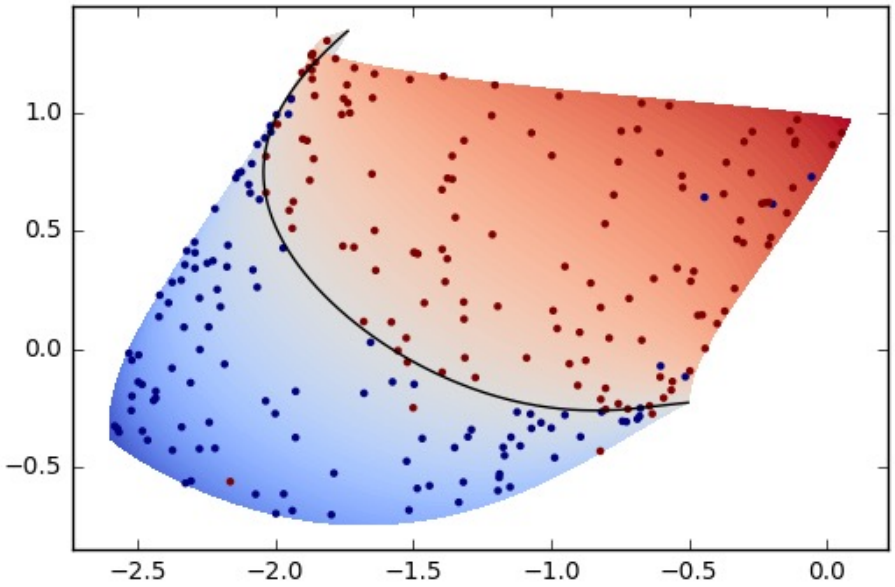
Lecture 9: Probability Review



Gavin Kerrigan

Spring 2023

Some materials courtesy Padhraic Smyth, Alex Ihler, Stephan Mandt.

Announcements

- HW2 due in one week (next Friday)
  - Can already attempt Problem 1 / Problem 2
  - Problem 3 covered by Monday

- No in-person lectures next week (M, W, F)
- Lectures will be recorded and posted on Canvas
  - Check Ed for announcement

- Midterm in 2 weeks (5/5)
  - More details early next week
  - Practice midterm + solutions

- Project details released early next week
  - Start thinking about forming teams (3 people)
  - Will release a spreadsheet to help find teammates

Comparing Classifiers

Probability in Machine Learning

Intro to Logistic Classifiers

Reminders from Last Lecture

- Wrapped up discussion of kNN classifiers
  - More flexible than nearest centroids
  - Requires tuning number of neighbors k

- Time-Space complexity analysis of nearest centroids, kNN
  - Why do we care?
  - Want models that *scale* well to large problems (i.e. big datasets with very many features)

- Curse of dimensionality
  - Amount of ”space” grows exponentially with number of features
  - So we need exponentially more data to “explore” the data space well

Time and Space Complexity of Classifiers

- Variables:
  - n = number of examples
  - d = dimensionality of each feature vector
  - C = number of classes
  - (for kNN, k = number of neighbors)
  - We will assume that n >> k and n >> C

- "Big O" notation: how an algorithm scales in the worst case
  - E.g., $O(n\,d)$ => linear in n and in d
  - E.g., $O(d\,n^2 + C\,n)$ => quadratic in n

- We are interested in
  - Both time and space complexity
  - For both training and prediction phases of classification

Baseline: Majority Classifier

- Always predict the most common label in the dataset
  - (assume label frequencies in train and test are roughly equal)

- What is the error rate of this "classifier"
  - Let maxP = probability (rel. frequency) of the most common label
  - Error of majority classifier = 1 – maxP

- This is not a real classifier, but its useful as a baseline
  - E.g., binary classification, P(Class 1) = 0.95
  - So the error rate of the majority classifier is 5%
  - Any classifier we build should have error rate at least as low as this

Example of a Majority Classifier

Training data with 100 examples:
60 from class 1, 30 from class 2, 10 from class 3

Class probabilities:
P(Class 1) = 0.6, P(Class 2) = 0.3, P(Class 3) = 0.1

Majority class is Class 1, with P(Class 1) = 0.6

Accuracy of Majority Classifier = P(Class 1) = 0.6
Error Rate of Majority Classifier = 1 – P(Class 1) = 0.4

Note: with C equally likely classes
Error Rate of Majority Classifier = 1 – 1/C = (C-1)/C
e.g., 90% for MNIST digits, and 66% for Wine

Classifiers discussed so far…

Majority Classifier: simple baseline, high error

Binned Classifier: simple baseline, impractical for d > 10 or 20

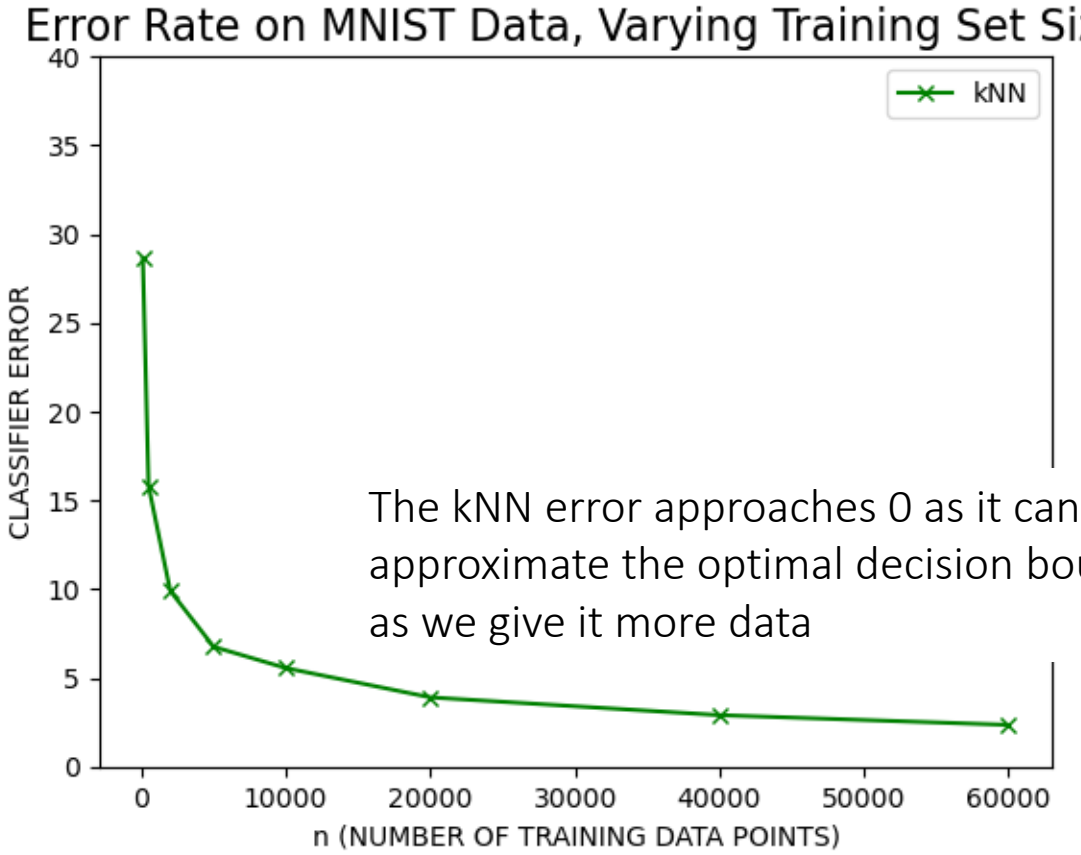
Nearest-Centroid: simple decision boundaries

kNN: can have complex decision boundaries (depends on k)

Comparing Classifiers

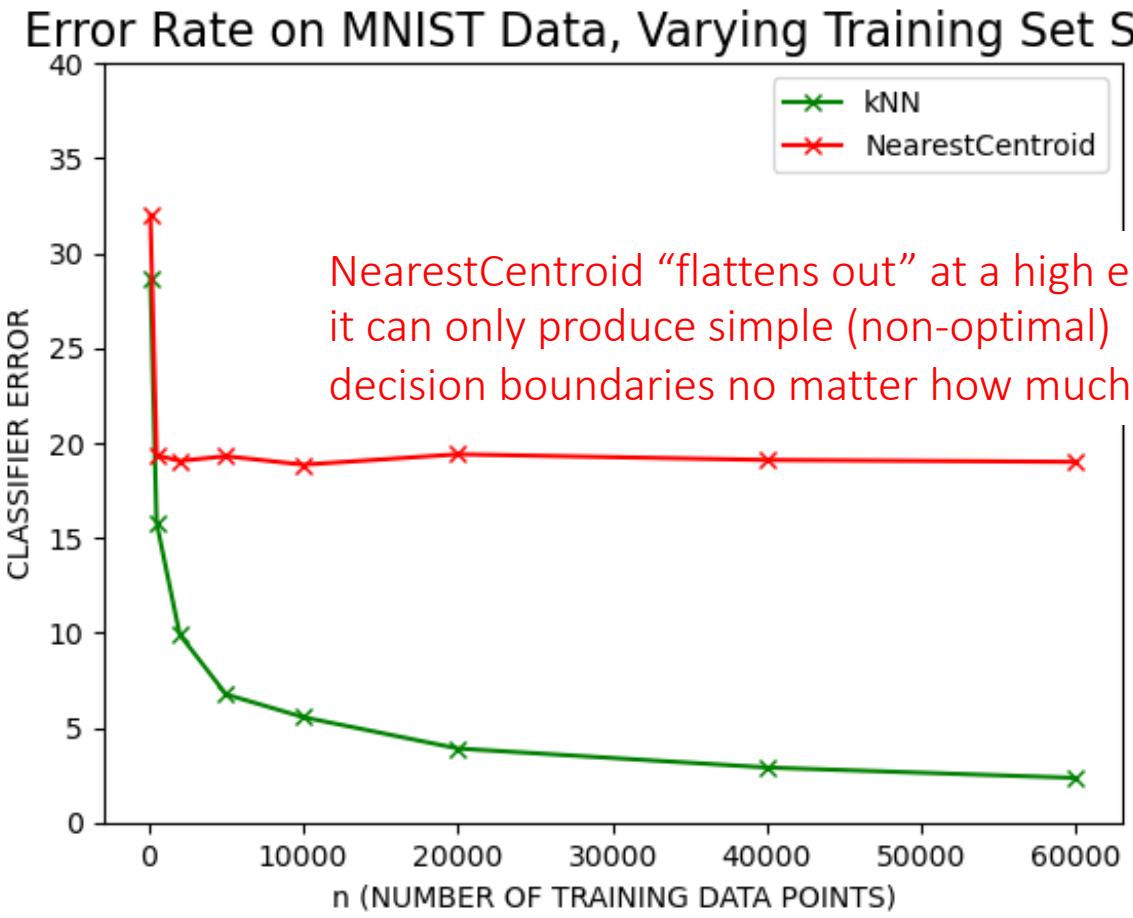
Classifier	Majority	Binned	Nearest Centroid	kNN
Prediction space complexity	O(1)	$O(B^d)$	O(C d)	O(nd)
Prediction time complexity	O(1)	O(1)	O(C d)	O(nd + nk)
Decision boundaries	None	Complex	Simple	Complex for small k
Distance-based?	No	No	Yes	Yes
Error on MNIST	90%	?	~ 20%	~ 3%
Error on Wine (2d)	66%	?	~ 22%	~ 22%

Learning Curve for kNN



The kNN error approaches 0 as it can better approximate the optimal decision boundaries as we give it more data

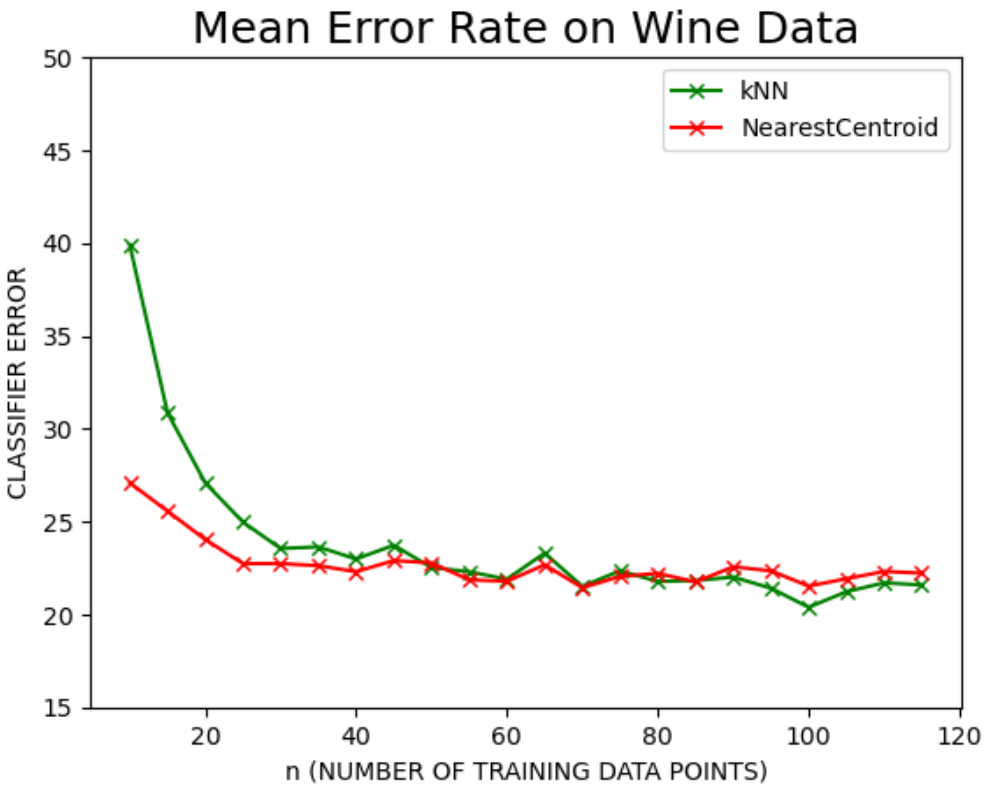
Learning Curve for kNN and NearestCentroid



NearestCentroid "flattens out" at a high error rate:
it can only produce simple (non-optimal)
decision boundaries no matter how much data we give it

On a different problem, the Wine Dataset, both kNN and NearestCentroid "flatten out" as training set increases

With more data we might see than kNN is less biased (closer to optimal) than NC, but we don't have enough data here to really tell.



Results generated as follows:
K = 5 for kNN
For each value of n
Dataset split into n training points and 40 test points
Repeated 100 times (randomly) and average reported

Questions?

Comparing Classifiers

Probability in Machine Learning

Intro to Logistic Classifiers

Why do we need to use probability in ML?

The real world is full of *uncertainty,* due to

- Randomness
- Overwhelming complexity
- Lack of knowledge
- ….

Modeling this uncertainty is critical in many applications

- e.g. classifying whether or not a patient has a disease
- Not enough to predict a label – also should know how certain you are about that label

Why do we need to use probability in ML?

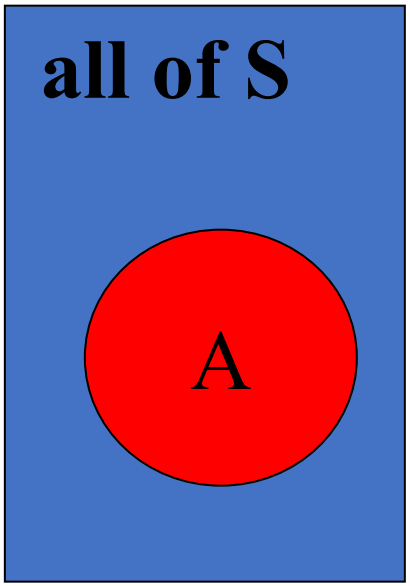
The real world is full of *uncertainty*

Using the language of probability gives us…

- a general quantitative framework for handling uncertainty
- rules for computing and manipulating uncertainty
- tools to build ML algorithms which incorporate randomness

Basic Probability Theory

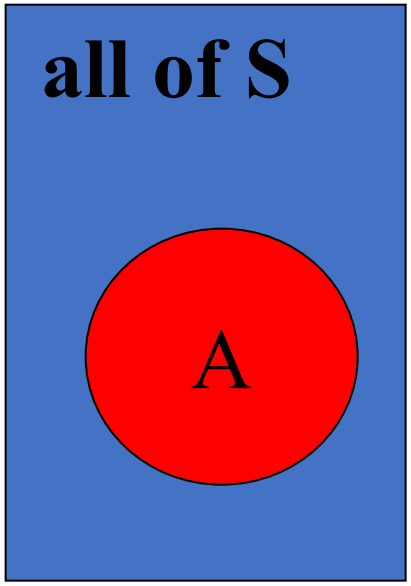
- S is some space of "events"
  - All possible outcomes we're interested in
- An event "A" is a particular subset of S



Example: Flipping a coin 5 times

- S is all possible Head/Tail strings of length 5
- A could be…
  - "HTTHT"
  - "Heads on the first flip"
  - "More heads than tails"

Basic Probability Theory



- S is some space of "events"
  - All possible outcomes we're interested in
- An event "A" is a particular subset of S

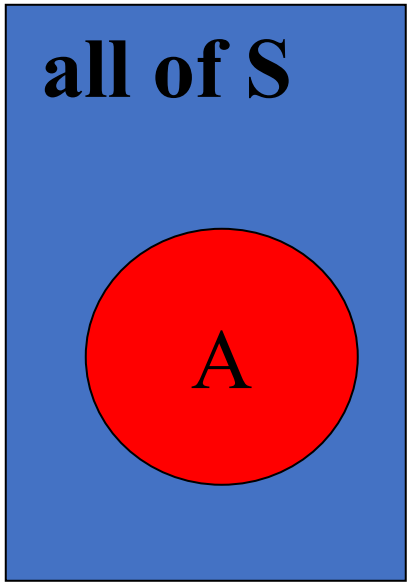
Pr[A] is the probability of seeing the event A

- If I randomly pick an element from S, how likely is it to come from A?

Example:

- S represents all possible outcomes from flipping a coin 5 times
- A represents "Heads on the first flip"
- Pr[A] = 1/2

Basic Probability Theory



- S is some space of "events"
  - All possible outcomes we're interested in
- An event "A" is a particular subset of S

Pr[A] is the probability of seeing the event A

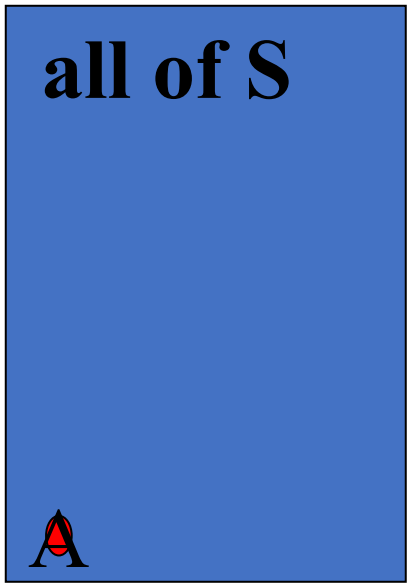
- If I randomly pick an element from S, how likely is it to come from A?

Can think of Pr[A] as the "area" of A in S

Basic Probability Theory

Axioms of probability:

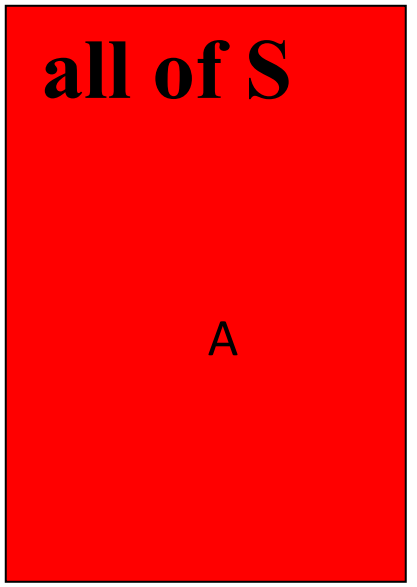
- $0 \leq \Pr[A] \leq 1$
  - Fraction of "area" corresponding to A can't go below zero



Basic Probability Theory

Axioms of probability:

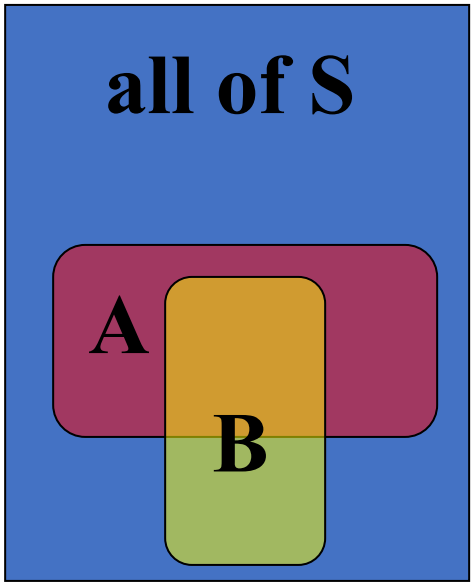
- $0 \leq \Pr[A] \leq 1$
  - Fraction of "area" corresponding to A can't go below zero
  - … or above 1
- $\Pr[S] = 1$
- $\Pr[\{\}] = 0$
  - probability of no event happening is zero



Basic Probability Theory

Axioms of probability:

- $0 \leq \Pr[A] \leq 1$
  - Fraction of "area" corresponding to A can't go below zero
  - … or above 1
- $\Pr[S] = 1$
- $\Pr[\{\}] = 0$
  - probability of no event happening is zero
- $\Pr[A \cup B] = \Pr[A] + \Pr[B] - \Pr[A \cap B]$



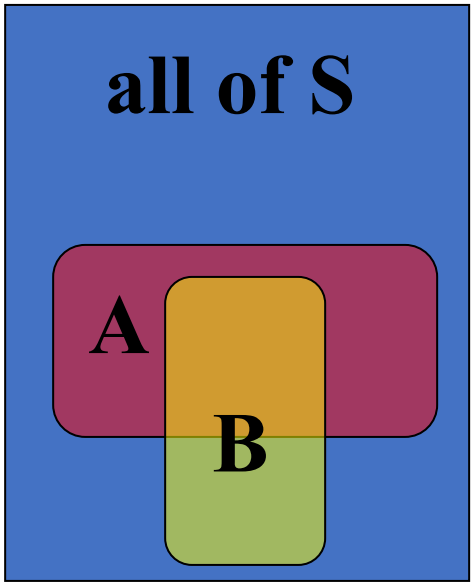
Probability of A or B happening is the "total area" corresponding to A and B

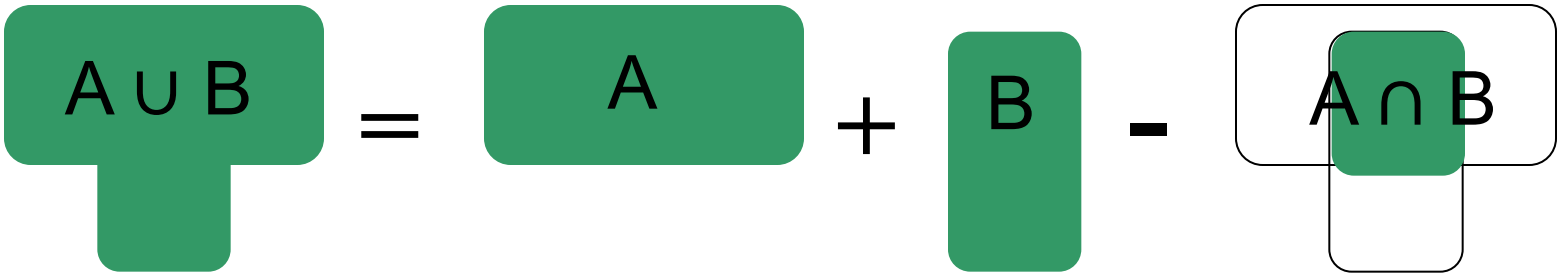
Need to subtract off the probability that A and B happen simultaneously – it is "double counted"

Basic Probability Theory

Axioms of probability:

- $0 \leq \Pr[A] \leq 1$
  - Fraction of "area" corresponding to A can't go below zero
  - … or above 1
- $\Pr[S] = 1$
- $\Pr[\{\}] = 0$
  - probability of no event happening is zero
- $\Pr[A \cup B] = \Pr[A] + \Pr[B] - \Pr[A \cap B]$





Questions?

Discrete Random Variables

A **discrete random variable** X:

- represents the outcome of an experiment
- Can take values in some finite set
- This finite set is *disjoint* and *exhaustive,* i.e. X takes exactly one value from this set

Example:

- S = {“Heads”, “Tails”} is the sample space for a coin flip
- X is a random variable representing the outcome of the flip

Discrete Random Variables

A **discrete random variable** X:

- represents the outcome of an experiment
- Can take values in some finite set

Generally X takes values in S = {a_1, a_2, …, a_d}

Pr[X = a_i] is defined for each value in S and satisfies:

$$0 \leq \Pr[X = a_i] \leq 1 \qquad \sum_{i=1}^{d} \Pr[X = a_i] = 1$$

The probability of *any* subset A of S is:

$$\Pr[X \in A \subseteq S] = \sum_{a_i \in A} Pr[X = a_i]$$

Discrete Random Variables

A **discrete random variable** X:

- represents the outcome of an experiment
- Can take values in some finite set

Generally X takes values in $S = \{a_1, a_2, \ldots, a_d\}$

$Pr[X = a_i]$ is defined for each value in S

- This is the "probability mass function" (pmf) associated with the random variable X
- Tells you how likely each outcome is

Examples of Random Variables

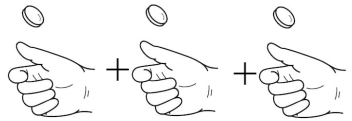
Bernoulli RV: "coin toss"

$X \in \{0, 1\}$ $\quad \Pr[X = 1] = \rho$

$\Pr[X = 0] = 1 - \rho$

Binomial(p,n): toss the coin n times & count

$$Y = \sum_{i=1}^{n} X_i$$



Discrete(d) or Categorical(d): d-sided die roll

$X \in \{1, \ldots, d\}$ $\quad \Pr[X = 1] = \rho_1$

$\vdots$

$\Pr[X = d] = \rho_d$

$$\sum_i \rho_i = 1$$

Multinomial(d,n): roll the die n times and count outcomes

$$Y = [\#\{X_i = 1\}, \ldots, \#\{X_i = d\}]$$

PMF for Bernoulli Distribution

- Discrete random variables
  - Typically represent PMF as a table
  - Sometimes useful to express as a function
    - Later: take derivatives, fit to data, etc.

- Ex: Bernoulli, X = 0 or 1

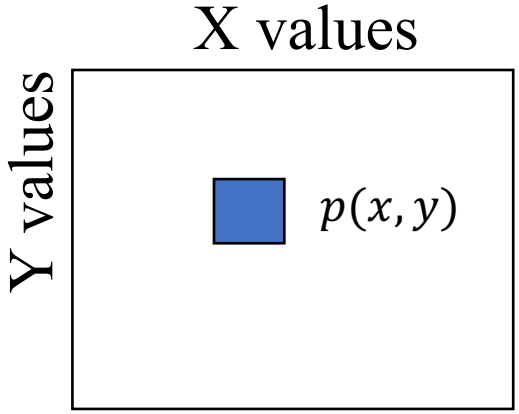
$$p(x) = (\rho)^x \cdot (1-\rho)^{(1-x)}$$

$$= \begin{cases} (\rho)^1 \cdot (1-\rho)^0 = \rho & \text{if } x = 1 \\ (\rho)^0 \cdot (1-\rho)^1 = (1-\rho) & \text{if } x = 0 \end{cases}$$

Joint Distributions

Joint distributions Pr[X=x, Y=y] represent the probability of two random variables taking particular values simultaneously

- Write a probability mass function for x and y together
- Can express as table of joint values
- Sum over all elements in table sums to one.



- Law of total probability: $p(x) = \sum_y p(x, y)$
  - *Some* value of y must have occurred
  - Since we don't know which one, we sum over all of them
  - E.g., for binary y:
- $p(x) = p(x, y = 0) + p(x, y = 1)$

Joint Distributions

- Often, we want to reason about multiple variables
- Example: dentist
  - T: have a toothache
  - D: dental probe catches
  - C: have a cavity

- Joint distribution
  - Assigns each event (T=t, D=d, C=c) a probability
  - Probabilities sum to 1.0

T	D	C	P(T,D,C)
0	0	0	0.576
0	0	1	0.008
0	1	0	0.144
0	1	1	0.072
1	0	0	0.064
1	0	1	0.012
1	1	0	0.016
1	1	1	0.108

Example from Russell & Norvig

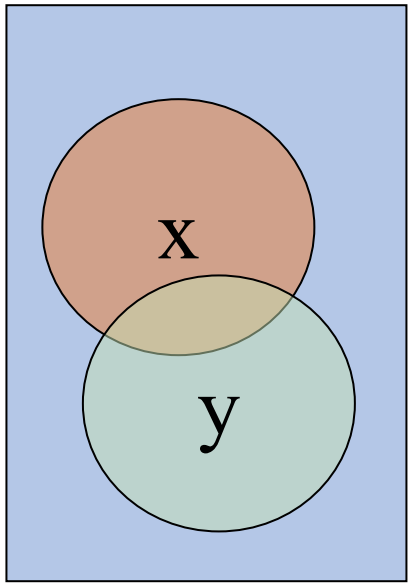
Law of total probability:

$$p(C = 1) = \sum_{t,p} P(T = t, D = d, C = 1)$$

= 0.008 + 0.072 + 0.012 + 0.108 = 0.20

- *Some* value of (T,D) must occur; values disjoint
- "Marginal probability" of C; "marginalize" or "sum over" T,D

Conditional Probability



Conditional probabilities represent the distribution of a random variable *given that* another random variable takes a particular value

- Chain rule: $p(X = x, Y = y) = p(X = x)p(Y = y|X = x)$
  - E.g., p(rain, cloudy) = p(rain|cloudy)p(cloudy)
  - p(X=x,Y=y) : probability that both X=x and Y=y
  - p(X=x) : probability that X=x (regardless of Y)
  - P(Y=y|X=x): probability that Y=y given X=x has occured
  - If p(X) > 0 : $p(Y|X) = \frac{p(X,Y)}{p(X)}$

- More generally: $p(X, Y, Z) = p(X)\ p(Y|X)\ p(Z|X, Y)$

$$p(W, X, Y, Z) = p(X)\ p(Y|X)\ p(Z|X, Y)\ p(W|X, Y, Z)$$

The Effect of Evidence

- Example: dentist
  - T: have a toothache
  - D: dental probe catches
  - C: have a cavity

- Recall p(C=1) = 0.20
- Suppose we observe D=0, T=0. Does this change our belief in C=1?

$$p(C=1|D=0,T=0) = \frac{p(C=1,D=0,T=0)}{p(D=0,T=0)}$$

$$= \frac{0.008}{0.576+0.008} = 0.012$$

- Observe D=1, T=1?

$$= \frac{0.108}{0.016+0.108} = 0.871$$

Called *posterior probabilities*

T	D	C	P(T,D,C)
0	0	0	0.576
0	0	1	0.008
0	1	0	0.144
0	1	1	0.072
1	0	0	0.064
1	0	1	0.012
1	1	0	0.016
1	1	1	0.108

The Effect of Evidence

- Example: dentist
  - T: have a toothache
  - D: dental probe catches
  - C: have a cavity

T	D	C	P(T,D,C)
0	0	0	0.576
0	0	1	0.008
0	1	0	0.144
0	1	1	0.072
1	0	0	0.064
1	0	1	0.012
1	1	0	0.016
1	1	1	0.108

$$p(C = 1|T = 1) = \frac{p(C = 1, T = 1)}{p(T = 1)}$$

$$= \frac{0.012 + 0.108}{0.064 + 0.012 + 0.016 + 0.108} = 0.60$$

$$p(T = 1) = 0.20$$

Called the *probability of evidence*

Bayes' Rule

Rev. Thomas Bayes
(1701-1761)

From the chain rule, we know

$$p(Y|X)\ p(X) = p(X,Y) = p(X|Y)\ p(Y)$$

- Lets us calculate posterior given evidence

$$\Rightarrow \quad p(Y|X) = \frac{p(X|Y)\ p(Y)}{p(X)} \quad \text{"Bayes' rule"}$$

- Example: flu
  - P(F), P(H|F)

F	P(F)
0	0.95
1	0.05

F	H	P(H|F)
0	0	0.80
0	1	0.20
1	0	0.50
1	1	0.50

  - $P(F = 1 \mid H = 1)$

$$= \frac{p(H=1|F=1)p(F=1)}{p(H=1|F=1)p(F=1) + p(H=1|F=0)p(F=0)}$$

$$= \frac{0.50 * 0.05}{0.50 * 0.05 \ + \ 0.20 * 0.95} = 0.116$$

Continuous Random Variables

We have now seen how to work with *discrete* random variables

What about random variables that take continuous values?

Example: X is a random variable representing the temperature on a particular day

- Expressions like Pr[X = x] don't make sense anymore
- e.g. how could we sum over all the values of x and get 1, if there are continuously many values for x?

Continuous Random Variables

The cumulative distribution function is well-defined for continuous R.V.s:

$$\mathrm{F(x)} = \Pr[X \leq x]$$

The probability density function (pdf) is the derivative

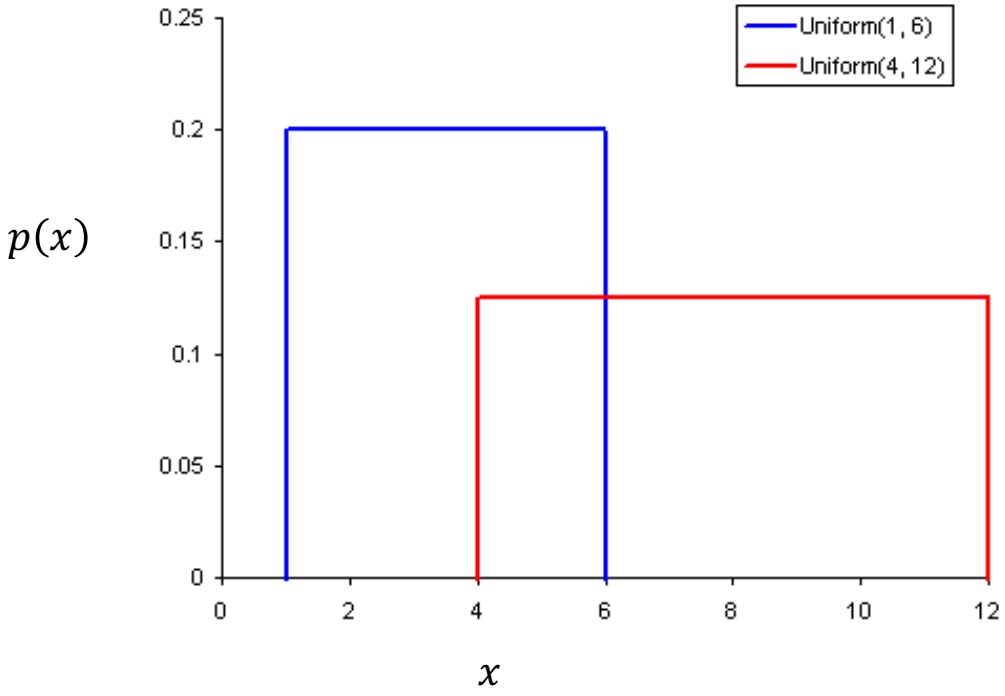
$$\mathrm{p(x)} = \frac{d}{dx} F(\mathrm{x})$$

- Intuitively similar to the probability that X=x… but not quite the same
- e.g. $0 \leq \Pr[X \leq x] \leq 1$
  - but $0 \leq p(x)$ can take values larger than one
- You can *integrate* the pdf to get probabilities

Continuous Random Variables

Examples of PDFs:

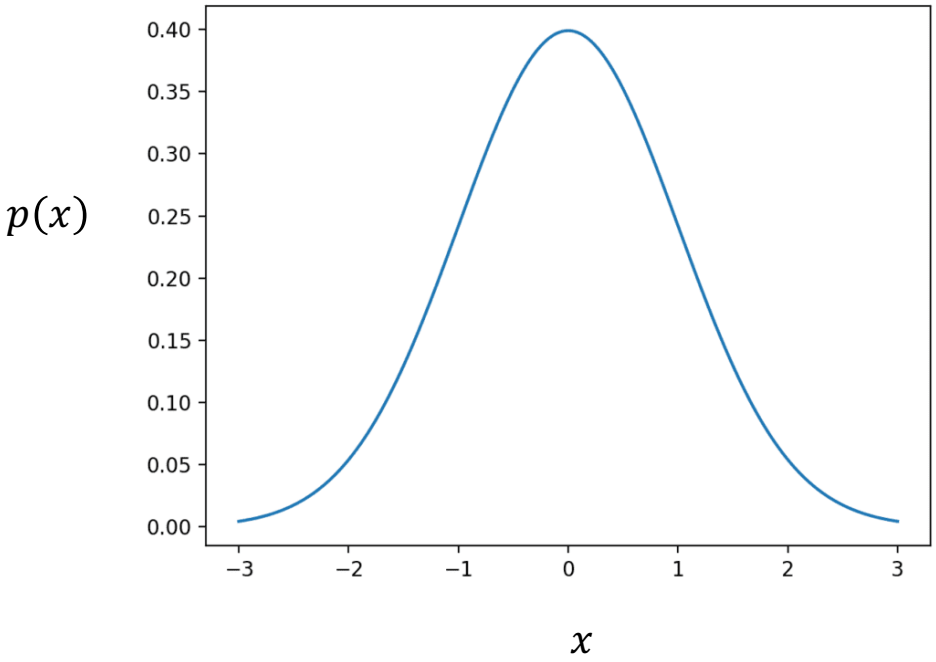
X has Uniform(a, b) pdf if it takes values between a, b with equal probability



Continuous Random Variables

Examples of PDFs:

Gaussian (or Normal) distribution is classic bell-curve:

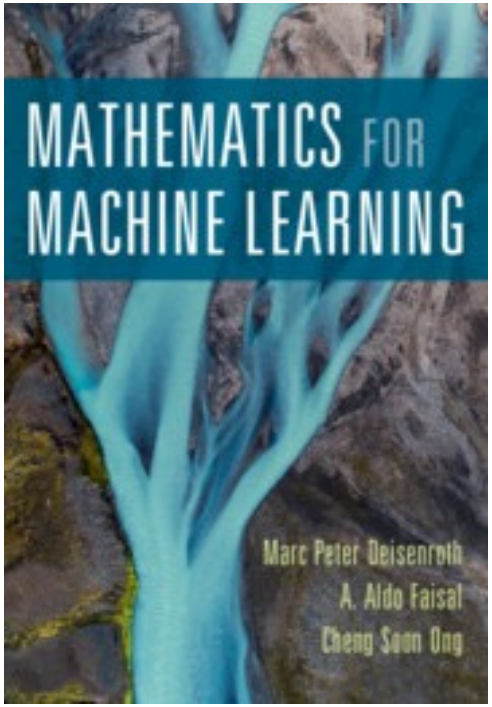


$$p(x) = \mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2)}{2\sigma^2}\right)$$

Additional Resource

Resource for more details on probability in ML

- Linked in Canvas page; freely available
- Good knowledge of probability is one of the most useful skills in more advanced ML



Questions?

Comparing Classifiers

Probability in Machine Learning

Intro to Logistic Classifiers

Using Probabilities for Classification

Say we want to predict a binary label y from a feature vector **x**

How can we model the uncertainty in our prediction?

That is, we are interested in the *conditional probabilities*

$$p(y = 1 \mid \boldsymbol{x})$$

$$p(y = 0 \mid \boldsymbol{x}) = 1 - p(y = 0 \mid \boldsymbol{x})$$

- Sufficient to model $p(y = 1 \mid x)$
- Classifiers we have seen so far (kNN, nearest centroids) only produce labels, not probabilities
- We need a new model!

Logistic Classifier Intuition

Key idea:

for every feature vector **x**, produce a numerical *score* $f(\boldsymbol{x} \mid \theta)$

- Here θ represents some parameters of our model
- High values for $f(\boldsymbol{x} \mid \theta)$ indicate y=1 is more likely
- Low values for $f(\boldsymbol{x} \mid \theta)$ indicate y=0 is more likely
- $f(\boldsymbol{x} \mid \theta)$ can take arbitrary real values

To model $p(y = 1 \mid \boldsymbol{x})$, we then need to turn our score into a probability

- Recall probabilities are between 0 and 1
- So we will need to somehow transform our score into this range

Logistic Classifier Intuition

So, we have several questions:

- How should we produce a real-valued score $f(\boldsymbol{x} \mid \theta)$?
  - Any guesses?
  - Score could be a *linear model*: $f(\boldsymbol{x} \mid \theta) = \theta_0 + \theta_1 x_1 + \cdots + \theta_d x_d$
- How should we turn this score into a probability (i.e. a number between 0 and 1)?
  - Many possible choices here
  - We will use the *sigmoid* (or *logistic*) function
- How can we fit this model to data?
  - We will derive the *cross-entropy* loss function, which can be minimized with gradient descent

Summary and Wrapup

- Comparing classifiers
  - Not enough to think about just accuracy
  - Also need to consider e.g.
    - training time
    - memory requirements
    - How model performance scales with amount of data

- Review of probability for ML
  - Discrete random variables and PMFs
  - Joint and conditional distributions

- Intro to Logistic Classifiers
  - More next week

Next Lecture

- Recorded & posted to canvas – no in-person lecture Monday
- Details on midterm & project
- More on logistic classifiers

Questions?
(Outside after lecture)