;)

Hey there!

- Jeppe Nørregaard
- PostDoc
- Works with Leon on misinformation research
- All our work is in NLP and majority uses Deep Learning

- Today's content is on Teams: *day_2*

Today's Deal

Lecture 2

Machine Learning Introduction

1. Machine Learning - what is it and what is it not?
2. Bayes theorem
3. Naive Bayes
4. Machine Learning Theory
5. Perceptron

Machine Learning

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*
   - We want computer to do things

Machine Learning

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*
    - We want computer to do things
    - We don't have time (or tools) to explicitly program them for everything

Machine Learning

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*
   - We want computer to do things
   - We don't have time (or tools) to explicitly program them for everything
   - So we "teach" them

Machine Learning

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*
   - We want computer to do things
   - We don't have time (or tools) to explicitly program them for everything
   - So we "teach" them
2. Could be considered "Empirical Computer Science" - perhaps amongst others

Machine Learning

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*
   - We want computer to do things
   - We don't have time (or tools) to explicitly program them for everything
   - So we "teach" them
2. Could be considered "Empirical Computer Science" - perhaps amongst others
   - In some parts of computer science we prove results

Machine Learning

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*
    - We want computer to do things
    - We don't have time (or tools) to explicitly program them for everything
    - So we "teach" them
2. Could be considered "Empirical Computer Science" - perhaps amongst others
    - In some parts of computer science we prove results
    - Machine learning is extremely empirical - we try things and see if they work

Machine Learning

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*
   - We want computer to do things
   - We don't have time (or tools) to explicitly program them for everything
   - So we "teach" them
2. Could be considered "Empirical Computer Science" - perhaps amongst others
   - In some parts of computer science we prove results
   - Machine learning is extremely empirical - we try things and see if they work
   - You will see how this is the case later

Machine Learning

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*
    - We want computer to do things
    - We don't have time (or tools) to explicitly program them for everything
    - So we "teach" them
2. Could be considered "Empirical Computer Science" - perhaps amongst others
    - In some parts of computer science we prove results
    - Machine learning is extremely empirical - we try things and see if they work
    - You will see how this is the case later
3. Machine learning models are not scientific models (yet anyways) and not statistical models

Machine Learning

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*
   - We want computer to do things
   - We don't have time (or tools) to explicitly program them for everything
   - So we "teach" them
2. Could be considered "Empirical Computer Science" - perhaps amongst others
   - In some parts of computer science we prove results
   - Machine learning is extremely empirical - we try things and see if they work
   - You will see how this is the case later
3. Machine learning models are not scientific models (yet anyways) and not statistical models
   - In statistics we statistically reject a null-hypothesis, in order to verify a hypothesis/model

Machine Learning

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*
    - We want computer to do things
    - We don't have time (or tools) to explicitly program them for everything
    - So we "teach" them
2. Could be considered "Empirical Computer Science" - perhaps amongst others
    - In some parts of computer science we prove results
    - Machine learning is extremely empirical - we try things and see if they work
    - You will see how this is the case later
3. Machine learning models are not scientific models (yet anyways) and not statistical models
    - In statistics we statistically reject a null-hypothesis, in order to verify a hypothesis/model
    - In science we attempt to find a correct, causal model

Machine Learning

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*
   - We want computer to do things
   - We don't have time (or tools) to explicitly program them for everything
   - So we "teach" them
2. Could be considered "Empirical Computer Science" - perhaps amongst others
   - In some parts of computer science we prove results
   - Machine learning is extremely empirical - we try things and see if they work
   - You will see how this is the case later
3. Machine learning models are not scientific models (yet anyways) and not statistical models
   - In statistics we statistically reject a null-hypothesis, in order to verify a hypothesis/model
   - In science we attempt to find a correct, causal model
   - In machine learning we knowingly pick a *wrong* model and fit it until it is *useful*

Machine Learning

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*
   - We want computer to do things
   - We don't have time (or tools) to explicitly program them for everything
   - So we "teach" them
2. Could be considered "Empirical Computer Science" - perhaps amongst others
   - In some parts of computer science we prove results
   - Machine learning is extremely empirical - we try things and see if they work
   - You will see how this is the case later
3. Machine learning models are not scientific models (yet anyways) and not statistical models
   - In statistics we statistically reject a null-hypothesis, in order to verify a hypothesis/model
   - In science we attempt to find a correct, causal model
   - In machine learning we knowingly pick a *wrong* model and fit it until it is *useful*
   - The *usefulness* of the model determines whether we keep it or not

Machine Learning

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*
   - We want computer to do things
   - We don't have time (or tools) to explicitly program them for everything
   - So we "teach" them
2. Could be considered "Empirical Computer Science" - perhaps amongst others
   - In some parts of computer science we prove results
   - Machine learning is extremely empirical - we try things and see if they work
   - You will see how this is the case later
3. Machine learning models are not scientific models (yet anyways) and not statistical models
   - In statistics we statistically reject a null-hypothesis, in order to verify a hypothesis/model
   - In science we attempt to find a correct, causal model
   - In machine learning we knowingly pick a *wrong* model and fit it until it is *useful*
   - The *usefulness* of the model determines whether we keep it or not
4. In machine learning we pick a huge class of functions (that are all wrong) and select the best one

Machine Learning

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*
    - We want computer to do things
    - We don't have time (or tools) to explicitly program them for everything
    - So we "teach" them
2. Could be considered "Empirical Computer Science" - perhaps amongst others
    - In some parts of computer science we prove results
    - Machine learning is extremely empirical - we try things and see if they work
    - You will see how this is the case later
3. Machine learning models are not scientific models (yet anyways) and not statistical models
    - In statistics we statistically reject a null-hypothesis, in order to verify a hypothesis/model
    - In science we attempt to find a correct, causal model
    - In machine learning we knowingly pick a *wrong* model and fit it until it is *useful*
    - The *usefulness* of the model determines whether we keep it or not
4. In machine learning we pick a huge class of functions (that are all wrong) and select the best one
5. Is machine learning glorified curve fitting?

What is machine learning?

1. Common definition: *Subfield of computer science that gives computers the ability to learn without being explicitly programmed*
   - We want computer to do things
   - We don't have time (or tools) to explicitly program them for everything
   - So we "teach" them
2. Could be considered "Empirical Computer Science" - perhaps amongst others
   - In some parts of computer science we prove results
   - Machine learning is extremely empirical - we try things and see if they work
   - You will see how this is the case later
3. Machine learning models are not scientific models (yet anyways) and not statistical models
   - In statistics we statistically reject a null-hypothesis, in order to verify a hypothesis/model
   - In science we attempt to find a correct, causal model
   - In machine learning we knowingly pick a *wrong* model and fit it until it is *useful*
   - The *usefulness* of the model determines whether we keep it or not
4. In machine learning we pick a huge class of functions (that are all wrong) and select the best one
5. Is machine learning glorified curve fitting?
   - Maybe, but it is useful, so who cares?

Machine Learning

- Deep Learning is a big subfield of Machine Learning

Machine Learning

- Deep Learning is a big subfield of Machine Learning
- We need you to get up to date with Machine Learning really quick, so we can get to Deep Learning :)

Concepts in Machine Learning

Concepts in Machine Learning

x, x_i	Sample / observation / instance	a single item of data

Concepts in Machine Learning

x, x_i	Sample / observation / instance	a single item of data
$x_j, x_{i,j}$	Features / descriptors	the information provided in the samples

Concepts in Machine Learning

x, x_i	Sample / observation / instance	a single item of data
$x_j, x_{i,j}$	Features / descriptors	the information provided in the samples
t, t_i	Targets	a special descriptor of the samples, which we want to predict

Concepts in Machine Learning

x, x_i	Sample / observation / instance	a single item of data
$\mathrm{x}_j, \mathrm{x}_{i,j}$	Features / descriptors	the information provided in the samples
t, t_i	Targets	a special descriptor of the samples, which we want to predict
$\mathcal{D}$	Dataset	a collection of samples. $\mathcal{D} = \left\{ (\mathrm{x}_i, \mathrm{t}_i) \ : \ i \in \mathbb{Z}, 0 < i < N \right\}$

Concepts in Machine Learning

x, x_i	Sample / observation / instance	a single item of data
$\mathrm{x}_j, \mathrm{x}_{i,j}$	Features / descriptors	the information provided in the samples
t, t_i	Targets	a special descriptor of the samples, which we want to predict
$\mathcal{D}$	Dataset	a collection of samples. $\mathcal{D} = \left\{(\mathrm{x}_i, \mathrm{t}_i) \ : \ i \in \mathbb{Z}, 0 < i < N\right\}$
$\mathcal{M}$	Model	a computational system that can predict something for us, based on features

Concepts in Machine Learning

x, x_i	Sample / observation / instance	a single item of data
$\mathrm{x}_j, \mathrm{x}_{i,j}$	Features / descriptors	the information provided in the samples
t, t_i	Targets	a special descriptor of the samples, which we want to predict
$\mathcal{D}$	Dataset	a collection of samples. $\mathcal{D} = \left\{(\mathrm{x}_i, \mathrm{t}_i) \ : \ i \in \mathbb{Z}, 0 < i < N\right\}$
$\mathcal{M}$	Model	a computational system that can predict something for us, based on features
y, y_i	Prediction	The prediction from model $\mathcal{M}$ made on sample x_i

Concepts in Machine Learning

x, x_i	Sample / observation / instance	a single item of data
$x_j, x_{i,j}$	Features / descriptors	the information provided in the samples
t, t_i	Targets	a special descriptor of the samples, which we want to predict
$\mathcal{D}$	Dataset	a collection of samples. $\mathcal{D} = \left\{(x_i, t_i) \ : \ i \in \mathbb{Z}, 0 < i < N\right\}$
$\mathcal{M}$	Model	a computational system that can predict something for us, based on features
y, y_i	Prediction	The prediction from model $\mathcal{M}$ made on sample x_i
$\ell(\mathcal{M}, \mathcal{D})$	Loss function	an evaluation of the performance of the model which we wish to *minimize*

Machine Learning

Supervised Learning

- We have a dataset of sample sample-target-pairs: $(x, t) \in \mathcal{D}$

Supervised Learning

- We have a dataset of sample sample-target-pairs: $(\mathsf{x}, \mathsf{t}) \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$, so that the expected loss on a prediction is low

$$\ell(\mathcal{M}, \mathcal{D}) = \mathbb{E}\left[\ell(\mathsf{y}, \mathsf{t})\right] = \mathbb{E}_{\mathsf{x},\mathsf{t}}\left[\ell(\mathcal{M}(\mathsf{x}), \mathsf{t})\right] \tag{1}$$

Supervised Learning

- We have a dataset of sample sample-target-pairs: $(\mathsf{x}, \mathsf{t}) \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$, so that the expected loss on a prediction is low

$$\ell(\mathcal{M}, \mathcal{D}) = \mathbb{E}\left[\ell(\mathsf{y}, \mathsf{t})\right] = \mathbb{E}_{\mathsf{x},\mathsf{t}}\left[\ell(\mathcal{M}(\mathsf{x}), \mathsf{t})\right] \tag{1}$$

- The loss can be broken into loss-per-sample

Supervised Learning

- We have a dataset of sample sample-target-pairs: $(\mathsf{x}, \mathsf{t}) \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$, so that the expected loss on a prediction is low

$$\ell(\mathcal{M}, \mathcal{D}) = \mathbb{E}\left[\ell(\mathsf{y}, \mathsf{t})\right] = \mathbb{E}_{\mathsf{x},\mathsf{t}}\left[\ell(\mathcal{M}(\mathsf{x}), \mathsf{t})\right] \tag{1}$$

- The loss can be broken into loss-per-sample

Supervised Learning

- We have a dataset of sample sample-target-pairs: $(\mathsf{x}, \mathsf{t}) \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$, so that the expected loss on a prediction is low

$$\ell(\mathcal{M}, \mathcal{D}) = \mathbb{E}\left[\ell(\mathsf{y}, \mathsf{t})\right] = \mathbb{E}_{\mathsf{x},\mathsf{t}}\left[\ell(\mathcal{M}(\mathsf{x}), \mathsf{t})\right] \tag{1}$$

- The loss can be broken into loss-per-sample

Unsupervised Learning

- We have a dataset of sample samples: $\mathsf{x} \in \mathcal{D}$

Supervised Learning

- We have a dataset of sample sample-target-pairs: $(\mathrm{x}, \mathrm{t}) \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$, so that the expected loss on a prediction is low

$$\ell(\mathcal{M}, \mathcal{D}) = \mathbb{E}\left[\ell(\mathrm{y}, \mathrm{t})\right] = \mathbb{E}_{\mathrm{x},\mathrm{t}}\left[\ell(\mathcal{M}(\mathrm{x}), \mathrm{t})\right] \tag{1}$$

- The loss can be broken into loss-per-sample

Unsupervised Learning

- We have a dataset of sample samples: $\mathrm{x} \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$ that can make some kind of useful prediction on new data

Supervised Learning

- We have a dataset of sample sample-target-pairs: $(\mathsf{x}, \mathsf{t}) \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$, so that the expected loss on a prediction is low

$$\ell(\mathcal{M}, \mathcal{D}) = \mathbb{E}\left[\ell(\mathsf{y}, \mathsf{t})\right] = \mathbb{E}_{\mathsf{x},\mathsf{t}}\left[\ell(\mathcal{M}(\mathsf{x}), \mathsf{t})\right] \tag{1}$$

- The loss can be broken into loss-per-sample

Unsupervised Learning

- We have a dataset of sample samples: $\mathsf{x} \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$ that can make some kind of useful prediction on new data
- The prediction can be

Supervised Learning

- We have a dataset of sample sample-target-pairs: $(\mathrm{x}, \mathrm{t}) \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$, so that the expected loss on a prediction is low

$$\ell(\mathcal{M}, \mathcal{D}) = \mathbb{E}\left[\ell(\mathrm{y}, \mathrm{t})\right] = \mathbb{E}_{\mathrm{x},\mathrm{t}}\left[\ell(\mathcal{M}(\mathrm{x}), \mathrm{t})\right] \tag{1}$$

- The loss can be broken into loss-per-sample

Unsupervised Learning

- We have a dataset of sample samples: $\mathrm{x} \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$ that can make some kind of useful prediction on new data
- The prediction can be
    - Predicting a missing feature

Supervised Learning

- We have a dataset of sample sample-target-pairs: $(\mathrm{x}, \mathrm{t}) \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$, so that the expected loss on a prediction is low

$$\ell(\mathcal{M}, \mathcal{D}) = \mathbb{E}\left[\ell(\mathrm{y}, \mathrm{t})\right] = \mathbb{E}_{\mathrm{x},\mathrm{t}}\left[\ell(\mathcal{M}(\mathrm{x}), \mathrm{t})\right] \tag{1}$$

- The loss can be broken into loss-per-sample

Unsupervised Learning

- We have a dataset of sample samples: $\mathrm{x} \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$ that can make some kind of useful prediction on new data
- The prediction can be
  - Predicting a missing feature
  - Simulating new data (generate a new sample from scratch)

Supervised Learning

- We have a dataset of sample sample-target-pairs: $(x, t) \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$, so that the expected loss on a prediction is low

$$\ell(\mathcal{M}, \mathcal{D}) = \mathbb{E}\left[\ell(y, t)\right] = \mathbb{E}_{x,t}\left[\ell(\mathcal{M}(x), t)\right] \tag{1}$$

- The loss can be broken into loss-per-sample

Unsupervised Learning

- We have a dataset of sample samples: $x \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$ that can make some kind of useful prediction on new data
- The prediction can be
  - Predicting a missing feature
  - Simulating new data (generate a new sample from scratch)
  - Predict structures in the data

Supervised Learning

- We have a dataset of sample sample-target-pairs: $(\mathrm{x}, \mathrm{t}) \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$, so that the expected loss on a prediction is low

$$\ell(\mathcal{M}, \mathcal{D}) = \mathbb{E}\left[\ell(\mathrm{y}, \mathrm{t})\right] = \mathbb{E}_{\mathrm{x},\mathrm{t}}\left[\ell(\mathcal{M}(\mathrm{x}), \mathrm{t})\right] \tag{1}$$

- The loss can be broken into loss-per-sample

Unsupervised Learning

- We have a dataset of sample samples: $\mathrm{x} \in \mathcal{D}$
- We wish to make a model $\mathcal{M}$ that can make some kind of useful prediction on new data
- The prediction can be
    - Predicting a missing feature
    - Simulating new data (generate a new sample from scratch)
    - Predict structures in the data
- Loss function depends a bit on purpose

Two main types of supervised learning

Two main types of supervised learning

Regression Real valued outputs. For example $y \in \mathbb{R}$, $y \in [0, \infty]$ or $y \in [-5, 5]$

Two main types of supervised learning

Regression Real valued outputs. For example $y \in \mathbb{R}$, $y \in [0, \infty]$ or $y \in [-5, 5]$

Classification Discrete valued outputs. For example $y \in \{0, 1, 2, 3, 4\}$

Two main types of supervised learning

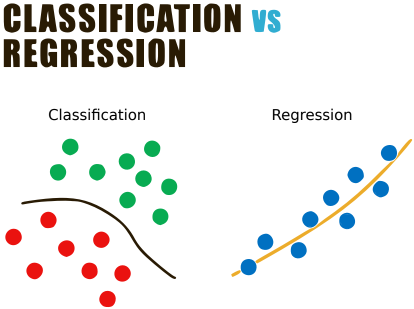
Regression Real valued outputs. For example $y \in \mathbb{R}$, $y \in [0, \infty]$ or $y \in [-5, 5]$

Classification Discrete valued outputs. For example $y \in \{0, 1, 2, 3, 4\}$

Two main types of supervised learning

Regression Real valued outputs. For example $y \in \mathbb{R}$, $y \in [0, \infty]$ or $y \in [-5, 5]$

Classification Discrete valued outputs. For example $y \in \{0, 1, 2, 3, 4\}$



Probability Theory and Bayes Theorem

Probability Theory and Bayes Theorem

$P(x)$	Probability of observing event x.	$P(\boxdot) = \frac{1}{6}$

$P(x)$	Probability of observing event x.	$P(\boxdot) = \frac{1}{6}$
$P(\neg x)$	Probability of *not* observing event x.	$P(\neg\boxdot) = \frac{5}{6}$

Probability Theory and Bayes Theorem

$P(x)$	Probability of observing event x.	$P(\boxdot) = \frac{1}{6}$
$P(\neg x)$	Probability of *not* observing event x.	$P(\neg\boxdot) = \frac{5}{6}$
$P(x, y) = P(x \cap y)$	*Joint* probability of observing both events x and y.	

Probability Theory and Bayes Theorem

$P(x)$	Probability of observing event x.	$P(\boxdot) = \frac{1}{6}$
$P(\neg x)$	Probability of *not* observing event x.	$P(\neg\boxdot) = \frac{5}{6}$
$P(x, y) = P(x \cap y)$	*Joint* probability of observing both events x and y.	
$P(x \mid y)$	*Conditional* probability of observing event x given that we have already observed y.	$P(\boxdot \mid \boxed{?} < 4) = \frac{1}{3}$

Probability Theory and Bayes Theorem

$P(x)$	Probability of observing event x.	$P(⚀) = \frac{1}{6}$
$P(\neg x)$	Probability of *not* observing event x.	$P(\neg ⚀) = \frac{5}{6}$
$P(x, y) = P(x \cap y)$	*Joint* probability of observing both events x and y.	
$P(x \mid y)$	*Conditional* probability of observing event x given that we have already observed y.	$P(⚀ \mid \boxed{?} < 4) = \frac{1}{3}$

Normality	For any x	$0 \leq P(x) \leq 1$

Probability Theory and Bayes Theorem

$P(x)$	Probability of observing event x.	$P(⚀) = \frac{1}{6}$
$P(\neg x)$	Probability of *not* observing event x.	$P(\neg ⚀) = \frac{5}{6}$
$P(x, y) = P(x \cap y)$	*Joint* probability of observing both events x and y.	
$P(x \mid y)$	*Conditional* probability of observing event x given that we have already observed y.	$P(⚀ \mid \boxed{?} < 4) = \frac{1}{3}$

Normality	For any x	$0 \leq P(x) \leq 1$	
Independence	x and y are independent iff.	$P(x, y) = P(x)\ P(y)$	$P(⚀, ⚀) = P(⚀) \times P(⚀) = \frac{1}{6} \times \frac{1}{6} = \frac{1}{36}$

Proving Bayes Theorem

$$P(\mathcal{A}, \mathcal{B}) = P(\mathcal{A}, \mathcal{B})$$

Proving Bayes Theorem

$$
\begin{aligned}
P(\mathcal{A}, \mathcal{B}) &= P(\mathcal{A}, \mathcal{B}) \\
P(\mathcal{A} \mid \mathcal{B})\ P(\mathcal{B}) &= P(\mathcal{B} \mid \mathcal{A})\ P(\mathcal{A})
\end{aligned}
$$

Proving Bayes Theorem

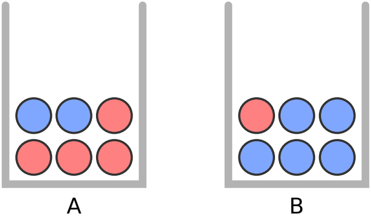
$$
\begin{aligned}
P(\mathcal{A}, \mathcal{B}) &= P(\mathcal{A}, \mathcal{B}) \\
P(\mathcal{A} \mid \mathcal{B})\; P(\mathcal{B}) &= P(\mathcal{B} \mid \mathcal{A})\; P(\mathcal{A}) \\
P(\mathcal{A} \mid \mathcal{B}) &= \frac{P(\mathcal{B} \mid \mathcal{A})\; P(\mathcal{A})}{P(\mathcal{B})}
\end{aligned}
$$

Proving Bayes Theorem

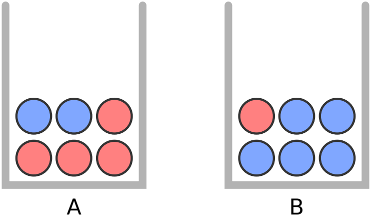
$$P(\mathcal{A}, \mathcal{B}) = P(\mathcal{A}, \mathcal{B})$$
$$P(\mathcal{A} \mid \mathcal{B})\; P(\mathcal{B}) = P(\mathcal{B} \mid \mathcal{A})\; P(\mathcal{A})$$
$$P(\mathcal{A} \mid \mathcal{B}) = \frac{P(\mathcal{B} \mid \mathcal{A})\; P(\mathcal{A})}{P(\mathcal{B})}$$

$$P(\mathcal{M} \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \mathcal{M})\; P(\mathcal{M})}{P(\mathcal{D})}$$

P(M I D) = P(D I M)P(M)

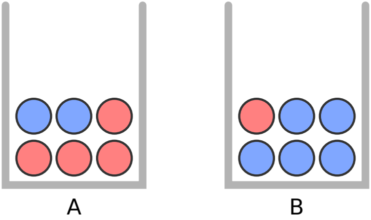


- I pick a random bag and draw a ball: it is red
- What is the probability that I picked bag A?



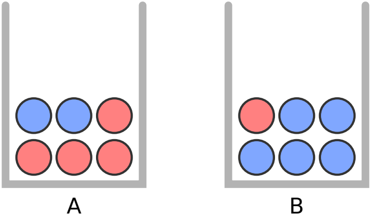
- I pick a random bag and draw a ball: it is red
- What is the probability that I picked bag A?
  - $P(A) = \dfrac{P(red \mid A)\ P(A)}{P(red \mid A)\ P(A) + P(red \mid B)\ P(B)}$

Probability Theory and Bayes Theorem



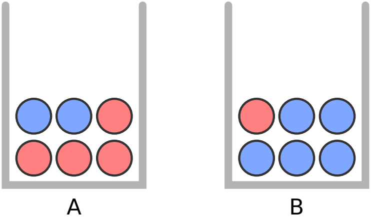
- I pick a random bag and draw a ball: it is red
- What is the probability that I picked bag A?
  - $P(A) = \dfrac{P(red \mid A)\ P(A)}{P(red \mid A)\ P(A) + P(red \mid B)\ P(B)}$
  - $P(A) = \dfrac{\frac{2}{3}\ \frac{1}{2}}{\frac{2}{3}\ \frac{1}{2} + \frac{1}{6}\ \frac{1}{2}}$

Probability Theory and Bayes Theorem



- I pick a random bag and draw a ball: it is red
- What is the probability that I picked bag A?
  - $P(A) = \dfrac{P(red \mid A)\ P(A)}{P(red \mid A)\ P(A) + P(red \mid B)\ P(B)}$
  - $P(A) = \dfrac{\frac{2}{3}\ \frac{1}{2}}{\frac{2}{3}\ \frac{1}{2} + \frac{1}{6}\ \frac{1}{2}}$
  - $P(A) = \dfrac{\frac{2}{6}}{\frac{2}{6} + \frac{1}{12}}$

Probability Theory and Bayes Theorem



- I pick a random bag and draw a ball: it is red
- What is the probability that I picked bag A?
  - $P(A) = \dfrac{P(red \mid A)\ P(A)}{P(red \mid A)\ P(A) + P(red \mid B)\ P(B)}$
  - $P(A) = \dfrac{\frac{2}{3}\ \frac{1}{2}}{\frac{2}{3}\ \frac{1}{2} + \frac{1}{6}\ \frac{1}{2}}$
  - $P(A) = \dfrac{\frac{2}{6}}{\frac{2}{6} + \frac{1}{12}}$
  - $P(A) = \dfrac{\frac{2}{6}}{\frac{5}{12}} = \dfrac{24}{30} = \dfrac{12}{15}$

$P(\mathcal{D} \mid \mathcal{M})$

The *likelihood* of our model is "running our model". We designed it for prediction, so we can usually evaluate it quite easily.

Probability Theory and Bayes Theorem

$P(\mathcal{D} \mid \mathcal{M})$

The *likelihood* of our model is "running our model". We designed it for prediction, so we can usually evaluate it quite easily.

$P(\mathcal{M})$

The *prior* is something we need choose. It is usually fairly simple and is crucial for controlling our model.

Probability Theory and Bayes Theorem

$P(\mathcal{D} \mid \mathcal{M})$

The *likelihood* of our model is "running our model". We designed it for prediction, so we can usually evaluate it quite easily.

$P(\mathcal{M})$

The *prior* is something we need choose. It is usually fairly simple and is crucial for controlling our model.

$$P(\mathcal{M} \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \mathcal{M})\, P(\mathcal{M})}{P(\mathcal{D})}$$

The *posterior* is what we really would like to compute, but the denominator is annoying.

Probability Theory and Bayes Theorem

$P(\mathcal{D} \mid \mathcal{M})$

The *likelihood* of our model is "running our model". We designed it for prediction, so we can usually evaluate it quite easily.

$P(\mathcal{M})$

The *prior* is something we need choose. It is usually fairly simple and is crucial for controlling our model.

$$P(\mathcal{M} \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \mathcal{M})\, P(\mathcal{M})}{P(\mathcal{D})}$$

The *posterior* is what we really would like to compute, but the denominator is annoying.

$$P(\mathcal{D}) = \sum_i P(\mathcal{D} \mid \mathcal{M}_i)\, P(\mathcal{M}_i)$$

The denominator is the sum (or integral) over the model space.

Probability Theory and Bayes Theorem

$P(\mathcal{D} \mid \mathcal{M})$

The *likelihood* of our model is "running our model". We designed it for prediction, so we can usually evaluate it quite easily.

$P(\mathcal{M})$

The *prior* is something we need choose. It is usually fairly simple and is crucial for controlling our model.

$$P(\mathcal{M} \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \mathcal{M})\ P(\mathcal{M})}{P(\mathcal{D})}$$

The *posterior* is what we really would like to compute, but the denominator is annoying.

$$P(\mathcal{D}) = \sum_i P(\mathcal{D} \mid \mathcal{M}_i)\ P(\mathcal{M}_i)$$

The denominator is the sum (or integral) over the model space.

$$P(\mathcal{M} \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \mathcal{M})\ P(\mathcal{M})}{\sum_i P(\mathcal{D} \mid \mathcal{M}_i)\ P(\mathcal{M}_i)}$$

This is how the complete expression looks like. Notice that the denominator is independent of $\mathcal{M}$.

Probability Theory and Bayes Theorem

$P(\mathcal{D} \mid \mathcal{M})$

The *likelihood* of our model is "running our model". We designed it for prediction, so we can usually evaluate it quite easily.

$P(\mathcal{M})$

The *prior* is something we need choose. It is usually fairly simple and is crucial for controlling our model.

$$P(\mathcal{M} \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \mathcal{M})\, P(\mathcal{M})}{P(\mathcal{D})}$$

The *posterior* is what we really would like to compute, but the denominator is annoying.

$$P(\mathcal{D}) = \sum_i P(\mathcal{D} \mid \mathcal{M}_i)\, P(\mathcal{M}_i)$$

The denominator is the sum (or integral) over the model space.

$$P(\mathcal{M} \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \mathcal{M})\, P(\mathcal{M})}{\sum_i P(\mathcal{D} \mid \mathcal{M}_i)\, P(\mathcal{M}_i)}$$

This is how the complete expression looks like. Notice that the denominator is independent of $\mathcal{M}$.

$$P(\mathcal{M} \mid \mathcal{D}) \propto P(\mathcal{D} \mid \mathcal{M})\, P(\mathcal{M})$$

Maximizing this expression provides the same model! (we just don't know what the actual posteriors value was).

Probability Theory and Bayes Theorem

$P(\mathcal{D} \mid \mathcal{M})$		The *likelihood* of our model is "running our model". We designed it for prediction, so we can usually evaluate it quite easily.
$P(\mathcal{M})$		The *prior* is something we need choose. It is usually fairly simple and is crucial for controlling our model.
$P(\mathcal{M} \mid \mathcal{D})$	$= \dfrac{P(\mathcal{D} \mid \mathcal{M})\ P(\mathcal{M})}{P(\mathcal{D})}$	The *posterior* is what we really would like to compute, but the denominator is annoying.
$P(\mathcal{D})$	$= \displaystyle\sum_i P(\mathcal{D} \mid \mathcal{M}_i)\ P(\mathcal{M}_i)$	The denominator is the sum (or integral) over the model space.
$P(\mathcal{M} \mid \mathcal{D})$	$= \dfrac{P(\mathcal{D} \mid \mathcal{M})\ P(\mathcal{M})}{\sum_i P(\mathcal{D} \mid \mathcal{M}_i)\ P(\mathcal{M}_i)}$	This is how the complete expression looks like. Notice that the denominator is independent of $\mathcal{M}$.
$P(\mathcal{M} \mid \mathcal{D})$	$\propto P(\mathcal{D} \mid \mathcal{M})\ P(\mathcal{M})$	Maximizing this expression provides the same model! (we just don't know what the actual posteriors value was).

We need to optimize the product of the likelihood and the prior!

Probability Theory and Bayes Theorem

- Some methods only optimize the likelihood

- Some methods only optimize the likelihood
  - Common with frequentists approach

- Some methods only optimize the likelihood
  - Common with frequentists approach
  - Objective as no prior is chosen

Probability Theory and Bayes Theorem

- Some methods only optimize the likelihood
  - Common with frequentists approach
  - Objective as no prior is chosen
- Bayesians use the prior as well

Probability Theory and Bayes Theorem

- Some methods only optimize the likelihood
  - Common with frequentists approach
  - Objective as no prior is chosen
- Bayesians use the prior as well
  - Provably correct and generalization of frequentist method

- Some methods only optimize the likelihood
  - Common with frequentists approach
  - Objective as no prior is chosen
- Bayesians use the prior as well
  - Provably correct and generalization of frequentist method
  - Prior is chosen - so not objective?

Probability Theory and Bayes Theorem

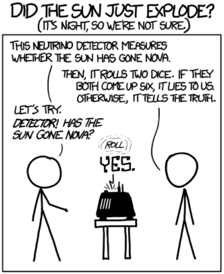
- Some methods only optimize the likelihood
  - Common with frequentists approach
  - Objective as no prior is chosen
- Bayesians use the prior as well
  - Provably correct and generalization of frequentist method
  - Prior is chosen - so not objective?
  - This is the side that is currently dominating AI

Probability Theory and Bayes Theorem

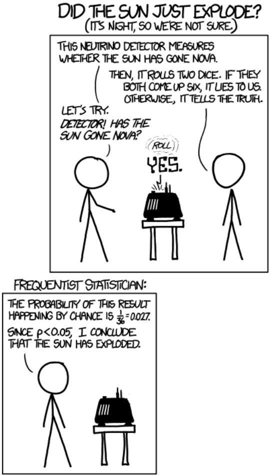
- Some methods only optimize the likelihood
  - Common with frequentists approach
  - Objective as no prior is chosen
- Bayesians use the prior as well
  - Provably correct and generalization of frequentist method
  - Prior is chosen - so not objective?
  - This is the side that is currently dominating AI
- Bayesians and frequentists will frequently fight about who's right

- Some methods only optimize the likelihood
  - Common with frequentists approach
  - Objective as no prior is chosen
- Bayesians use the prior as well
  - Provably correct and generalization of frequentist method
  - Prior is chosen - so not objective?
  - This is the side that is currently dominating AI
- Bayesians and frequentists will frequently fight about who's right

- Some methods only optimize the likelihood
  - Common with frequentists approach
  - Objective as no prior is chosen
- Bayesians use the prior as well
  - Provably correct and generalization of frequentist method
  - Prior is chosen - so not objective?
  - This is the side that is currently dominating AI
- Bayesians and frequentists will frequently fight about who's right

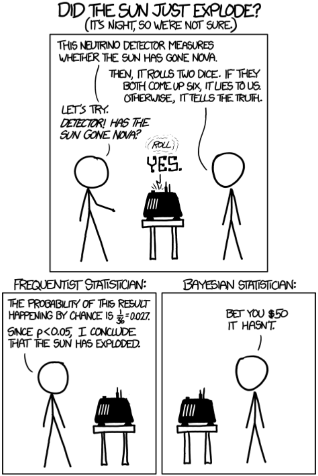


- Some methods only optimize the likelihood
  - Common with frequentists approach
  - Objective as no prior is chosen
- Bayesians use the prior as well
  - Provably correct and generalization of frequentist method
  - Prior is chosen - so not objective?
  - This is the side that is currently dominating AI
- Bayesians and frequentists will frequently fight about who's right



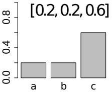
Probability Theory and Bayes Theorem

- Some methods only optimize the likelihood
  - Common with frequentists approach
  - Objective as no prior is chosen
- Bayesians use the prior as well
  - Provably correct and generalization of frequentist method
  - Prior is chosen - so not objective?
  - This is the side that is currently dominating AI
- Bayesians and frequentists will frequently fight about who's right



Bayes Rule as hypothesis probabilities

Bayes Rule as hypothesis probabilities



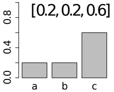
$P(M)$
Prior

Sums to 1

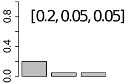
✓

Bayes Rule as hypothesis probabilities

Sums to 1



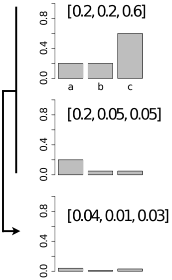
$P(M)$
Prior



$P(\mathcal{D} \mid \mathcal{M})$
Likelihood

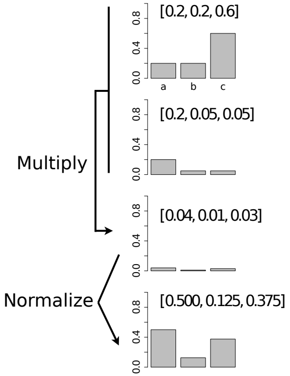
Bayes Rule as hypothesis probabilities

Multiply



		Sums to 1
$P(M)$	Prior	✓
$P(\mathcal{D} \mid \mathcal{M})$	Likelihood	✗
$P(\mathcal{D} \mid \mathcal{M})\,P(M)$	Unnorm. Posterior	✗

Bayes Rule as hypothesis probabilities



Sums to 1

$P(M)$
Prior ✓

$P(\mathcal{D} \mid \mathcal{M})$
Likelihood ✗

$P(\mathcal{D} \mid \mathcal{M})\, P(M)$
Unnorm. Posterior ✗

$P(\mathcal{M} \mid \mathcal{D})$
Posterior ✓

A more important case!

A patient takes a cancer test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer. What is the probability that the patient has cancer?

A more important case!

A patient takes a cancer test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer. What is the probability that the patient has cancer?

- Prior:

A more important case!

A patient takes a cancer test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer. What is the probability that the patient has cancer?

- Prior: $P(\text{cancer}) =$

A more important case!

A patient takes a cancer test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer. What is the probability that the patient has cancer?

- Prior: $P(\text{cancer}) = 0.008$

A more important case!

A patient takes a cancer test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer. What is the probability that the patient has cancer?

- Prior: $P(\text{cancer}) = 0.008$
- Likelihood:

A more important case!

A patient takes a cancer test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer. What is the probability that the patient has cancer?

- Prior: $P(\text{cancer}) = 0.008$
- Likelihood: $P(\text{positive} \mid \text{cancer}) =$

A more important case!

A patient takes a cancer test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer. What is the probability that the patient has cancer?

- Prior: $P(\text{cancer}) = 0.008$
- Likelihood: $P(\text{positive} \mid \text{cancer}) = 0.98$

A more important case!

A patient takes a cancer test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer. What is the probability that the patient has cancer?

- Prior: $P(\text{cancer}) = 0.008$
- Likelihood: $P(\text{positive} \mid \text{cancer}) = 0.98$
- Probability of positive (normalizer):

$$P(\text{positive}) = P(\text{positive} \mid \text{cancer})\ P(\text{cancer}) + P(\text{positive} \mid \neg\text{cancer})\ P(\neg\text{cancer})$$

A more important case!

A patient takes a cancer test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer. What is the probability that the patient has cancer?

- Prior: $P(\text{cancer}) = 0.008$
- Likelihood: $P(\text{positive} \mid \text{cancer}) = 0.98$
- Probability of positive (normalizer):

$$P(\text{positive}) = P(\text{positive} \mid \text{cancer})\ P(\text{cancer}) + P(\text{positive} \mid \neg\text{cancer})\ P(\neg\text{cancer})$$

$$= 0.98 \times 0.008 + (1 - 0.97) \times (1 - 0.008) = 0.0376$$

Probability Theory and Bayes Theorem

A more important case!

A patient takes a cancer test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer. What is the probability that the patient has cancer?

- Prior: $P(\text{cancer}) = 0.008$
- Likelihood: $P(\text{positive} \mid \text{cancer}) = 0.98$
- Probability of positive (normalizer):

$$\begin{aligned} P(\text{positive}) &= P(\text{positive} \mid \text{cancer})\ P(\text{cancer}) + P(\text{positive} \mid \neg\text{cancer})\ P(\neg\text{cancer}) \\ &= 0.98 \times 0.008 + (1 - 0.97) \times (1 - 0.008) = 0.0376 \end{aligned}$$

- Posterior probability of having cancer:

$$P(\text{cancer} \mid \text{positive}) = \frac{P(\text{positive} \mid \text{cancer})\ P(\text{cancer})}{P(\text{positive})}$$

A more important case!

A patient takes a cancer test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer. What is the probability that the patient has cancer?

- Prior: $P(\text{cancer}) = 0.008$
- Likelihood: $P(\text{positive} \mid \text{cancer}) = 0.98$
- Probability of positive (normalizer):

$$
\begin{aligned}
P(\text{positive}) &= P(\text{positive} \mid \text{cancer})\ P(\text{cancer}) + P(\text{positive} \mid \neg\text{cancer})\ P(\neg\text{cancer}) \\
&= 0.98 \times 0.008 + (1 - 0.97) \times (1 - 0.008) = 0.0376
\end{aligned}
$$

- Posterior probability of having cancer:

$$
P(\text{cancer} \mid \text{positive}) = \frac{P(\text{positive} \mid \text{cancer})\ P(\text{cancer})}{P(\text{positive})}
$$

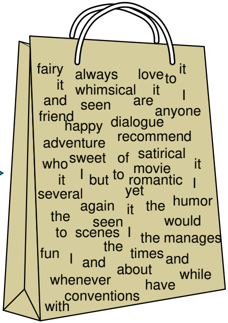
$$
= \frac{0.98 \times 0.008}{0.0376} = 0.2085
$$

Naive Bayes

Naive Bayes

Bag-of-Words

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

Bag-of-Words

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

Disregards order of words!

Tokenization

- We call splitting a text into small segments *tokenization*

Tokenization

- We call splitting a text into small segments *tokenization*
- Each segment is a *token*.

Tokenization

- We call splitting a text into small segments *tokenization*
- Each segment is a *token*.
- Above I used words as tokens - why?

Tokenization

- We call splitting a text into small segments *tokenization*
- Each segment is a *token*.
- Above I used words as tokens - why?
    - Not necessarily the best level

Tokenization

- We call splitting a text into small segments *tokenization*
- Each segment is a *token*.
- Above I used words as tokens - why?
  - Not necessarily the best level
  - There are character-level models!

Tokenization

- We call splitting a text into small segments *tokenization*
- Each segment is a *token*.
- Above I used words as tokens - why?
    - Not necessarily the best level
    - There are character-level models!
- Turns out the best models in the fields uses a mix of words and *subwords*

Tokenization

- We call splitting a text into small segments *tokenization*
- Each segment is a *token*.
- Above I used words as tokens - why?
    - Not necessarily the best level
    - There are character-level models!
- Turns out the best models in the fields uses a mix of words and *subwords*
    - Instead of making this decision, we make models learn the level they operate on

Tokenization

- We call splitting a text into small segments *tokenization*
- Each segment is a *token*.
- Above I used words as tokens - why?
  - Not necessarily the best level
  - There are character-level models!
- Turns out the best models in the fields uses a mix of words and *subwords*
  - Instead of making this decision, we make models learn the level they operate on
- We will learn more about this later, but for now our operating level is *words*

Naive Bayes

Documents:

- an1 *The domestic dog is a domesticated descendant of the wolf.*
- an2 *The cat is a domestic species of small carnivorous mammal.*
- pl1 *Saturn is the sixth planet from the Sun and the second-largest in the Solar System.*
- an3 *Jerboas are hopping desert rodents found throughout Northern Africa and Asia.*
- pl2 *Mars is the fourth planet from the Sun and the second-smallest planet in the Solar System, being larger than only Mercury.*

Naive Bayes

Bag-of-words

	an1	an2	pl1	an3	pl2
africa	0	0	0	1	0
and	0	0	1	1	1
are	0	0	0	1	0
asia	0	0	0	1	0
being	0	0	0	0	1
carnivorous	0	1	0	0	0
cat	0	1	0	0	0
descendant	1	0	0	0	0
desert	0	0	0	1	0
dog	1	0	0	0	0
domestic	1	1	0	0	0
domesticated	1	0	0	0	0
found	0	0	0	1	0
fourth	0	0	0	0	1
from	0	0	1	0	1
hopping	0	0	0	1	0
in	0	0	1	0	1
is	1	1	1	0	1
jerboas	0	0	0	1	0
larger	0	0	0	0	1
largest	0	0	1	0	0
mammal	0	1	0	0	0
mars	0	0	0	0	1
mercury	0	0	0	0	1
northern	0	0	0	1	0
of	1	1	0	0	0
only	0	0	0	0	1
planet	0	0	1	0	2
rodents	0	0	0	1	0
saturn	0	0	1	0	0
second	0	0	1	0	1
sixth	0	0	1	0	0
small	0	1	0	0	0
smallest	0	0	0	0	1
solar	0	0	1	0	1
species	0	1	0	0	0
sun	0	0	1	0	1
system	0	0	1	0	1
than	0	0	0	0	1
the	2	1	4	0	4
throughout	0	0	0	1	0
wolf	1	0	0	0	0

Documents:

- an1 *The domestic dog is a domesticated descendant of the wolf.*
- an2 *The cat is a domestic species of small carnivorous mammal.*
- pl1 *Saturn is the sixth planet from the Sun and the second-largest in the Solar System.*
- an3 *Jerboas are hopping desert rodents found throughout Northern Africa and Asia.*
- pl2 *Mars is the fourth planet from the Sun and the second-smallest planet in the Solar System, being larger than only Mercury.*

Naive Bayes

- Make a simple model of the probability of words in a class

$$P(\text{word} \mid \text{class}) = \frac{\text{count of word in class}}{\text{total words in class}}$$

For example if "cat" occurs 3 times in class "anim" in our data, and there are a total of 20 words in that class, then we have

$$P(\text{cat} \mid \text{anim}) = \frac{3}{20}$$

Naive Bayes

Naive Bayes

- Make a simple model of the probability of words in a class

$$P(\text{word} \mid \text{class}) = \frac{\text{count of word in class}}{\text{total words in class}}$$

  For example if "cat" occurs 3 times in class "anim" in our data, and there are a total of 20 words in that class, then we have

$$P(\text{cat} \mid \text{anim}) = \frac{3}{20}$$

- *Likelihood*: Approximate the probability of a document to be the joint probability of the words in the document, **assuming independence**

$$P(\text{the cat is small} \mid \text{anim}) \underbrace{\approx}_{\text{assumption}} P(\text{the} \mid \text{anim}) \times P(\text{cat} \mid \text{anim}) \times P(\text{is} \mid \text{anim}) \times P(\text{small} \mid \text{anim})$$

Naive Bayes

- Make a simple model of the probability of words in a class

$$P(\text{word} \mid \text{class}) = \frac{\text{count of word in class}}{\text{total words in class}}$$

For example if "cat" occurs 3 times in class "anim" in our data, and there are a total of 20 words in that class, then we have

$$P(\text{cat} \mid \text{anim}) = \frac{3}{20}$$

- *Likelihood*: Approximate the probability of a document to be the joint probability of the words in the document, **assuming independence**

$$P(\text{the cat is small} \mid \text{anim}) \underbrace{\approx}_{\text{assumption}} P(\text{the} \mid \text{anim}) \times P(\text{cat} \mid \text{anim}) \times P(\text{is} \mid \text{anim}) \times P(\text{small} \mid \text{anim})$$

- This assumption is why we call it naive

Naive Bayes

- Make a simple model of the probability of words in a class

$$P(\text{word} \mid \text{class}) = \frac{\text{count of word in class}}{\text{total words in class}}$$

For example if "cat" occurs 3 times in class "anim" in our data, and there are a total of 20 words in that class, then we have

$$P(\text{cat} \mid \text{anim}) = \frac{3}{20}$$

- *Likelihood*: Approximate the probability of a document to be the joint probability of the words in the document, **assuming independence**

$$P(\text{the cat is small} \mid \text{anim}) \underbrace{\approx}_{\text{assumption}} P(\text{the} \mid \text{anim}) \times P(\text{cat} \mid \text{anim}) \times P(\text{is} \mid \text{anim}) \times P(\text{small} \mid \text{anim})$$

- This assumption is why we call it naive
- *Prior*: Weight the classes according to how often we see them

Naive Bayes

- Make a simple model of the probability of words in a class

$$P(\text{word} \mid \text{class}) = \frac{\text{count of word in class}}{\text{total words in class}}$$

  For example if "cat" occurs 3 times in class "anim" in our data, and there are a total of 20 words in that class, then we have

$$P(\text{cat} \mid \text{anim}) = \frac{3}{20}$$

- *Likelihood*: Approximate the probability of a document to be the joint probability of the words in the document, **assuming independence**

$$P(\text{the cat is small} \mid \text{anim}) \underbrace{\approx}_{\text{assumption}} P(\text{the} \mid \text{anim}) \times P(\text{cat} \mid \text{anim}) \times P(\text{is} \mid \text{anim}) \times P(\text{small} \mid \text{anim})$$

- This assumption is why we call it naive
- *Prior*: Weight the classes according to how often we see them
- *Posterior*: compute from likelihood and prior according to Bayes theorem

Naive Bayes

Bag-of-words

	an1	an2	pl1	an3	pl2
africa	0	0	0	1	0
and	0	0	1	1	1
are	0	0	0	1	0
asia	0	0	0	1	0
being	0	0	0	0	1
carnivorous	0	1	0	0	0
cat	0	1	0	0	0
descendant	1	0	0	0	0
desert	0	0	0	1	0
dog	1	0	0	0	0
domestic	1	1	0	0	0
domesticated	1	0	0	0	0
found	0	0	0	1	0
fourth	0	0	0	0	1
from	0	0	1	0	1
hopping	0	0	0	1	0
in	0	0	1	0	1
is	1	1	1	0	1
jerboas	0	0	0	1	0
larger	0	0	0	0	1
largest	0	0	1	0	0
mammal	0	1	0	0	0
mars	0	0	0	0	1
mercury	0	0	0	0	1
northern	0	0	0	1	0
of	1	1	0	0	0
only	0	0	0	0	1
planet	0	0	1	0	2
rodents	0	0	0	1	0
saturn	0	0	1	0	0
second	0	0	1	0	1
sixth	0	0	1	0	0
small	0	1	0	0	0
smallest	0	0	0	0	1
solar	0	0	1	0	1
species	0	1	0	0	0
sun	0	0	1	0	1
system	0	0	1	0	1
than	0	0	0	0	1
the	2	1	4	0	4
throughout	0	0	0	1	0
wolf	1	0	0	0	0

Documents:

- an1 *The domestic dog is a domesticated descendant of the wolf.*
- an2 *The cat is a domestic species of small carnivorous mammal.*
- pl1 *Saturn is the sixth planet from the Sun and the second-largest in the Solar System.*
- an3 *Jerboas are hopping desert rodents found throughout Northern Africa and Asia.*
- pl2 *Mars is the fourth planet from the Sun and the second-smallest planet in the Solar System, being larger than only Mercury.*

Naive Bayes

Bag-of-words

	an1	an2	pl1	an3	pl2
africa	0	0	0	1	0
and	0	0	1	1	1
are	0	0	0	1	0
asia	0	0	0	1	0
being	0	0	0	0	1
carnivorous	0	1	0	0	0
cat	0	1	0	0	0
descendant	1	0	0	0	0
desert	0	0	0	1	0
dog	1	0	0	0	0
domestic	1	1	0	0	0
domesticated	1	0	0	0	0
found	0	0	0	1	0
fourth	0	0	0	0	1
from	0	0	1	0	1
hopping	0	0	0	1	0
in	0	0	1	0	1
is	1	1	1	0	1
jerboas	0	0	0	1	0
larger	0	0	0	0	1
largest	0	0	1	0	0
mammal	0	1	0	0	0
mars	0	0	0	0	1
mercury	0	0	0	0	1
northern	0	0	0	1	0
of	1	1	0	0	0
only	0	0	0	0	1
planet	0	0	1	0	2
rodents	0	0	0	1	0
saturn	0	0	1	0	0
second	0	0	1	0	1
sixth	0	0	1	0	0
small	0	1	0	0	0
smallest	0	0	0	0	1
solar	0	0	1	0	1
species	0	1	0	0	0
sun	0	0	1	0	1
system	0	0	1	0	1
than	0	0	0	0	1
the	2	1	4	0	4
throughout	0	0	0	1	0
wolf	1	0	0	0	0

Naive Bayes

Bag-of-words

	an1	an2	pl1	an3	pl2
africa	0	0	0	1	0
and	0	0	1	1	1
are	0	0	0	1	0
asia	0	0	0	1	0
being	0	0	0	0	1
carnivorous	0	1	0	0	0
cat	0	1	0	0	0
descendant	1	0	0	0	0
desert	0	0	0	1	0
dog	1	0	0	0	0
domestic	1	1	0	0	0
domesticated	1	0	0	0	0
found	0	0	0	1	0
fourth	0	0	0	0	1
from	0	0	1	0	1
hopping	0	0	0	1	0
in	0	0	1	0	1
is	1	1	1	0	1
jerboas	0	0	0	1	0
larger	0	0	0	0	1
largest	0	0	1	0	0
mammal	0	1	0	0	0
mars	0	0	0	0	1
mercury	0	0	0	0	1
northern	0	0	0	1	0
of	1	1	0	0	0
only	0	0	0	0	1
planet	0	0	1	0	2
rodents	0	0	0	1	0
saturn	0	0	1	0	0
second	0	0	1	0	1
sixth	0	0	1	0	0
small	0	1	0	0	0
smallest	0	0	0	0	1
solar	0	0	1	0	1
species	0	1	0	0	0
sun	0	0	1	0	1
system	0	0	1	0	1
than	0	0	0	0	1
the	2	1	4	0	4
throughout	0	0	0	1	0
wolf	1	0	0	0	0

Word counts in classes

	animal	planet
africa	1	0
and	1	2
are	1	0
asia	1	0
being	0	1
carnivorous	1	0
cat	1	0
descendant	1	0
desert	1	0
dog	1	0
domestic	2	0
domesticated	1	0
found	1	0
fourth	0	1
from	0	2
hopping	1	0
in	0	2
is	2	2
jerboas	1	0
larger	0	1
largest	0	1
mammal	1	0
mars	0	1
mercury	0	1
northern	1	0
of	2	0
only	0	1
planet	0	3
rodents	1	0
saturn	0	1
second	0	2
sixth	0	1
small	1	0
smallest	0	1
solar	0	2
species	1	0
sun	0	2
system	0	2
than	0	1
the	3	8
throughout	1	0
wolf	1	0

Naive Bayes

Bag-of-words

	an1	an2	pl1	an3	pl2
africa	0	0	0	1	0
and	0	0	1	1	1
are	0	0	0	1	0
asia	0	0	0	1	0
being	0	0	0	0	1
carnivorous	0	1	0	0	0
cat	0	1	0	0	0
descendant	1	0	0	0	0
desert	0	0	0	1	0
dog	1	0	0	0	0
domestic	1	1	0	0	0
domesticated	1	0	0	0	0
found	0	0	0	1	0
fourth	0	0	0	0	1
from	0	0	1	0	1
hopping	0	0	0	1	0
in	0	0	1	0	1
is	1	1	1	0	1
jerboas	0	0	0	1	0
larger	0	0	0	0	1
largest	0	0	1	0	0
mammal	0	1	0	0	0
mars	0	0	0	0	1
mercury	0	0	0	0	1
northern	0	0	0	1	0
of	1	1	0	0	0
only	0	0	0	0	1
planet	0	0	1	0	2
rodents	0	0	0	1	0
saturn	0	0	1	0	0
second	0	0	1	0	1
sixth	0	0	1	0	0
small	0	1	0	0	0
smallest	0	0	0	0	1
solar	0	0	1	0	1
species	0	1	0	0	0
sun	0	0	1	0	1
system	0	0	1	0	1
than	0	0	0	0	1
the	2	1	4	0	4
throughout	0	0	0	1	0
wolf	1	0	0	0	0

Word counts in classes

	animal	planet
africa	1	0
and	1	2
are	1	0
asia	1	0
being	0	1
carnivorous	1	0
cat	1	0
descendant	1	0
desert	1	0
dog	1	0
domestic	2	0
domesticated	1	0
found	1	0
fourth	0	1
from	0	2
hopping	1	0
in	0	2
is	2	2
jerboas	1	0
larger	0	1
largest	0	1
mammal	1	0
mars	0	1
mercury	0	1
northern	1	0
of	2	0
only	0	1
planet	0	3
rodents	1	0
saturn	0	1
second	0	2
sixth	0	1
small	1	0
smallest	0	1
solar	0	2
species	1	0
sun	0	2
system	0	2
than	0	1
the	3	8
throughout	1	0
wolf	1	0

Word probabilities in classes

	animal	planet
africa	0.03	0.00
and	0.03	0.05
are	0.03	0.00
asia	0.03	0.00
being	0.00	0.03
carnivorous	0.03	0.00
cat	0.03	0.00
descendant	0.03	0.00
desert	0.03	0.00
dog	0.03	0.00
domestic	0.07	0.00
domesticated	0.03	0.00
found	0.03	0.00
fourth	0.00	0.03
from	0.00	0.05
hopping	0.03	0.00
in	0.00	0.05
is	0.07	0.05
jerboas	0.03	0.00
larger	0.00	0.03
largest	0.00	0.03
mammal	0.03	0.00
mars	0.00	0.03
mercury	0.00	0.03
northern	0.03	0.00
of	0.07	0.00
only	0.00	0.03
planet	0.00	0.08
rodents	0.03	0.00
saturn	0.00	0.03
second	0.00	0.05
sixth	0.00	0.03
small	0.03	0.00
smallest	0.00	0.03
solar	0.00	0.05
species	0.03	0.00
sun	0.00	0.05
system	0.00	0.05
than	0.00	0.03
the	0.10	0.21
throughout	0.03	0.00
wolf	0.03	0.00

Naive Bayes

- **New datapoint:** *the cat is larger*

Word probabilities in classes

	animal	planet
africa	0.03	0.00
and	0.03	0.05
are	0.03	0.00
asia	0.03	0.00
being	0.00	0.03
carnivorous	0.03	0.00
cat	0.03	0.00
descendant	0.03	0.00
desert	0.03	0.00
dog	0.03	0.00
domestic	0.07	0.00
domesticated	0.03	0.00
found	0.03	0.00
fourth	0.00	0.03
from	0.00	0.05
hopping	0.03	0.00
in	0.00	0.05
is	0.07	0.05
jerboas	0.03	0.00
larger	0.00	0.03
largest	0.00	0.03
mammal	0.03	0.00
mars	0.00	0.03
mercury	0.00	0.03
northern	0.03	0.00
of	0.07	0.00
only	0.00	0.03
planet	0.00	0.08
rodents	0.03	0.00
saturn	0.00	0.03
second	0.00	0.05
sixth	0.00	0.03
small	0.03	0.00
smallest	0.00	0.03
solar	0.00	0.05
species	0.03	0.00
sun	0.00	0.05
system	0.00	0.05
than	0.00	0.03
the	0.10	0.21
throughout	0.03	0.00
wolf	0.03	0.00

Naive Bayes

- **New datapoint:** *the cat is larger*
- Get probabilities!

Word probabilities in classes

	animal	planet
africa	0.03	0.00
and	0.03	0.05
are	0.03	0.00
asia	0.03	0.00
being	0.00	0.03
carnivorous	0.03	0.00
cat	0.03	0.00
descendant	0.03	0.00
desert	0.03	0.00
dog	0.03	0.00
domestic	0.07	0.00
domesticated	0.03	0.00
found	0.03	0.00
fourth	0.00	0.03
from	0.00	0.05
hopping	0.03	0.00
in	0.00	0.05
is	0.07	0.05
jerboas	0.03	0.00
larger	0.00	0.03
largest	0.00	0.03
mammal	0.03	0.00
mars	0.00	0.03
mercury	0.00	0.03
northern	0.03	0.00
of	0.07	0.00
only	0.00	0.03
planet	0.00	0.08
rodents	0.03	0.00
saturn	0.00	0.03
second	0.00	0.05
sixth	0.00	0.03
small	0.03	0.00
smallest	0.00	0.03
solar	0.00	0.05
species	0.03	0.00
sun	0.00	0.05
system	0.00	0.05
than	0.00	0.03
the	0.10	0.21
throughout	0.03	0.00
wolf	0.03	0.00

Naive Bayes

- **New datapoint:** *the cat is larger*
- Get probabilities!

animal

word	*probability*
the	0.10
cat	0.03
is	0.07
larger	0.00

Word probabilities in classes

	animal	planet
africa	0.03	0.00
and	0.03	0.05
are	0.03	0.00
asia	0.03	0.00
being	0.00	0.03
carnivorous	0.03	0.00
cat	0.03	0.00
descendant	0.03	0.00
desert	0.03	0.00
dog	0.03	0.00
domestic	0.07	0.00
domesticated	0.03	0.00
found	0.03	0.00
fourth	0.00	0.03
from	0.00	0.05
hopping	0.03	0.00
in	0.00	0.05
is	0.07	0.05
jerboas	0.03	0.00
larger	0.00	0.03
largest	0.00	0.03
mammal	0.03	0.00
mars	0.00	0.03
mercury	0.00	0.03
northern	0.03	0.00
of	0.07	0.00
only	0.00	0.03
planet	0.00	0.08
rodents	0.03	0.00
saturn	0.00	0.03
second	0.00	0.05
sixth	0.00	0.03
small	0.03	0.00
smallest	0.00	0.03
solar	0.00	0.05
species	0.03	0.00
sun	0.00	0.05
system	0.00	0.05
than	0.00	0.03
the	0.10	0.21
throughout	0.03	0.00
wolf	0.03	0.00

Naive Bayes

- **New datapoint:** *the cat is larger*
- Get probabilities!

animal

word	*probability*
the	0.10
cat	0.03
is	0.07
larger	0.00

planet

word	*probability*
the	0.21
cat	0.00
is	0.05
larger	0.03

Word probabilities in classes

	animal	planet
africa	0.03	0.00
and	0.03	0.05
are	0.03	0.00
asia	0.03	0.00
being	0.00	0.03
carnivorous	0.03	0.00
cat	0.03	0.00
descendant	0.03	0.00
desert	0.03	0.00
dog	0.03	0.00
domestic	0.07	0.00
domesticated	0.03	0.00
found	0.03	0.00
fourth	0.00	0.03
from	0.00	0.05
hopping	0.03	0.00
in	0.00	0.05
is	0.07	0.05
jerboas	0.03	0.00
larger	0.00	0.03
largest	0.00	0.03
mammal	0.03	0.00
mars	0.00	0.03
mercury	0.00	0.03
northern	0.03	0.00
of	0.07	0.00
only	0.00	0.03
planet	0.00	0.08
rodents	0.03	0.00
saturn	0.00	0.03
second	0.00	0.05
sixth	0.00	0.03
small	0.03	0.00
smallest	0.00	0.03
solar	0.00	0.05
species	0.03	0.00
sun	0.00	0.05
system	0.00	0.05
than	0.00	0.03
the	0.10	0.21
throughout	0.03	0.00
wolf	0.03	0.00

Naive Bayes

- **New datapoint:** *the cat is larger*
- Get probabilities!

animal

word	*probability*
the	0.10
cat	0.03
is	0.07
larger	0.00

planet

word	*probability*
the	0.21
cat	0.00
is	0.05
larger	0.03

- Likelihood:

$$P(\mathrm{x} \mid \mathrm{animal}) = 0.10 \times 0.03 \times 0.07 \times 0.00 = 0.00$$

$$P(\mathrm{x} \mid \mathrm{planet}) = 0.21 \times 0.00 \times 0.05 \times 0.03 = 0.00$$

Word probabilities in classes

	animal	planet
africa	0.03	0.00
and	0.03	0.05
are	0.03	0.00
asia	0.03	0.00
being	0.00	0.03
carnivorous	0.03	0.00
cat	0.03	0.00
descendant	0.03	0.00
desert	0.03	0.00
dog	0.03	0.00
domestic	0.07	0.00
domesticated	0.03	0.00
found	0.03	0.00
fourth	0.00	0.03
from	0.00	0.05
hopping	0.03	0.00
in	0.00	0.05
is	0.07	0.05
jerboas	0.03	0.00
larger	0.00	0.03
largest	0.00	0.03
mammal	0.03	0.00
mars	0.00	0.03
mercury	0.00	0.03
northern	0.03	0.00
of	0.07	0.00
only	0.00	0.03
planet	0.00	0.08
rodents	0.03	0.00
saturn	0.00	0.03
second	0.00	0.05
sixth	0.00	0.03
small	0.03	0.00
smallest	0.00	0.03
solar	0.00	0.05
species	0.03	0.00
sun	0.00	0.05
system	0.00	0.05
than	0.00	0.03
the	0.10	0.21
throughout	0.03	0.00
wolf	0.03	0.00

Naive Bayes

- **New datapoint:** *the cat is larger*
- Get probabilities!

animal

word	*probability*
the	0.10
cat	0.03
is	0.07
larger	0.00

planet

word	*probability*
the	0.21
cat	0.00
is	0.05
larger	0.03

- Likelihood:

$$P(\mathrm{x} \mid \text{animal}) = 0.10 \times 0.03 \times 0.07 \times 0.00 = 0.00$$
$$P(\mathrm{x} \mid \text{planet}) = 0.21 \times 0.00 \times 0.05 \times 0.03 = 0.00$$

- We have a problem: any unseen word in class will "crash" the probabilities!

Word probabilities in classes

	animal	planet
africa	0.03	0.00
and	0.03	0.05
are	0.03	0.00
asia	0.03	0.00
being	0.00	0.03
carnivorous	0.03	0.00
cat	0.03	0.00
descendant	0.03	0.00
desert	0.03	0.00
dog	0.03	0.00
domestic	0.07	0.00
domesticated	0.03	0.00
found	0.03	0.00
fourth	0.00	0.03
from	0.00	0.05
hopping	0.03	0.00
in	0.00	0.05
is	0.07	0.05
jerboas	0.03	0.00
larger	0.00	0.03
largest	0.00	0.03
mammal	0.03	0.00
mars	0.00	0.03
mercury	0.00	0.03
northern	0.03	0.00
of	0.07	0.00
only	0.00	0.03
planet	0.00	0.08
rodents	0.03	0.00
saturn	0.00	0.03
second	0.00	0.05
sixth	0.00	0.03
small	0.03	0.00
smallest	0.00	0.03
solar	0.00	0.05
species	0.03	0.00
sun	0.00	0.05
system	0.00	0.05
than	0.00	0.03
the	0.10	0.21
throughout	0.03	0.00
wolf	0.03	0.00

Naive Bayes

- **New datapoint:** *the cat is larger*
- Get probabilities!

animal

word	*probability*
the	0.10
cat	0.03
is	0.07
larger	0.00

planet

word	*probability*
the	0.21
cat	0.00
is	0.05
larger	0.03

- Likelihood:

$$P(x \mid \text{animal}) = 0.10 \times 0.03 \times 0.07 \times 0.00 = 0.00$$
$$P(x \mid \text{planet}) = 0.21 \times 0.00 \times 0.05 \times 0.03 = 0.00$$

- We have a problem: any unseen word in class will "crash" the probabilities!
- We need some "base probability" of any word.

Word probabilities in classes

	animal	planet
africa	0.03	0.00
and	0.03	0.05
are	0.03	0.00
asia	0.03	0.00
being	0.00	0.03
carnivorous	0.03	0.00
cat	0.03	0.00
descendant	0.03	0.00
desert	0.03	0.00
dog	0.03	0.00
domestic	0.07	0.00
domesticated	0.03	0.00
found	0.03	0.00
fourth	0.00	0.03
from	0.00	0.05
hopping	0.03	0.00
in	0.00	0.05
is	0.07	0.05
jerboas	0.03	0.00
larger	0.00	0.03
largest	0.00	0.03
mammal	0.03	0.00
mars	0.00	0.03
mercury	0.00	0.03
northern	0.03	0.00
of	0.07	0.00
only	0.00	0.03
planet	0.00	0.08
rodents	0.03	0.00
saturn	0.00	0.03
second	0.00	0.05
sixth	0.00	0.03
small	0.03	0.00
smallest	0.00	0.03
solar	0.00	0.05
species	0.03	0.00
sun	0.00	0.05
system	0.00	0.05
than	0.00	0.03
the	0.10	0.21
throughout	0.03	0.00
wolf	0.03	0.00

Naive Bayes

- This time we have a base-probability of any word!
- Get probabilities!

Word probabilities in classes

	animal	planet
africa	0.03	0.01
and	0.03	0.05
are	0.03	0.01
asia	0.03	0.01
being	0.01	0.03
carnivorous	0.03	0.01
cat	0.03	0.01
descendant	0.03	0.01
desert	0.03	0.01
dog	0.03	0.01
domestic	0.06	0.01
domesticated	0.03	0.01
found	0.03	0.01
fourth	0.01	0.03
from	0.01	0.05
hopping	0.03	0.01
in	0.01	0.05
is	0.06	0.05
jerboas	0.03	0.01
larger	0.01	0.03
largest	0.01	0.03
mammal	0.03	0.01
mars	0.01	0.03
mercury	0.01	0.03
northern	0.03	0.01
of	0.06	0.01
only	0.01	0.03
planet	0.01	0.07
rodents	0.03	0.01
saturn	0.01	0.03
second	0.01	0.05
sixth	0.01	0.03
small	0.03	0.01
smallest	0.01	0.03
solar	0.01	0.05
species	0.03	0.01
sun	0.01	0.05
system	0.01	0.05
than	0.01	0.03
the	0.08	0.17
throughout	0.03	0.01
wolf	0.03	0.01

Naive Bayes

- This time we have a base-probability of any word!
- Get probabilities!

animal

word	*probability*
the	0.08
cat	0.03
is	0.06
larger	0.01

Word probabilities in classes

	animal	planet
africa	0.03	0.01
and	0.03	0.05
are	0.03	0.01
asia	0.03	0.01
being	0.01	0.03
carnivorous	0.03	0.01
cat	0.03	0.01
descendant	0.03	0.01
desert	0.03	0.01
dog	0.03	0.01
domestic	0.06	0.01
domesticated	0.03	0.01
found	0.03	0.01
fourth	0.01	0.03
from	0.01	0.05
hopping	0.03	0.01
in	0.01	0.05
is	0.06	0.05
jerboas	0.03	0.01
larger	0.01	0.03
largest	0.01	0.03
mammal	0.03	0.01
mars	0.01	0.03
mercury	0.01	0.03
northern	0.03	0.01
of	0.06	0.01
only	0.01	0.03
planet	0.01	0.07
rodents	0.03	0.01
saturn	0.01	0.03
second	0.01	0.05
sixth	0.01	0.03
small	0.03	0.01
smallest	0.01	0.03
solar	0.01	0.05
species	0.03	0.01
sun	0.01	0.05
system	0.01	0.05
than	0.01	0.03
the	0.08	0.17
throughout	0.03	0.01
wolf	0.03	0.01

Naive Bayes

- This time we have a base-probability of any word!
- Get probabilities!

animal

word	*probability*
the	0.08
cat	0.03
is	0.06
larger	0.01

planet

word	*probability*
the	0.17
cat	0.01
is	0.05
larger	0.03

Word probabilities in classes

	animal	planet
africa	0.03	0.01
and	0.03	0.05
are	0.03	0.01
asia	0.03	0.01
being	0.01	0.03
carnivorous	0.03	0.01
cat	0.03	0.01
descendant	0.03	0.01
desert	0.03	0.01
dog	0.03	0.01
domestic	0.06	0.01
domesticated	0.03	0.01
found	0.03	0.01
fourth	0.01	0.03
from	0.01	0.05
hopping	0.03	0.01
in	0.01	0.05
is	0.06	0.05
jerboas	0.03	0.01
larger	0.01	0.03
largest	0.01	0.03
mammal	0.03	0.01
mars	0.01	0.03
mercury	0.01	0.03
northern	0.03	0.01
of	0.06	0.01
only	0.01	0.03
planet	0.01	0.07
rodents	0.03	0.01
saturn	0.01	0.03
second	0.01	0.05
sixth	0.01	0.03
small	0.03	0.01
smallest	0.01	0.03
solar	0.01	0.05
species	0.03	0.01
sun	0.01	0.05
system	0.01	0.05
than	0.01	0.03
the	0.08	0.17
throughout	0.03	0.01
wolf	0.03	0.01

Naive Bayes

- This time we have a base-probability of any word!
- Get probabilities!

animal

word	*probability*
the	0.08
cat	0.03
is	0.06
larger	0.01

planet

word	*probability*
the	0.17
cat	0.01
is	0.05
larger	0.03

- Likelihood:

$$P(\mathrm{x} \mid \text{animal}) = 0.08 \times 0.03 \times 0.06 \times 0.01 = 0.00000144$$
$$P(\mathrm{x} \mid \text{planet}) = 0.17 \times 0.01 \times 0.05 \times 0.03 = 0.00000255$$

Word probabilities in classes

	animal	planet
africa	0.03	0.01
and	0.03	0.05
are	0.03	0.01
asia	0.03	0.01
being	0.01	0.03
carnivorous	0.03	0.01
cat	0.03	0.01
descendant	0.03	0.01
desert	0.03	0.01
dog	0.03	0.01
domestic	0.06	0.01
domesticated	0.03	0.01
found	0.03	0.01
fourth	0.01	0.03
from	0.01	0.05
hopping	0.03	0.01
in	0.01	0.05
is	0.06	0.05
jerboas	0.03	0.01
larger	0.01	0.03
largest	0.01	0.03
mammal	0.03	0.01
mars	0.01	0.03
mercury	0.01	0.03
northern	0.03	0.01
of	0.06	0.01
only	0.01	0.03
planet	0.01	0.07
rodents	0.03	0.01
saturn	0.01	0.03
second	0.01	0.05
sixth	0.01	0.03
small	0.03	0.01
smallest	0.01	0.03
solar	0.01	0.05
species	0.03	0.01
sun	0.01	0.05
system	0.01	0.05
than	0.01	0.03
the	0.08	0.17
throughout	0.03	0.01
wolf	0.03	0.01

Naive Bayes

- This time we have a base-probability of any word!
- Get probabilities!

animal

word	*probability*
the	0.08
cat	0.03
is	0.06
larger	0.01

planet

word	*probability*
the	0.17
cat	0.01
is	0.05
larger	0.03

- Likelihood:

$$P(\mathrm{x} \mid \mathrm{animal}) = 0.08 \times 0.03 \times 0.06 \times 0.01 = 0.00000144$$

$$P(\mathrm{x} \mid \mathrm{planet}) = 0.17 \times 0.01 \times 0.05 \times 0.03 = 0.00000255$$

- Multiply by prior:

$$P(\mathrm{animal} \mid \mathrm{x}) \propto P(\mathrm{x} \mid \mathrm{animal}) \times P(\mathrm{animal})$$

$$= 0.00000144 \times \frac{3}{5} = 0.00000086$$

$$P(\mathrm{planet} \mid \mathrm{x}) \propto P(\mathrm{x} \mid \mathrm{planet}) \times P(\mathrm{planet})$$

$$= 0.00000255 \times \frac{2}{5} = 0.00000102$$

Word probabilities in classes

	animal	planet
africa	0.03	0.01
and	0.03	0.05
are	0.03	0.01
asia	0.03	0.01
being	0.01	0.03
carnivorous	0.03	0.01
cat	0.03	0.01
descendant	0.03	0.01
desert	0.03	0.01
dog	0.03	0.01
domestic	0.06	0.01
domesticated	0.03	0.01
found	0.03	0.01
fourth	0.01	0.03
from	0.01	0.05
hopping	0.03	0.01
in	0.01	0.05
is	0.06	0.05
jerboas	0.03	0.01
larger	0.01	0.03
largest	0.01	0.03
mammal	0.03	0.01
mars	0.01	0.03
mercury	0.01	0.03
northern	0.03	0.01
of	0.06	0.01
only	0.01	0.03
planet	0.01	0.07
rodents	0.03	0.01
saturn	0.01	0.03
second	0.01	0.05
sixth	0.01	0.03
small	0.03	0.01
smallest	0.01	0.03
solar	0.01	0.05
species	0.03	0.01
sun	0.01	0.05
system	0.01	0.05
than	0.01	0.03
the	0.08	0.17
throughout	0.03	0.01
wolf	0.03	0.01

- The "added probability" is called α and can be interpreted as *having seen any word some small number of times in all classes*
- For example if $\alpha = 1$ then the interpretation is that any class has seen any word once + the number of occurrences in our dataset

Machine Learning Theory

Probabilities

- They are great, because we understand what they mean and then allow us to show/prove cool things!

Probabilities

- They are great, because we understand what they mean and then allow us to show/prove cool things!
- They can be difficult to do computations with

Probabilities

- They are great, because we understand what they mean and then allow us to show/prove cool things!
- They can be difficult to do computations with
- Example

Probabilities

- They are great, because we understand what they mean and then allow us to show/prove cool things!
- They can be difficult to do computations with
- Example
  - You have a vocabulary of 10,000 words

Probabilities

- They are great, because we understand what they mean and then allow us to show/prove cool things!
- They can be difficult to do computations with
- Example
  - You have a vocabulary of 10,000 words
  - You have a document with 1,000 words

Probabilities

- They are great, because we understand what they mean and then allow us to show/prove cool things!
- They can be difficult to do computations with
- Example
  - You have a vocabulary of 10,000 words
  - You have a document with 1,000 words
  - The average probability of a word is $\frac{1}{10000}$

Probabilities

- They are great, because we understand what they mean and then allow us to show/prove cool things!
- They can be difficult to do computations with
- Example
  - You have a vocabulary of 10,000 words
  - You have a document with 1,000 words
  - The average probability of a word is $\frac{1}{10000}$
  - The Naive Bayes probability of the document is something like

$$P(\mathcal{D} \mid \mathcal{M}) = \left(\frac{1}{10000}\right)^{1000} \approx \frac{1}{1.134 \times 10^{602}}$$

Probabilities

- They are great, because we understand what they mean and then allow us to show/prove cool things!
- They can be difficult to do computations with
- Example
    - You have a vocabulary of 10,000 words
    - You have a document with 1,000 words
    - The average probability of a word is $\frac{1}{10000}$
    - The Naive Bayes probability of the document is something like

$$P(\mathcal{D} \mid \mathcal{M}) = \left(\frac{1}{10000}\right)^{1000} \approx \frac{1}{1.134 \times 10^{602}}$$

    - There are $\approx 3.28 \times 10^{80}$ particles in the universe

Probabilities

- They are great, because we understand what they mean and then allow us to show/prove cool things!
- They can be difficult to do computations with
- Example
  - You have a vocabulary of 10,000 words
  - You have a document with 1,000 words
  - The average probability of a word is $\frac{1}{10000}$
  - The Naive Bayes probability of the document is something like

$$P(\mathcal{D} \mid \mathcal{M}) = \left(\frac{1}{10000}\right)^{1000} \approx \frac{1}{1.134 \times 10^{602}}$$

  - There are $\approx 3.28 \times 10^{80}$ particles in the universe
  - Any computer working with that number will say 0: underflow

Probabilities

- They are great, because we understand what they mean and then allow us to show/prove cool things!
- They can be difficult to do computations with
- Example
  - You have a vocabulary of 10,000 words
  - You have a document with 1,000 words
  - The average probability of a word is $\frac{1}{10000}$
  - The Naive Bayes probability of the document is something like

$$P(\mathcal{D} \mid \mathcal{M}) = \left(\frac{1}{10000}\right)^{1000} \approx \frac{1}{1.134 \times 10^{602}}$$

  - There are $\approx 3.28 \times 10^{80}$ particles in the universe
  - Any computer working with that number will say 0: underflow
  - We divide by something that is suuuuper small: everything explodes

Log-probabilities

- Solution: we work in the log-domain

$$\begin{aligned}
\log\left[\left(\frac{1}{10000}\right)^{1000}\right] &= 1000\log\left(\frac{1}{10000}\right)\\
&= 1000(\log(1) - \log(1000))\\
&= 1000(0 - 9.9658)\\
&= -9965.8
\end{aligned}$$

Log-probabilities

- Solution: we work in the log-domain

$$\log\left[\left(\frac{1}{10000}\right)^{1000}\right] = 1000 \log\left(\frac{1}{10000}\right)$$
$$= 1000(\log(1) - \log(1000))$$
$$= 1000(0 - 9.9658)$$
$$= -9965.8$$

  - Small numbers becomes large (in absolute terms) negative numbers

Log-probabilities

- Solution: we work in the log-domain

$$\begin{aligned}\log\left[\left(\frac{1}{10000}\right)^{1000}\right] &= 1000\log\left(\frac{1}{10000}\right)\\ &= 1000(\log(1) - \log(1000))\\ &= 1000(0 - 9.9658)\\ &= -9965.8\end{aligned}$$

- Small numbers becomes large (in absolute terms) negative numbers
- Models like Naive Bayes can be implemented in log-probabilities

Log-probabilities

- Solution: we work in the log-domain

$$
\begin{aligned}
\log\left[\left(\frac{1}{10000}\right)^{1000}\right] &= 1000 \log\left(\frac{1}{10000}\right) \\
&= 1000(\log(1) - \log(1000)) \\
&= 1000(0 - 9.9658) \\
&= -9965.8
\end{aligned}
$$

- Small numbers becomes large (in absolute terms) negative numbers
- Models like Naive Bayes can be implemented in log-probabilities
- Our computers like this :)

Log-probabilities

- Solution: we work in the log-domain

$$
\begin{aligned}
\log\left[\left(\frac{1}{10000}\right)^{1000}\right] &= 1000\log\left(\frac{1}{10000}\right) \\
&= 1000(\log(1) - \log(1000)) \\
&= 1000(0 - 9.9658) \\
&= -9965.8
\end{aligned}
$$

  - Small numbers becomes large (in absolute terms) negative numbers
  - Models like Naive Bayes can be implemented in log-probabilities
  - Our computers like this :)
- There is an information theoretical reason for log-probabilities

Log-probabilities

- Solution: we work in the log-domain

$$\begin{aligned} \log\left[\left(\frac{1}{10000}\right)^{1000}\right] &= 1000 \log\left(\frac{1}{10000}\right) \\ &= 1000(\log(1) - \log(1000)) \\ &= 1000(0 - 9.9658) \\ &= -9965.8 \end{aligned}$$

  - Small numbers becomes large (in absolute terms) negative numbers
  - Models like Naive Bayes can be implemented in log-probabilities
  - Our computers like this :)
- There is an information theoretical reason for log-probabilities
- You will often with with terms like: log-likelihood, log-posterior

$$\log P(\mathcal{M} \mid \mathcal{D}) \propto \log P(\mathcal{D} \mid \mathcal{M}) + \log P(\mathcal{M})$$

Machine Learning Theory

- Consider again the (bad) model with $\alpha = 0$

Word probabilities in classes

	animal	planet
africa	0.03	0.00
and	0.03	0.05
are	0.03	0.00
asia	0.03	0.00
being	0.00	0.03
carnivorous	0.03	0.00
cat	0.03	0.00
descendant	0.03	0.00
desert	0.03	0.00
dog	0.03	0.00
domestic	0.07	0.00
domesticated	0.03	0.00
found	0.03	0.00
fourth	0.00	0.03
from	0.00	0.05
hopping	0.03	0.00
in	0.00	0.05
is	0.07	0.05
jerboas	0.03	0.00
larger	0.00	0.03
largest	0.00	0.03
mammal	0.03	0.00
mars	0.00	0.03
mercury	0.00	0.03
northern	0.03	0.00
of	0.07	0.00
only	0.00	0.03
planet	0.00	0.08
rodents	0.03	0.00
saturn	0.00	0.03
second	0.00	0.05
sixth	0.00	0.03
small	0.03	0.00
smallest	0.00	0.03
solar	0.00	0.05
species	0.03	0.00
sun	0.00	0.05
system	0.00	0.05
than	0.00	0.03
the	0.10	0.21
throughout	0.03	0.00
wolf	0.03	0.00

Machine Learning Theory

- Consider again the (bad) model with $\alpha = 0$
- One of the original samples is: *The domestic dog is a domesticated descendant of the wolf.*

Word probabilities in classes

	animal	planet
africa	0.03	0.00
and	0.03	0.05
are	0.03	0.00
asia	0.03	0.00
being	0.00	0.03
carnivorous	0.03	0.00
cat	0.03	0.00
descendant	0.03	0.00
desert	0.03	0.00
dog	0.03	0.00
domestic	0.07	0.00
domesticated	0.03	0.00
found	0.03	0.00
fourth	0.00	0.03
from	0.00	0.05
hopping	0.03	0.00
in	0.00	0.05
is	0.07	0.05
jerboas	0.03	0.00
larger	0.00	0.03
largest	0.00	0.03
mammal	0.03	0.00
mars	0.00	0.03
mercury	0.00	0.03
northern	0.03	0.00
of	0.07	0.00
only	0.00	0.03
planet	0.00	0.08
rodents	0.03	0.00
saturn	0.00	0.03
second	0.00	0.05
sixth	0.00	0.03
small	0.03	0.00
smallest	0.00	0.03
solar	0.00	0.05
species	0.03	0.00
sun	0.00	0.05
system	0.00	0.05
than	0.00	0.03
the	0.10	0.21
throughout	0.03	0.00
wolf	0.03	0.00

Machine Learning Theory

- Consider again the (bad) model with $\alpha = 0$
- One of the original samples is: *The domestic dog is a domesticated descendant of the wolf.*
- All these words have probability > 0

Word probabilities in classes

	animal	planet
africa	0.03	0.00
and	0.03	0.05
are	0.03	0.00
asia	0.03	0.00
being	0.00	0.03
carnivorous	0.03	0.00
cat	0.03	0.00
descendant	0.03	0.00
desert	0.03	0.00
dog	0.03	0.00
domestic	0.07	0.00
domesticated	0.03	0.00
found	0.03	0.00
fourth	0.00	0.03
from	0.00	0.05
hopping	0.03	0.00
in	0.00	0.05
is	0.07	0.05
jerboas	0.03	0.00
larger	0.00	0.03
largest	0.00	0.03
mammal	0.03	0.00
mars	0.00	0.03
mercury	0.00	0.03
northern	0.03	0.00
of	0.07	0.00
only	0.00	0.03
planet	0.00	0.08
rodents	0.03	0.00
saturn	0.00	0.03
second	0.00	0.05
sixth	0.00	0.03
small	0.03	0.00
smallest	0.00	0.03
solar	0.00	0.05
species	0.03	0.00
sun	0.00	0.05
system	0.00	0.05
than	0.00	0.03
the	0.10	0.21
throughout	0.03	0.00
wolf	0.03	0.00

- Consider again the (bad) model with $\alpha = 0$
- One of the original samples is: *The domestic dog is a domesticated descendant of the wolf.*
- All these words have probability > 0
- The model works fine on the training data - so what's the problem?

Word probabilities in classes

	animal	planet
africa	0.03	0.00
and	0.03	0.05
are	0.03	0.00
asia	0.03	0.00
being	0.00	0.03
carnivorous	0.03	0.00
cat	0.03	0.00
descendant	0.03	0.00
desert	0.03	0.00
dog	0.03	0.00
domestic	0.07	0.00
domesticated	0.03	0.00
found	0.03	0.00
fourth	0.00	0.03
from	0.00	0.05
hopping	0.03	0.00
in	0.00	0.05
is	0.07	0.05
jerboas	0.03	0.00
larger	0.00	0.03
largest	0.00	0.03
mammal	0.03	0.00
mars	0.00	0.03
mercury	0.00	0.03
northern	0.03	0.00
of	0.07	0.00
only	0.00	0.03
planet	0.00	0.08
rodents	0.03	0.00
saturn	0.00	0.03
second	0.00	0.05
sixth	0.00	0.03
small	0.03	0.00
smallest	0.00	0.03
solar	0.00	0.05
species	0.03	0.00
sun	0.00	0.05
system	0.00	0.05
than	0.00	0.03
the	0.10	0.21
throughout	0.03	0.00
wolf	0.03	0.00

- This is one of the most fundamental problems of machine learning

Machine Learning Theory

- This is one of the most fundamental problems of machine learning
- We wish to have a model $\mathcal{M}$ where

$$\int \ell[\mathcal{M}(x), t] \cdot p(x, t) \, dx \, dt \tag{2}$$

is as small as possible

- This is one of the most fundamental problems of machine learning
- We wish to have a model $\mathcal{M}$ where

$$\int \ell[\mathcal{M}(\mathrm{x}), \mathrm{t}] \cdot p(\mathrm{x}, \mathrm{t}) \, \mathrm{dx} \, \mathrm{dt} \tag{2}$$

  is as small as possible
- We call this the *generalization error*

- This is one of the most fundamental problems of machine learning
- We wish to have a model $\mathcal{M}$ where

$$\int \ell[\mathcal{M}(\mathrm{x}), \mathrm{t}] \cdot p(\mathrm{x}, \mathrm{t}) \, \mathrm{dx} \, \mathrm{dt} \tag{2}$$

  is as small as possible
- We call this the *generalization error*
- A model with low generalization error *generalizes well*

- This is one of the most fundamental problems of machine learning
- We wish to have a model $\mathcal{M}$ where

$$\int \ell[\mathcal{M}(\mathrm{x}), \mathrm{t}] \cdot p(\mathrm{x}, \mathrm{t}) \, \mathrm{dx} \, \mathrm{dt} \tag{2}$$

  is as small as possible
- We call this the *generalization error*
- A model with low generalization error *generalizes well*
- How do we compute the generalization error? - we can't

- This is one of the most fundamental problems of machine learning
- We wish to have a model $\mathcal{M}$ where

$$\int \ell[\mathcal{M}(\mathrm{x}), \mathrm{t}] \cdot p(\mathrm{x}, \mathrm{t}) \, \mathrm{dx} \, \mathrm{dt} \tag{2}$$

  is as small as possible
- We call this the *generalization error*
- A model with low generalization error *generalizes well*
- How do we compute the generalization error? - we can't
- We approximate by

$$\int \ell[\mathcal{M}(\mathrm{x}), \mathrm{t}] \cdot p(\mathrm{x}, \mathrm{t}) \, \mathrm{dx} \, \mathrm{dt} \approx \sum_{(\mathrm{x}_i, \mathrm{t}_i) \in \mathcal{D}_{\mathrm{test}}} \ell[\mathcal{M}(\mathrm{x}_i), \mathrm{t}_i] \tag{3}$$

- We split our data into:

Training set

Used for optimizing our algorithms.
Abuse this as much as you want :)

Test set

Used for testing our algorithms.
Can be used for ABSOLUTELY NOTHING else.
Must represent new data.

- We split our data into:

Training set

Used for optimizing our algorithms.
Abuse this as much as you want :)

Validation set (optional)

Used for validation our algorithms during training.
For example for deciding between 3 competing algorithms and tuning hyperparameters.

Test set

Used for testing our algorithms.
Can be used for ABSOLUTELY NOTHING else.
Must represent new data.

- We split our data into:

Training set	Validation set (optional)	Test set
Used for optimizing our algorithms. Abuse this as much as you want :)	Used for validation our algorithms during training. For example for deciding between 3 competing algorithms and tuning hyperparameters.	Used for testing our algorithms. Can be used for ABSOLUTELY NOTHING else. Must represent new data.

- People make improper split quite often - and it is really not a good idea

- We split our data into:

Training set

Used for optimizing our algorithms.
Abuse this as much as you want :)

Validation set (optional)

Used for validation our algorithms during training.
For example for deciding between 3 competing algorithms and tuning hyperparameters.

Test set

Used for testing our algorithms.
Can be used for ABSOLUTELY NOTHING else.
Must represent new data.

- People make improper split quite often - and it is really not a good idea
- You will almost ALWAYS see something like 0.96 performance for training set and 0.91 for test set

- We split our data into:

Training set

Used for optimizing our algorithms.
Abuse this as much as you want :)

Validation set (optional)

Used for validation our algorithms during training.
For example for deciding between 3 competing algorithms and tuning hyperparameters.

Test set

Used for testing our algorithms.
Can be used for ABSOLUTELY NOTHING else.
Must represent new data.

- People make improper split quite often - and it is really not a good idea
- You will almost ALWAYS see something like 0.96 performance for training set and 0.91 for test set
    - If they are switched, I would assume there is an error in your model/implementation

- We split our data into:

Training set

Used for optimizing our algorithms.
Abuse this as much as you want :)

Validation set (optional)

Used for validation our algorithms during training.
For example for deciding between 3 competing algorithms and tuning hyperparameters.

Test set

Used for testing our algorithms.
Can be used for ABSOLUTELY NOTHING else.
Must represent new data.

- People make improper split quite often - and it is really not a good idea
- You will almost ALWAYS see something like 0.96 performance for training set and 0.91 for test set
    - If they are switched, I would assume there is an error in your model/implementation (technically it can happen by luck, but almost never will)

- We split our data into:

Training set	Validation set (optional)	Test set
Used for optimizing our algorithms. Abuse this as much as you want :)	Used for validation our algorithms during training. For example for deciding between 3 competing algorithms and tuning hyperparameters.	Used for testing our algorithms. Can be used for ABSOLUTELY NOTHING else. Must represent new data.

- People make improper split quite often - and it is really not a good idea
- You will almost ALWAYS see something like 0.96 performance for training set and 0.91 for test set
  - If they are switched, I would assume there is an error in your model/implementation (technically it can happen by luck, but almost never will)
  - We have methods for attempting to close the gap between training and test performance

- We split our data into:

Training set

Used for optimizing our algorithms.
Abuse this as much as you want :)

Validation set (optional)

Used for validation our algorithms during training.
For example for deciding between 3 competing algorithms and tuning hyperparameters.

Test set

Used for testing our algorithms.
Can be used for ABSOLUTELY NOTHING else.
Must represent new data.

- People make improper split quite often - and it is really not a good idea
- You will almost ALWAYS see something like 0.96 performance for training set and 0.91 for test set
  - If they are switched, I would assume there is an error in your model/implementation (technically it can happen by luck, but almost never will)
  - We have methods for attempting to close the gap between training and test performance
- Training and test set must always come from the same distribution

Mini-quiz

- I have a collection of images of dogs and cats. I shuffle all the images and split them into training and test set - good enough?

Mini-quiz

- I have a collection of images of dogs and cats. I shuffle all the images and split them into training and test set - good enough?
- I have a training set with users from London and a test set with users from Copenhagen - good enough?

Mini-quiz

- I have a collection of images of dogs and cats. I shuffle all the images and split them into training and test set - good enough?
- I have a training set with users from London and a test set with users from Copenhagen - good enough?
- I have a collection of Amazon users and a their item-reviews. I shuffle all reviews and split them into training and test set - good enough?

Mini-quiz

- I have a collection of images of dogs and cats. I shuffle all the images and split them into training and test set - good enough?
- I have a training set with users from London and a test set with users from Copenhagen - good enough?
- I have a collection of Amazon users and a their item-reviews. I shuffle all reviews and split them into training and test set - good enough?
- I want to predict weather 1 month in advance. I have 1 year of data. I shuffle all days and split them into training and test set - good enough?

Mini-quiz

- I have a collection of images of dogs and cats. I shuffle all the images and split them into training and test set - good enough?
- I have a training set with users from London and a test set with users from Copenhagen - good enough?
- I have a collection of Amazon users and a their item-reviews. I shuffle all reviews and split them into training and test set - good enough?
- I want to predict weather 1 month in advance. I have 1 year of data. I shuffle all days and split them into training and test set - good enough?
- You have 3 models trained. You evaluate them using the test data. One is better than the others so you wish to use that one and report its test-score - good enough?

Mini-quiz

- I have a collection of images of dogs and cats. I shuffle all the images and split them into training and test set - good enough?
- I have a training set with users from London and a test set with users from Copenhagen - good enough?
- I have a collection of Amazon users and a their item-reviews. I shuffle all reviews and split them into training and test set - good enough?
- I want to predict weather 1 month in advance. I have 1 year of data. I shuffle all days and split them into training and test set - good enough?
- You have 3 models trained. You evaluate them using the test data. One is better than the others so you wish to use that one and report its test-score - good enough?
  - How about 10 models?

Mini-quiz

- I have a collection of images of dogs and cats. I shuffle all the images and split them into training and test set - good enough?
- I have a training set with users from London and a test set with users from Copenhagen - good enough?
- I have a collection of Amazon users and a their item-reviews. I shuffle all reviews and split them into training and test set - good enough?
- I want to predict weather 1 month in advance. I have 1 year of data. I shuffle all days and split them into training and test set - good enough?
- You have 3 models trained. You evaluate them using the test data. One is better than the others so you wish to use that one and report its test-score - good enough?
    - How about 10 models?
    - How about 100 models?

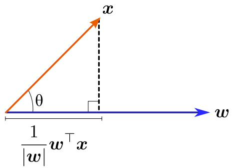
Mini-quiz

- I have a collection of images of dogs and cats. I shuffle all the images and split them into training and test set - good enough?
- I have a training set with users from London and a test set with users from Copenhagen - good enough?
- I have a collection of Amazon users and a their item-reviews. I shuffle all reviews and split them into training and test set - good enough?
- I want to predict weather 1 month in advance. I have 1 year of data. I shuffle all days and split them into training and test set - good enough?
- You have 3 models trained. You evaluate them using the test data. One is better than the others so you wish to use that one and report its test-score - good enough?
  - How about 10 models?
  - How about 100 models?
  - How about 10000 models?

Perceptron

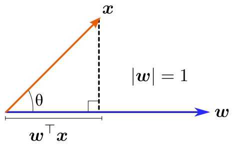
Perceptron

- Component 1: dot-product



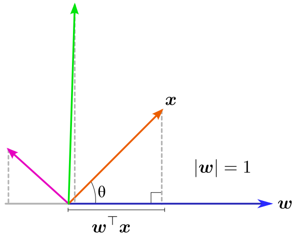
Perceptron

- Component 1: dot-product
  - Dot-product measures how much a vector "aligns" with a weight vector



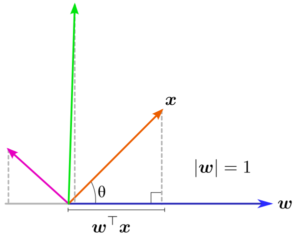
Perceptron

- Component 1: dot-product
  - Dot-product measures how much a vector "aligns" with a weight vector



Perceptron

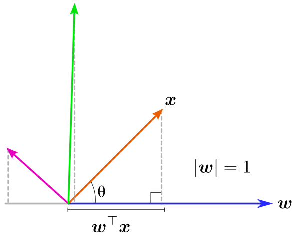
- Component 1: dot-product
  - Dot-product measures how much a vector "aligns" with a weight vector
- Component 2: step function



Perceptron

- Component 1: dot-product
  - Dot-product measures how much a vector "aligns" with a weight vector

- Component 2: step function
  - Returns the sign of a value

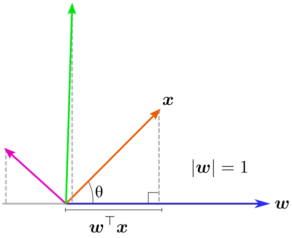


Perceptron

- Component 1: dot-product
  - Dot-product measures how much a vector "aligns" with a weight vector
- Component 2: step function
  - Returns the sign of a value
- We model the predicted class to be

$$y(\mathrm{x}) = \mathrm{sign}(\mathrm{w}^T\mathrm{x})$$

$$\mathrm{sign}(a) = \begin{cases} 1 & a > 0 \\ -1 & a < 0 \end{cases}$$



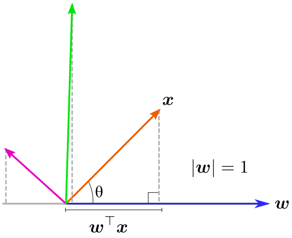
Perceptron

- Component 1: dot-product
  - Dot-product measures how much a vector "aligns" with a weight vector
- Component 2: step function
  - Returns the sign of a value
- We model the predicted class to be

$$y(\mathrm{x}) = \mathrm{sign}(\mathrm{w}^T \mathrm{x})$$

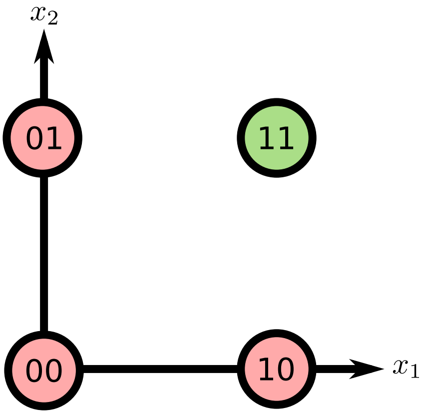
$$\mathrm{sign}(a) = \begin{cases} 1 & a > 0 \\ -1 & a < 0 \end{cases}$$

- What about 0? - do what you want



Example - **and**-operator

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

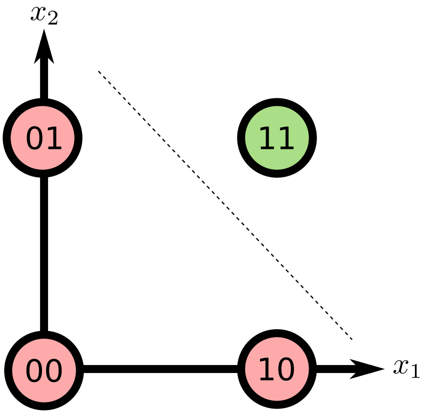


Perceptron

Example - **and**-operator

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

$$y = \sigma(w_0 + w_1 x_1 + w_2 x_2)$$



Example - **and**-operator

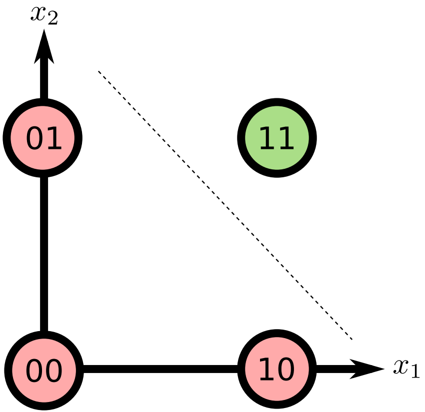
x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

$$y = \sigma(w_0 + w_1 x_1 + w_2 x_2)$$

$$w_0 = -15$$

$$w_2 = 10$$

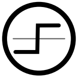
$$w_1 = 10$$

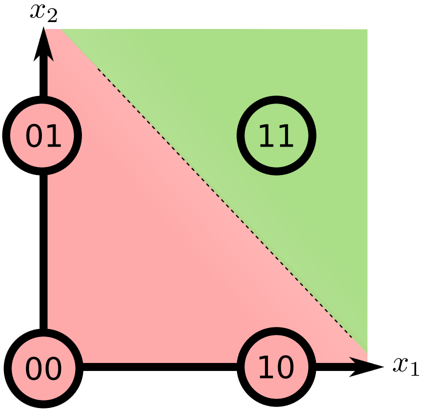


Perceptron

Example - **and**-operator

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1





$$y = f(w_0 + w_1 x_1 + w_2 x_2)$$

$$w_0 = -15$$

$$w_2 = 10$$

$$w_1 = 10$$

Perceptron

Example - **or**-operator

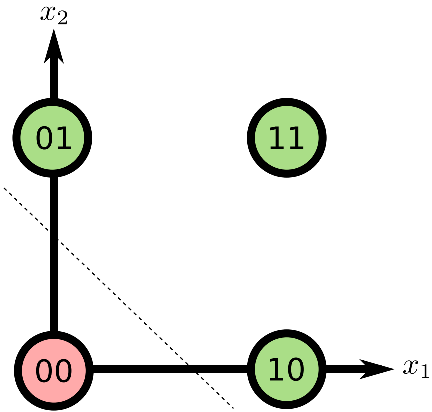
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

$$y = \sigma(w_0 + w_1 x_1 + w_2 x_2)$$

$$w_0 = -5$$

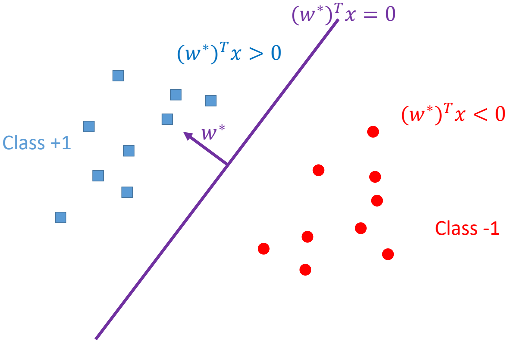
$$w_2 = 10$$

$$w_1 = 10$$



Perceptron

Task

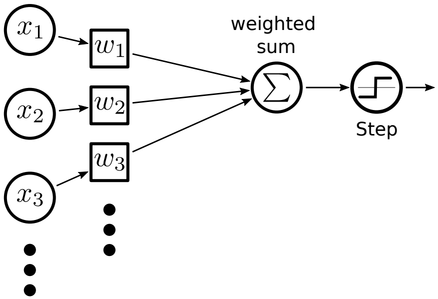


Perceptron

Graphical depiction of perceptron:

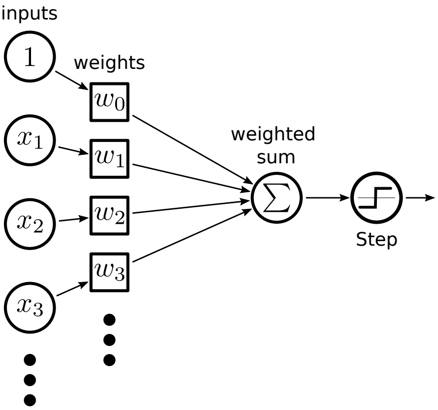
inputs

weights



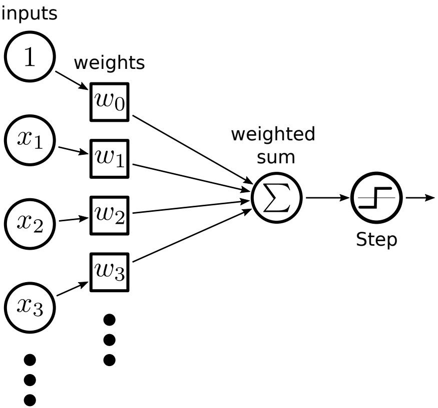
Perceptron

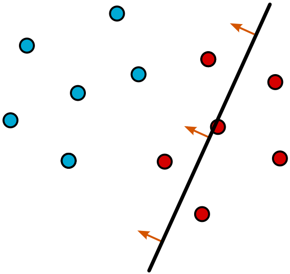
Graphical depiction of perceptron:



Perceptron

Graphical depiction of perceptron:





Perceptron

Learning

We wish to optimize this perceptron somehow
(we don't want to set the weights manually)

Learning

We wish to optimize this perceptron somehow
(we don't want to set the weights manually)

Perceptron learning algorithm:

1. Randomly initialize w

Learning

We wish to optimize this perceptron somehow
(we don't want to set the weights manually)

Perceptron learning algorithm:

1. Randomly initialize w
2. For each *epoch*

Perceptron

Learning

We wish to optimize this perceptron somehow
(we don't want to set the weights manually)

Perceptron learning algorithm:

1. Randomly initialize w
2. For each *epoch*
   - For each sample (x_i, t_i)

Perceptron

Learning

We wish to optimize this perceptron somehow
(we don't want to set the weights manually)

Perceptron learning algorithm:

1. Randomly initialize w
2. For each *epoch*
   - For each sample (x_i, t_i)
     - Compute output: $y_i = \mathrm{sign}(\mathrm{w}^T \mathrm{x}_i)$

Learning

We wish to optimize this perceptron somehow
(we don't want to set the weights manually)

Perceptron learning algorithm:

1. Randomly initialize w
2. For each *epoch*
   - For each sample (x_i, t_i)
     - Compute output: $y_i = \mathrm{sign}(\mathrm{w}^T \mathrm{x}_i)$
     - Update weights: $\mathrm{w} \leftarrow \mathrm{w} + \eta \times (t_i - y_i)\mathrm{x}$

Perceptron

Learning

We wish to optimize this perceptron somehow
(we don't want to set the weights manually)

Perceptron learning algorithm:

1. Randomly initialize w
2. For each *epoch*
    - For each sample (x_i, t_i)
        - Compute output: $y_i = \mathrm{sign}(\mathrm{w}^T \mathrm{x}_i)$
        - Update weights: $\mathrm{w} \leftarrow \mathrm{w} + \eta \times (t_i - y_i)\mathrm{x}$

Perceptron

Learning

We wish to optimize this perceptron somehow
(we don't want to set the weights manually)

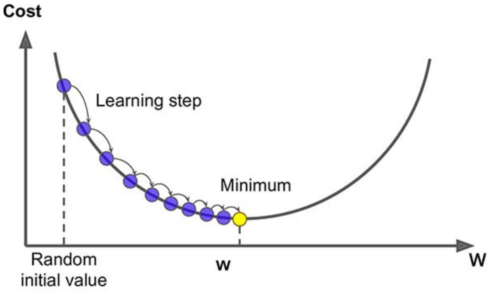
Perceptron learning algorithm:

1. Randomly initialize w
2. For each *epoch*
   - For each sample (x_i, t_i)
     - Compute output: $y_i = \mathrm{sign}(\mathrm{w}^T \mathrm{x}_i)$
     - Update weights: $\mathrm{w} \leftarrow \mathrm{w} + \eta \times (t_i - y_i)\mathrm{x}$

- Epoch: one training-iteration through entire dataset
- η: learning rate

Perceptron

Slowly get better and better solutions



Exercises

Exercise

- Script ex_2_1.py
  - Complete the implementation of Naive Bayes
  - Use log-probabilities to avoid underflow
  - Predict the probabilities of the test-set
  - How does α affect the prediction
- Script ex_2_2.py
  - Use sklearn's MultinomialNB and CountVectorizer to fit a Naive Bayes model on the 20-newsgroups dataset
  - Plot a visualization of the confusion matrix of the model's performance on the test set
  - Run predictions on the extra documents
  - Can you determine the best value for α?
- Script ex_2_3.py
  - Implement update-rule for perceptron to fit 2D problem
    - Remember to have a constant 1 as "first feature"
  - Can you fit to problems with more dimensions as well?
  - What is the decision boundary for the perceptron in higher dimensions?