistant Position - Susan Sharp
Attachments:

Normal Arial 10

Dear Hiring Manager,

I would like to express my interest in a position as editorial assistant for your publishing company. As a recent graduate with writing, editing, and administrative experience, I believe I am a strong candidate for a position at the 123 Publishing Company.

You specify that you are looking for someone with strong writing skills. As an English major, a writing tutor, and an editorial intern for both a government magazine and a college marketing office, I have become a skilled writer with a variety of experience.

Although I am a recent college graduate, my maturity, practical experience, and eagerness to enter the publishing business will make me an excellent editorial assistant. I would love to begin my career with your company, and am confident that I would be a beneficial addition to the 123 Publishing Company.

I have attached my resume. Thank you so much for your time and consideration.

Sincerely,

Susan Sharp

Susan Sharp
123 Main Street
XYZ Town, NY 11111
Email: susan.sharp@mail.com
Cell: 555-555-5555

Content-related features

- Use of certain words
- Word frequencies
- Language
- Sentence

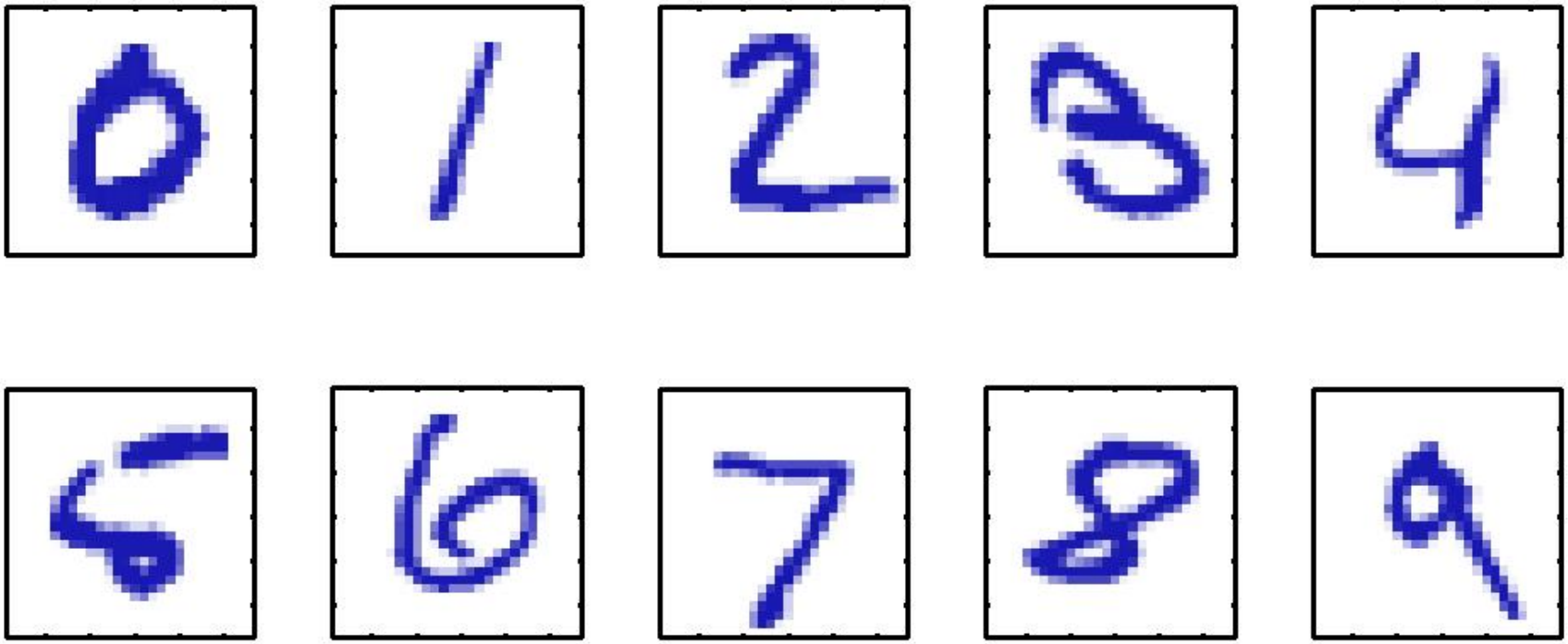
Structural features

- Sender IP address
- IP blacklist
- DNS information
- Email server
- URL links (non-matching)

Binary classification: SPAM or HAM

Example 2

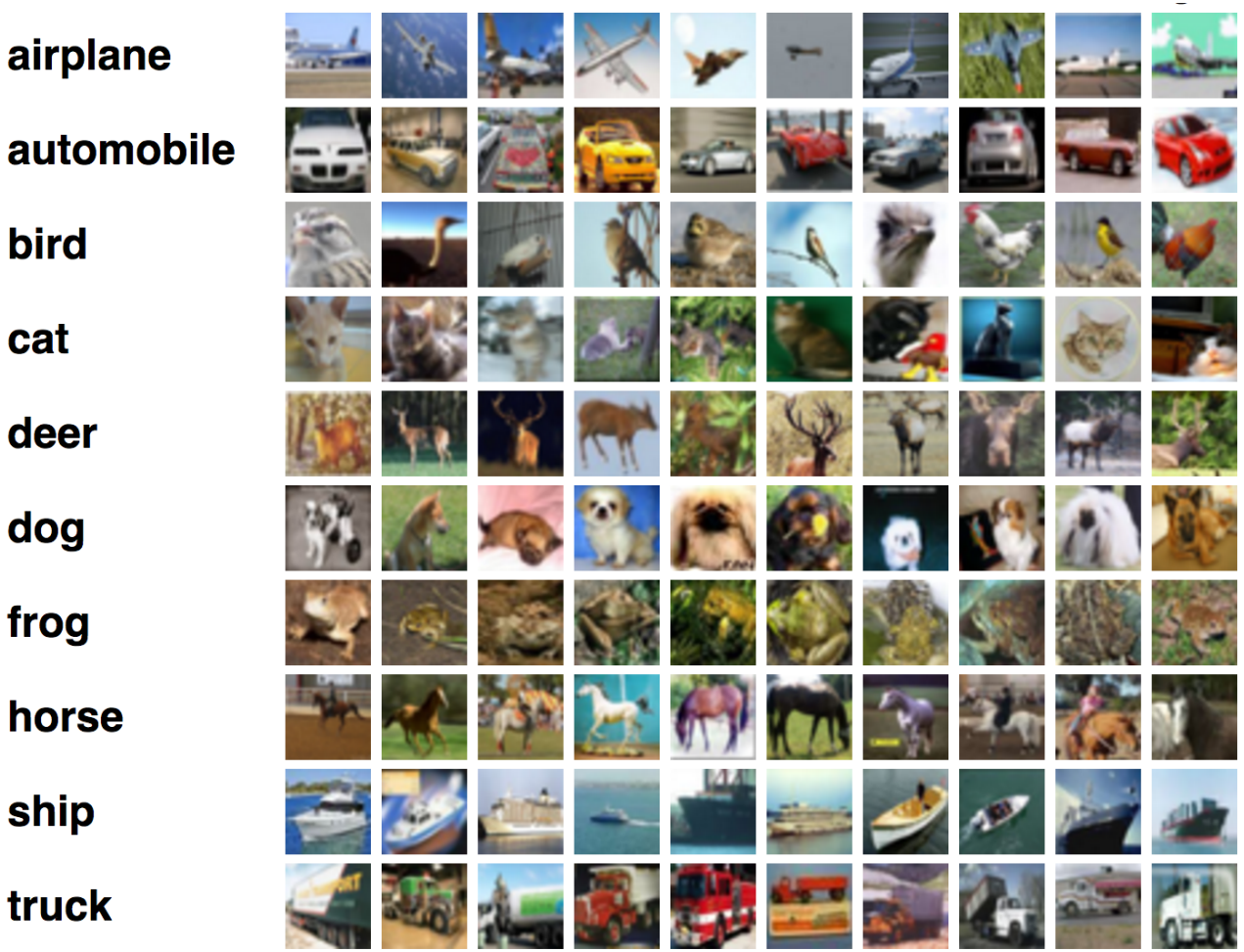
Handwritten Digit Recognition



Multi-class classification

Example 3

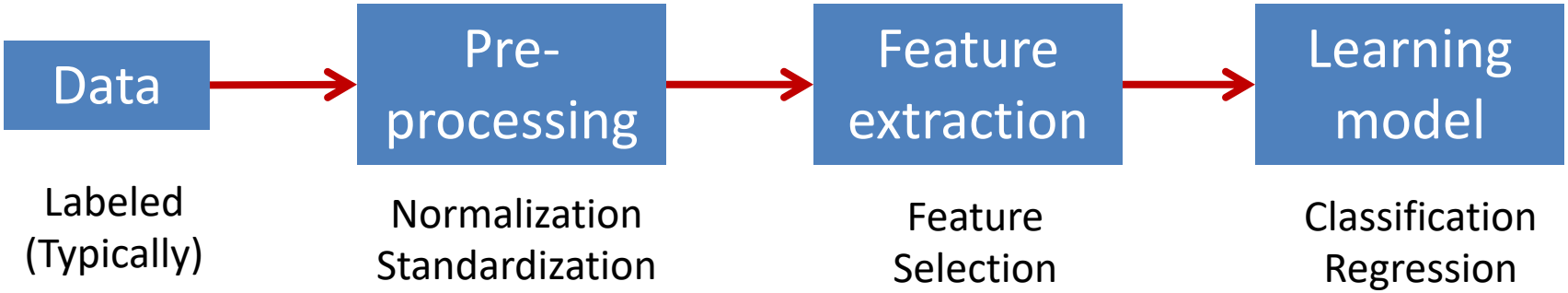
Image classification



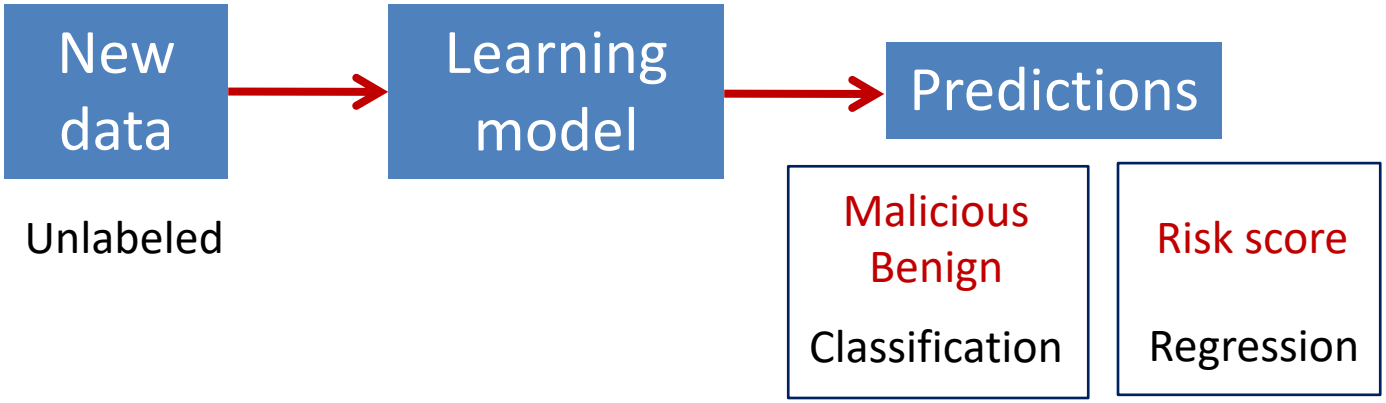
Multi-class classification

Supervised Learning Process

Training



Testing



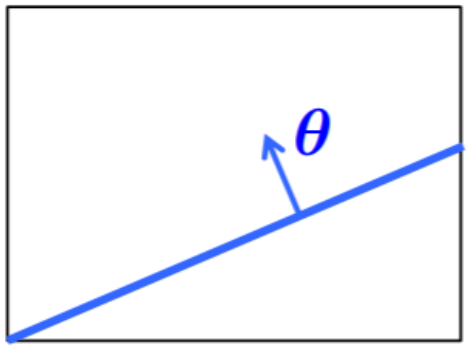
History of Perceptrons

- They were popularised by Frank Rosenblatt in the early 1960's.
  - They appeared to have a very powerful learning algorithm.
  - Lots of grand claims were made for what they could learn to do.
- In 1969, Minsky and Papert published a book called "Perceptrons" that analysed what they could do and showed their limitations.
  - Many people thought these limitations applied to all neural network models.
- The perceptron learning procedure is still widely used today for tasks with enormous feature vectors that contain many millions of features.

They are the basic building blocks for
Deep Neural Networks

Linear classifiers

- A **hyperplane** partitions space into 2 half-spaces
  - Defined by the normal vector $\boldsymbol{\theta} \in \mathbb{R}^{d+1}$
    - $\boldsymbol{\theta}$ is orthogonal to any vector lying on the hyperplane



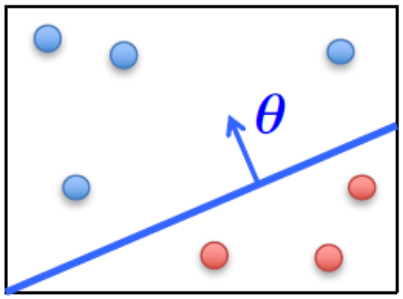
  - Assumed to pass through the origin
    - This is because we incorporated bias term θ_0 into it by $x_0 = 1$

- Consider classification with +1, -1 labels ...

Linear classifiers

- **Linear classifiers**: represent decision boundary by hyperplane

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \quad \boldsymbol{x}^\intercal = \begin{bmatrix} 1 & x_1 & \dots & x_d \end{bmatrix}$$



$$h(\boldsymbol{x}) = \text{sign}(\boldsymbol{\theta}^\intercal \boldsymbol{x}) \text{ where } \quad \text{sign}(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{if } z < 0 \end{cases}$$

– Note that: $\boldsymbol{\theta}^\intercal \boldsymbol{x} > 0 \implies y = +1$

$\boldsymbol{\theta}^\intercal \boldsymbol{x} < 0 \implies y = -1$

All the points x on the hyperplane satisfy: $\theta^T x = 0$

Example: Spam

- Imagine 3 features (spam is "positive" class):
  1. free (number of occurrences of "free")
  2. money (occurrences of "money")
  3. BIAS (intercept, always has value 1)

$$\sum_{i=0}^{d} x_i \theta_i$$

"free money"

x	
BIAS	1
free	1
money	1
...	

θ	
BIAS	-3
free	4
money	2
...	

$$(1)(-3) + (1)(4) + (1)(2) + \ldots = 3$$

$\sum_i x_i \theta_i > 0$ ➔ SPAM!!!

The Perceptron

$$h(\boldsymbol{x}) = \text{sign}(\boldsymbol{\theta}^\intercal \boldsymbol{x}) \quad \text{where} \quad \text{sign}(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{if } z < 0 \end{cases}$$

- The perceptron uses the following update rule each time it receives a new training instance (x_i, y_i)

$$\theta_j \leftarrow \theta_j - \frac{1}{2}(h_\theta(x_i) - y_i)x_{ij}$$

either 2 or -2

  - If the prediction matches the label, make no change
  - Otherwise, adjust $\boldsymbol{\theta}$

The Perceptron

- The perceptron uses the following update rule each time it receives a new training instance (x_i, y_i)

$$\theta_j \leftarrow \theta_j - \frac{1}{2}(h_\theta(x_i) - y_i)x_{ij}$$

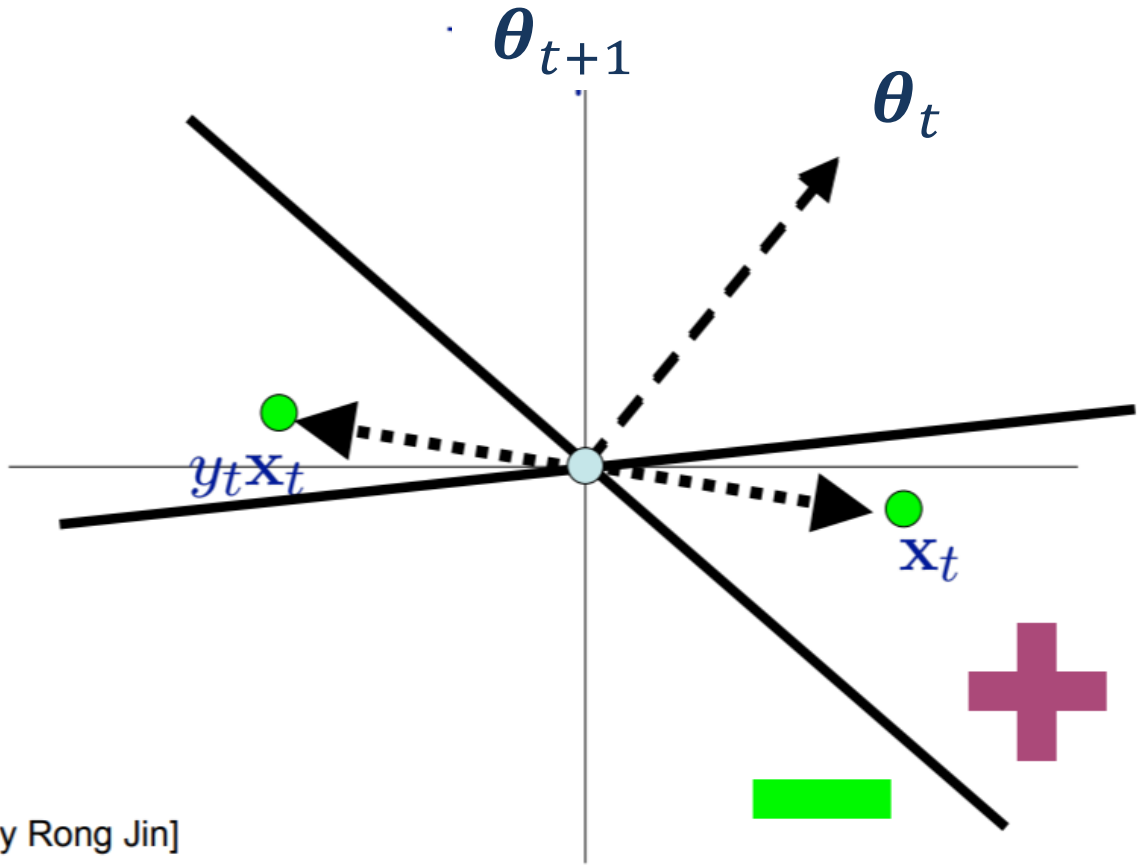
either 2 or -2

- Re-write as $\theta_j \leftarrow \theta_j + y_i x_{ij}$ (only upon misclassification)

Perceptron Rule: If x_i is misclassified, do

$$\theta \leftarrow \theta + y_i\, x_i$$

Geometric interpretation



[Slide by Rong Jin]

Online Perceptron

Let $\theta \leftarrow$[0,0,…,0]
Repeat:
    Receive training example (x_i, y_i)
    If $y_i \theta^T x_i \leq 0$ // prediction is incorrect
        $\theta \leftarrow \theta + y_i \, x_i$

Online learning – the learning mode where the model update is performed each time a single observation is received

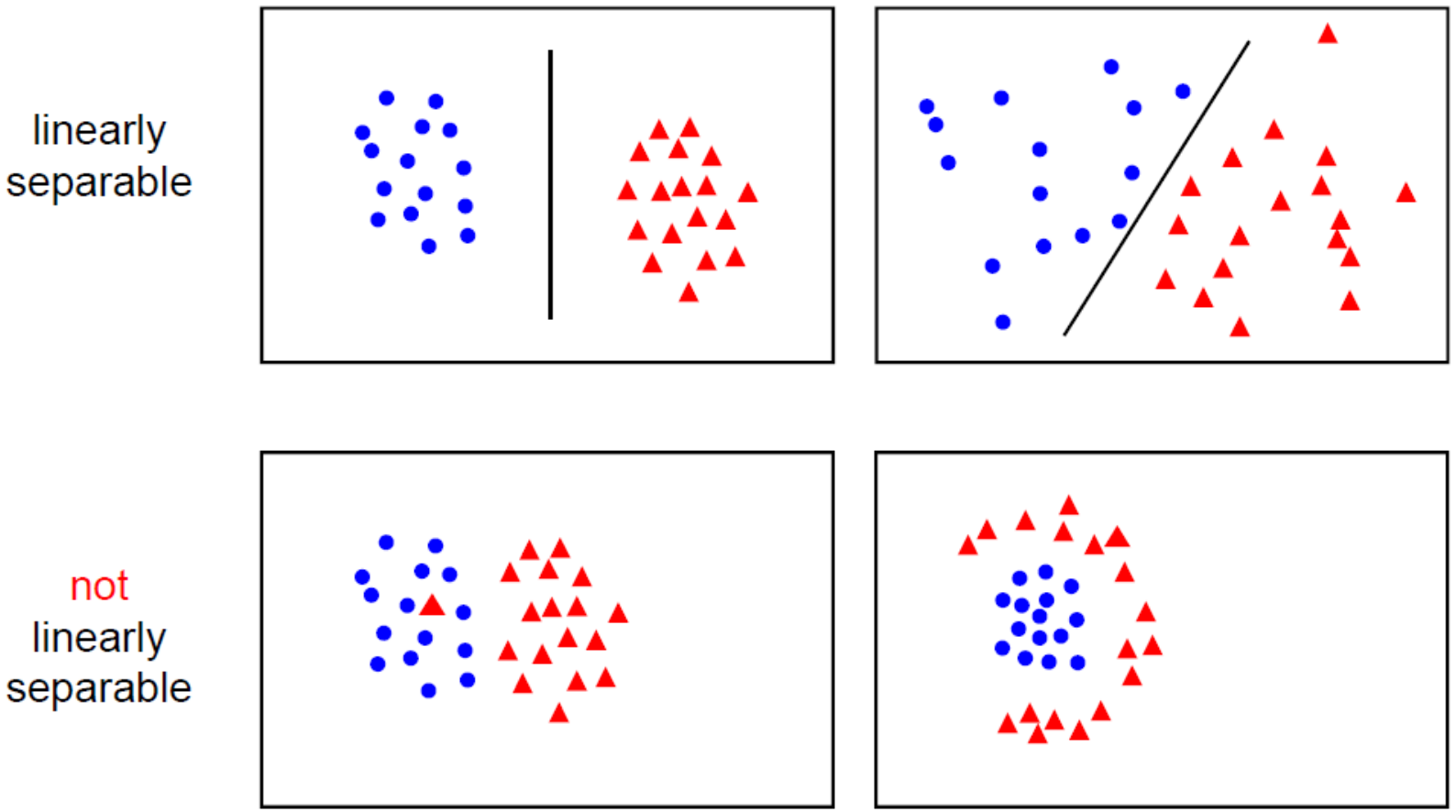
Batch learning – the learning mode where the model update is performed after observing the entire training set

Batch Perceptron

Given training data $\{x_i, y_i\}_{i=1}^{n}$
Let $\boldsymbol{\theta} \leftarrow [0, 0, \ldots, 0]$
Repeat:
    Let $\boldsymbol{\Delta} \leftarrow [0, 0, \ldots, 0]$
    for $i = 1 \ldots n$, do
        if $y_i \theta^T x_i \leq 0$ // prediction for ith instance is incorrect
            $\boldsymbol{\Delta} \leftarrow \boldsymbol{\Delta} + y_i x_i$
    $\boldsymbol{\Delta} \leftarrow \boldsymbol{\Delta}/n$ // compute average update
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{\Delta}$
Until $\|\boldsymbol{\Delta}\|_2 < \epsilon$

Guaranteed to find separating hyperplane if data is linearly separable

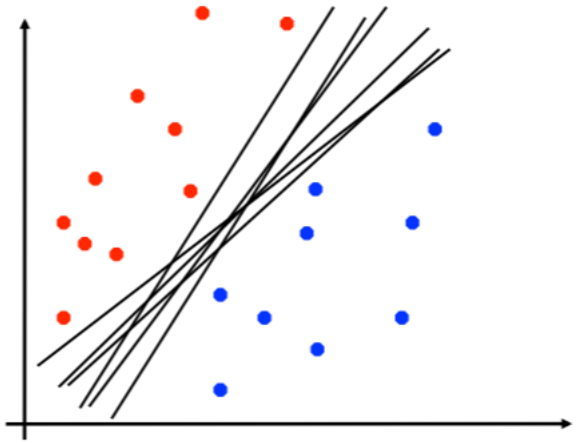
Linear separability



- For linearly separable data, can prove bounds on perceptron error (depends on how well separated the data is)

Perceptron Limitations

- Is dependent on starting point
- It could take many steps for convergence
- Perceptron can overfit
  - Move the decision boundary for every example



Which of this is optimal?

Improving the Perceptron

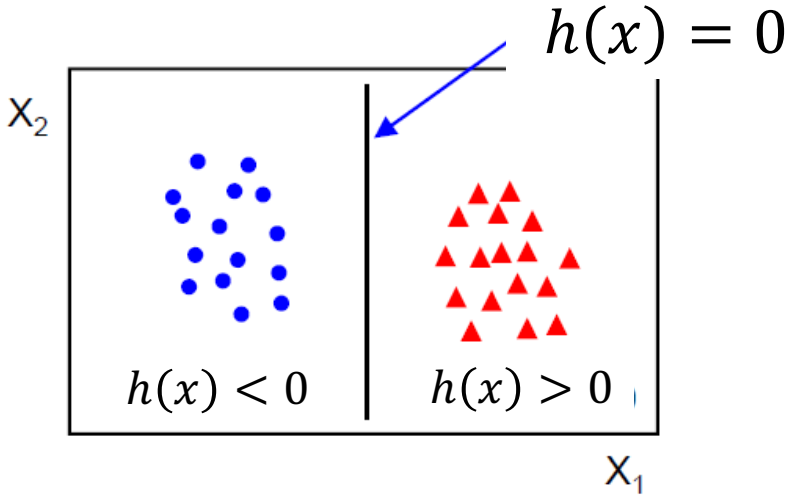
- The Perceptron produces many θ's during training
- The standard Perceptron simply uses the final θ at test time
  - This may sometimes not be a good idea!
  - Some other θ may be correct on 1,000 consecutive examples, but one mistake ruins it!

- **Idea:** Use a combination of multiple perceptrons
  - (i.e., neural networks!)
- **Idea:** Use the intermediate θ's
  - **Voted Perceptron**: vote on predictions of the intermediate θ's
  - **Averaged Perceptron**: average the intermediate θ's

Linear classifiers

A linear classifier has the form

$$h_\theta(x) = f(\theta^T x)$$



- Properties
  - $(\theta_0, \theta_1, \ldots, \theta_d)$ = model parameters
  - Perceptron is a special case with $f = sign$
- Pros
  - Very compact model (size d)
  - Perceptron is fast
- Cons
  - Does not work for data that is not linearly separable

LDA

- Classify to one of k classes
- Logistic regression computes directly
  - $\mathrm{P}[Y = 1|X = x]$
  - Assume sigmoid function
- LDA uses Bayes Theorem to estimate it
  - $\mathrm{P}[Y = k|X = x] = \frac{\mathrm{P}[X = x|Y = k]\mathrm{P}[Y=k]}{\mathrm{P}[X=x]}$
  - Let $\pi_k = \mathrm{P}[Y = k]$ be the prior probability of class k and $f_k(x) = \mathrm{P}[X = x|Y = k]$

LDA

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}.$$

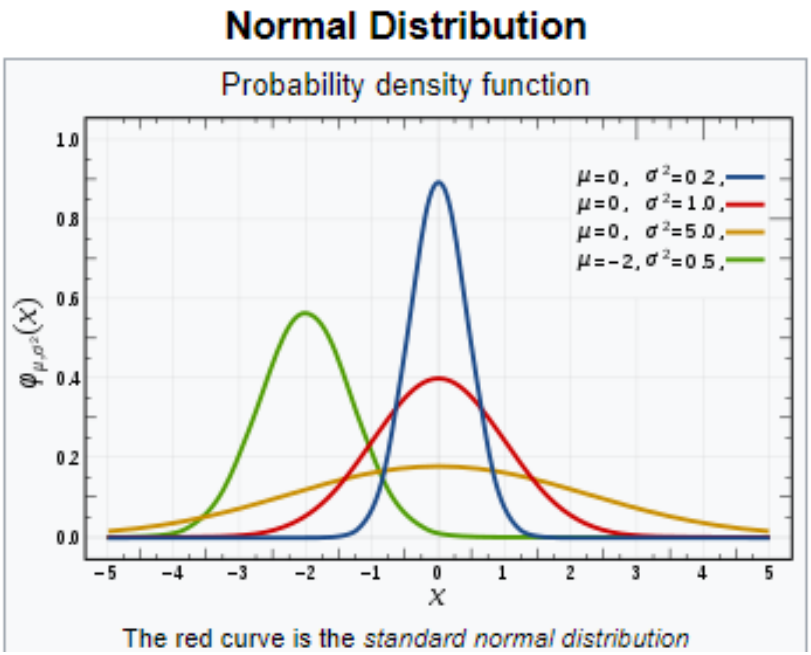
Assume $f_k(x)$ is Gaussian!
Unidimensional case (d=1)

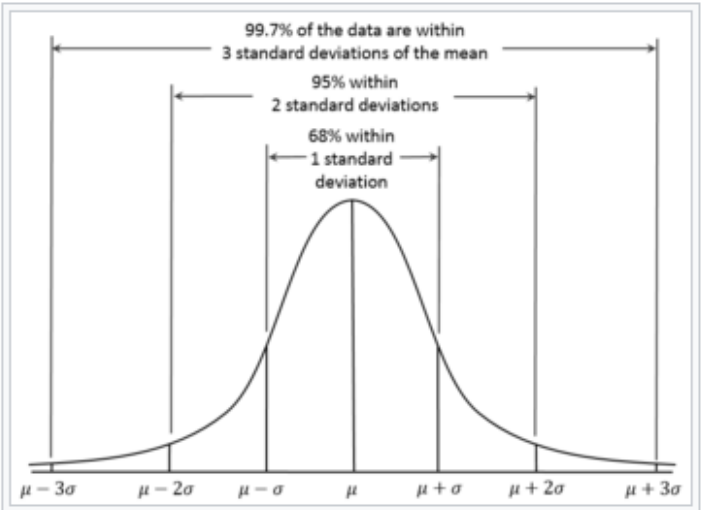
$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x - \mu_k)^2\right)$$

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_l)^2\right)}.$$

Assumption: $\sigma_1 = \ldots \sigma_k = \sigma$

Gaussian Distribution





For the normal distribution, the values less than one standard deviation away from the mean account for 68.27% of the set; while two standard deviations from the mean account for 95.45%; and three standard deviations account for 99.73%.

Notation	$\mathcal{N}(\mu, \sigma^2)$
Parameters	$\mu \in \mathbb{R}$ = mean (location) $\sigma^2 > 0$ = variance (squared scale)
Support	$x \in \mathbb{R}$
PDF	$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

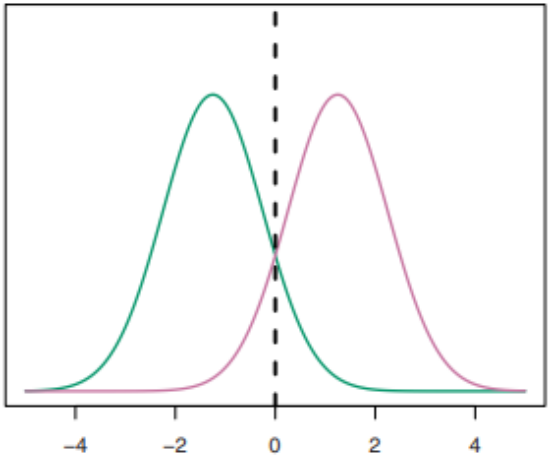
LDA decision boundary

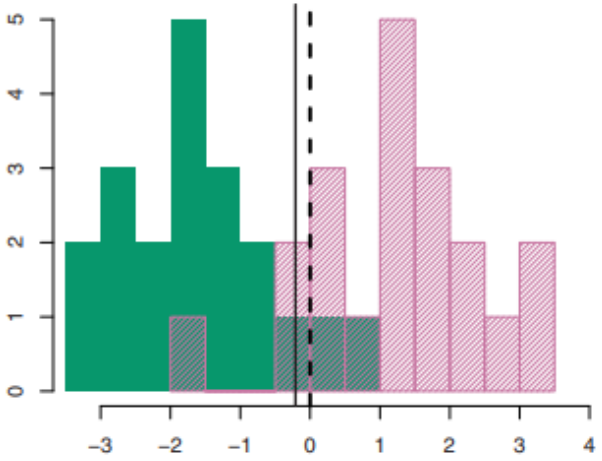
Pick class k to maximize

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

Example: $k = 2, \pi_1 = \pi_2$

Classify as class 1 if $x > \frac{\mu_1 + \mu_2}{2\sigma}$





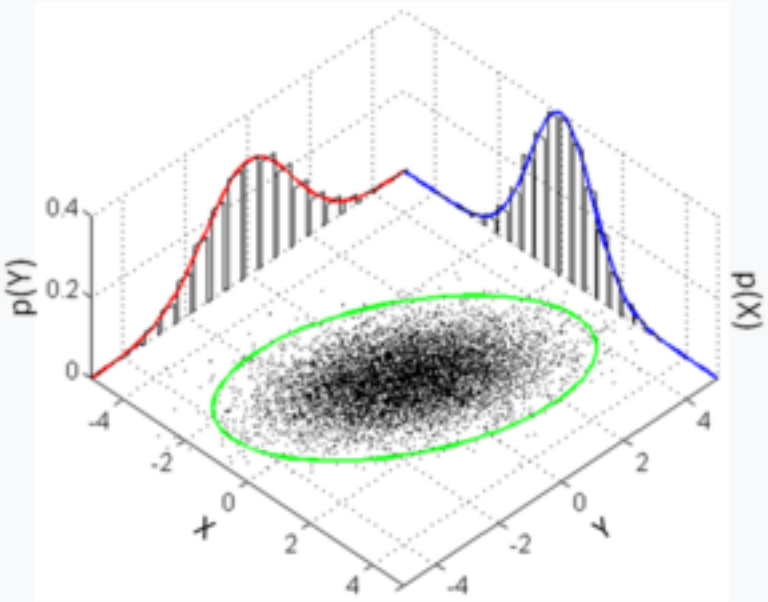
True decision boundary

Estimated decision boundary

Multi-Variate Normal

Multivariate normal

Probability density function



Many sample points from a multivariate normal distribution with $\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 3/5 \\ 3/5 & 2 \end{bmatrix}$, shown along with the 3-sigma ellipse, the two marginal distributions, and the two 1-d histograms.

$$\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

with *k*-dimensional mean vector

$$\boldsymbol{\mu} = \mathrm{E}[\mathbf{X}] = [\mathrm{E}[X_1], \mathrm{E}[X_2], \ldots, \mathrm{E}[X_k]]^{\mathrm{T}},$$

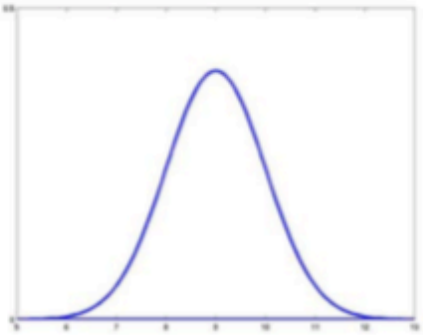
and $k \times k$ covariance matrix

$$\Sigma_{i,j} =: \mathrm{E}[(X_i - \mu_i)(X_j - \mu_j)] = \mathrm{Cov}[X_i, X_j]$$

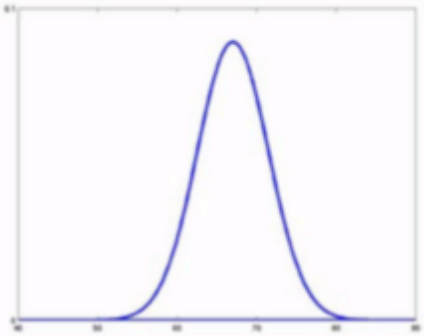
$$\boldsymbol{\Sigma} =: \mathrm{E}[(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^{\mathrm{T}}] = [\mathrm{Cov}[X_i, X_j]; 1 \leq i, j \leq k].$$

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \boldsymbol{\Sigma}_k^{-1}(x-\mu_k)}.$$

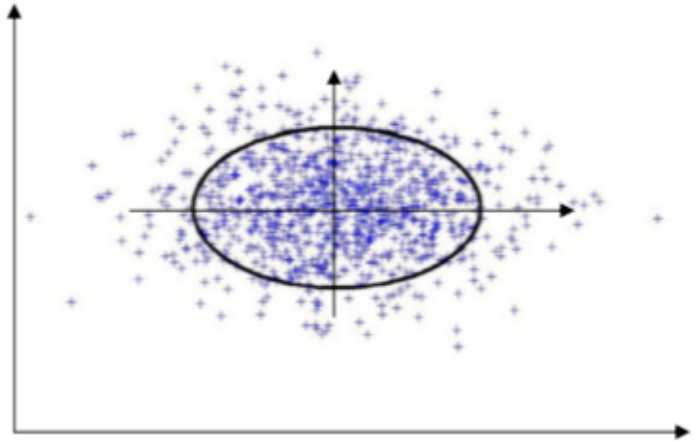
Example: Independent variables



people's heights:
$X \sim \mathrm{N}(67, 20)$



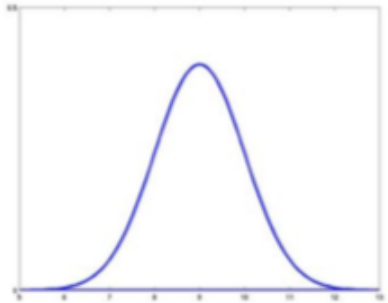
time people woke up this morning: $Y \sim \mathrm{N}(9, 1)$



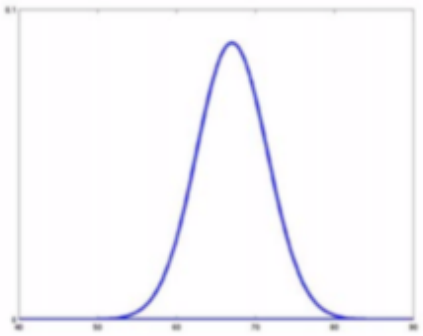
Co-variance matrix

$$\begin{bmatrix} 20 & 0 \\ 0 & 9 \end{bmatrix}$$

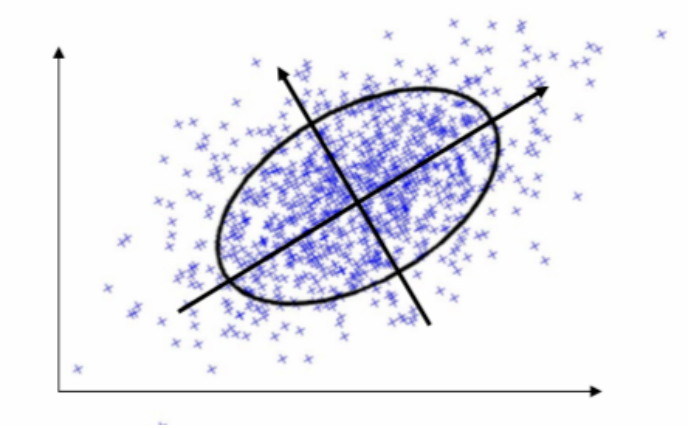
Example: Correlated variables



people's heights:
$X \sim \mathrm{N}(67, 20)$



People's weight
$Y \approx N(177,40)$



Co-variance matrix

$$\begin{bmatrix} 20 & 5 \\ 5 & 40 \end{bmatrix}$$

Multi-variate LDA

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}.$$

Assume $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$

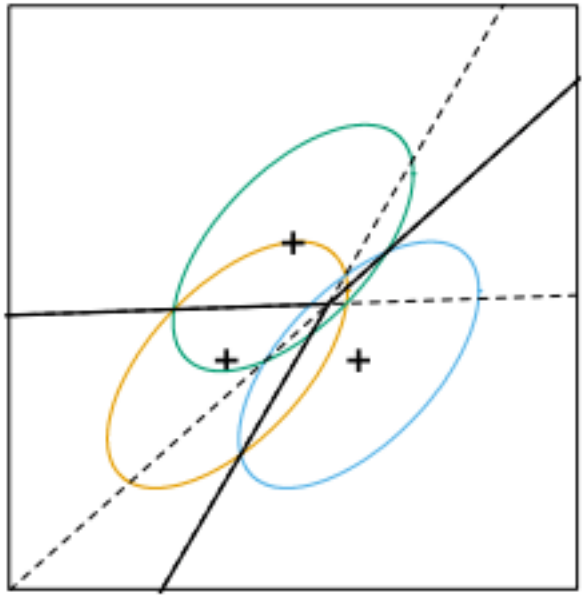
$$\log \frac{\Pr(Y = k|X = x)}{\Pr(Y = l|X = x)z} = \log \frac{f_k(x)}{f_\ell(x)} + \log \frac{\pi_k}{\pi_\ell}$$
$$= \log \frac{\pi_k}{\pi_\ell} - \frac{1}{2}(\mu_k + \mu_\ell)^T \boldsymbol{\Sigma}^{-1}(\mu_k - \mu_\ell)$$
$$+ x^T \boldsymbol{\Sigma}^{-1}(\mu_k - \mu_\ell),$$

Linear decision boundary between classes *k* and *l*

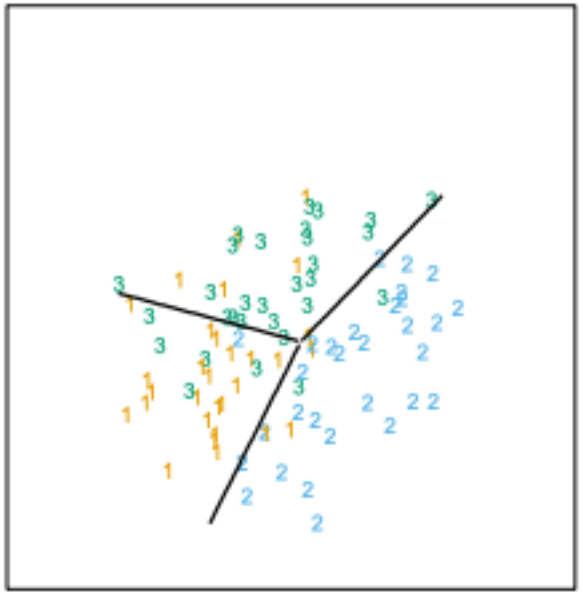
Linear discriminant functions $\delta_k(x) = x^T \boldsymbol{\Sigma}^{-1} \mu_k - \frac{1}{2} \mu_k^T \boldsymbol{\Sigma}^{-1} \mu_k + \log \pi_k$

Given x, classify to class k: $argmax_k \delta_k(x)$

Example 3 classes



3 Normal distributions with same co-variance, but different means



LDA decision boundary

LDA in practice

Given training data $(x_i, y_i), i = 1, \ldots, n, y_i \in \{1, \ldots, K\}$

1. Estimate mean and variance

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

2. Estimate prior

$$\hat{\pi}_k = n_k/n.$$

Given testing point x, predict k that maximizes:

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

Multi-variate LDA

Given training data $(x_i, y_i), i = 1, \ldots, n, y_i \in \{1, \ldots, K\}$

1. Estimate mean and variance

- $\hat{\pi}_k = N_k/N$, where N_k is the number of class-k observations;
- $\hat{\mu}_k = \sum_{g_i=k} x_i/N_k$;
- $\hat{\mathbf{\Sigma}} = \sum_{k=1}^{K} \sum_{g_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T/(N - K)$.

2. Estimate prior

Given testing point x, predict k that maximizes:

$$\delta_k(x) = x^T \mathbf{\Sigma}^{-1} \mu_k - \frac{1}{2} \mu_k^T \mathbf{\Sigma}^{-1} \mu_k + \log \pi_k$$

Acknowledgements

- Slides made using resources from:
  - Andrew Ng
  - Eric Eaton
  - David Sontag
- Thanks!