CS480 – Introduction to Artificial Intelligence

Topic: Learning

Mustafa Bilgic

http://www.cs.iit.edu/~mbilgic

https://twitter.com/bilgicm

LEARNING

- What's learning?
- Intro to Chapter 18: *"In which we describe agents that can improve their behavior through diligent study of their own experiences."*
- We do not make any philosophical statements about whether the agent is *truly* learning
- *"An agent is learning if it improves its performance on future tasks after making observations about the world."*

Why Learn and not program directly?

- We cannot anticipate all possible situations that the agent might find itself in
- Time/location/context changes knowledge and rules
- We might not know the solution crisp enough to program it
- We might not have time to encode all the knowledge

WHAT TO LEARN?

- Which action to take in a state (state $\rightarrow$ action)
- Outcomes of our actions (action $\rightarrow$ state)
- Mapping percepts to world states (percept $\rightarrow$ state)
- Utility of the states (state $\rightarrow$ utility)
- and more…

FEEDBACK

- Unsupervised learning
  - No feedback; the agent discovers patterns in the data
  - E.g., clustering, dimensionality reduction, outlier detection
- Supervised learning
  - Feedback: input-output pairs
  - E.g., classification, regression, ranking
- Reinforcement learning
  - Feedback: rewards

Episodic vs Sequential

- Supervised and unsupervised learning are often episodic
  - E.g., speech recognition, medical diagnosis, credit score prediction, …
- Reinforcement learning is often sequential
  - E.g., game playing

Machine Learning

- ML is used to supplement several applications of AI
- Even though all the rage is now about deep learning, DL is a subfield of ML, and ML is a subfield of AI
- Example
  - Agents can combine the powers of search and ML to play games
  - Robots can use ML to make sense of their percepts and model the world, but they need to use search and planning to achieve goals

We'll Cover

1. Bayesian network parameter estimation
2. Supervised learning
3. Reinforcement learning

Bayesian Network Parameter Estimation

Bayesian network parameter estimation

- Given:
  - A set of random variables, V_i
    - E.g., age, gender, cholesterol level, etc.
  - A Bayesian network structure over these variables
    - E.g., a doctor can point out the most important correlations and causations
  - Data
    - E.g., existing patient records, where some or all V_i are known
- Goal:
  - Estimate the parameters needed for the Bayesian network, i.e., $P(V_i \mid \text{parentsOf}(V_i))$

Known Bayesian network structure

- In this class, we assume the structure is given
- How reasonable is this assumption?
  - In some domains, the expert might provide a reasonable structure to start with
- There are many methods that learn the structure of the Bayesian network from data
  - Those topics are covered in the CS583 – Probabilistic Graphical Models course in detail

Parameter estimation for BNs

- Assume the network structure is given over variables V_i
- Let d_j be a fully observed instance
  - $d_j = \langle V_1 = \text{t}, V_2 = \text{f}, \ldots, V_n = \text{t} \rangle$
- The data $\mathcal{D}$ consists of fully observed instances
  - $\mathcal{D} = \{d_1, d_2, \ldots, d_m\}$
- Estimate the network parameters $\text{P}(V_i \mid \text{parents}(V_i))$
- Two approaches
  1. Maximum likelihood estimation
  2. Bayesian estimation

Simplest case – one variable

- Imagine we have a thumbtack
- Flip it, and it comes as heads or tails

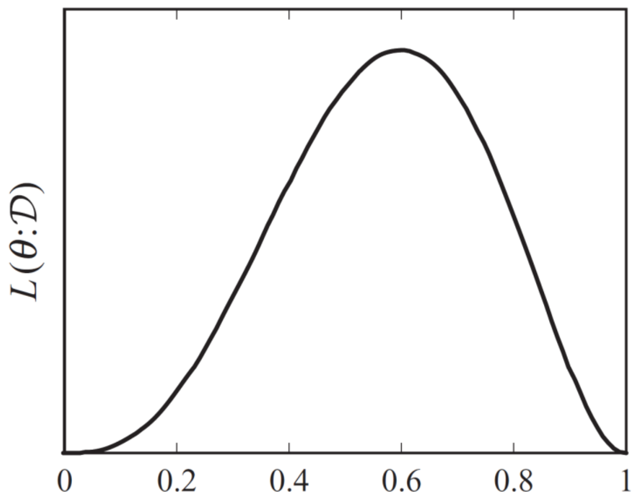
- P(Heads) = θ, P(Tails) = 1- θ
- Assume we flip it 100 times and it comes head 30 times
- What is θ?

Thumbtack tosses

- Assume we have a set of thumbtack tosses
  - $\mathcal{D} = \{d_1, d_2, \ldots, d_{100}\}$
- Assume we have 30 heads and 70 tails
- P(Heads) = θ, P(Tails) = 1- θ
- θ can be any number between 0 and 1
- We have an infinite number of choices
  - θ=0, …, θ=0.3, …, θ=0.5, …, θ=1
- We want to formulate an objective function f(θ: $\mathcal{D}$), where, given 30 heads and 70 tails, f(θ: $\mathcal{D}$) achieves its maximum when θ=0.3
  - Any ideas?

LIKELIHOOD

- What is the probability, or *likelihood*, of seeing the sequence H, T, T, H, H?
  - $\theta * (1-\theta) * (1-\theta) * \theta * \theta = \theta^3 (1-\theta)^2$



When is $L(\theta : \mathcal{D})$ maximum?

LIKELIHOOD/LOG-LIKELIHOOD

- Number of heads = k, number of tails = m-k
- Likelihood: $L(\theta:\mathcal{D}) = \theta^k(1-\theta)^{m-k}$
- Log-likelihood: $l(\theta:\mathcal{D}) = k\log\theta+(m-k)\log(1-\theta)$
- Note that $L(\theta:\mathcal{D})$ achieves its maximum for θ that maximizes $l(\theta:\mathcal{D})$
- Find θ that maximizes the log-likelihood
- Take derivate of $l(\theta:\mathcal{D})$ w.r.t. θ and set it to zero

Maximum likelihood for a multinomial

- Domain of X is {A, B, C}
- We see A a times, B b times, and C c times.
- P(X=A) is p, P(X=B) is q, and P(C) = $1 - p - q$
- What are p and q?
- Proof?

Let's see a few examples

- Simple structure
  - X→Y
- General structure
  - The key is that the parameters for each variable can be optimized independently
  - Examples

Bayesian estimation

- Assume we flip a coin 10 times and we get 4 Heads, 6 Tails
  - What is P(C=H)?
- What if we repeat the flips 10M times and we get 4M Heads and 6M Tails?
- Bayesian estimation will let us encode our *prior knowledge*

To cut it short, (I mean really short)

- We'll encode our prior knowledge as a set of "imaginary" counts
- For example, we will assume that we have already seen α heads and β tails
- Assume we flip a coin 10 times and we get 4 Heads, 6 Tails
  - $P(C=heads) = (4 + \alpha) / (10 + \alpha + \beta)$
  - $\alpha = 0, \beta = 0; P(C=h) = 4/10 = 0.4$
  - $\alpha = 1, \beta = 1; P(C=h) = 5/12 = 0.417$
  - $\alpha = 10, \beta = 10; P(C=h) = 14/30 = 0.467$
  - $\alpha = 100, \beta = 100; P(C=h) = 104/210 = 0.495$
- Assume we flip a coin 1000 times and we get 400 Heads, 600 Tails
  - $P(C=heads) = (400 + \alpha) / (1000 + \alpha + \beta)$
  - $\alpha = 0, \beta = 0; P(C=h) = 400/1000 = 0.4$
  - $\alpha = 1, \beta = 1; P(C=h) = 401/1002 = 0.4002$
  - $\alpha = 10, \beta = 10; P(C=h) = 410/1020 = 0.402$
  - $\alpha = 100, \beta = 100; P(C=h) = 500/1200 = 0.417$

Imaginary counts

- Note that imaginary counts can be applied to any categorical variable, not necessarily just binary variables
- Also helps with dealing zero probabilities
- When all imaginary counts are 1, this is called Laplace smoothing
  - E.g, $\alpha = 1$, $\beta = 1$
- Let's see some examples

Supervised Learning

FEEDBACK

- Unsupervised learning
  - No feedback; the agent discovers patterns in the data
  - E.g., clustering, dimensionality reduction, outlier detection
- Supervised learning
  - Feedback: input-output pairs
  - E.g., classification, regression, ranking
- Reinforcement learning
  - Feedback: rewards

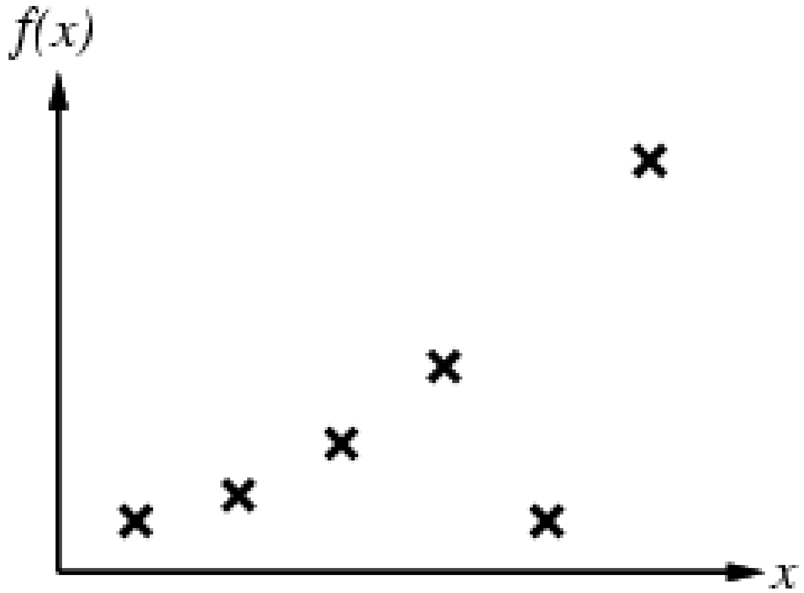
Supervised learning

- Given objects with their labels, <X,Y>
- Learn a function f that maps objects, X, to labels, Y
- We want f to perform well on unseen objects
- Several applications
  - Face recognition, speech recognition, medical diagnosis, fraud detection, credit scoring, home value prediction, temperature prediction, …
- If Y is
  - Discrete, the task is called classification
  - Continuous, the task is called regression

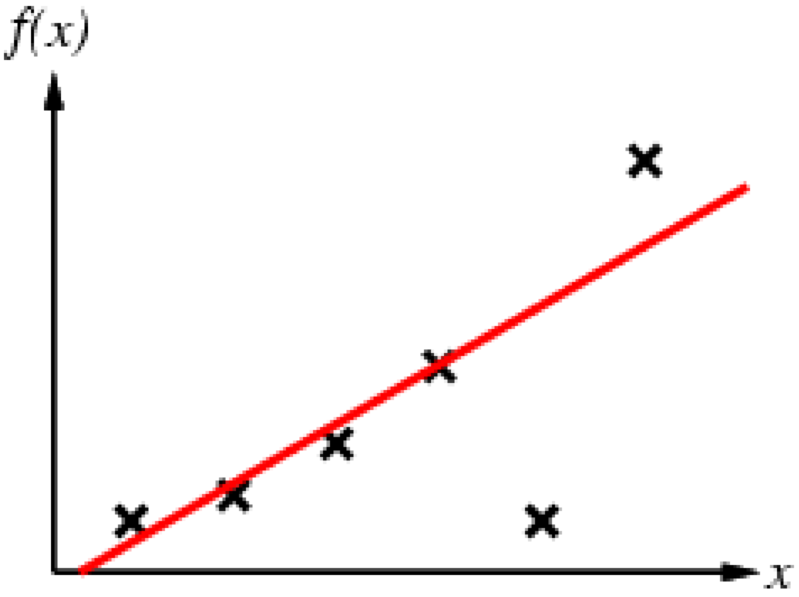
Function Fitting?

- Isn't classification/regression simply "function fitting?"
- Yes and No
- The purpose is to generalize and perform well on unseen data
- We don't want to underfit or overfit to the training data

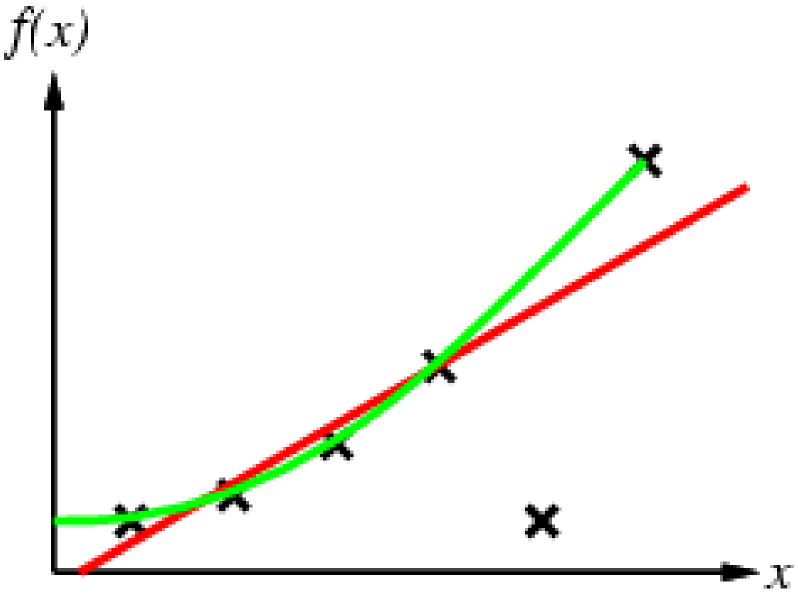
Curve Fitting



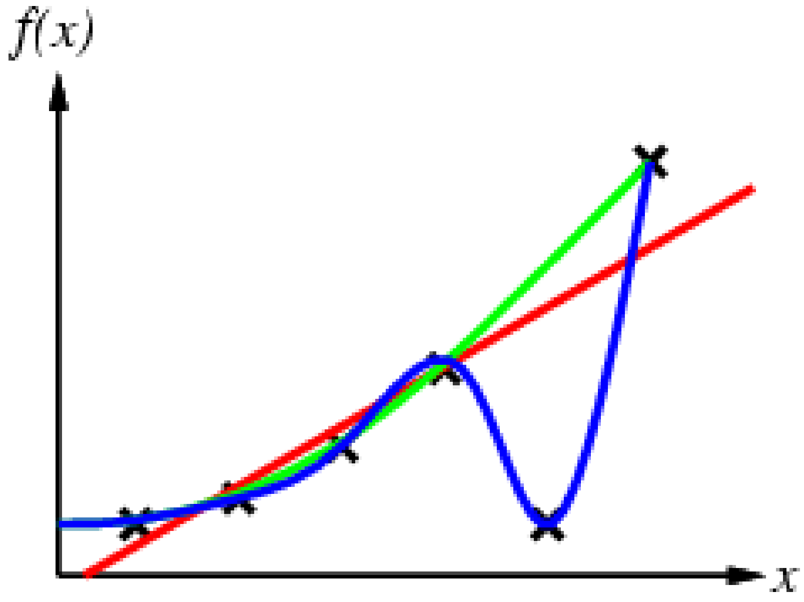
Curve Fitting



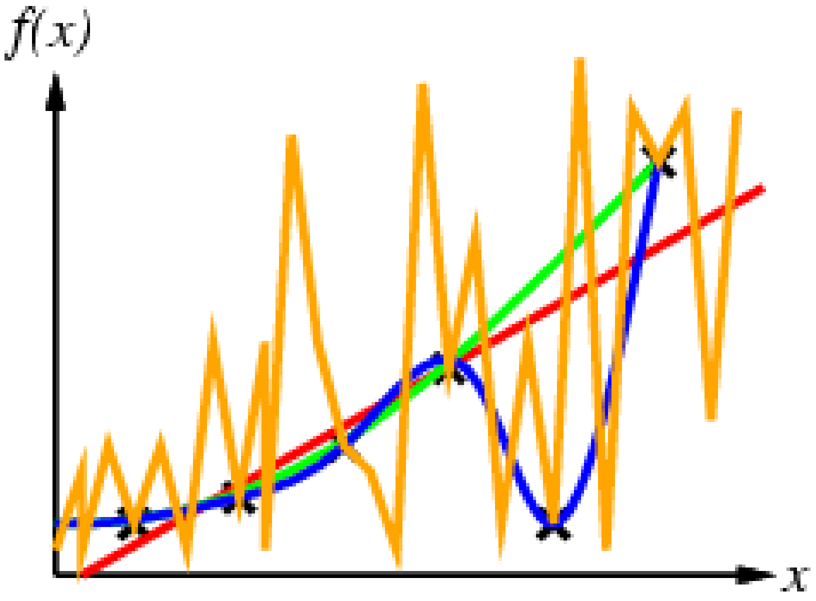
Curve Fitting



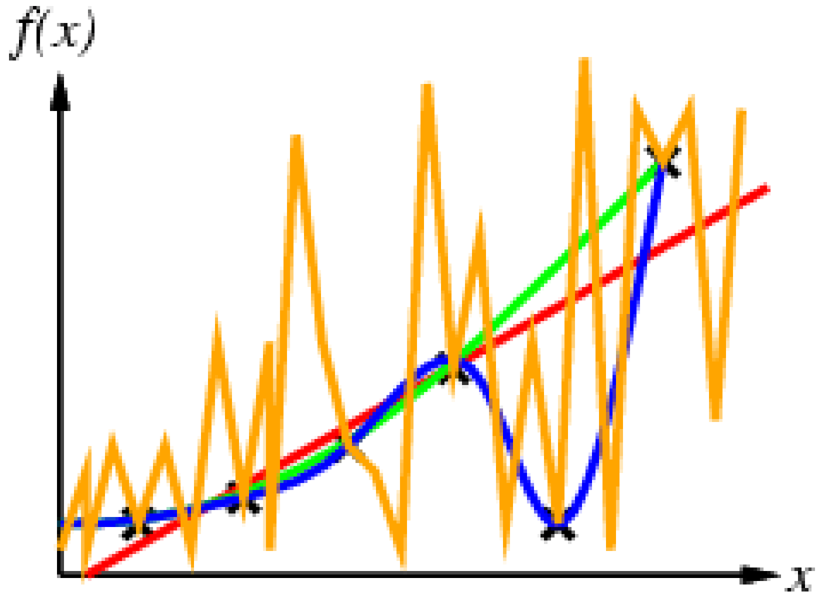
Curve Fitting



Curve Fitting



Curve Fitting



So, which function is the "right" one?

Some Classification Models

1. Naïve Bayes
2. Logistic regression

   Note: it's called regression, but it is a classification model
3. Decision trees
4. Support vector machines
5. Neural networks

Naïve Bayes

Task

- Classify emails as spam (s) / not-spam (~s) based on the words they contain
- You look at 100 random emails; 40 of them are spam, 60 of them are not-spam
- What is P(s) for a new email?

Features

- Assume you'll look into the emails' contents; you've decided that the word Nigeria[1] seems to correlate well with spam. You group the 100 emails as follows

Nigeria	Spam	Count
t	s	30
f	s	10
t	~s	10
f	~s	50

If the word Nigeria appears in the new email, then what is P(s | Nigeria=t)?

1. Why "Nigeria?" https://www.google.com/search?q=nigeria+scam+emails

Nigeria=t

Nigeria	Spam	Count
t	s	30
f	s	10
t	~s	10
f	~s	50

If the word Nigeria appears in the new email, then what is P(s | Nigeria=t)?

$$P(s \mid N=t) = \frac{P(s, N=t)}{P(N=t)} = \frac{30/100}{(30+10)/100} = \frac{30}{40}$$

Add Admission into your vocabulary

Nigeria	Adm.	Spam	Count
t	t	s	10
t	f	s	20
f	t	s	3
f	f	s	7
t	t	~s	8
t	f	~s	2
f	t	~s	40
f	f	~s	10

What is P(s | N=t, A=f)? What about P(s | N=t, A=t)?

ADD ADMISSION INTO YOUR VOCABULARY

Nigeria	Adm.	Spam	Count
t	t	s	10
t	f	s	20
f	t	s	3
f	f	s	7
t	t	~s	8
t	f	~s	2
f	t	~s	40
f	f	~s	10

What is P(s | N=t, A=f)? What about P(s | N=t, A=t)?

$$P(s \mid N=t, A=f) = \frac{P(s, N=t, A=f)}{P(N=t, A=f)} = \frac{20/100}{(20+2)/100} = \frac{20}{22}$$

P(s | N=t) was 0.75. P(s | N=t, A=f) is 0.91

ADD ADMISSION INTO YOUR VOCABULARY

Nigeria	Adm.	Spam	Count
t	t	s	10
t	f	s	20
f	t	s	3
f	f	s	7
t	t	~s	8
t	f	~s	2
f	t	~s	40
f	f	~s	10

What is P(s | N=t, A=f)? What about P(s | N=t, A=t)?

$$P(s \mid N=t, A=t) = \frac{P(s, N=t, A=f)}{P(N=t, A=f)} = \frac{10/100}{(10+8)/100} = \frac{10}{18}$$

P(s | N=t) was 0.75. P(s | N=t, A=f) is 0.91. P(s | N=t, A=t) = 0.56.

Now assume we add 998 more words

W_1	W_2	…	W_{1000}	Spam	Count
t	t	…	t	s	
t	t	…	f	s	
…	…	…	…	…	
f	f	…	f	~s	

Q: How many entries are there in this table?

A: $2^{1001} \approx 2 \times 10^{301}$

We have 100 emails. If all emails are distinct, 100 entries will be 1;
The rest will be 0.

Q: What is $P(s \mid W_1=t, W_2=f, \ldots, W_{1000}=t)$?

A: Either 1 or 0 if it is in D, otherwise, it is NaN

Q: How big of a training data do we need?

NAÏVE BAYES

- Given $X_1, X_2, \ldots, X_n$, and class Y
- Assume $X_i \perp X_j \mid Y$

Bayes

naive

$$P(Y|X_1, X_2, \ldots, X_n) = \frac{P(X_1, X_2, \ldots, X_n|Y)P(Y)}{P(X_1, X_2, \ldots, X_n)} = \frac{P(Y)\prod_{i=1}^{n} P(X_i|Y)}{P(X_1, X_2, \ldots, X_n)}$$

We need to estimate $\underline{P(Y)}$ and $\underline{P(X_i \mid Y)}$

What is the Bayesian network representation of Naïve Bayes?

NAÏVE BAYES

Nigeria	Adm.	Spam	Count
t	t	s	10
t	f	s	20
f	t	s	3
f	f	s	7
t	t	~s	8
t	f	~s	2
f	t	~s	40
f	f	~s	10

What is P(S)?

What is P(N|S)?

What is P(A|S)?

Naïve Bayes

Nigeria	Adm.	Spam	Count
t	t	s	10
t	f	s	20
f	t	s	3
f	f	s	7
t	t	~s	8
t	f	~s	2
f	t	~s	40
f	f	~s	10

What is P(S)?

Spam	P(S)
s	40/100
~s	60/100

What is P(N|S)?

Nigeria	Spam	P(N,S)	P(N|S)
t	s	30/100	30/40
f	s	10/100	10/40
t	~s	10/100	10/60
f	~s	50/100	50/60

What is P(A|S)?

Adm.	Spam	P(A,S)	P(A|S)
t	s	13/100	13/40
f	s	27/100	27/40
t	~s	48/100	48/60
f	~s	12/100	12/60

$P(\sim s \mid t,f) = 8/89$ $\qquad \frac{81}{400} \Big/ \frac{89}{400} = 81/89$

INFERENCE IN NAÏVE BAYES

- What is P(s | N=t, A=f)? $\propto P(s)\,P(N{=}t \mid s)\,P(A{=}f \mid s)$

$$\frac{\cancel{40}}{100} \times \frac{3\cancel{0}}{\cancel{40}} \times \frac{27}{4\cancel{0}} = \frac{81}{400}$$

$$P(\sim s \mid N{=}t, A{=}f) \propto P(\sim s)\,P(N{=}t \mid \sim s)\,P(A{=}f \mid \sim s)$$

$$\frac{\cancel{60}}{100} \times \frac{1\cancel{0}}{\cancel{60}} \times \frac{\cancel{12}^{\,2}}{\cancel{6}\cancel{0}} = \frac{2}{100} = \frac{8}{400}$$

$$P(N{=}t, A{=}f) = \frac{81}{400} + \frac{8}{400} = \frac{89}{400}$$

Zero probabilities

- We have n features, X_1 through X_n
- If $P(X_i|C)$ is zero for any feature and class combination, we would be in trouble
- Example
  - Assume that X_{592} is a weird feature that is rarely *true* in the world. Assume that X_{592} is always *false* in our training data, no matter what the class is
    - $P(X_{592} = f \mid C = t) = 1$; $P(X_{592} = t \mid C = t) = 0$
    - $P(X_{592} = f \mid C = f) = 1$; $P(X_{592} = t \mid C = f) = 0$
  - In one of the objects in our test data, X_{592} is *true*.
    - What is $P(C \mid X_1, X_2, \ldots, X_{592} = t, \ldots X_n)$?

Other Classifiers - Overview

Some Classification Models

1. Naïve Bayes
2. Logistic regression

   Note: it's called regression, but it is a classification model
3. Decision trees
4. Support vector machines
5. Neural networks

Logistic Regression

- Learns $P(Y|\boldsymbol{X})$ directly, without going through $P(\boldsymbol{X}|Y)$ and $P(Y)$
- Assumes $P(Y|\boldsymbol{X})$ follows the logistic function

$$P(Y = false \mid X_1, X_2, \cdots, X_n) = \frac{1}{1 + e^{w_0 + \sum_{i=1}^{n} w_i X_i}}$$

$$P(Y = true \mid X_1, X_2, \cdots, X_n) = \frac{e^{w_0 + \sum_{i=1}^{n} w_i X_i}}{1 + e^{w_0 + \sum_{i=1}^{n} w_i X_i}}$$

- Learning: estimate the weights $w_0, w_1, \ldots, w_n$

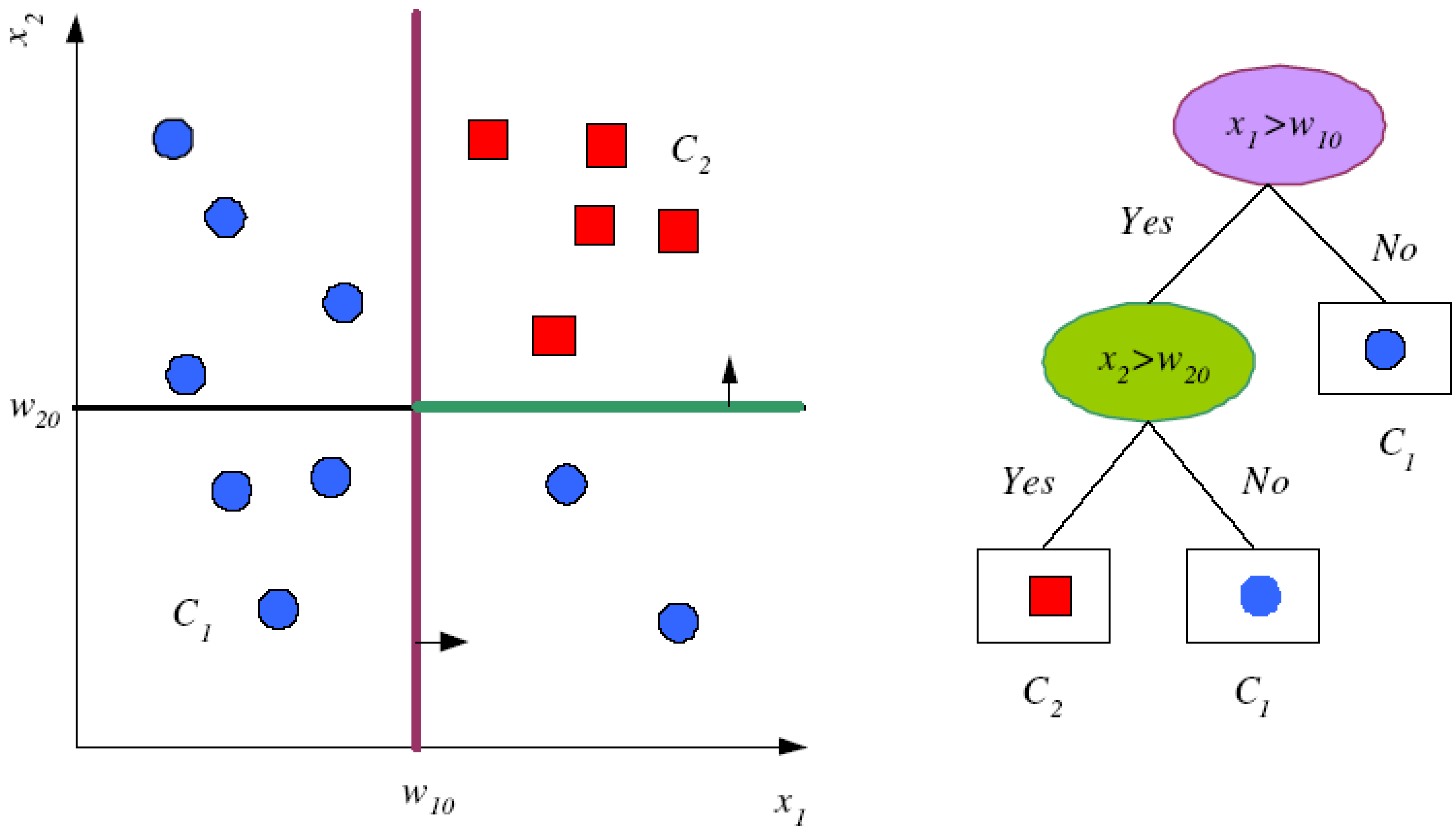
Learning – Parameter Estimation

- Maximize (conditional) log-likelihood

$$W \leftarrow \underset{W}{\operatorname{argmax}} \prod P(Y^{(d)} | \boldsymbol{X}^{(d)})$$

$$W \leftarrow \underset{W}{\operatorname{argmax}} \sum \ln P(Y^{(d)} | \boldsymbol{X}^{(d)})$$

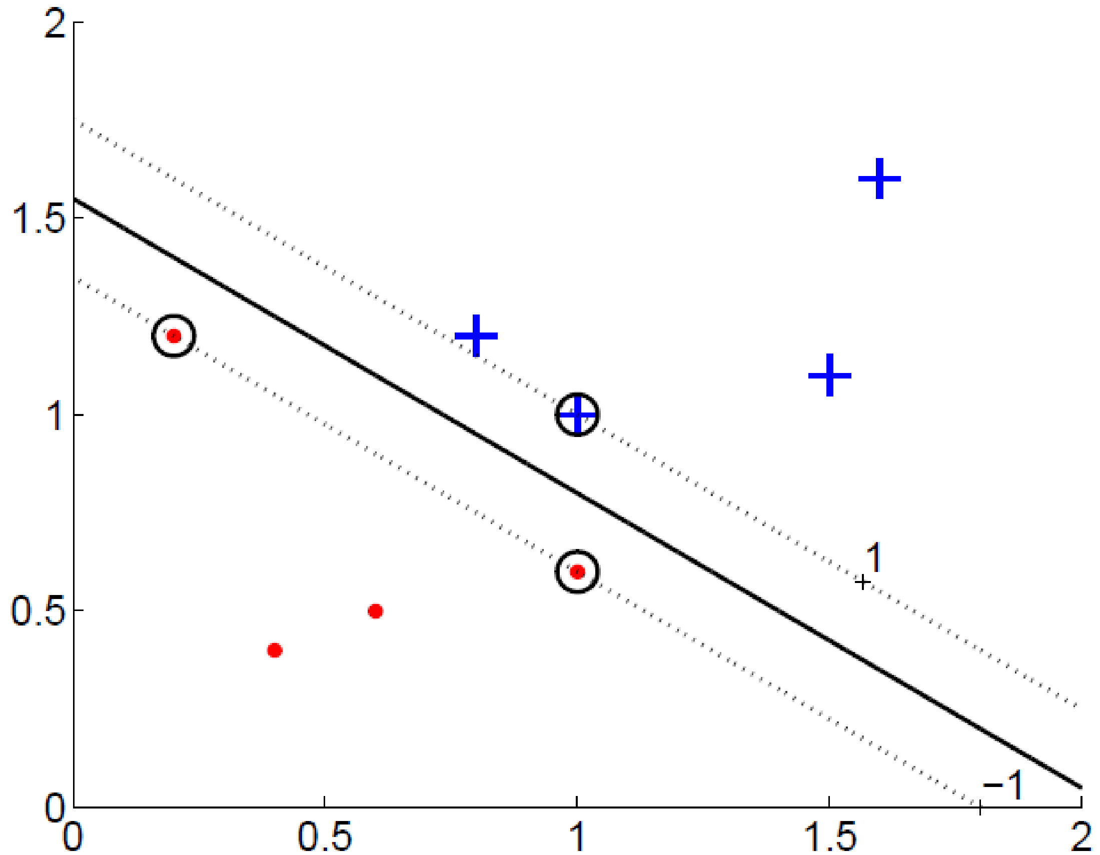
Decision Trees



Learning: how do you learn a small tree that generalizes to unseen data?

Image credit: Ethem Alpaydin. Introduction to Machine Learning. 3rd Edition. http://www.cmpe.boun.edu.tr/~ethem/i2ml3e

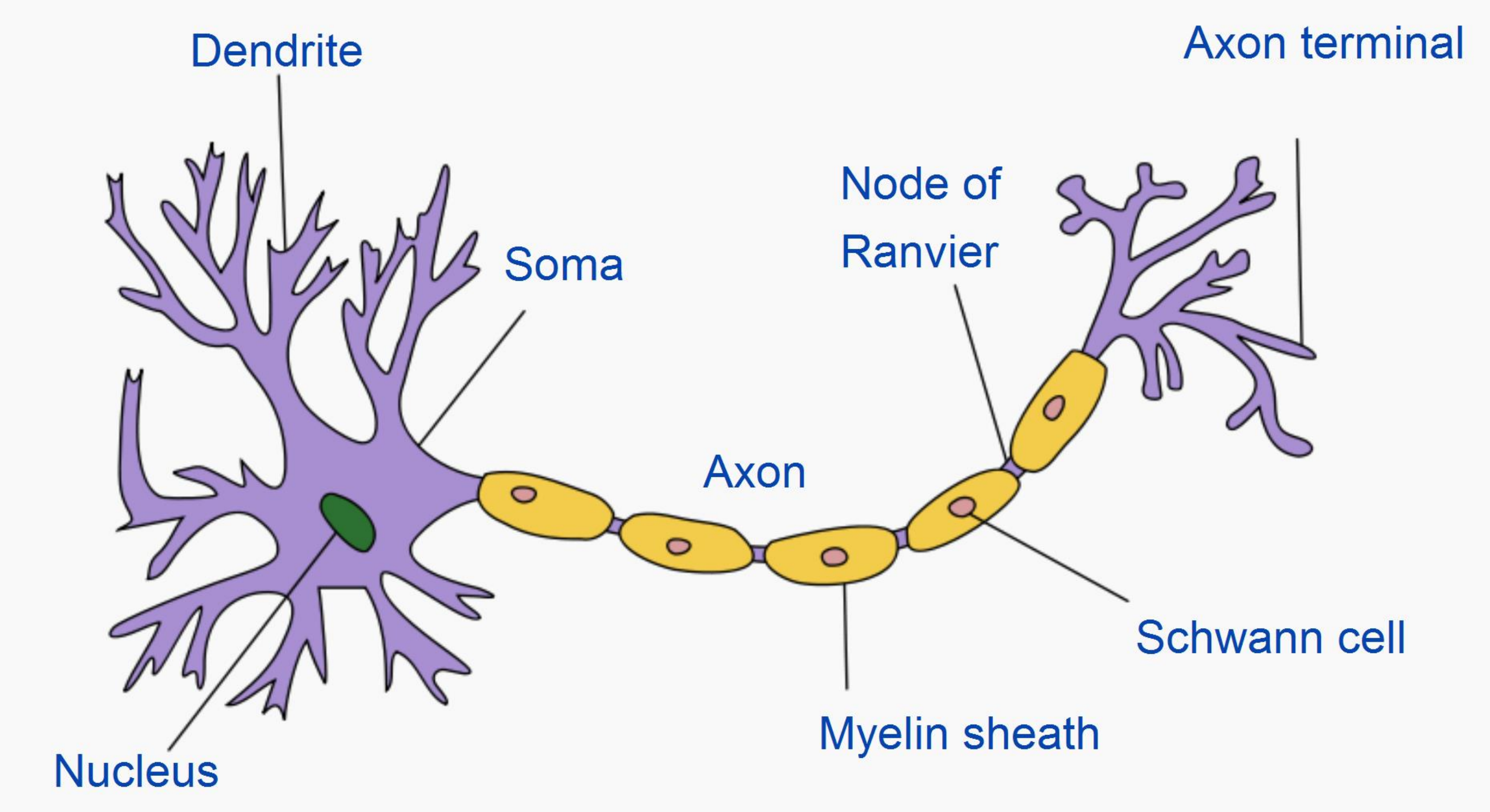
Support Vector Machines



$$\min \frac{1}{2} w^T w \text{ subject to } y^{(d)}\left(w^T x^{(d)} + b\right) \geq +1$$

Image credit: Ethem Alpaydin. Introduction to Machine Learning. 3rd Edition. http://www.cmpe.boun.edu.tr/~ethem/i2ml3e

NEURON



By Quasar Jarosz at English Wikipedia, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=7616130

What an Artificial Neuron Does

- Takes a weighted sum of its inputs
  - $w_0 + \sum_{i=1}^{k} w_i x_i$
  - Assume that there is always a constant input 1, that is, $x_0 = 1$. Then,
  - $\sum_{i=0}^{k} w_i x_i$
- Passes this sum through its activation function
  - $f\left(\sum_{i=0}^{k} w_i x_i\right)$

Multilayer Neural Networks

- An input layer
- One or more hidden layers
- An output layer

- Learning: estimate the weights

Two Common NN Types

- Feedforward NNs
  - E.g., Convolutional neural networks (CNNs)
  - E.g., image data
- Recurrent neural networks
  - E.g., Long Short-Term Memory (LSTM)
  - E.g., text data

Scikit-learn Code Examples

- https://scikit-learn.org/stable/
- Naïve Bayes
  - https://scikit-learn.org/stable/modules/naive_bayes.html
- Logistic regression
  - https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
- Decision Trees
  - https://scikit-learn.org/stable/modules/tree.html
- Support vector machines
  - https://scikit-learn.org/stable/modules/svm.html
- Neural networks
  - https://scikit-learn.org/stable/modules/neural_networks_supervised.html

Reinforcement Learning

Reinforcement Learning

- Let's first refresh our memory with Chapter 17 – sequential decision making
- In Ch17, we assumed we knew the transition model $P(s' | s, a)$ and the reward function $R(s)$
- Reinforcement learning makes no such assumption

Passive Reinforcement Learning

- The agent has a fixed policy π
- The goal is to learn how good policy π is
  - That is, compute $U^{\pi}(s)$ for each state s
- This was straightforward in chapter 17, because we assumed we knew $P(s' \mid s, a)$ and $R(s)$
  - Here, RL makes no such assumptions
- An example approach
  - Carry out experiments / trials, following the policy π
  - Use **temporal-difference learning** algorithm to compute the utilities

Active Reinforcement Learning

- In passive RL, we evaluate a fixed policy π
- In active RL, no such policy is given
- The agent needs to learn an optimal policy
  - Again, $P(s'|\, s, a)$ and $R(s)$ are not given
- **Exploration versus exploitation trade-off**
  - The agent needs to explore various actions, even if they are suboptimal
  - The agent needs to exploit what it knows and choose what it thinks is the optimal action
- A typical example: multi-armed bandit problems

Deep Reinforcement Learning

- Combine the power of DL and RL
- Example
  - AlphaGo
- DeepMind Blog on deep reinforcement learning
  - https://deepmind.com/blog/article/deep-reinforcement-learning

WE'~~LL~~ COVER*ED*

1. Bayesian network parameter estimation
2. Supervised learning
3. Reinforcement learning