Machine Learning

Introduction (cont')

Dan Goldwasser

dgoldwas@purdue.edu

Goal for Today's class

Finish introduction to Classification

and

<u>My first learning algorithm</u>
<u>(sort-of *)</u>

<u>KNN</u>

* Reason for disclaimer will be clear by the end of this lecture!

Review: *10,000 feet view*

- *We will look into several learning **protocols**, using different learning **algorithms**, for learning different **models**.*
  - **Model**: function mapping inputs to outputs
    - Instance space
    - Hypothesis space
  - **Algorithm**: used for constructing a model, based on data
  - **Protocol**: the settings in which the algorithm learns.
    - Supervised/Unsupervised/..
- **Learning**: searching through the hypothesis space using data

Review: Learning Model

- We think about learning as producing a function mapping input to outputs, based on data
  - E.g., spam: email $\rightarrow$ Boolean
- Model is the type of function the learner looks at
  - Linear Classifier
  - Decision Trees
  - ..
- All of these models can be characterized and discussed formally in terms of their expressivity.

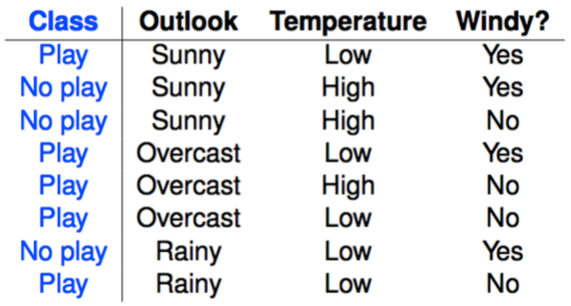
Review: Learning Protocols

- Supervised learning
    - Human (teacher) supplies a labeled examples
    - Learner has to learn a model using this data
- Unsupervised learning
    - No teacher, learner has only unlabeled examples
    - Data mining: finding patterns in unlabeled data
- Semi-supervised learning
    - Learner has access to both labeled and unlabeled examples
- Active learning
    - Learner and teacher interact
    - Learner can ask questions
- Reinforcement learning
    - Learner learns by interacting with the environment
    - Why is it different/similar to Supervised learning? Active learning?

Review: Learning Algorithms

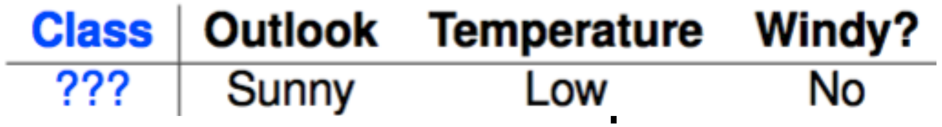
- Learning Algorithms generate a model, they work under the settings of a specific protocol
- Supervised vs. Semi-Supervised vs. Unsupervised
- Online vs. Batch
  - Online algorithm: learning is done one example at a time
    - Winnow, Perceptron,..
  - Batch algorithm: learning is done over entire dataset
    - SVM, Logistic Regression, Decision Trees, …
- How can we compare learning algorithms?

Review: Classification Dataset and Notation



Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

- A labeled dataset is a collection of (x,y) pairs
- Task: use dataset to predict new instance class
- Generalizatio



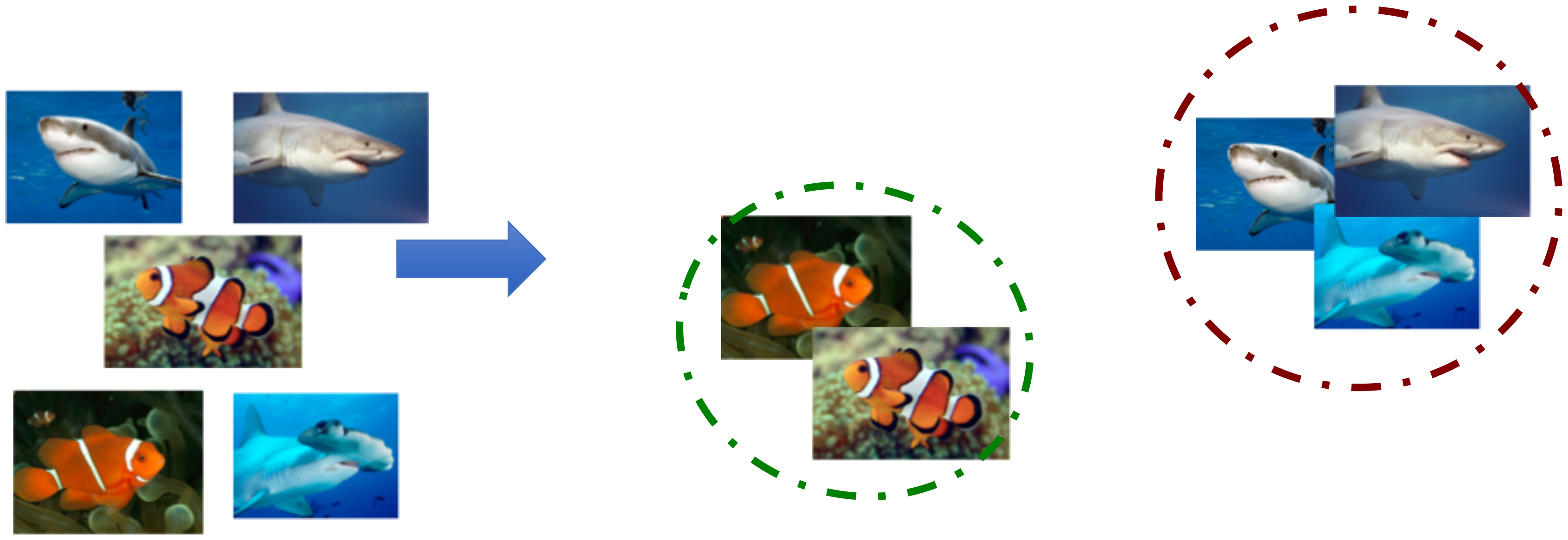
Class	Outlook	Temperature	Windy?
???	Sunny	Low	No

Review: Machine Learning Tasks

- Supervised Classification
  - Most popular scenario
  - Most of this class is about classification
- Regression
- Clustering (unsupervised)
- Structured Prediction
- Reinforcement learning

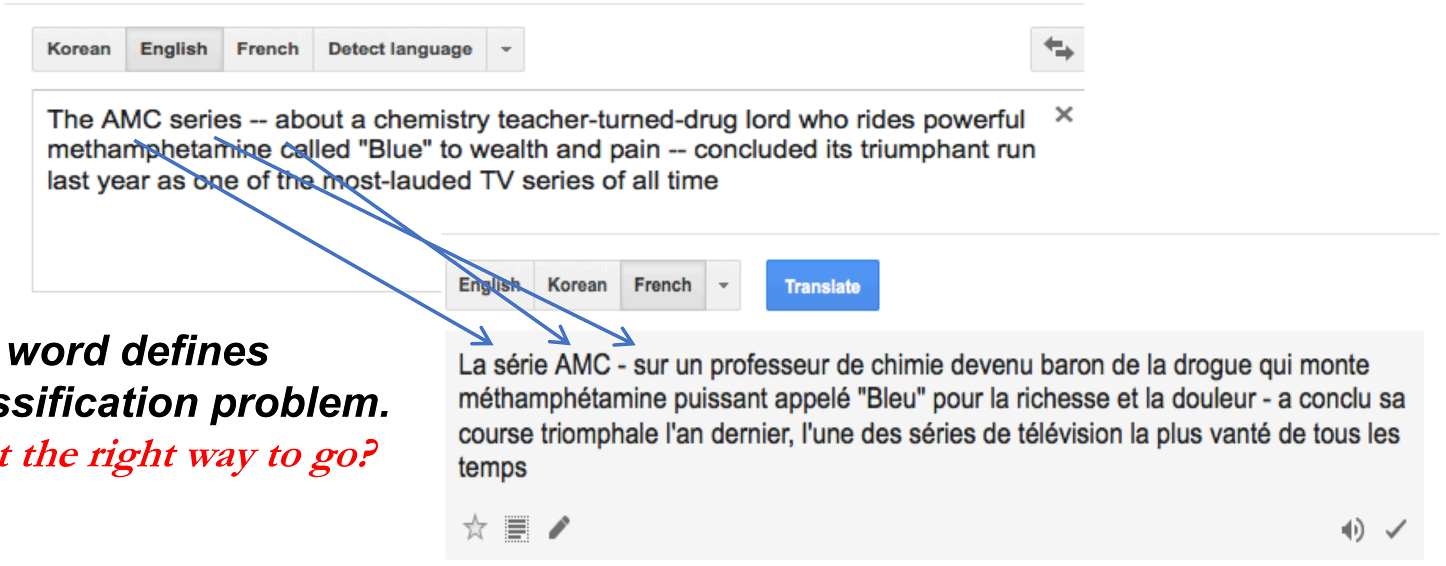
Clustering

- Clustering: Group similar instances.
  - Define a similarity metric
  - Questions:
    - How can you tell how many groups are in the data?
    - What is the ”meaning” of the formed clusters?



Structured Prediction

- Mapping Between **complex objects (structures)**
  - Translation, Protein sequence
- Reduction to classification
  - Multiple **interdependent** decisions
    - We need to model the interactions between decisions!

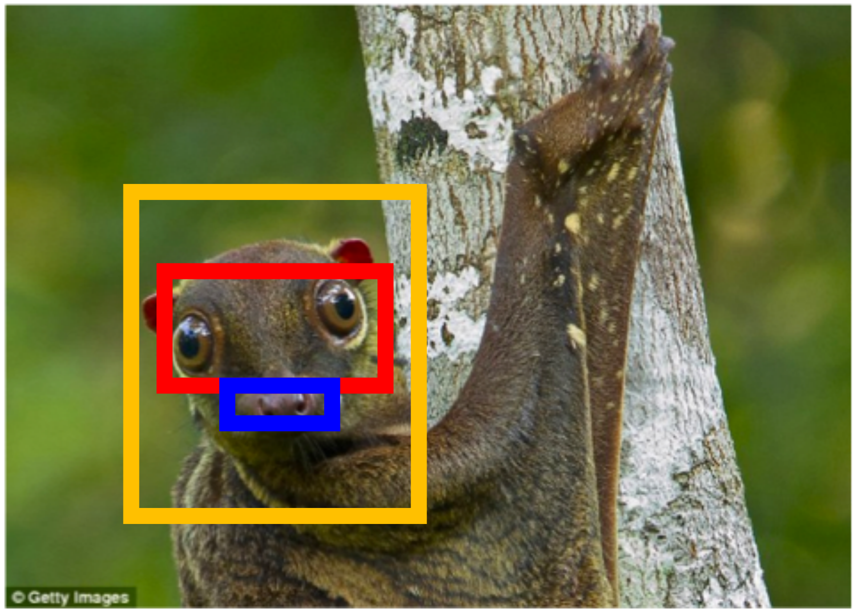


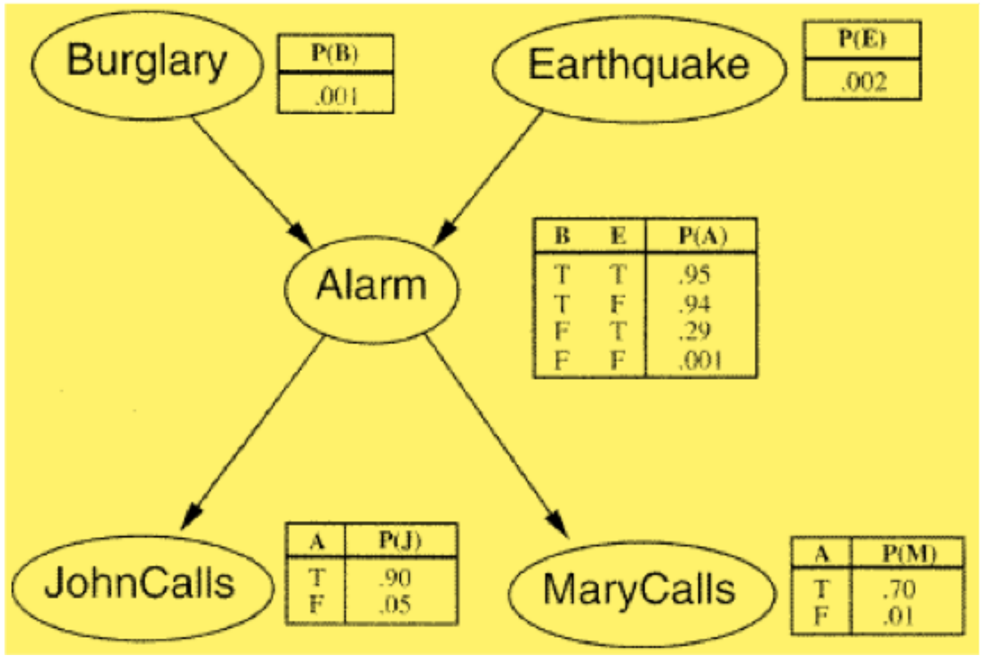
The AMC series -- about a chemistry teacher-turned-drug lord who rides powerful methamphetamine called "Blue" to wealth and pain -- concluded its triumphant run last year as one of the most-lauded TV series of all time

La série AMC - sur un professeur de chimie devenu baron de la drogue qui monte méthamphétamine puissant appelé "Bleu" pour la richesse et la douleur - a conclu sa course triomphale l'an dernier, l'une des séries de télévision la plus vanté de tous les temps

Each word defines a classification problem.
Is that the right way to go?

Structured Prediction





P(B)
.001

P(E)
.002

B	E	P(A)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J)
T	.90
F	.05

A	P(M)
T	.70
F	.01

Our focus: Supervised Learning

- Recall the "**Badge Game**"
  - Each member gets a badge with a +/- label and a name
- **Assumptions:** *the label is a function of the name*
  - Strong assumption! *Meaning that future data will behave in the same way!*
- **Our goal**: approximate the true function using the available data.
  - Learning is about "removing uncertainty" about what is a good approximation.
- ***Let's formalize these concepts!***

Our Focus: **Supervised** Learning

Input

$x \in \mathcal{X}$

An instance **x** drawn from an **instance space**

$y = f(x)$

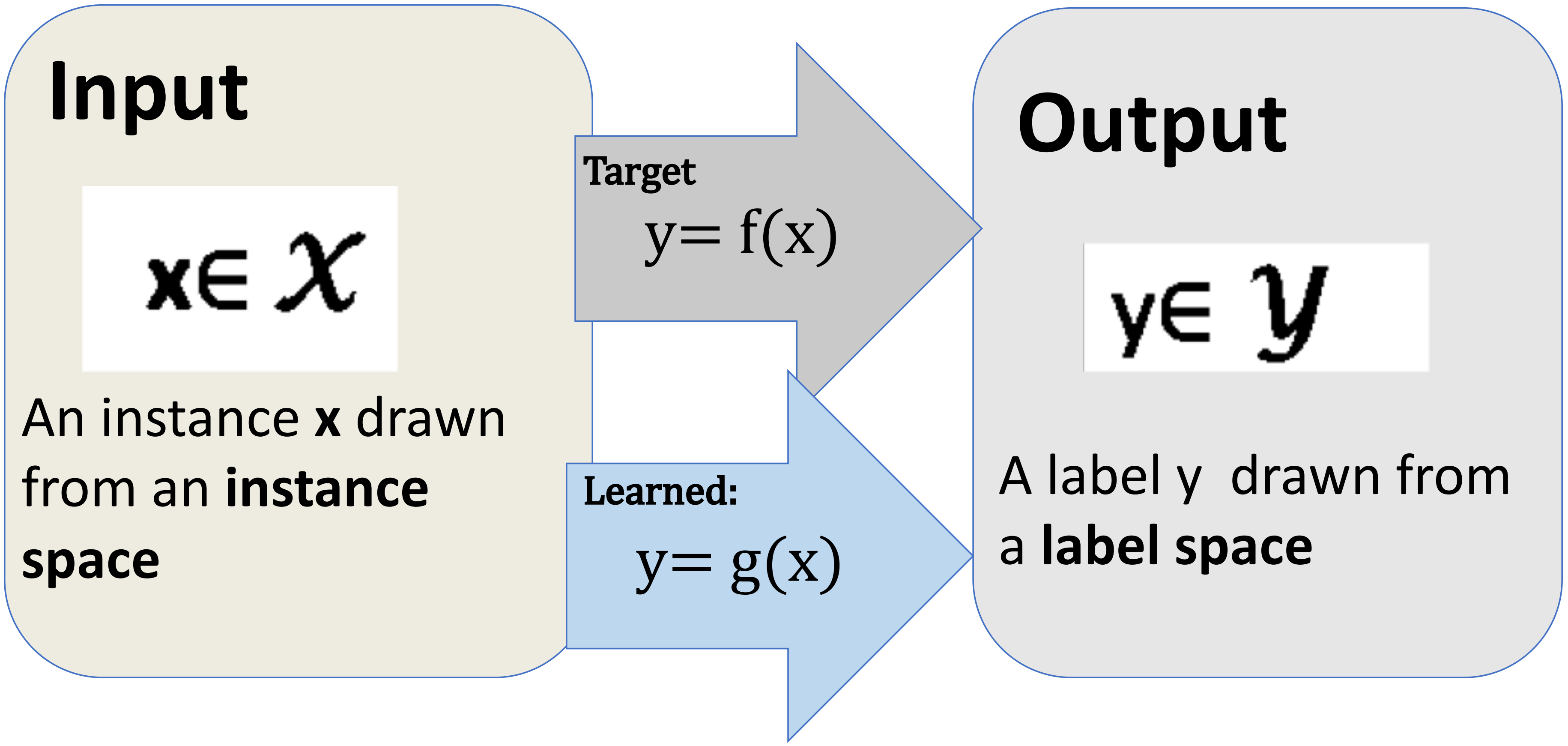
Output

$y \in \mathcal{Y}$

A label y drawn from a **label space**

- In the supervised settings we want to find f(x) based on examples

Supervised Learning

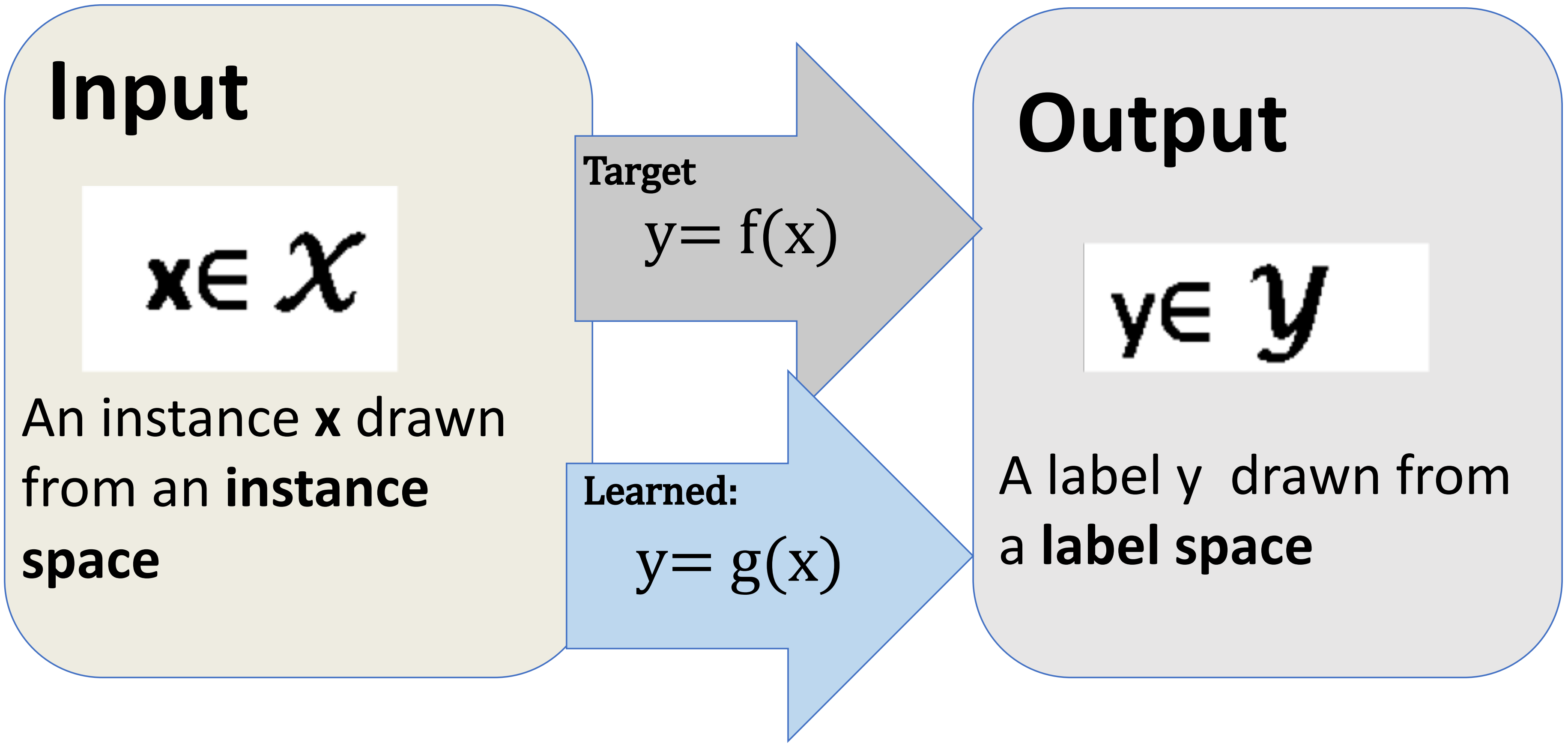


An instance **x** drawn from an **instance space**

A label y drawn from a **label space**

- The learning algorithm sees a small subset of the instance space, with their labels

Supervised Learning



Input

x∈ $\mathcal{X}$

An instance **x** drawn from an **instance space**

Target

y= f(x)

Learned:

y= g(x)

Output

y∈ $\mathcal{Y}$

A label y drawn from a **label space**

- The learned function may not be identical to the target function: for a given x, f(x) $\overset{?}{=}$ g(x)

Supervised Learning Settings

- **Input**: a set of examples <**x**,f(**x**)>, where f(**x**) is the unknown target function
  - f(x) can be binary, categorical, or continuous
- The input **x** is represented in a ***feature space***
  - *Typically* $\boldsymbol{x} \in \{0,1\}^n$,or $\boldsymbol{x} \in \{0,1,..,K\}^n$,or $\boldsymbol{x} \in R^n$
- **Learning**: find g(x) that approximates f(x) well.
  - The space from which g(x) is chosen is called ***hypothesis space***
  - ***Learning*** *is* ***searching*** *this space for a "good" g(x)*
- ***Question***: *does the hypothesis space of f(x) and g(x) has to be similar?*

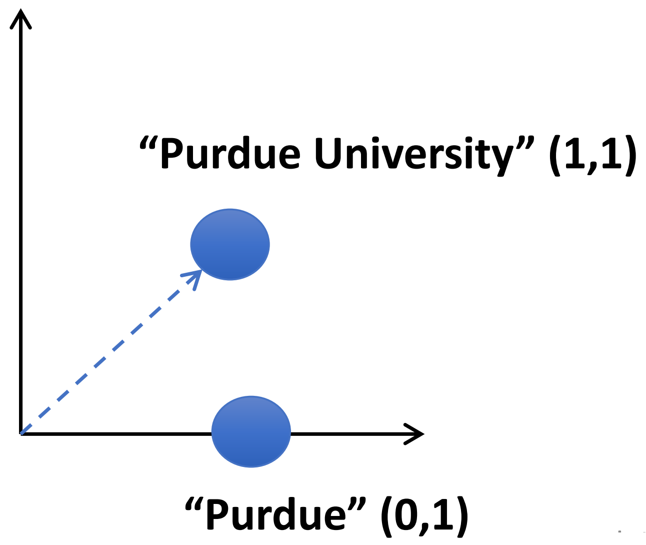
The Instance Space

- ***How you choose to represent examples matters!***
  - Good representations should capture aspect of the input object relevant for classification
- ***What is a good representation for the badge game?***
  - Boolean features: "2nd letter is a vowel"
  - Numerical: "how many vowels?"

- This is a domain specific process
  - **Know your problem domain!**

The Instance Space

- Feature functions map inputs to Boolean/numeric values
- The input objects are represented using a feature vector
  - X is a N-dimensional vector space
  - Each dimension is one feature
  - Each instance **x** is a point in the vector space



It's convenient to think about templates:

"what's the 2nd character in the name?"

John $\rightarrow$ <0,1,0,… >
James $\rightarrow$ <1,0,0,… >
Claire $\rightarrow$ <0,0,1,… >

The Hypothesis Space

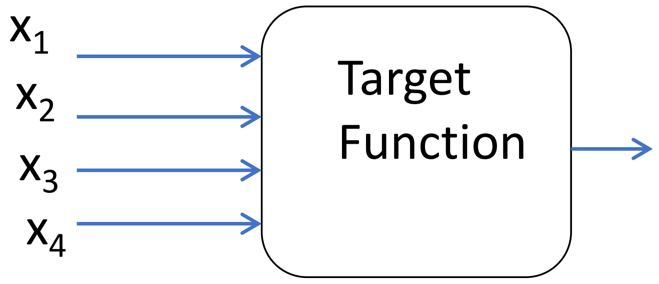
- What functions should the learning algorithm consider?
    - **Does not** have to search over the same space as f(x)
    - Should it be a **more** expressive space? **Less** expressive?

- **Does it matter?**

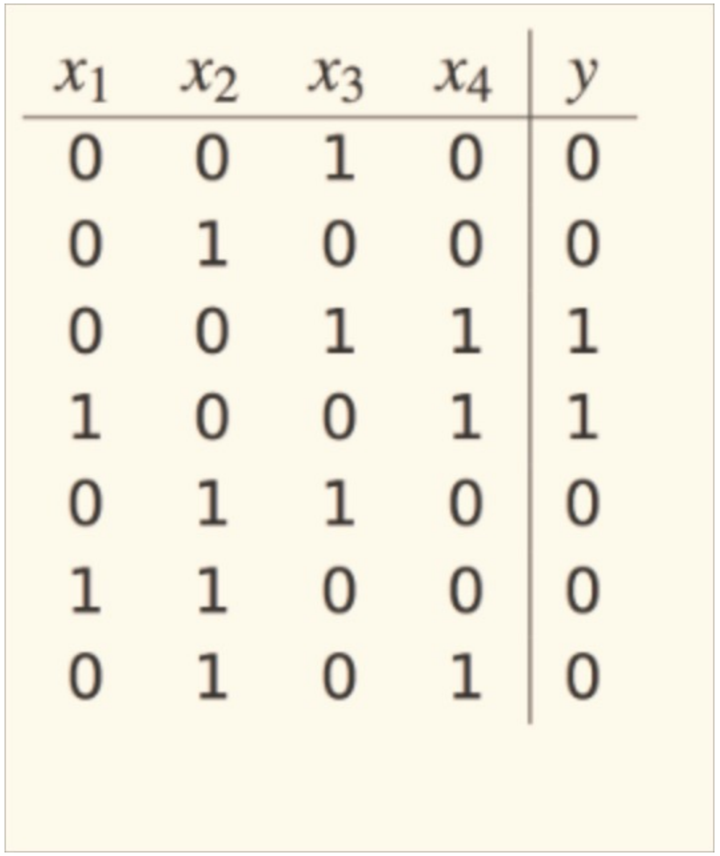
The Hypothesis Space

We want to find the target functions based on the training examples

Can you find the target function?



$y=f(x_1,x_2,x_3,x_4)$



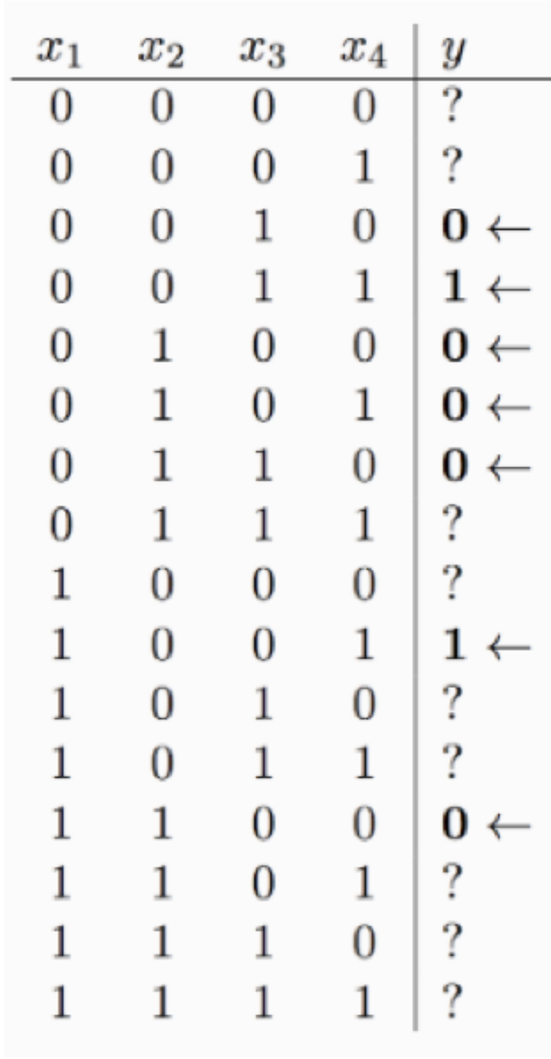
x_1	x_2	x_3	x_4	y
0	0	1	0	0
0	1	0	0	0
0	0	1	1	1
1	0	0	1	1
0	1	1	0	0
1	1	0	0	0
0	1	0	1	0

Is learning possible?

- How many Boolean functions are there over 4 inputs?
- 2^{16} = 65536 functions (**why?**)
  - 16 possible outputs.
  - Two possibilities for each output
- Without any data, 2^{16} options

- ***Does the data identify the right function?***
- The training data contains 7 examples
  - We still have 2^9 options,

Is learning even possible?



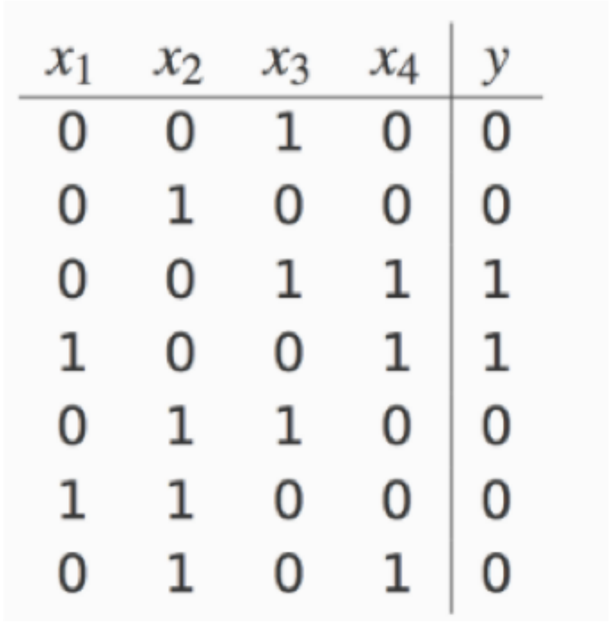
x_1	x_2	x_3	x_4	y
0	0	0	0	?
0	0	0	1	?
0	0	1	0	**0** ←
0	0	1	1	**1** ←
0	1	0	0	**0** ←
0	1	0	1	**0** ←
0	1	1	0	**0** ←
0	1	1	1	?
1	0	0	0	?
1	0	0	1	**1** ←
1	0	1	0	?
1	0	1	1	?
1	1	0	0	**0** ←
1	1	0	1	?
1	1	1	0	?
1	1	1	1	?

Hypothesis Space

- A ***hypothesis space*** is the set of possible functions we consider
  - **The Problem**: *we were looking at the space of **all** Boolean functions*
  - **The Solution**: *Instead choose a hypothesis space that is smaller than the space of all Boolean functions:*
    - *Only simple conjunctions* (with four variables, there are only 16 conjunctions without negations)
    - *Simple disjunctions*
    - *m-of-n rules*: Fix a set of n variables. At least m of them must be true
    - *Linear functions*

Let's try a different Hypothesis space!

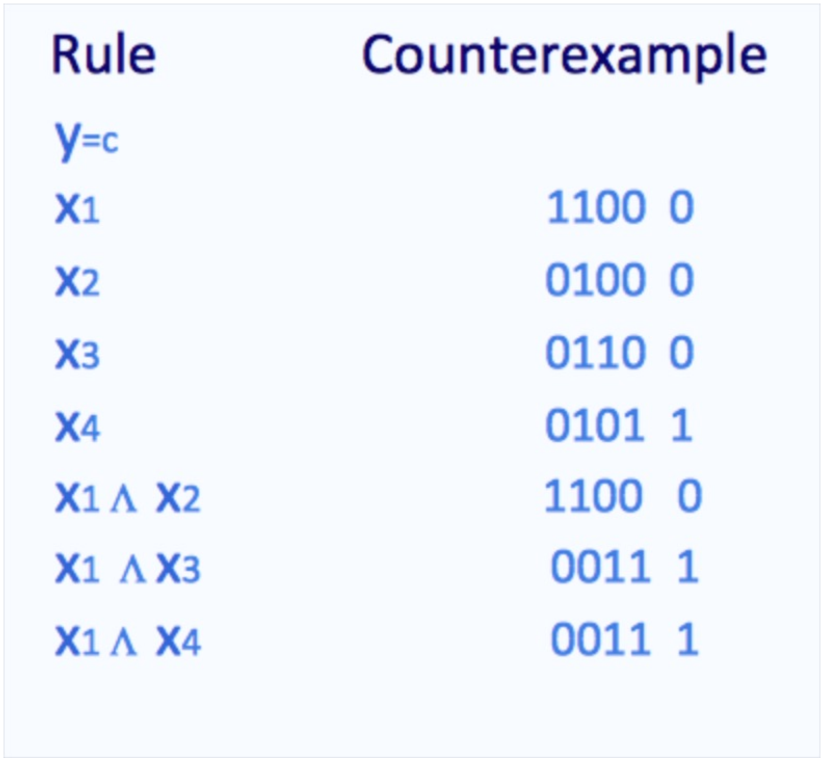
- Simple Conjunctions: *very small subset of Boolean functions*
  - Only 16 possible conjunction of the form: $y=x_i \wedge \ldots x_j$
    - **Why?**
- ***Can you find a consistent Hypothesis in this space?***



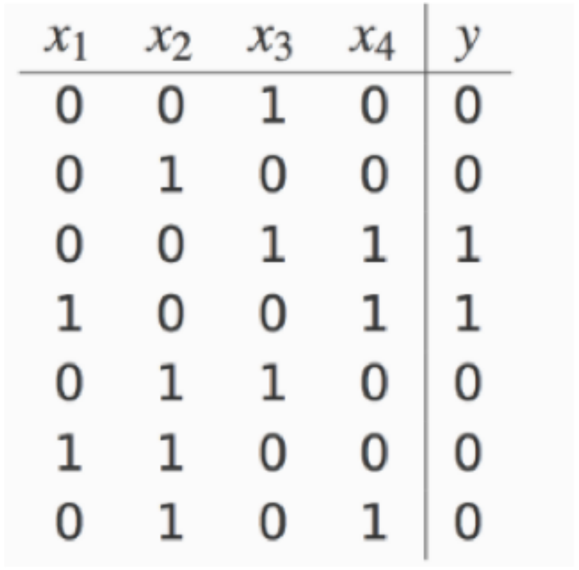
x_1	x_2	x_3	x_4	y
0	0	1	0	0
0	1	0	0	0
0	0	1	1	1
1	0	0	1	1
0	1	1	0	0
1	1	0	0	0
0	1	0	1	0

Simple Conjunctions

- Simple Conjunctions



Rule	Counterexample
$y=c$	
x_1	1100 0
x_2	0100 0
x_3	0110 0
x_4	0101 1
$x_1 \land x_2$	1100 0
$x_1 \land x_3$	0011 1
$x_1 \land x_4$	0011 1

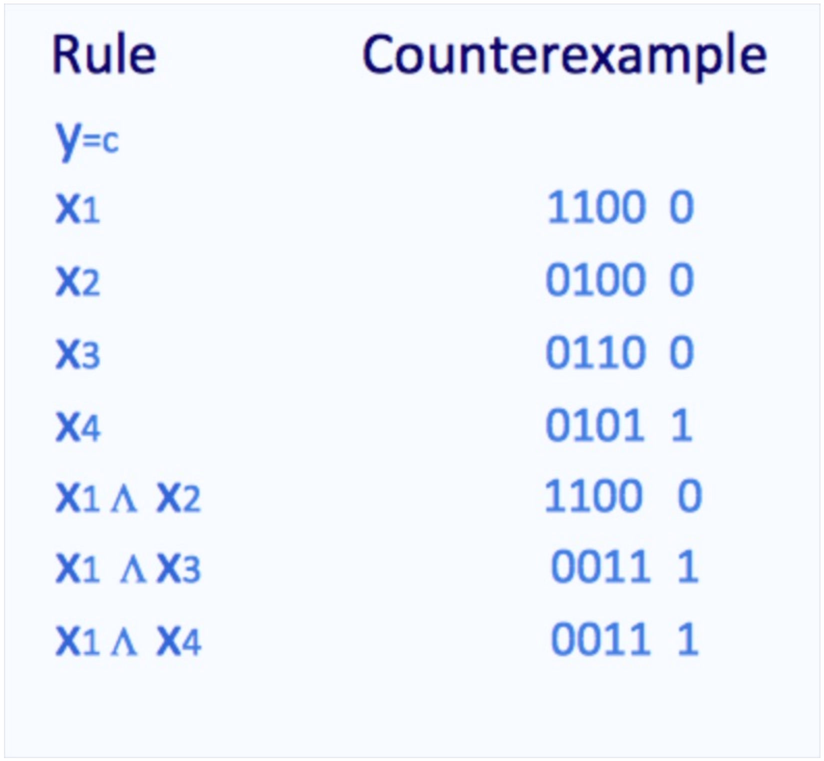


x_1	x_2	x_3	x_4	y
0	0	1	0	0
0	1	0	0	0
0	0	1	1	1
1	0	0	1	1
0	1	1	0	0
1	1	0	0	0
0	1	0	1	0

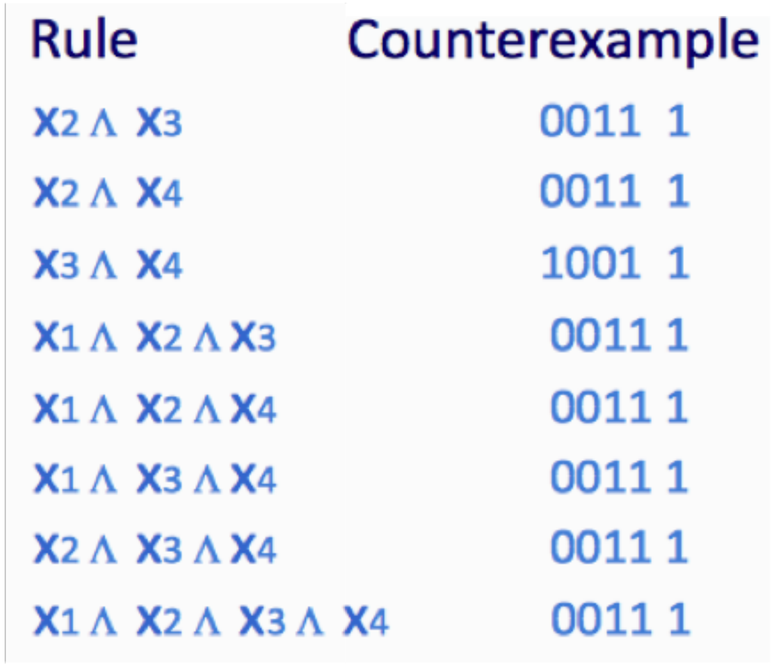
.. let's keep going..

Simple Conjunctions

- Simple Conjunctions



Rule	Counterexample
$y=c$	
x_1	1100 0
x_2	0100 0
x_3	0110 0
x_4	0101 1
$x_1 \land x_2$	1100 0
$x_1 \land x_3$	0011 1
$x_1 \land x_4$	0011 1



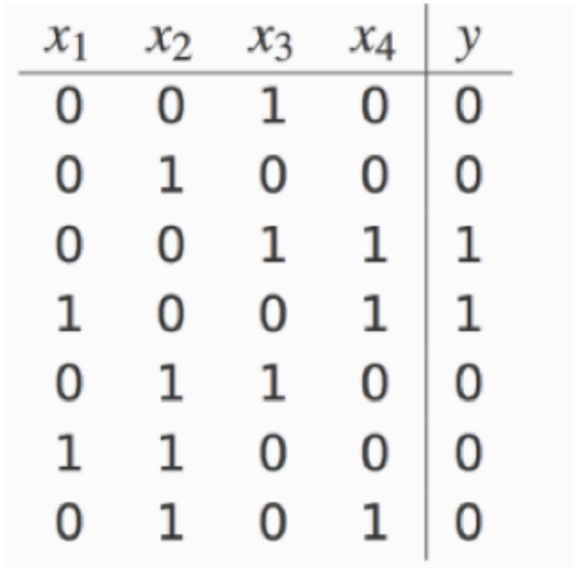
Rule	Counterexample
$x_2 \land x_3$	0011 1
$x_2 \land x_4$	0011 1
$x_3 \land x_4$	1001 1
$x_1 \land x_2 \land x_3$	0011 1
$x_1 \land x_2 \land x_4$	0011 1
$x_1 \land x_3 \land x_4$	0011 1
$x_2 \land x_3 \land x_4$	0011 1
$x_1 \land x_2 \land x_3 \land x_4$	0011 1

No simple conjunction can explain this data!

Let's try a different Hypothesis space!

- **The class of simple conjunctions is not expressive enough for our functions**
- **How can we pick a better space?**
  - Prior knowledge about the problem
  - Sufficiently "flexible"
- Let's try another space – ***m-of-n rules***
  - Rules of the form: *"y = 1 if and only if at least m of the following n variables are 1"*
    - How many are there for 4 Boolean variables?
    - **Is there a consistent hypothesis?**

M-of-N rules



x_1	x_2	x_3	x_4	y
0	0	1	0	0
0	1	0	0	0
0	0	1	1	1
1	0	0	1	1
0	1	1	0	0
1	1	0	0	0
0	1	0	1	0

- m-of-n rules
  - Examples:
    - 1 out of {x1}
    - 2 out of {x1, x3}
    - …
- **Is there a consistent hypothesis?**
  - Check!
  - For example, let's check: *"2 out of {x1,x2,x3,x4}"*

  ➔ *Exactly one hypothesis is consistent with the data!*

Learning

- Learning is removal of remaining uncertainty
  - *If we know that the function is a "m-out-of-n", data can help find a function from that class*
- **Finding a good hypothesis class is essential!**
  - You can start small, and enlarge it until you can find a hypothesis that fits the data

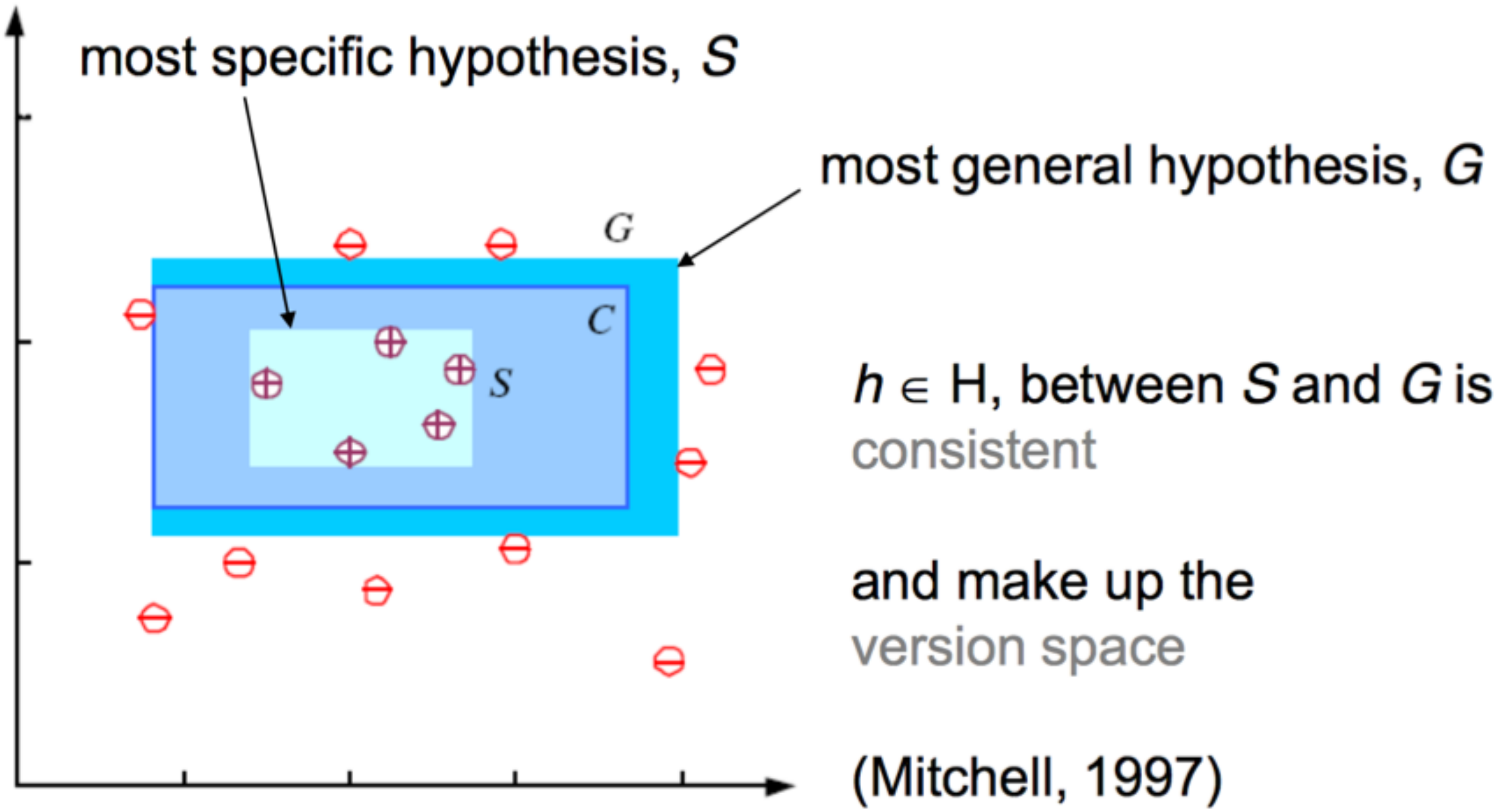
Question: *Can there be more than one function that is consistent with the data*?

How do you choose between them?

Terminology

- **Classifier**: discrete-valued function, possible output are called classes.
- **Hypothesis space:** The space of all hypotheses than can be the output of a learning algorithm.
- **Version space:** The space of all hypotheses in the hypothesis space, consistent with the data

Terminology



h ∈ H, between *S* and *G* is consistent

and make up the version space

(Mitchell, 1997)

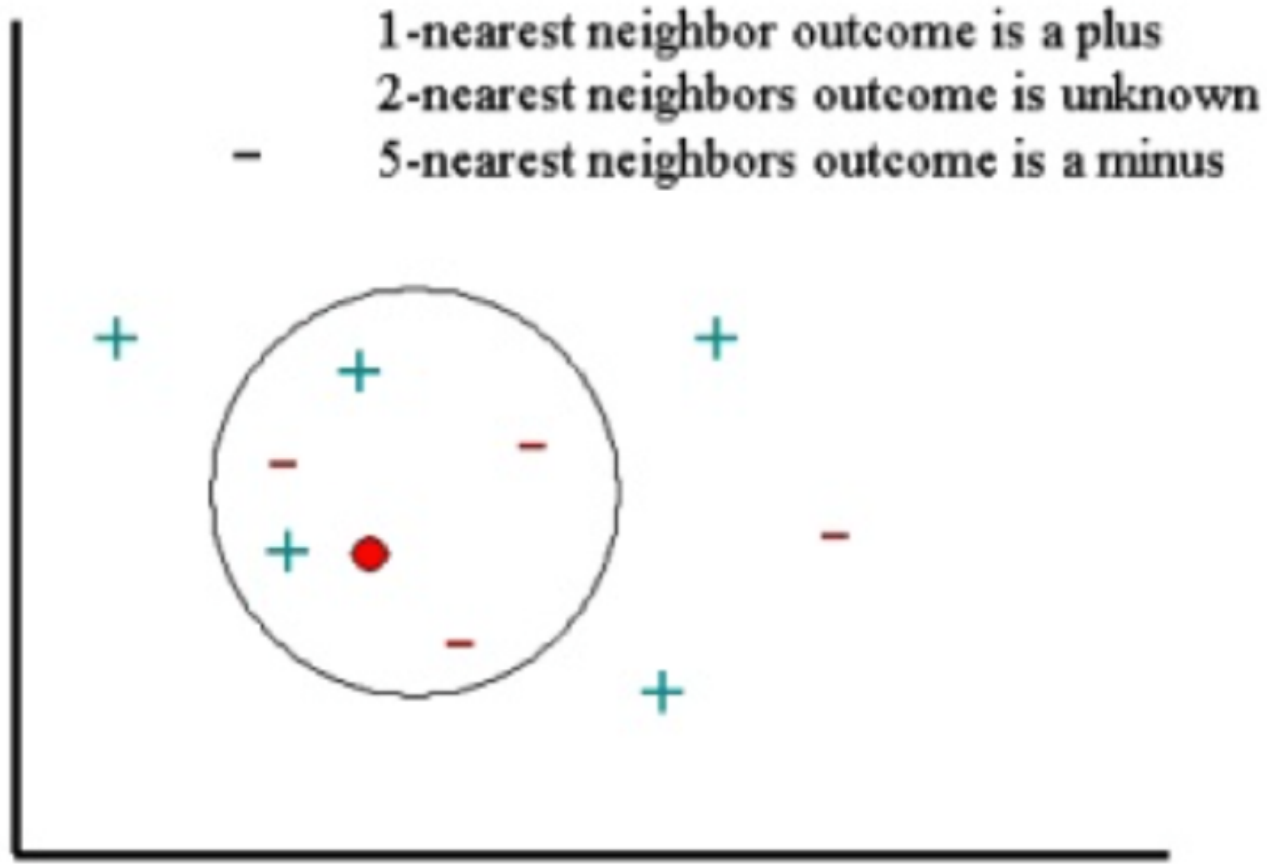
My first Classifier!

- Let’s take a look at one of the simplest classifiers
  - *Basically, just maintain the training data (**ideas?**)*
- Assume we have a training set $(x_1,y_1)\ldots(x_n,y_n)$
  - **Simplest solution**: given a new sample, x, find the most similar example in the training data, and predict the same value.
    - If you liked “*Fast and Furious*” you’ll like “2 *fast* 2 *furious*”
  - Key decision: find a good distance metric

$$d(x_1, x_2) = 1 - \frac{x_1 \bigcap x_2}{x_1 \bigcup x_2} \qquad d(x_1, x_2) = \sqrt[2]{(x_1 - x_2)^2}$$

  - ***Can we do better?***
  - *We can make the decision by looking at several near examples, not just one. **Why would it be better**?*

KNN


1-nearest neighbor outcome is a plus
2-nearest neighbors outcome is unknown
5-nearest neighbors outcome is a minus

K Nearest Neighbors

- Learning: just storing the training examples
- Prediction:
  - Find the K training example closest to **x**
    - Classification: majority vote
    - Regression: mean value

- KNN is a type of *instance based learning*
  - *Learning is requires storing the training set*
  - *Prediction using similar stored examples*
- This is called ***lazy*** learning, since most of the computation is done at prediction time

Let’s analyze KNN

- **What are the advantages/disadvantages?**
  - *What are the important aspects when analyzing learning algorithms?*

- Complexity
  - *Space*
  - *Time (training and testing)*
- Expressivity
  - *What kind of functions can we learn?*

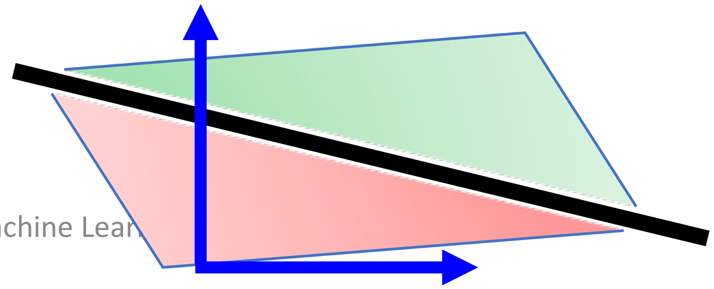
Let’s analyze KNN

- **Time Complexity**
  - *Training is very fast*
  - *Prediction is slower..*
    - *O(dN) for N examples, of dimensionality d*
    - *Increases with number of training examples!*
- Space Complexity
  - *KNN needs to maintain all training examples*

- *Let’s Compare to the m-of-n rules*

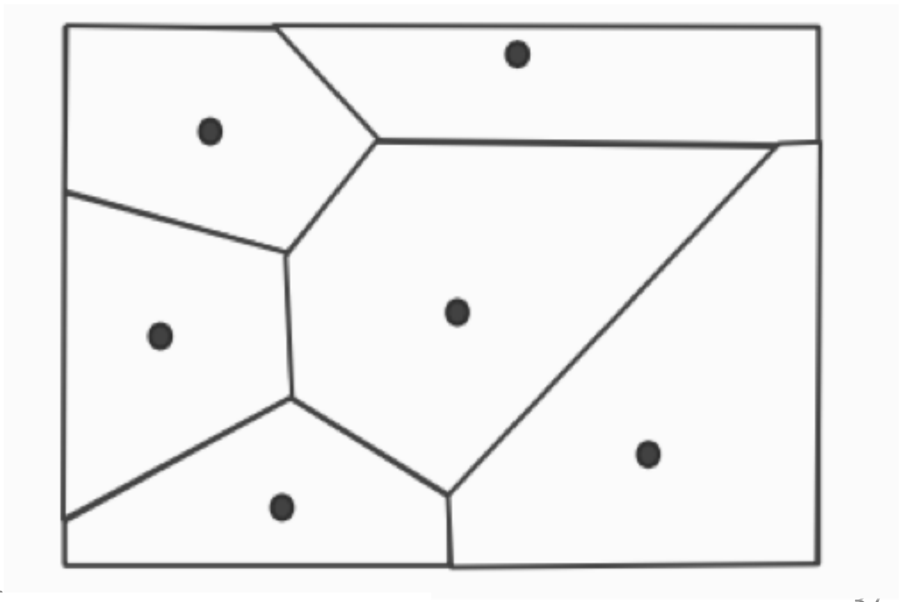
Let's analyze KNN

- <u>Expressivity</u>
  - Can learn very complex decisions (*more on that later*)
    - Very sensitive to feature representation choice
    - Irrelevant attributes can fool the classifier
- KNN does not construct an explicit hypothesis
- We can try to characterize the learned function using its *decision boundary*
  - *Visualize which elements will be classified as positive/negative*
  - *Decision Boundary is the curve separating the negative and positive regions*



Decision Boundary: Voronoi Diagram

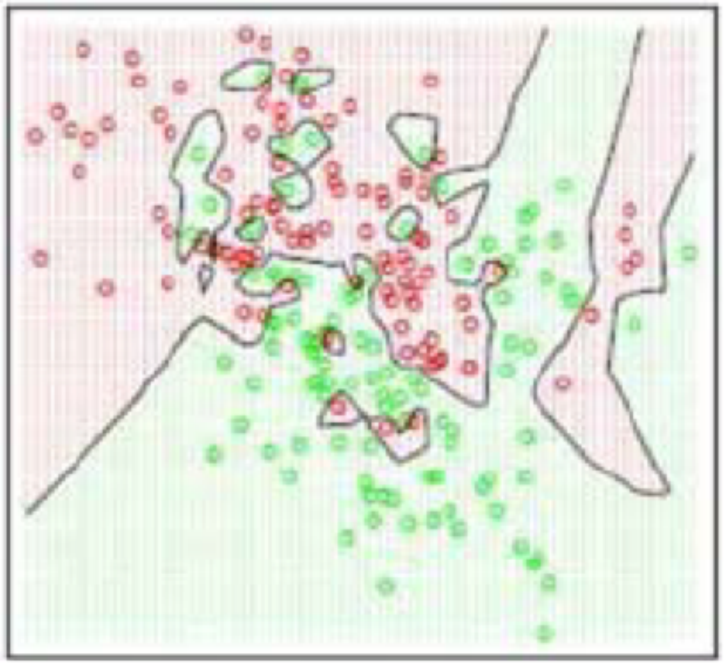
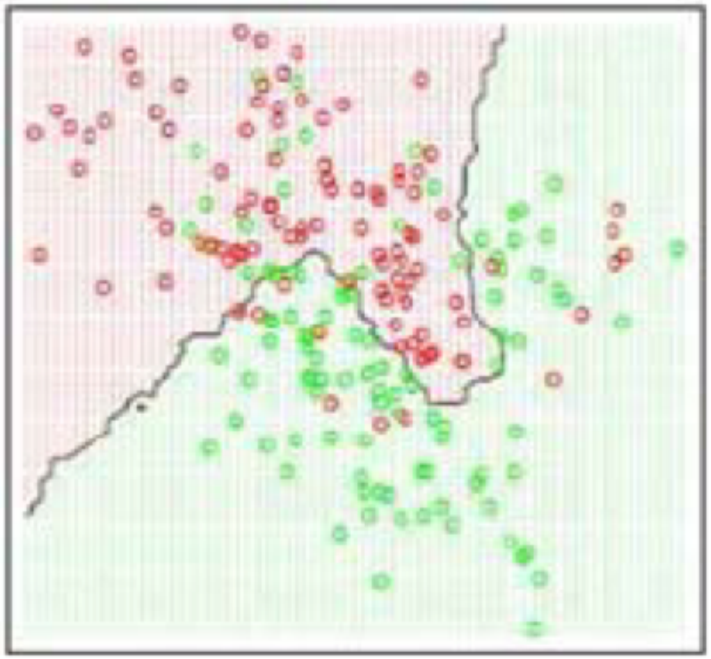
- Each point x in the training set defines a Voronoi cell
  - The polyhedron consisting of all the points closest to x
- The Voronoi diagram is the union of all these cells
- For 1-NN (with Euclidean distance) this is the decision boundary



KNN: choosing the right K

Let’s take a closer look at the learned function

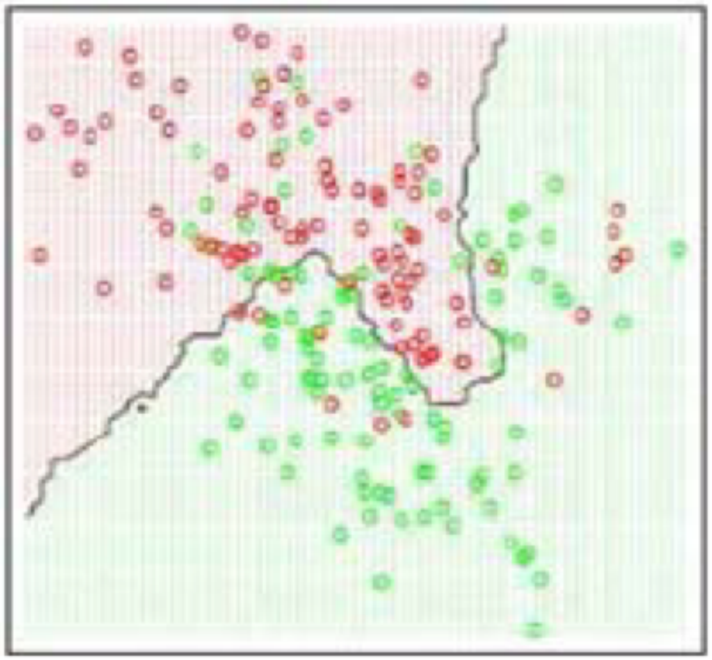
→ ***High sensitivity to noise!***

Higher k values results in smoother decision boundaries

Figures from Hastie, Tibshirani and Friedman (Elements of Statistical Learning)

Question



Higher k values results in smoother decision boundaries

What will happen if we keep increasing K?

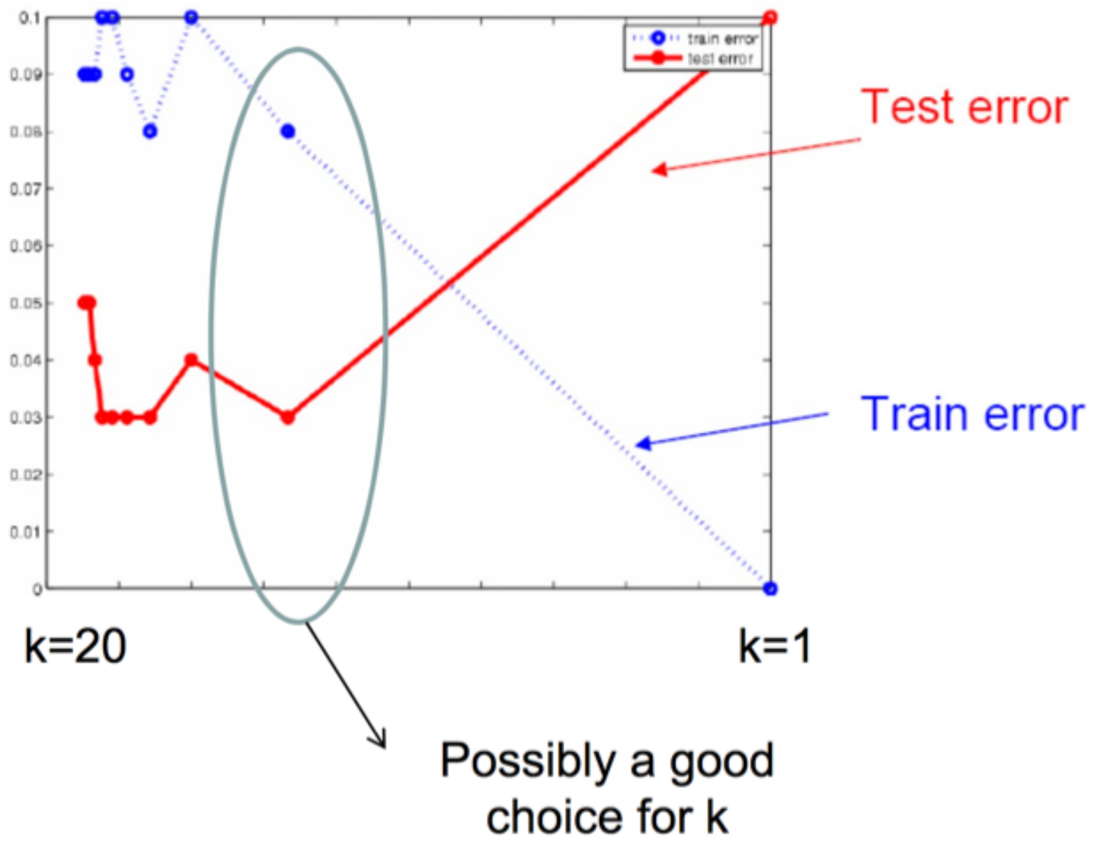
→ Extreme scenario: if k = N?

How should we set the value of K?

- **Option 1**: Pick the K that minimizes the training error.
  - Training Error: number of errors on the training set, *AFTER* we learned a classifier
- **Question**: *What is the training error of 1-NN? 10-NN?*
- **Option 2**: choose k to minimize mistakes on test error
  - Test Error: number of errors on the test set, after we learned a classifier
  - ***Note****: It's better to use a validation set, and not the test set. More on that later in the class..*

How should we set the value of K?

How would the test and train error change with K?



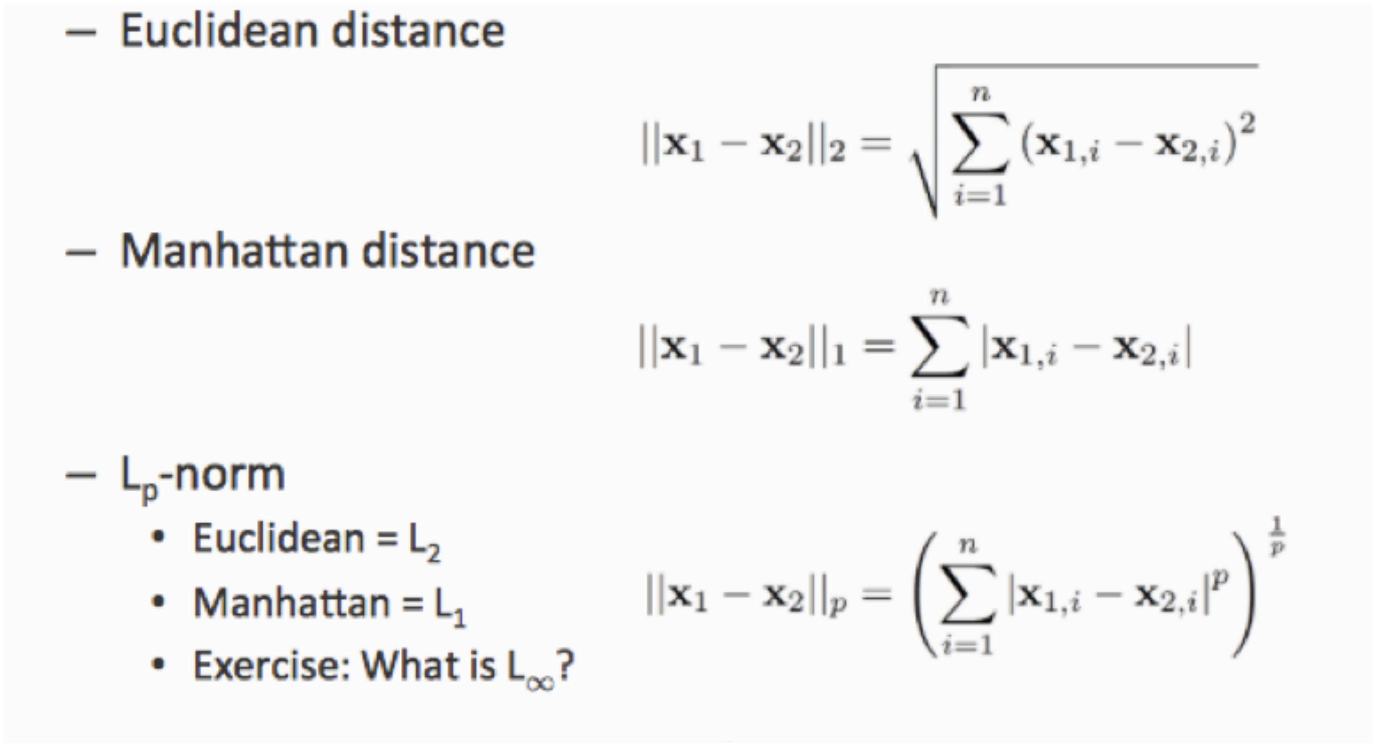
In general – using the training error to tune parameters will always result in a more complex hypothesis! **(why?)**

KNN Practical Considerations

- Very simple to understand and implement
- An odd value for k is better (why?)
- How can we find the right value for k?
  - Using a held out set
- Feature normalization is important! (why?)
  - Different scales for different features

KNN Practical Considerations

- Choosing the right features is important
  - Very sensitive to irrelevant attributes
  - Sensitive to the number of dimensions
- Choosing the right distance metric is important



- Euclidean distance

$$||\mathbf{x}_1 - \mathbf{x}_2||_2 = \sqrt{\sum_{i=1}^{n} (\mathbf{x}_{1,i} - \mathbf{x}_{2,i})^2}$$

- Manhattan distance

$$||\mathbf{x}_1 - \mathbf{x}_2||_1 = \sum_{i=1}^{n} |\mathbf{x}_{1,i} - \mathbf{x}_{2,i}|$$

- L_p-norm
  - Euclidean = L_2
  - Manhattan = L_1
  - Exercise: What is L_∞?

$$||\mathbf{x}_1 - \mathbf{x}_2||_p = \left(\sum_{i=1}^{n} |\mathbf{x}_{1,i} - \mathbf{x}_{2,i}|^p\right)^{\frac{1}{p}}$$

Review Questions

- What is a hypothesis space?
- What is inductive bias?
- When deciding how to design the hypothesis space, what decisions can we make?
- What are the advantages of a simpler, or more complex space?
- How does KNN work? how do we learn and make predictions?
- If the test performance of KNN is low, should we increase or decrease K?

Further reading

Machine Learning. Tom Mitchell.

Chapter 8

A Course in Machine Learning. Hal Daumé III.

Chapter 2.2 and 2.3

http://ciml.info/dl/v0_9/ciml-v0_9-ch02.pdf