AIN311

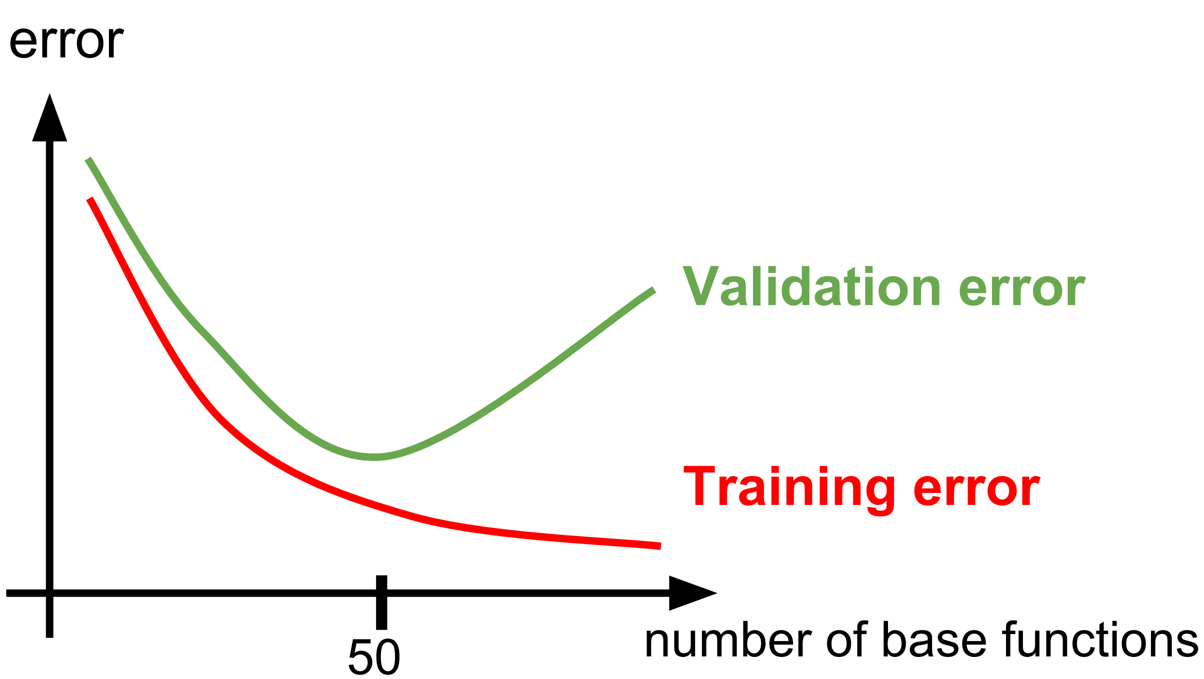
Fundamentals of Machine Learning

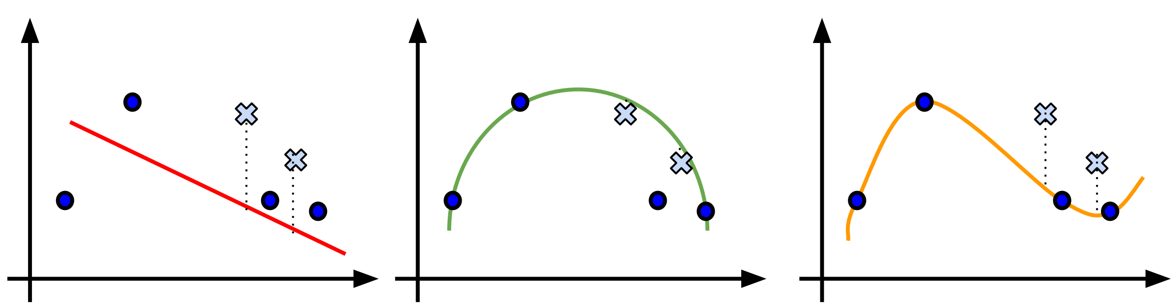
Lecture 6:
Learning theory
Probability Review

HACETTEPE UNIVERSITY COMPUTER VISION LAB

Erkut Erdem // Hacettepe University // Fall 2023

Last time… Regularization, Cross-Validation

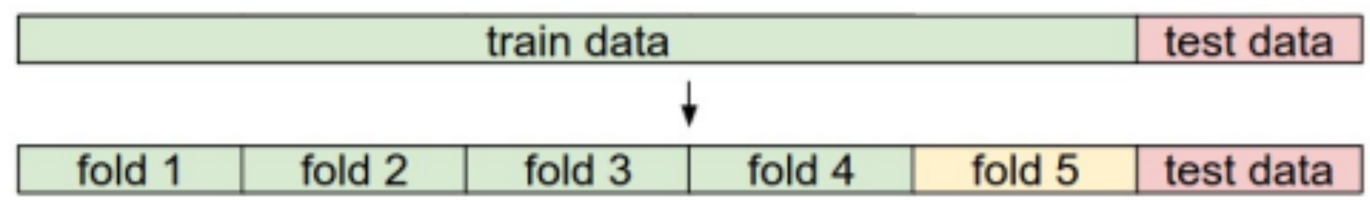


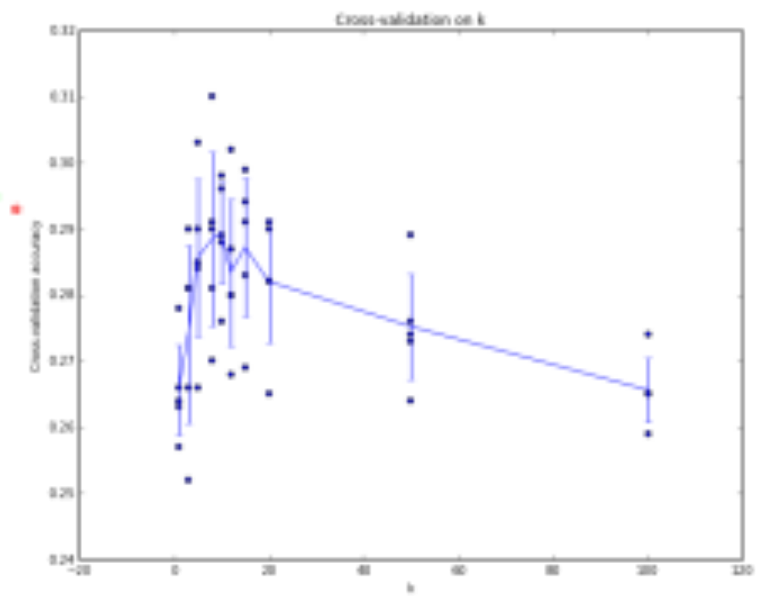


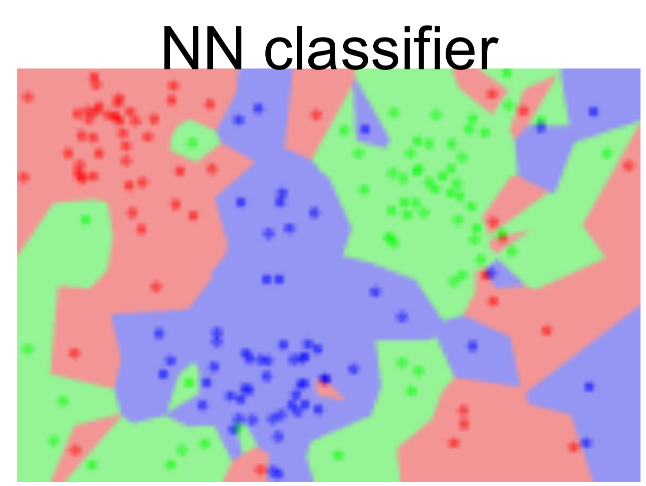
Underfitting
- large training error
- large validation error

Just Right
- small training error
- small validation error

Overfitting
- small training error
- large validation error







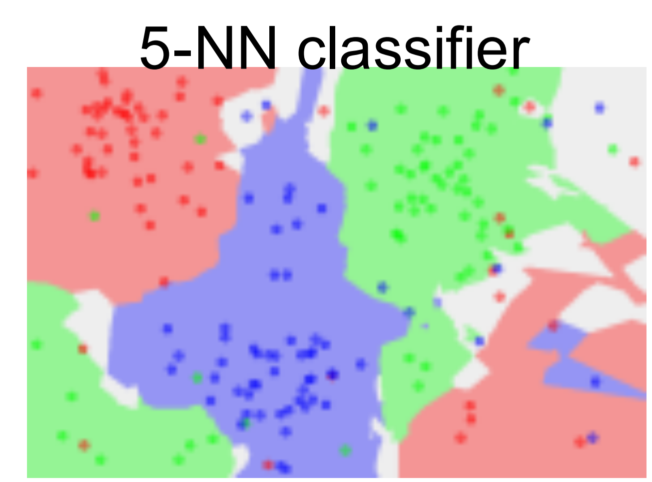


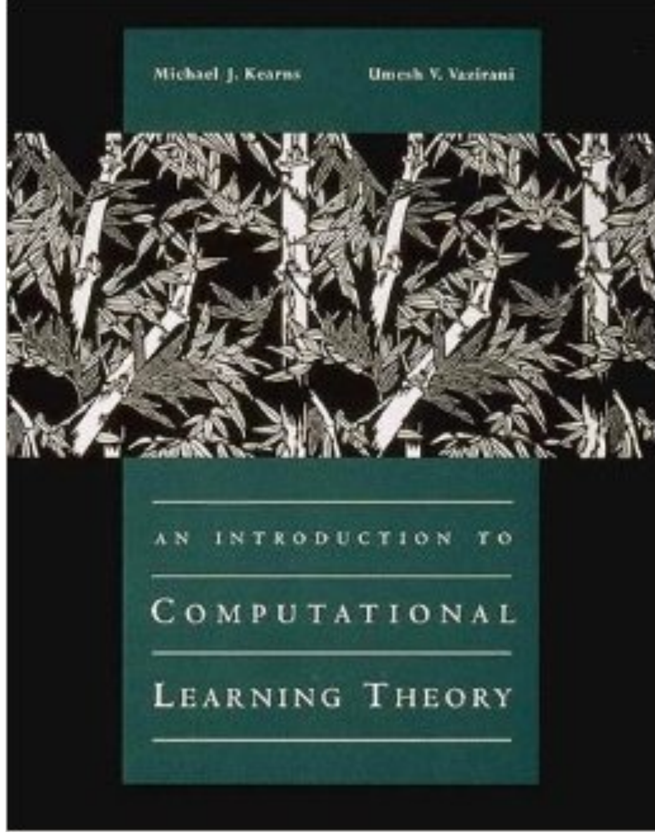
Figure credit: Fei-Fei Li, Andrej Karpathy, Justin Johnson

Today

- Learning Theory
- Probability Review

Learning Theory: Why ML Works

Computational Learning Theory



- Entire subfield devoted to the mathematical analysis of machine learning algorithms

- Has led to several practical methods:
  - PAC (probably approximately correct) learning
    → boosting
  - VC (Vapnik–Chervonenkis) theory
    → support vector machines

Annual conference: Conference on Learning Theory (COLT)

slide by Eric Eaton

The Role of Theory

- Theory can serve two roles:
  - It can justify and help understand why common practice works.

    theory after

  - It can also serve to suggest new algorithms and approaches that turn out to work well in practice.

    theory before

adapted from Hal Daume III

Often, it turns out to be a mix!

The Role of Theory

- Practitioners discover something that works surprisingly well.
- Theorists figure out why it works and prove something about it.
  - In the process, they make it better or find new algorithms.
- Theory can also help you understand **what's possible** and **what's not possible.**

adapted from Hal Daume III

Learning and Inference

The inductive inference process:

1. Observe a phenomenon
2. Construct a model of the phenomenon
3. Make predictions

- This is more or less the definition of natural sciences !

- The goal of Machine Learning is to **automate** this process

- The goal of Learning Theory is to **formalize** it.

slide by Olivier Bousquet

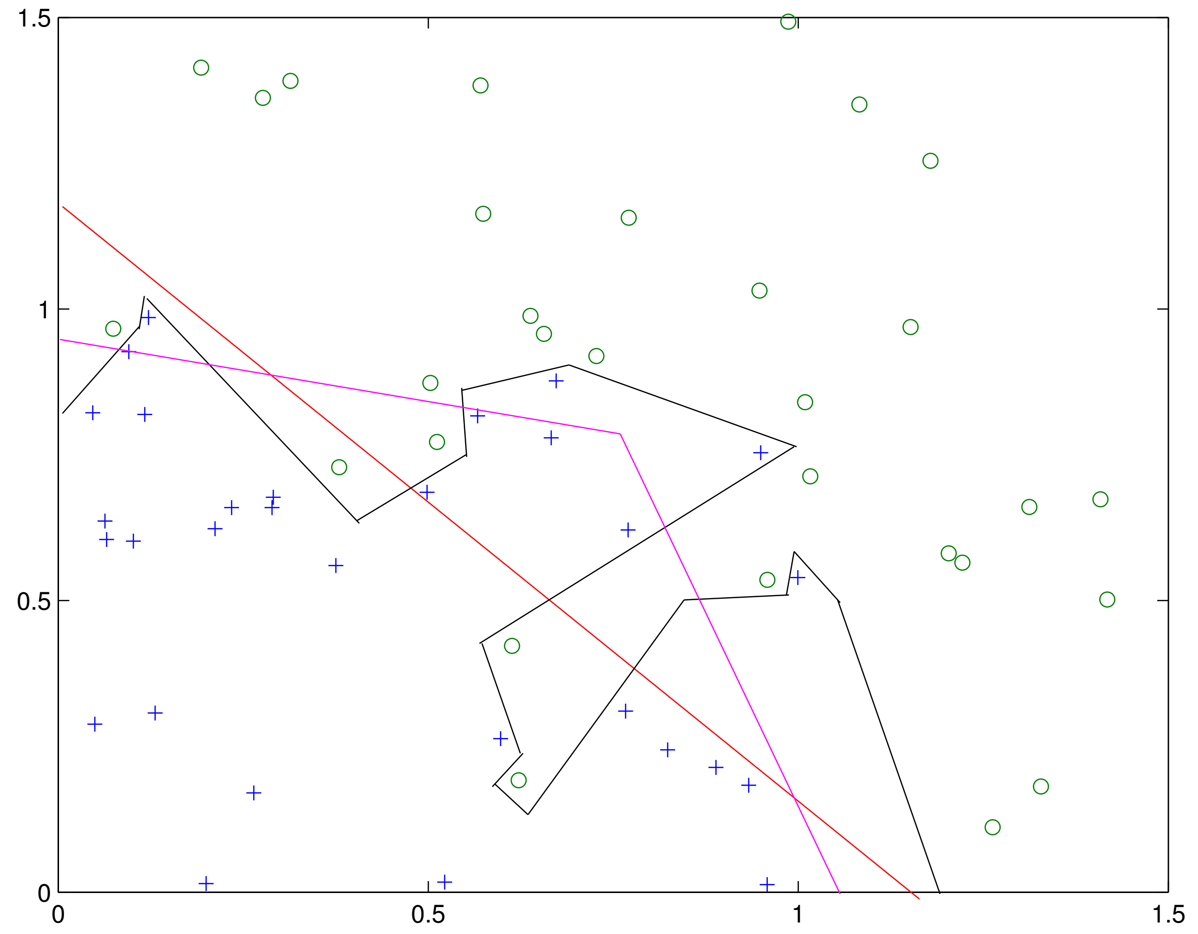
Pattern recognition

- We consider here the **supervised learning** framework for pattern recognition:
  - Data consists of pairs (instance, label)
  - Label is +1 or –1
  - Algorithm constructs a function (instance → label)
  - Goal: make few mistakes on future unseen instances

slide by Olivier Bousquet

Approximation/Interpolation

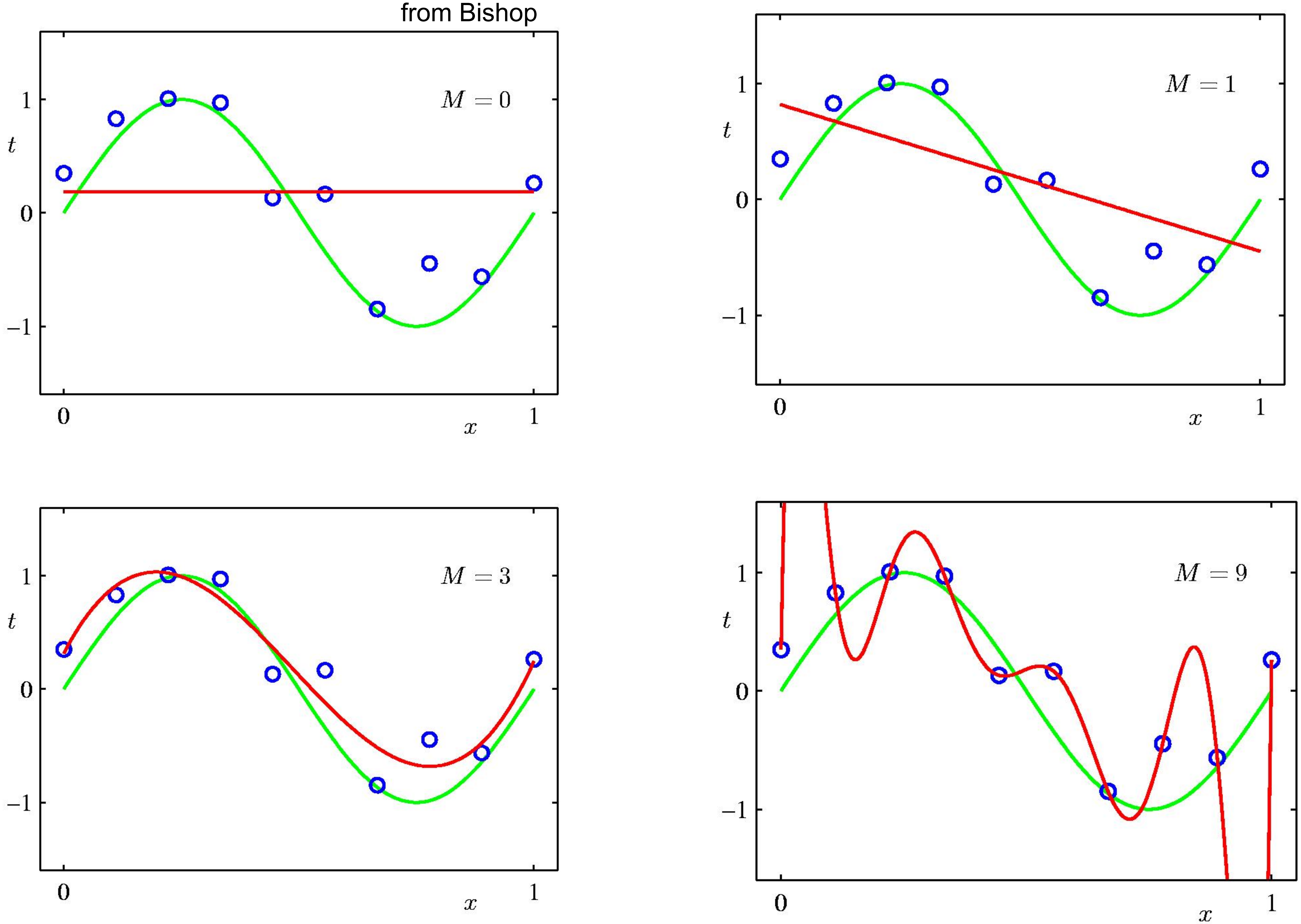
- It is always possible to build a function that fits exactly the data.



- But is it reasonable?

Which Fit is Best?

from Bishop



slide by Sanja Fidler

Occam's Razor

William of Occam
(c. 1288 – c. 1348)

- Idea: look for **regularities** in the observed phenomenon

  These can be **generalized** from the observed past to the future

  ⇒ choose the simplest consistent model

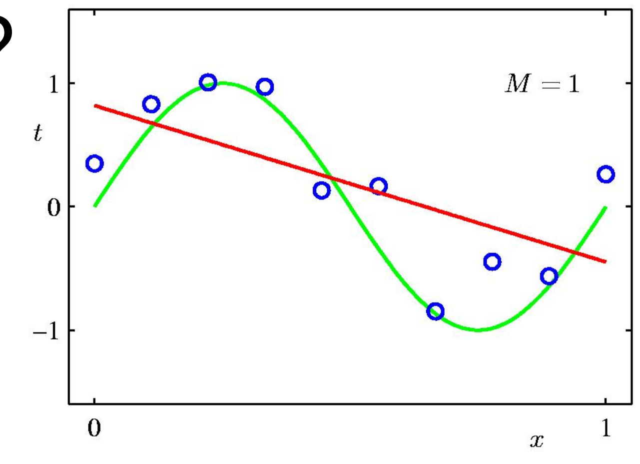
- How to measure simplicity ?
  - Physics: number of constants
  - Description length
  - Number of parameters
  - ...

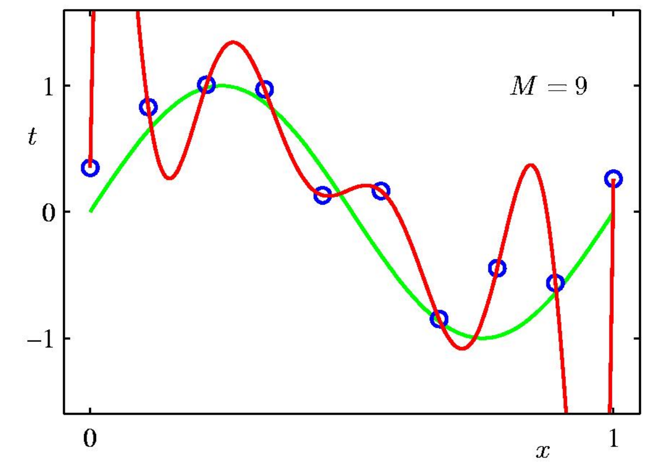
No Free Lunch

- **No Free Lunch**
  - if there is no assumption on how the **past** is related to the future, prediction is impossible
  - if there is no **restriction** on the possible phenomena, generalization is impossible
- We need to make assumptions
- Simplicity is not absolute
- Data will never replace knowledge
- Generalization = data + knowledge

Recall from last week… **Some key concepts**

- Data fits – is linear model best (**model selection**)?
  - Simplest models do not capture all the important variations (signal) in the data: **underfit**
  - More complex model may **overfit** the training data (fit not only the signal but also the **noise** in the data), especially if not enough data to constrain model



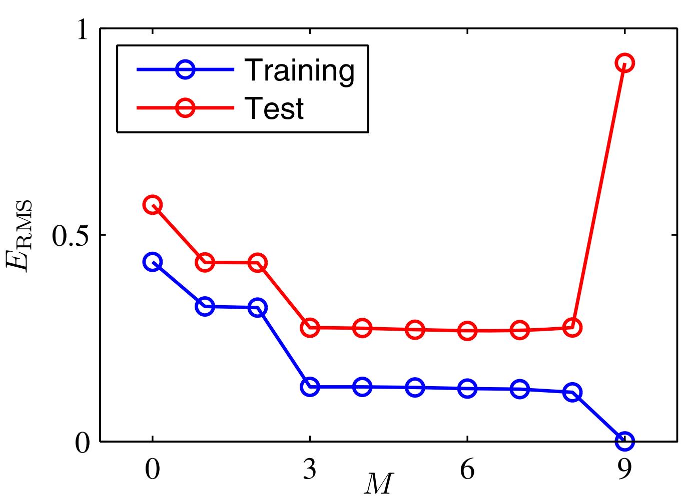


- One method of assessing fit:
  - test **generalization** = model's ability to predict the held out data
- **Regularization**

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

$$\|\mathbf{w}\|^2 \equiv \mathbf{w}^{\mathrm{T}}\mathbf{w} = w_0^2 + w_1^2 + \ldots + w_M^2$$

	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01



slide by Richard Zemel

Probably Approximately Correct (PAC) Learning

- A formalism based on the realization that the best we can hope of an algorithm is that
  - It does a good job most of the time (**probably approximately correct**)

adapted from Hal Daume III

Probably Approximately Correct (PAC) Learning

- Consider a hypothetical learning algorithm
  - We have 10 different binary classification data sets.
  - For each one, it comes back with functions $f_1, f_2, \ldots, f_{10}$.
    - ✦ For some reason, whenever you run f_4 on a test point, it crashes your computer. For the other learned functions, their performance on test data is always at most 5% error.
    - ✦ If this situtation is guaranteed to happen, then this hypothetical learning algorithm is a PAC learning algorithm.
      - ✤ It satisfies probably because it only failed in one out of ten cases, and it's approximate because it achieved low, but non-zero, error on the remainder of the cases.

adapted from Hal Daume III

PAC Learning

Definitions 1. *An algorithm $\mathcal{A}$ is an (ϵ, δ)-PAC learning algorithm if*, for all *distributions $\mathcal{D}$: given samples from $\mathcal{D}$, the probability that it returns a "bad function" is at most δ; where a "bad" function is one with test error rate more than ϵ on $\mathcal{D}$.*

adapted from Hal Daume III

PAC Learning

- Two notions of efficiency
  - **Computational complexity:** Prefer an algorithm that runs quickly to one that takes forever
  - **Sample complexity:** The number of examples required for your algorithm to achieve its goals

Definition: An algorithm $\mathcal{A}$ is an efficient (ϵ, δ)-PAC learning algorithm if it is an (ϵ, δ)-PAC learning algorithm whose runtime is polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$.

In other words, to let your algorithm to achieve 4% error rather than 5%, the runtime required to do so should not go up by an exponential factor!

adapted from Hal Daume III

Example: PAC Learning of Conjunctions

- Data points are binary vectors, for instance $\mathbf{x} = \langle 0, 1, 1, 0, 1 \rangle$
- Some Boolean conjunction defines the true labeling of this data (e.g. $x_1 \wedge x_2 \wedge x_5$)
- There is some distribution $\mathcal{D}_X$ over binary data points (vectors) $\mathbf{x} = \langle x_1, x_2, \ldots, x_D \rangle$.
- There is a fixed concept conjunction c that we are trying to learn.
- There is no noise, so for any example x, its true label is simply $y = c(\mathbf{x})$

- **Example:**
  - Clearly, the true formula cannot include the terms $x_1, x_2, \neg x_3, \neg x_4$

y	x_1	x_2	x_3	x_4
+1	0	0	1	1
+1	0	1	1	1
-1	1	1	0	1

adapted from Hal Daume III

Example: PAC Learning of Conjunctions

y	x_1	x_2	x_3	x_4
+1	0	0	1	1
+1	0	1	1	1
-1	1	1	0	1

Algorithm 30 BinaryConjunctionTrain(**D**)

```
f ← x_1 ∧ ¬x_1 ∧ x_2 ∧ ¬x_2 ∧ ··· ∧ x_D ∧ ¬x_D      // initialize function
for all positive examples (x,+1) in D do
  for d = 1 ... D do
    if x_d = 0 then
      f ← f without term "x_d"
    else
      f ← f without term "¬x_d"
    end if
  end for
end for
return f
```

"Throw Out Bad Terms"

$$f^0(\mathbf{x}) = x_1 \wedge \neg x_1 \wedge x_2 \wedge \neg x_2 \wedge x_3 \wedge \neg x_3 \wedge x_4 \wedge \neg x_4$$

$$f^1(\mathbf{x}) = \neg x_1 \wedge \neg x_2 \wedge x_3 \wedge x_4$$

$$f^2(\mathbf{x}) = \neg x_1 \wedge x_3 \wedge x_4$$

$$f^3(\mathbf{x}) = \neg x_1 \wedge x_3 \wedge x_4$$

- After processing an example, it is **guaranteed to classify that example correctly** (provided that there is no noise)
- **Computationally very efficient**
  - Given a data set of N examples in D dimensions, it takes $O(ND)$ time to process the data. This is linear in the size of the data set.

adapted from Hal Daume III

Example: PAC Learning of Conjunctions

y	x_1	x_2	x_3	x_4
+1	0	0	1	1
+1	0	1	1	1
-1	1	1	0	1

Algorithm 30 BinaryConjunctionTrain(**D**)

```
f ← x_1 ∧ ¬x_1 ∧ x_2 ∧ ¬x_2 ∧ ··· ∧ x_D ∧ ¬x_D        // initialize function
for all positive examples (x,+1) in D do
  for d = 1 ... D do
    if x_d = 0 then
      f ← f without term "x_d"
    else
      f ← f without term "¬x_d"
    end if
  end for
end for
return f
```

"Throw Out Bad Terms"

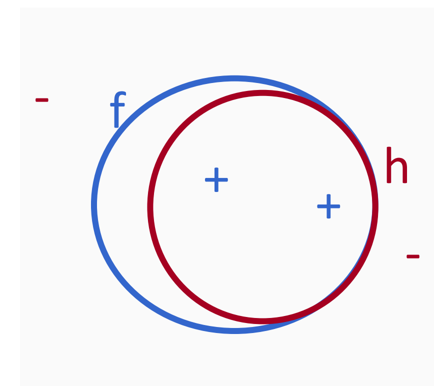
- Is this an efficient (ε, δ)-PAC learning algorithm?

adapted from Hal Daume III

Learning Conjunctions: Analysis

$f = x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$ $h = x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$

- **Claim 1:** Any hypothesis consistent with the training data will only make mistakes on positive future examples. **Why?**



- A mistake will occur only if some literal z (in our example x_1) is present in h but not in f
  - This mistake can cause a positive example to be predicted as negative by h. Specifically: $x_1 = 0$, $x_2 = 1$, $x_3 = 1$, $x_4 = 1$, $x_5 = 1$, $x_{100} = 1$
- The reverse situation can never happen
  - For an example to be predicted as positive in the training set, every relevant literal must have been present

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

- **Theorem:** Suppose we are learning a conjunctive concept with n dimensional Boolean features using m training examples. If

$$m > \frac{n}{\epsilon}\left(\log(n) + \log\left(\frac{1}{\delta}\right)\right)$$

Poly in n, $1/\delta$, $1/\epsilon$

then, with probability > 1 - δ, the error of the learned hypothesis $err_D(h)$ will be less than ϵ.

If we see these many training examples, then the algorithm will produce a conjunction that, with high probability, will make few errors

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

- **Theorem:** Suppose we are learning a conjunctive concept with n dimensional Boolean features using m training examples. If

$$m > \frac{n}{\epsilon}\left(\log(n) + \log\left(\frac{1}{\delta}\right)\right)$$

then, with probability > 1 - δ, the error of the learned hypothesis $err_D(h)$ will be less than ϵ.

Let's prove this assertion

Proof Intuition

$f = x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$

$h = x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$

- What kinds of examples would drive a hypothesis to make a mistake?
- Positive examples, where x_1 is absent
  - f would say true and h would say false
- None of these examples appeared during training
  - Otherwise x_1 would have been eliminated
- If they never appeared during training, maybe their appearance in the future would also be rare!
  - Let's quantify our surprise at seeing such examples

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

- Let p(z) be the probability that, in an example drawn from D, the feature z is absent but the example has a positive label
  - That is, after training is done, p(z) is the probability that in a randomly drawn example, the literal z causes a mistake
  - For any z in the target function, p(z) = 0

Learning Conjunctions: Analysis

- Let p(z) be the probability that, in an example drawn from D, the feature z is absent but the example has a positive label
  - That is, after training is done, p(z) is the probability that in a randomly drawn example, the literal z causes a mistake
  - For any z in the target function, p(z) = 0

Remember that there will only be mistakes on positive examples for this toy problem

Learning Conjunctions: Analysis

- Let p(z) be the probability that, in an example drawn from D, the feature z is absent but the example has a positive label
  - That is, after training is done, p(z) is the probability that in a randomly drawn example, the literal z causes a mistake
  - For any z in the target function, p(z) = 0

Remember that there will only be mistakes on positive examples for this toy problem

$$f = x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$$

$$h = x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$$

Learning Conjunctions: Analysis

- Let p(z) be the probability that, in an example drawn from D, the feature z is absent but the example has a positive label
  - That is, after training is done, p(z) is the probability that in a randomly drawn example, the literal z causes a mistake
  - For any z in the target function, p(z) = 0

Remember that there will only be mistakes on positive examples for this toy problem

$$f = x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$$

<(0,1,1,1,1,0,...0,1,1), 1>

$$h = x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$$

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

- Let p(z) be the probability that, in an example drawn from D, the feature z is absent but the example has a positive label
    - That is, after training is done, p(z) is the probability that in a randomly drawn example, the literal z causes a mistake
    - For any z in the target function, p(z) = 0

Remember that there will only be mistakes on positive examples for this toy problem

$$f = x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$$

$$h = x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$$

$<(0,1,1,1,1,0,\ldots 0,1,1),\ 1\ >$

$p(x_1)$: Probability that this situation occurs

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

- Let p(z) be the probability that, in an example drawn from D, the feature z is absent but the example has a positive label
  - That is, after training is done, p(z) is the probability that in a randomly drawn example, the literal z causes a mistake
  - For any z in the target function, p(z) = 0
- We know that $$err_D(h) \leq \sum_{z \in h} p(z)$$
  - via direct application of the union bound

Union bound
For a set of events, probability that at least one of them happens < the sum of the probabilities of the individual events

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

- Call a literal z *bad* if $p(z) > \frac{\epsilon}{n}$

n = dimensionality

- Intuitively, a bad literal is one that has a significant probability of not appearing with a positive example
    - (And, if it appears in all positive training examples, it can cause errors)

If there are no bad literals, then $err_D(h) < \epsilon$

Learning Conjunctions: Analysis

- Call a literal z *bad* if $p(z) > \frac{\epsilon}{n}$

n = dimensionality

- Intuitively, a bad literal is one that has a significant probability of not appearing with a positive example
  - (And, if it appears in all positive training examples, it can cause errors)

If there are no bad literals, then $err_D(h) < \epsilon$

- Why? Because $err_D(h) \leq \sum_{z \in h} p(z)$

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

n = dimensionality

- Call a literal z *bad* if $p(z) > \frac{\epsilon}{n}$
- Intuitively, a bad literal is one that has a significant probability of not appearing with a positive example
  - (And, if it appears in all positive training examples, it can cause errors)

If there are no bad literals, then $err_D(h) < \epsilon$

- Why? Because $$err_D(h) \leq \sum_{z \in h} p(z)$$

Let us try to see when this will not happen

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

- Call a literal z *bad* if $p(z) > \frac{\epsilon}{n}$

n = dimensionality

- Intuitively, a bad literal is one that has a significant probability of not appearing with a positive example
  - (And, if it appears in all positive training examples, it can cause errors)

What if there are bad literals?

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

- Call a literal z *bad* if $p(z) > \frac{\epsilon}{n}$

n = dimensionality

- Intuitively, a bad literal is one that has a significant probability of not appearing with a positive example
  - (And, if it appears in all positive training examples, it can cause errors)

---

adapted from D. Roth, A. Blum, T Mitchell and others

What if there are bad literals?

- Let z be a bad literal
- What is the probability that it will not be eliminated by one training example?

Learning Conjunctions: Analysis

- Call a literal z *bad* if $p(z) > \frac{\epsilon}{n}$

n = dimensionality

- Intuitively, a bad literal is one that has a significant probability of not appearing with a positive example
  - (And, if it appears in all positive training examples, it can cause errors)

---

adapted from D. Roth, A. Blum, T Mitchell and others

What if there are bad literals?

- Let z be a bad literal
- What is the probability that it will not be eliminated by one training example?

$$
\begin{aligned}
Pr(z \text{ survives one example}) &= 1 - Pr(z \text{ is eliminated by one example}) \\
&\leq 1 - p(z) \\
&< 1 - \frac{\epsilon}{n}
\end{aligned}
$$

Learning Conjunctions: Analysis

- Call a literal z *bad* if $p(z) > \frac{\epsilon}{n}$

n = dimensionality

- Intuitively, a bad literal is one that has a significant probability of not appearing with a positive example
  - (And, if it appears in all positive training examples, it can cause errors)

What if there are bad literals?

There was one example of this kind
<(1,1,1,1,1,0,...0,1,1), 1 >

- Let z be a bad literal
- What is the probability that it will not be eliminated by one training example?

$$
\begin{aligned}
Pr(z \text{ survives one example}) &= 1 - Pr(z \text{ is eliminated by one example}) \\
&\leq 1 - p(z) \\
&< 1 - \frac{\epsilon}{n}
\end{aligned}
$$

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

- What we know so far:

n = dimensionality

$$Pr(\text{A bad literal is not eliminated by one example}) < 1 - \frac{\epsilon}{n}$$

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

n = dimensionality

- What we know so far:

$$Pr(\text{A bad literal is not eliminated by one example}) < 1 - \frac{\epsilon}{n}$$

But say we have m training examples. Then

$$Pr(\text{A bad literal survives } m \text{ examples}) < \left(1 - \frac{\epsilon}{n}\right)^m$$

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

- What we know so far:

n = dimensionality

$$Pr(\text{A bad literal is not eliminated by one example}) < 1 - \frac{\epsilon}{n}$$

But say we have m training examples. Then

$$Pr(\text{A bad literal survives } m \text{ examples}) < \left(1 - \frac{\epsilon}{n}\right)^m$$

There are at most n bad literals. So

$$Pr(\text{Any bad literal survives } m \text{ examples}) < n\left(1 - \frac{\epsilon}{n}\right)^m$$

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

$$Pr(\text{Any bad literal survives } m \text{ examples}) < n\left(1 - \frac{\epsilon}{n}\right)^m$$

- We want this probability to be small
- Why? So that we can choose enough training examples so that the probability that any z survives all of them is less than some δ.

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

$$Pr(\text{Any bad literal survives } m \text{ examples}) < n\left(1 - \frac{\epsilon}{n}\right)^m$$

- We want this probability to be small
- Why? So that we can choose enough training examples so that the probability that any z survives all of them is less than some δ.
- That is, we want $n\left(1 - \frac{\epsilon}{n}\right)^m < \delta$

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

$$Pr(\text{Any bad literal survives } m \text{ examples}) < n\left(1 - \frac{\epsilon}{n}\right)^m$$

- We want this probability to be small
- Why? So that we can choose enough training examples so that the probability that any z survives all of them is less than some δ.
- That is, we want $n\left(1 - \frac{\epsilon}{n}\right)^m < \delta$
- We know that $1 - x < e^{-x}$. So it is sufficient to require $ne^{-\frac{m\epsilon}{n}} < \delta$

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

$$Pr(\text{Any bad literal survives } m \text{ examples}) < n\left(1 - \frac{\epsilon}{n}\right)^m$$

- We want this probability to be small
- Why? So that we can choose enough training examples so that the probability that any z survives all of them is less than some δ.
- That is, we want $n\left(1 - \frac{\epsilon}{n}\right)^m < \delta$
- We know that $1 - x < e^{-x}$. So it is sufficient to require $ne^{-\frac{m\epsilon}{n}} < \delta$
- Or equivalently,

$$m > \frac{n}{\epsilon}\left(\log(n) + \log\left(\frac{1}{\delta}\right)\right)$$

Learning Conjunctions: Analysis

- To guarantee a probability of failure (i.e, error > ϵ) that is less than δ, the number of examples we need is

$$m > \frac{n}{\epsilon}\left(\log(n) + \log\left(\frac{1}{\delta}\right)\right)$$

Poly in n, 1/δ, 1/ϵ

- That is, if m has this property, then
    - With probability 1 - δ, there will be no bad literals,
    - Or equivalently, with probability 1 - δ, we will have $err_D(h) < \epsilon$

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

- To guarantee a probability of failure (i.e, error > ϵ) that is less than δ, the number of examples we need is

$$m > \frac{n}{\epsilon}\left(\log(n) + \log\left(\frac{1}{\delta}\right)\right)$$

Poly in n, 1/δ, 1/ϵ

- That is, if m has this property, then
  - With probability 1 - δ, there will be no bad literals,
  - Or equivalently, with probability 1 - δ, we will have $err_D(h) < \epsilon$
- What does this mean:
  - If ϵ = 0.1 and δ = 0.1, then for n = 100, we need 6908 training examples

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

- To guarantee a probability of failure (i.e, error > ϵ) that is less than δ, the number of examples we need is

$$m > \frac{n}{\epsilon}\left(\log(n) + \log\left(\frac{1}{\delta}\right)\right)$$

Poly in n, $1/\delta$, $1/\epsilon$

- That is, if m has this property, then
  - With probability 1 - δ, there will be no bad literals,
  - Or equivalently, with probability 1 - δ, we will have $err_D(h) < \epsilon$

- What does this mean:
  - If ϵ = 0.1 and δ = 0.1, then for n = 100, we need 6908 training examples
  - If ϵ = 0.1 and δ = 0.1, then for n = 10, we need only 461 examples

adapted from D. Roth, A. Blum, T Mitchell and others

Learning Conjunctions: Analysis

- To guarantee a probability of failure (i.e, error > ϵ) that is less than δ, the number of examples we need is

$$m > \frac{n}{\epsilon}\left(\log(n) + \log\left(\frac{1}{\delta}\right)\right)$$

Poly in n, $1/\delta$, $1/\epsilon$

- That is, if m has this property, then
  - With probability 1 - δ, there will be no bad literals,
  - Or equivalently, with probability 1 - δ, we will have $err_D(h) < \epsilon$
- What does this mean:
  - If ϵ = 0.1 and δ = 0.1, then for n = 100, we need 6908 training examples
  - If ϵ = 0.1 and δ = 0.1, then for n = 10, we need only 461 examples
  - If ϵ = 0.1 and δ = 0.01, then for n = 10, we need 691 examples

Learning Conjunctions: Analysis

- To guarantee a probability of failure (i.e, error > ϵ) that is less than δ, the number of examples we need is

$$m > \frac{n}{\epsilon}\left(\log(n) + \log\left(\frac{1}{\delta}\right)\right)$$

Poly in n, $1/\delta$, $1/\epsilon$

- That is, if m has this property, then
  - With probability 1 - δ, there will be no bad literals,
  - Or equivalently, with probability 1 - δ, we will have $err_D(h) < \epsilon$

What we have here is a PAC guarantee

Our algorithm is ***Probably Approximately Correct*.**

adapted from D. Roth, A. Blum, T Mitchell and others

Vapnik-Chervonenkis (VC) Dimension

- A classic measure of complexity of infinite hypothesis classes based on this intuition.
- The VC dimension is a very classification-oriented notion of complexity
  - The idea is to look at a finite set of unlabeled examples
  - no matter how these points were labeled, would we be able to find a hypothesis that correctly classifies them
- The idea is that as you add more points, being able to represent an arbitrary labeling becomes harder and harder.

Definitions 2. *For data drawn from some space $\mathcal{X}$, the VC dimension of a hypothesis space $\mathcal{H}$ over $\mathcal{X}$ is the* maximal *K such that: there exists a set $X \subseteq \mathcal{X}$ of size $|X| = K$, such that for all binary labelings of X, there exists a function $f \in \mathcal{H}$ that matches this labeling.*

adapted from Hal Daume III

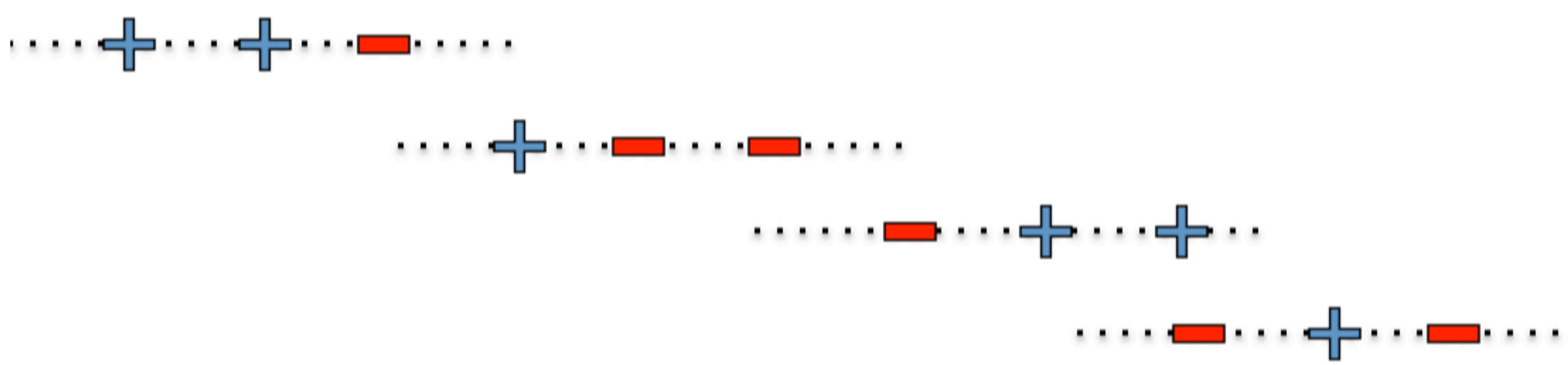
How many points can a linear boundary classify exactly? (1-D)

- 2 points:

  Yes!

- 3 points:

  No!



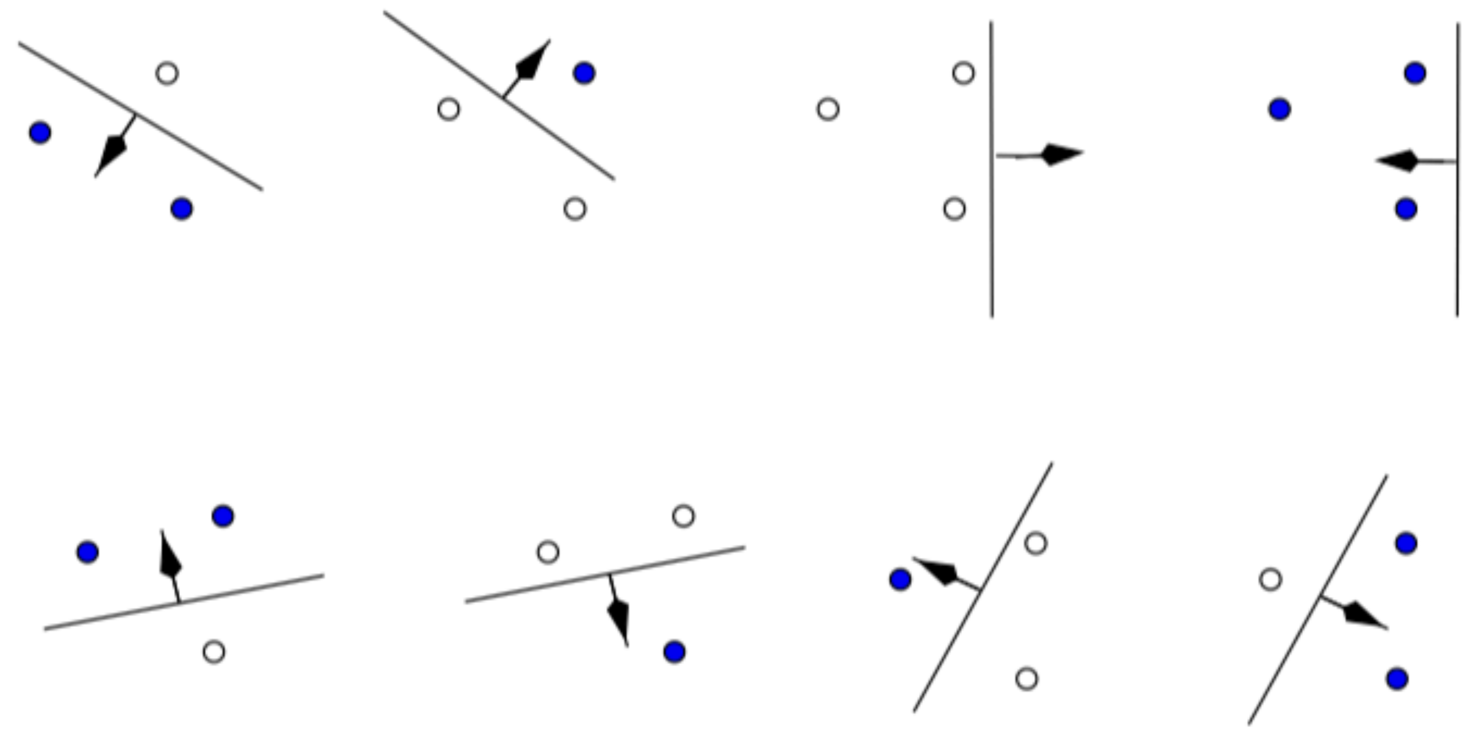
etc (8 total)

VC-dimension = 2

slide by David Sontag

How many points can a linear boundary classify exactly? (2-D)

- 3 points:
  Yes!



- 4 points:
  No!

etc.

VC-dimension = 3

slide by David Sontag

figure credit: Chris Burges

Basic Probability Review

Probability

- A is non-deterministic event
  - Can think of A as a boolean-valued variable

- Examples
  - A = your next patient has cancer
  - A = Max Verstappen wins United States Grand Prix 2023

slide by Dhruv Batra

Interpreting Probabilities

If I flip this coin, the probability that it will come up heads is 0.5

- **Frequentist Interpretation***:* If we flip this coin many times, it will come up heads about half the time. *Probabilities are the expected frequencies of events over repeated trials.*
- **Bayesian Interpretation***:* I believe that my next toss of this coin is equally likely to come up heads or tails. *Probabilities quantify subjective beliefs about single events.*
- Viewpoints play complementary roles in **machine learning***:*
  - Bayesian view used to build models based on domain knowledge, and automatically derive learning algorithms
  - Frequentist view used to analyze worst case behavior of learning algorithms, in limit of large datasets
- From either view, basic mathematics is the same!

slide by Erik Sudderth

slide by Andrew Moore

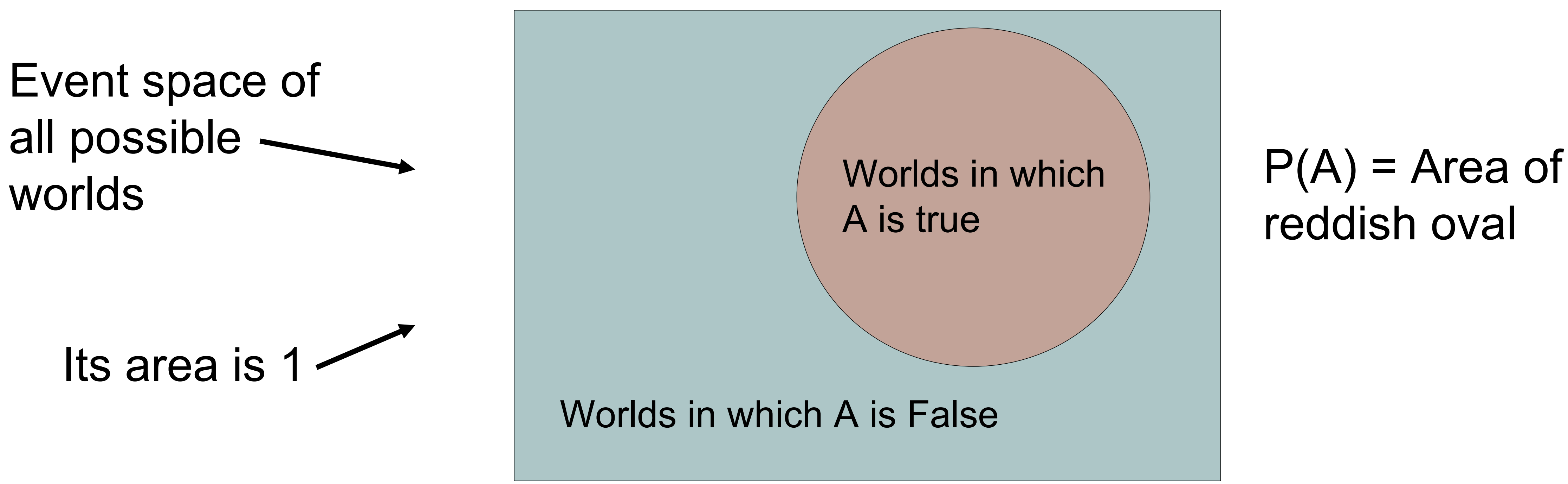
Axioms of Probability

- 0<= P(A) <= 1
- P(empty-set) = 0
- P(everything) = 1
- P(A or B) = P(A) + P(B) – P(A and B)

slide by Dhruv Batra

Interpreting the Axioms

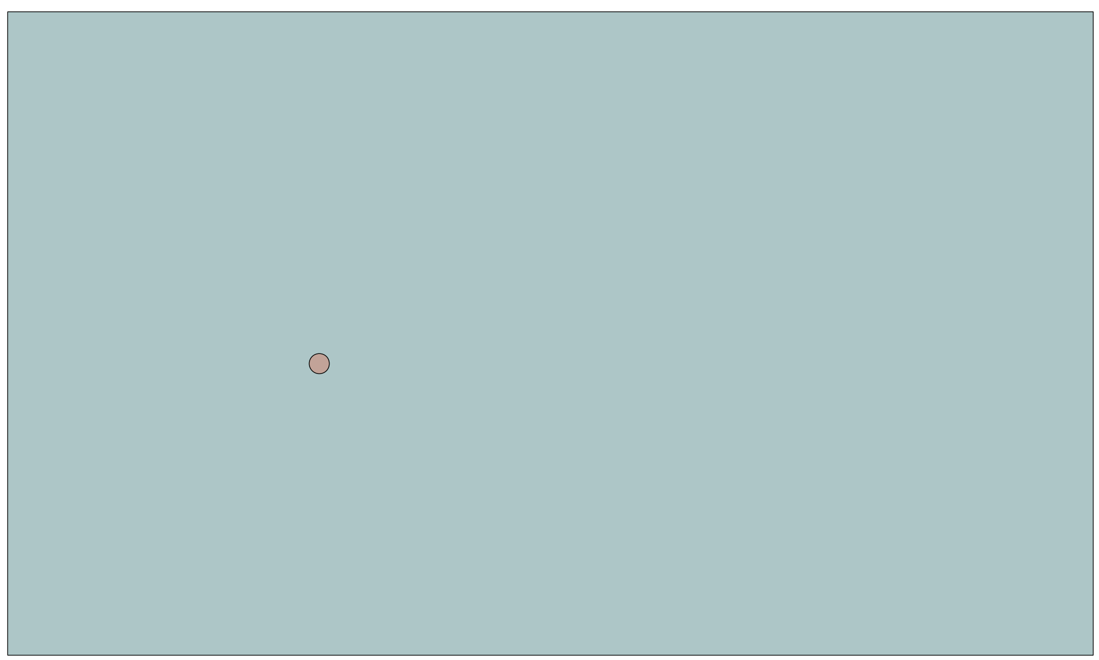
- 0<= P(A) <= 1
- P(empty-set) = 0
- P(everything) = 1
- P(A or B) = P(A) + P(B) – P(A and B)



slide by Dhruv Batra

Interpreting the Axioms

- 0<= P(A) <= 1
- P(empty-set) = 0
- P(everything) = 1
- P(A or B) = P(A) + P(B) – P(A and B)



The area of A can t get any smaller than 0

And a zero area would mean no world could ever have A true

slide by Dhruv Batra

Interpreting the Axioms

- 0<= P(A) <= 1
- P(empty-set) = 0
- P(everything) = 1
- P(A or B) = P(A) + P(B) – P(A and B)

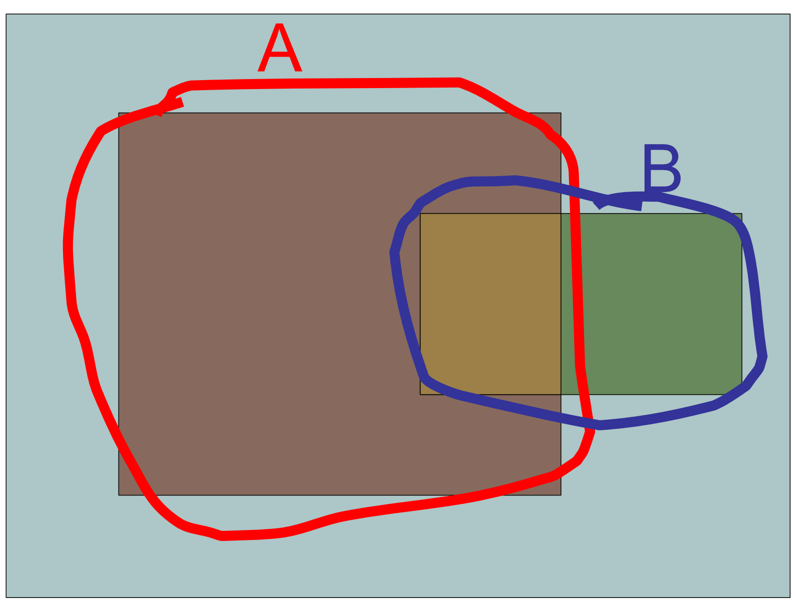
The area of A can t get any bigger than 1

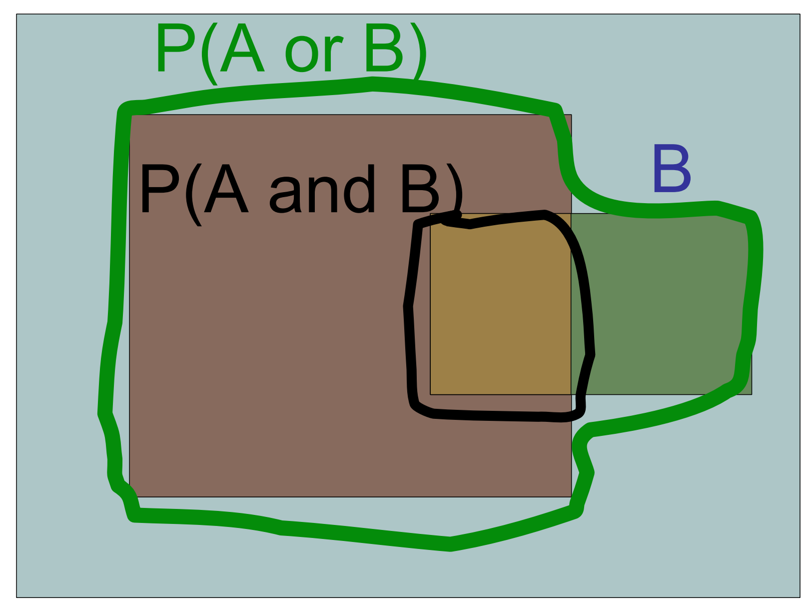
And an area of 1 would mean all worlds will have A true

slide by Dhruv Batra

Interpreting the Axioms

- 0<= P(A) <= 1
- P(empty-set) = 0
- P(everything) = 1
- P(A or B) = P(A) + P(B) – P(A and B)





Simple addition and subtraction

slide by Dhruv Batra

Discrete Random Variables

X ⟶ discrete random variable

$\mathcal{X}$ ⟶ sample space of possible outcomes, which may be finite or countably infinite

$x \in \mathcal{X}$ ⟶ outcome of sample of discrete random variable

slide by Erik Suddherth

Discrete Random Variables

X → discrete random variable

$\mathcal{X}$ → sample space of possible outcomes, which may be finite or countably infinite

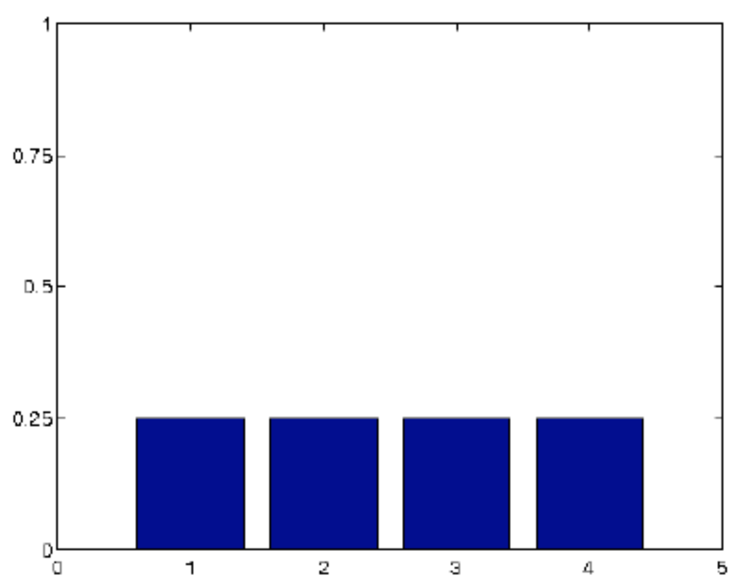
$x \in \mathcal{X}$ → outcome of sample of discrete random variable

$p(X = x)$ → probability distribution (probability mass function)

$p(x)$ → shorthand used when no ambiguity

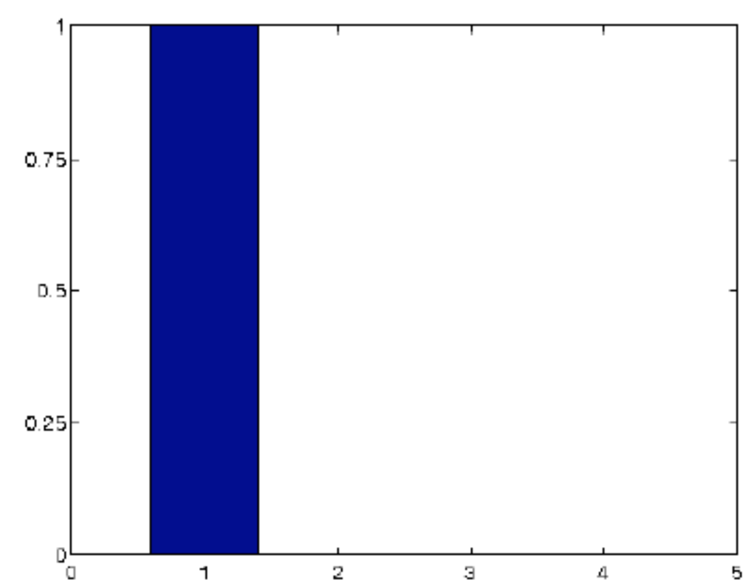
$$0 \leq p(x) \leq 1 \text{ for all } x \in \mathcal{X}$$

$$\sum_{x \in \mathcal{X}} p(x) = 1$$



uniform distribution

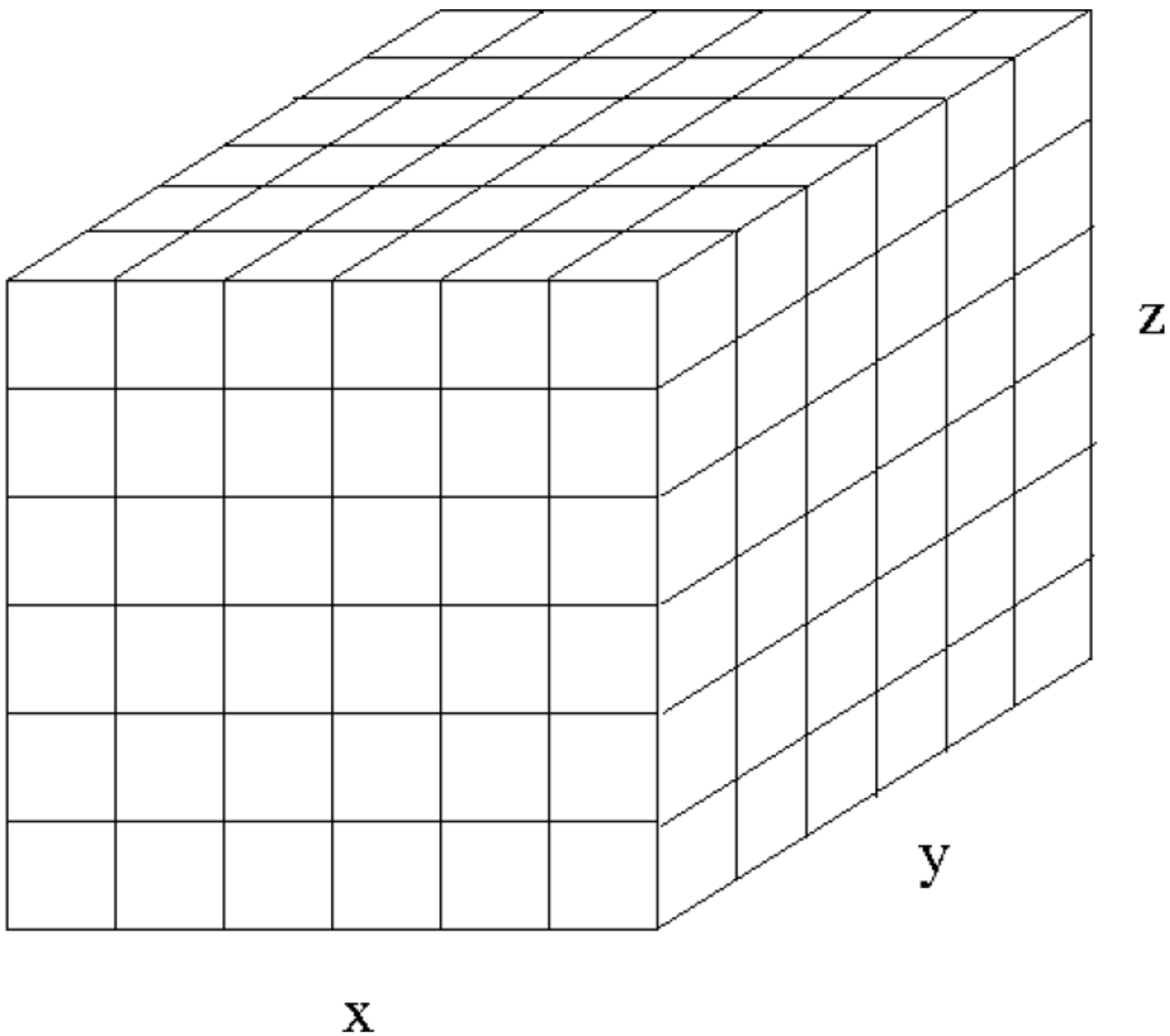
$\mathcal{X} = \{1, 2, 3, 4\}$



degenerate distribution

slide by Erik Sudderth

Joint Distribution

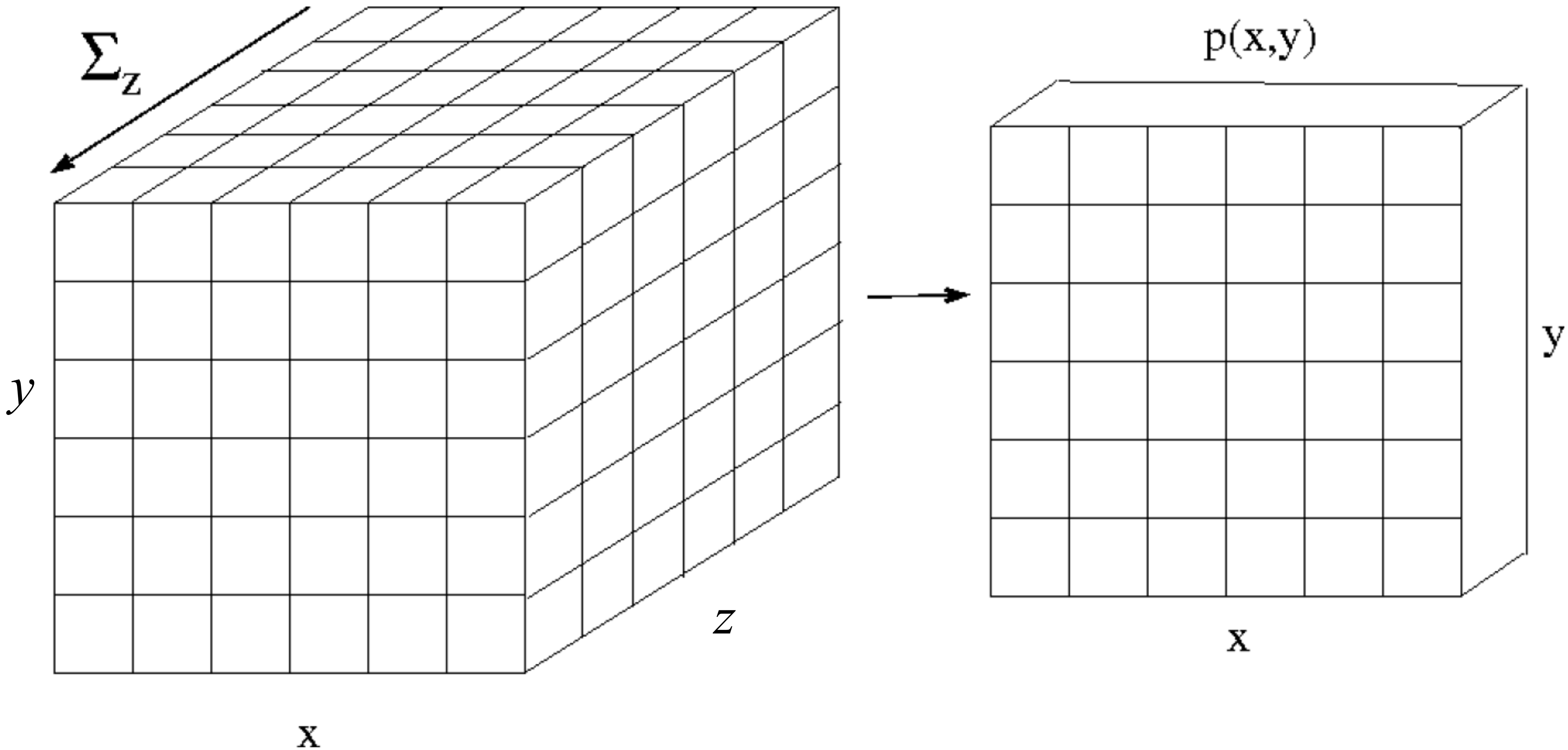


Marginalization

- Marginalization
  - Events: P(A) = P(A and B) + P(A and not B)
  - Random variables $P(X=x)=\sum_{y} P(X=x, Y=y)$

slide by Dhruv Batra

Marginal Distributions



$$p(x, y) = \sum_{z \in \mathcal{Z}} p(x, y, z)$$

$$p(x) = \sum_{y \in \mathcal{Y}} p(x, y)$$

slide by Erik Suddherth

Conditional Probabilities

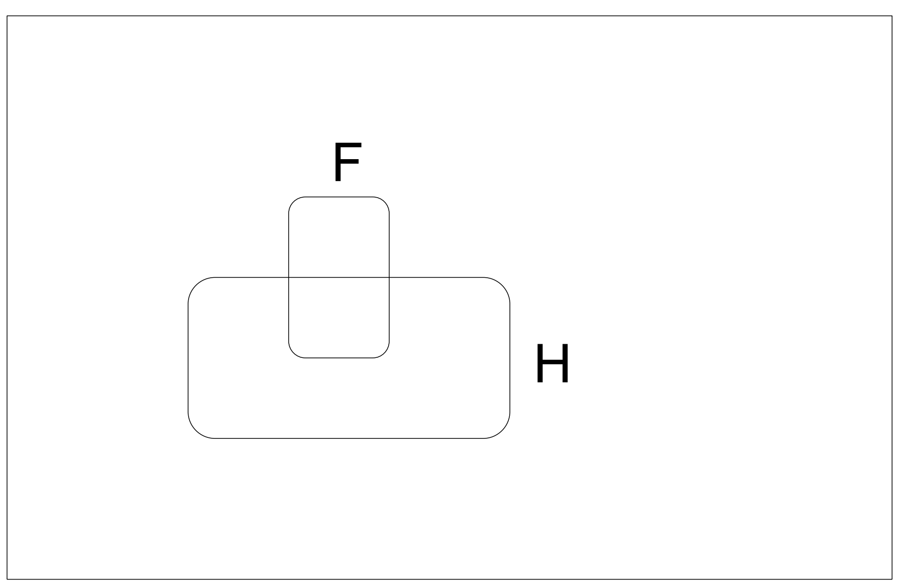
- P(Y=y | X=x)
- What do you believe about Y=y, if I tell you X=x?
- P(Max Verstappen winning the 2024 United States Grand Prix)?
- What if I tell you:
  - He has won the Formula One World Champion title for 2021, 2022, and 2023.
  - He has won the United States Grand Prix 3/8 he has raced there.

slide by Dhruv Batra

Conditional Probabilities

- P(A | B) = In worlds that where B is true,
  fraction where A is true

- Example
  - H: “Have a headache”
  - F: “Coming down with Flu”

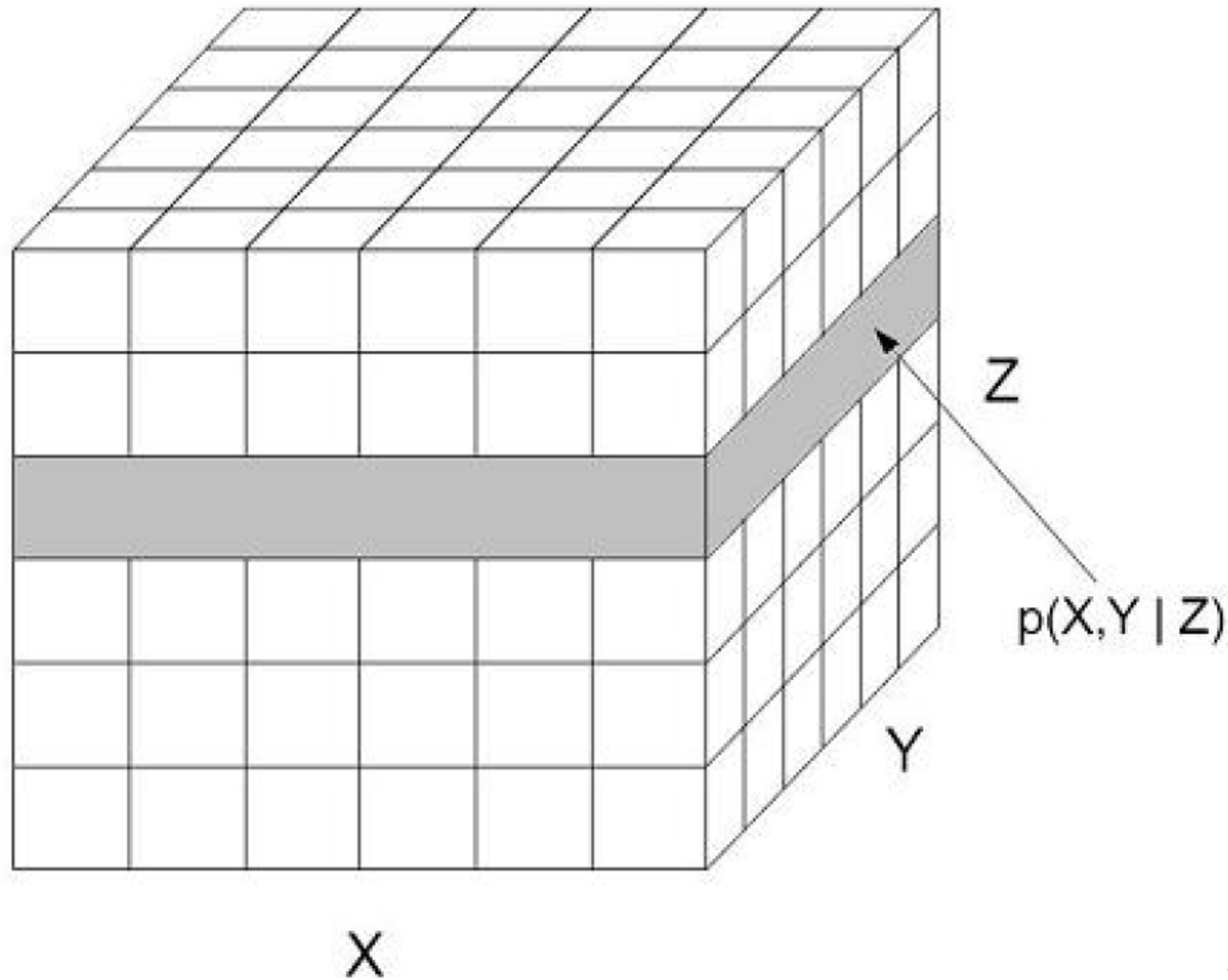


P(H) = 1/10
P(F) = 1/40
P(H|F) = 1/2

Headaches are rare and flu is rarer, but if you re coming down with flu there s a 50-50 chance you ll have a headache.

slide by Dhruv Batra

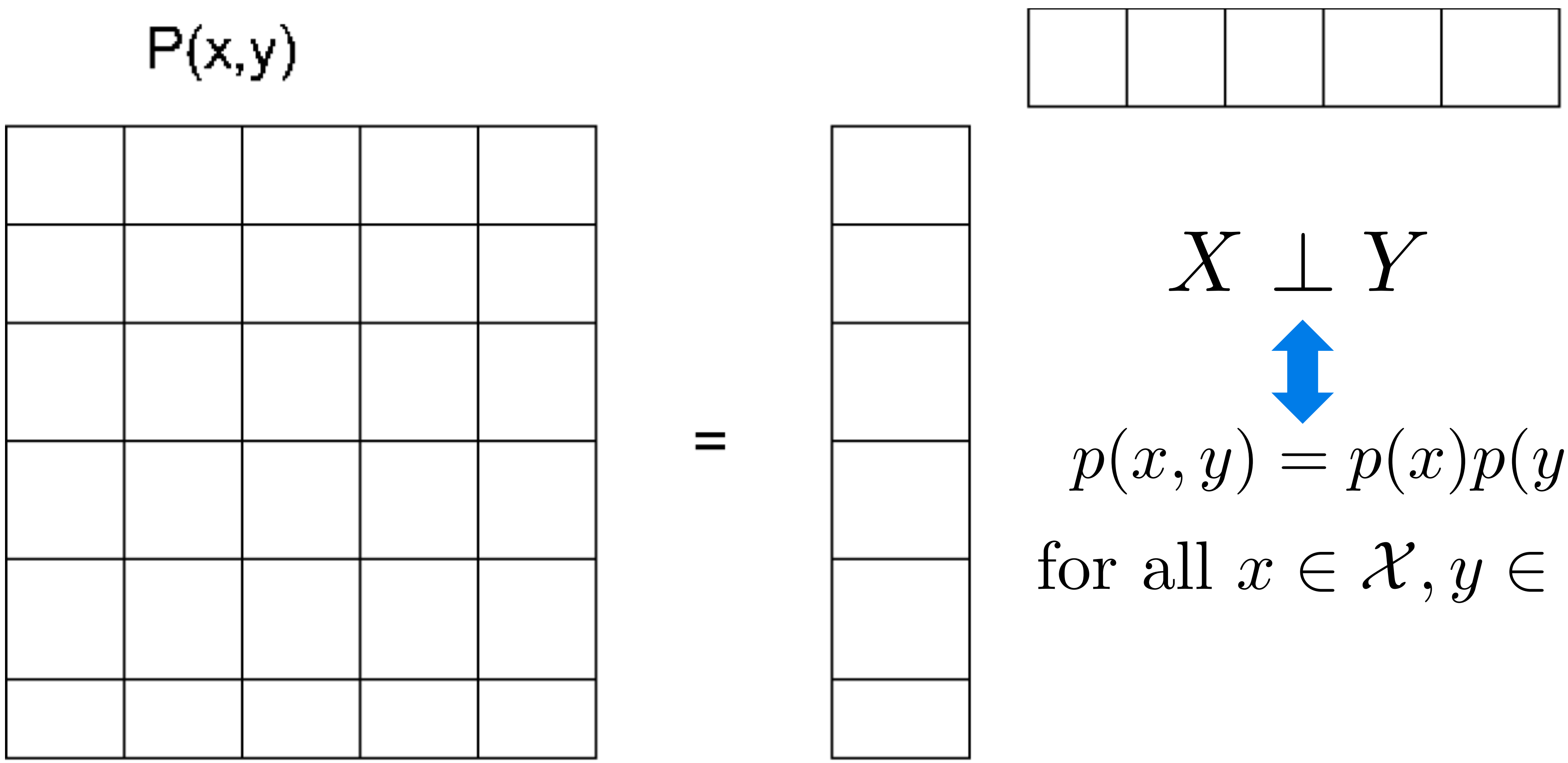
Conditional Distributions



$$p(x, y \mid Z = z) = \frac{p(x, y, z)}{p(z)}$$

slide by Erik Sudderth

Independent Random Variables



$$X \perp Y$$

⇕

$$p(x, y) = p(x)p(y)$$

$$\text{for all } x \in \mathcal{X}, y \in \mathcal{Y}$$

Equivalent conditions on conditional probabilities:

$$p(x \mid Y = y) = p(x) \text{ and } p(y) > 0 \text{ for all } y \in \mathcal{Y}$$

$$p(y \mid X = x) = p(y) \text{ and } p(x) > 0 \text{ for all } x \in \mathcal{X}$$

slide by Erik Sudderth

Bayes Rule (Bayes Theorem)

$$p(x, y) = p(x)p(y \mid x) = p(y)p(x \mid y)$$

$$p(y \mid x) = \frac{p(x, y)}{p(x)} = \frac{p(x \mid y)p(y)}{\sum_{y' \in \mathcal{Y}} p(y')p(x \mid y')}$$

$$\propto p(x \mid y)p(y)$$

- A basic identity from the definition of conditional probability
- Used in ways that have no thing to do with Bayesian statistics!
- Typical application to learning and data analysis:

$Y \longrightarrow$ unknown parameters we would like to infer

$X = x \longrightarrow$ observed data available for learning

$p(y) \longrightarrow$ prior distribution (domain knowledge)

$p(x \mid y) \longrightarrow$ likelihood function (measurement model)

$p(y \mid x) \longrightarrow$ posterior distribution (learned information)

slide by Erik Sudderth

Binary Random Variables

- Bernoulli Distribution: Single toss of a (possibly biased) coin

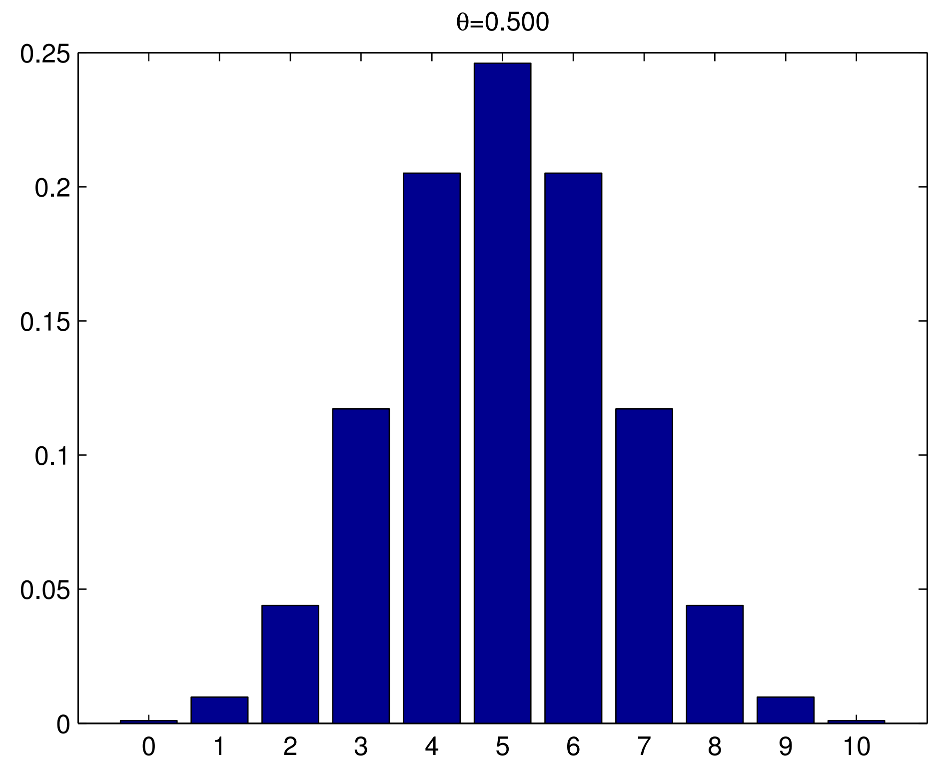
$$\mathcal{X} = \{0, 1\}$$
$$0 \leq \theta \leq 1$$
$$\mathrm{Ber}(x \mid \theta) = \theta^{\delta(x,1)}(1-\theta)^{\delta(x,0)}$$

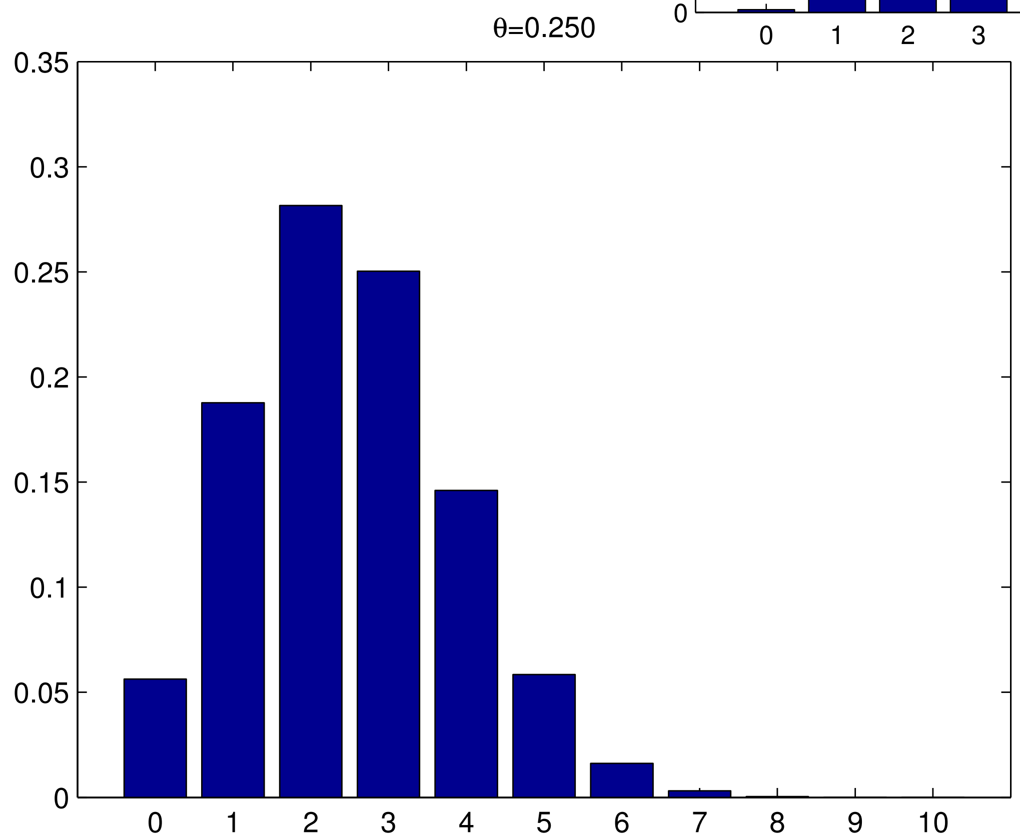
- Binomial Distribution: Toss a single (possibly biased) coin *n* times, and report the number *k* of times it comes up

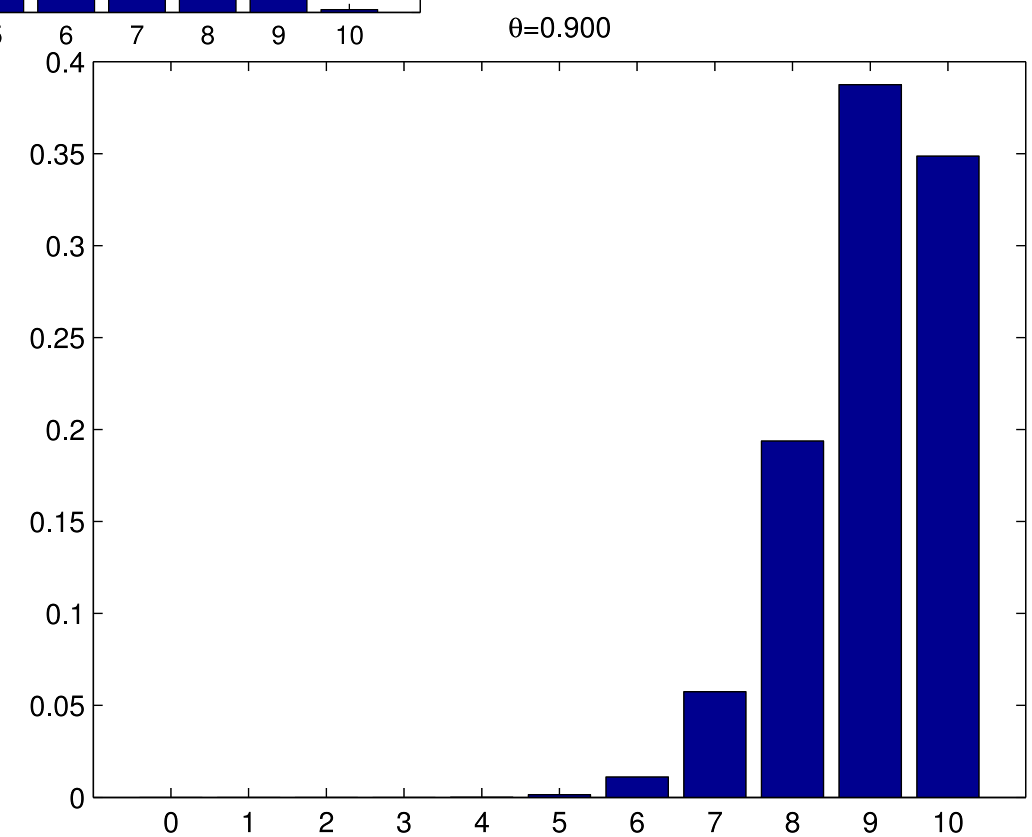
$$\mathcal{K} = \{0, 1, 2, \ldots, n\}$$
$$0 \leq \theta \leq 1$$
$$\mathrm{Bin}(k \mid n, \theta) = \binom{n}{k} \theta^k (1-\theta)^{n-k} \qquad \binom{n}{k} = \frac{n!}{(n-k)!k!}$$

slide by Erik Sudderth

Binomial Distributions







slide by Erik Sudderth

Bean Machine (Sir Francis Galton)

http://en.wikipedia.org/wiki/Bean_machine

Categorical Random Variables

- Multinoulli Distribution: Single roll of a (possibly biased) die

$$\mathcal{X} = \{0,1\}^K, \sum_{k=1}^{K} x_k = 1$$

binary vector encoding

$$\theta = (\theta_1, \theta_2, \ldots, \theta_K), \theta_k \geq 0, \sum_{k=1}^{K} \theta_k = 1$$

$$\mathrm{Cat}(x \mid \theta) = \prod_{k=1}^{K} \theta_k^{x_k}$$

- Multinomial Distribution: Roll a single (possibly biased) die n times, and report the number n_k of each possible outcome

$$\mathrm{Mu}(x \mid n, \theta) = \binom{n}{n_1 \ldots n_K} \prod_{k=1}^{K} \theta_k^{n_k}$$

$$n_k = \sum_{i=1}^{n} x_{ik}$$

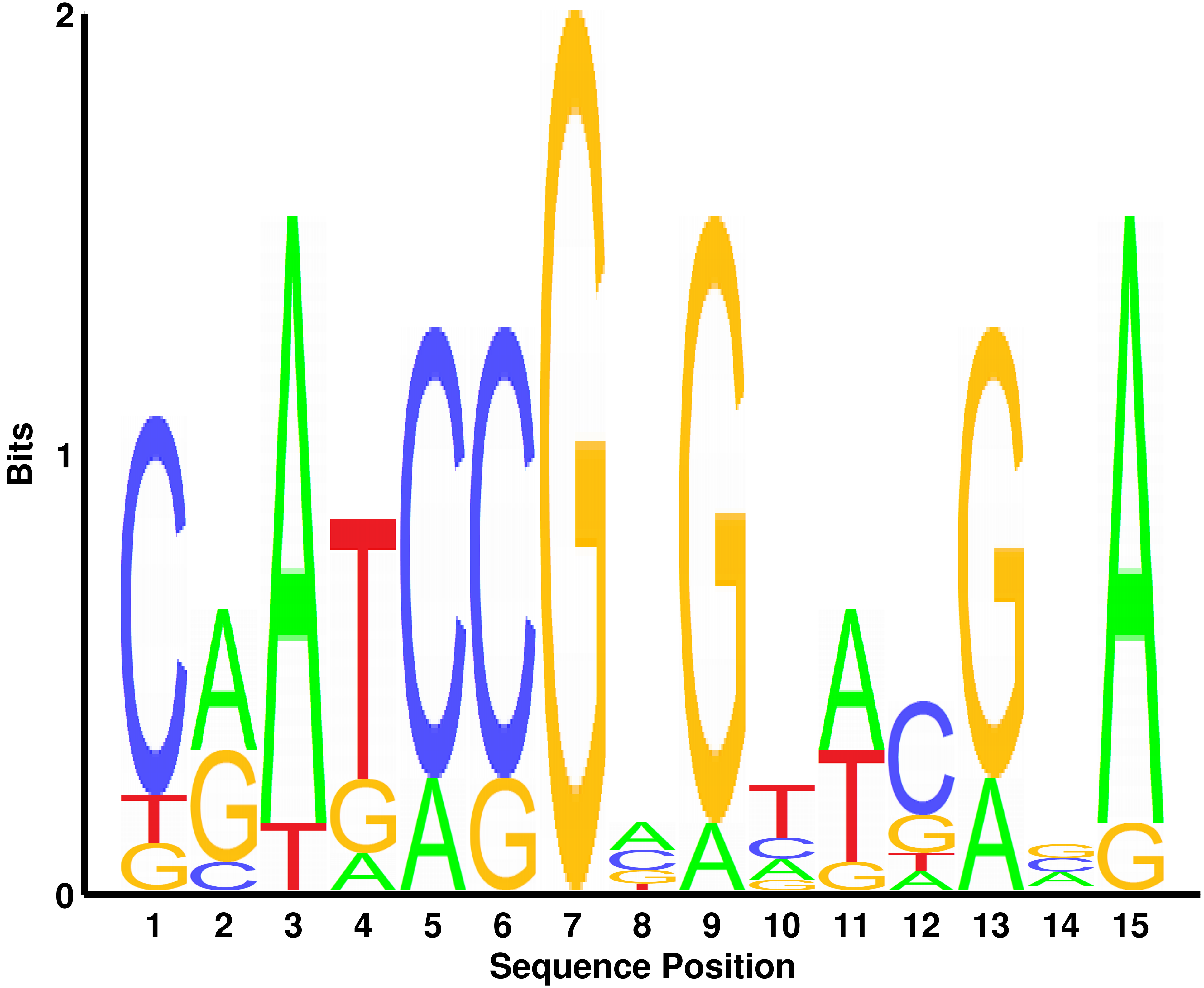
slide by Erik Sudderth

Aligned DNA Sequences

c	g	a	t	a	c	g	g	g	g	t	c	g	a	a
c	a	a	t	c	c	g	a	g	a	t	c	g	c	a
c	a	a	t	c	c	g	t	g	t	t	g	g	g	a
c	a	a	t	c	g	g	c	a	t	g	c	g	g	g
c	g	a	g	c	c	g	c	g	t	a	c	g	a	a
c	a	t	a	c	g	g	a	g	c	a	c	g	a	a
t	a	a	t	c	c	g	g	g	c	a	t	g	t	a
c	g	a	g	c	c	g	a	g	t	a	c	a	g	a
c	c	a	t	c	c	g	c	g	t	a	a	g	c	a
g	g	a	t	a	c	g	a	g	a	t	g	a	c	a

slide by Erik Suddherth

Multinomial Model of DNA



slide by Erik Suddherth

Next Lecture:

Maximum Likelihood Estimation (MLE)