Application: Naive Bayes Classifiers

Table of Contents

Predicting the world

Things we may want to do with our computer-science powers:

Predicting the world

Things we may want to do with our computer-science powers:

- Diagnose if a person has the flu/a cold/nothing based on their symptoms.

Predicting the world

Things we may want to do with our computer-science powers:

- Diagnose if a person has the flu/a cold/nothing based on their symptoms.
- Filter emails based on whether or not they are likely to be spam.

Classification problems

A lot of problems can be expressed in terms of **input** data and **output** labels...

Classification problems

A lot of problems can be expressed in terms of **input** data and **output** labels...

- Data: A list of symptoms for each patient.
  Labels: Flu/Cold/Healthy.

Classification problems

A lot of problems can be expressed in terms of **input** data and **output** labels...

- Data: A list of symptoms for each patient.
  Labels: Flu/Cold/Healthy.
- Data: The words constituting an email.
  Labels: Spam/Ham.

Classification problems

A lot of problems can be expressed in terms of **input** data and **output** labels...

- Data: A list of symptoms for each patient.
  Labels: Flu/Cold/Healthy.
- Data: The words constituting an email.
  Labels: Spam/Ham.

Definition

A **classification problem** is one where we classify data points by giving each one a label.

Classification problems (continued)

What are we actually doing, though?

Classification problems (continued)

What are we actually doing, though?

- Which is most likely, flu/cold/healthy, given the symptoms of a patient?

Classification problems (continued)

What are we actually doing, though?

- Which is most likely, flu/cold/healthy, given the symptoms of a patient?
- Which is most likely, spam/ham, given the words in an email?

Table of Contents

A philosophical question...

How does the world work?

Let "$\rightarrow$" suggest causation:

A philosophical question...

How does the world work?

Let "$\rightarrow$" suggest causation:

- Sickness (the disease/virus) $\rightarrow$ Symptoms

  OR

  Symptoms $\rightarrow$ Sickness (the name)

A philosophical question...

How does the world work?

Let "$\rightarrow$" suggest causation:

Sickness (the disease/virus) $\rightarrow$ Symptoms

- OR

Symptoms $\rightarrow$ Sickness (the name)

Spam (the intent) $\rightarrow$ Email

- OR

Email $\rightarrow$ Spam (the meaning)

A philosophical question... (continued)

It's not clear what direction causality goes!

A philosophical question... (continued)

It's not clear what direction causality goes!

In practice, use whichever works better/faster/whatever you care about.

You're sick!

Let's assume that the fact that someone has the flu causes them to manifest certain symptoms, i.e. Sickness $\rightarrow$ Symptoms.

You're sick!

Let's assume that the fact that someone has the flu causes them to manifest certain symptoms, i.e. Sickness $\rightarrow$ Symptoms.

We'd expect

You're sick!

Let's assume that the fact that someone has the flu causes them to manifest certain symptoms, i.e. Sickness $\rightarrow$ Symptoms.

We'd expect

- Healthy people do not have fevers

You're sick!

Let's assume that the fact that someone has the flu causes them to manifest certain symptoms, i.e. Sickness $\rightarrow$ Symptoms.

We'd expect

- Healthy people do not have fevers
- People with the flu have upset stomachs and sore throats

You're sick!

Let's assume that the fact that someone has the flu causes them to manifest certain symptoms, i.e. Sickness $\rightarrow$ Symptoms.

We'd expect

- Healthy people do not have fevers
- People with the flu have upset stomachs and sore throats
- People with a cold have a runny nose and a cough.

You're sick! (continued)

We could write code to try and model this:

Example

```python
def diagnose(symptoms):
    diagnoses = {'flu', 'cold', 'healthy'}
    if 'fever' in symptoms:
        # Patient has a fever, can't be healthy
        diagnoses.remove('healthy')
    if 'upset stomach' and 'sore throat' not in symptoms:
        # Patient doesn't have an upset stomach or a sore
        # throat, can't have the flu
        diagnoses.remove('flu')
    if 'runny nose' and 'cough' not in symptoms:
        # Patient doesn't have a runny nose or a cough,
        # can't have a cold
        diagnoses.remove('cold')
    return diagnoses
```

You're sick! (continued)

We could write code to try and model this:

Example

```
def diagnose(symptoms):
    diagnoses = {'flu', 'cold', 'healthy'}
    if 'fever' in symptoms:
        # Patient has a fever, can't be healthy
        diagnoses.remove('healthy')
    if 'upset stomach' and 'sore throat' not in symptoms:
        # Patient doesn't have an upset stomach or a sore
        # throat, can't have the flu
        diagnoses.remove('flu')
    if 'runny nose' and 'cough' not in symptoms:
        # Patient doesn't have a runny nose or a cough,
        # can't have a cold
        diagnoses.remove('cold')
    return diagnoses
```

This... sucks. Sometimes we're going to have multiple diagnoses we can't differentiate, sometimes we'll eliminate all the diagnoses.

You're sick! (continued)

We're taking 312 – we can deal with this using *probability*!

Example

```
def diagnose(symptoms):
    diagnoses = {'flu' : 1/3, 'cold' : 1/3, 'healthy' : 1/3}
    if 'fever' in symptoms:
        # Patient has a fever, very unlikely to be healthy
        diagnoses['healthy'] /= 5.0
        diagnoses.normalize()
    if 'upset stomach' and 'sore throat' not in symptoms:
        # Patient doesn't have an upset stomach or a sore
        # throat, unlikely to have the flu
        diagnoses['flu'] /= 2.0
        diagnoses.normalize()
    if 'runny nose' and 'cough' not in symptoms:
        # Patient doesn't have a runny nose or a cough
        # unlikely to have a cold
        diagnoses['cold'] /= 2.0
        diagnoses.normalize()
    return diagnoses
```

You're sick! (continued)

We're taking 312 – we can deal with this using *probability*!

Example

```python
def diagnose(symptoms):
    diagnoses = {'flu' : 1/3, 'cold' : 1/3, 'healthy' : 1/3}
    if 'fever' in symptoms:
        # Patient has a fever, very unlikely to be healthy
        diagnoses['healthy'] /= 5.0
        diagnoses.normalize()
    if 'upset stomach' and 'sore throat' not in symptoms:
        # Patient doesn't have an upset stomach or a sore
        # throat, unlikely to have the flu
        diagnoses['flu'] /= 2.0
        diagnoses.normalize()
    if 'runny nose' and 'cough' not in symptoms:
        # Patient doesn't have a runny nose or a cough
        # unlikely to have a cold
        diagnoses['cold'] /= 2.0
        diagnoses.normalize()
    return diagnoses
```

Instead of making it impossible for healthy people to have fevers, we just make it less likely the patient is healthy when they have a fever!

You're sick! (continued)

Example

```python
def diagnose(symptoms):
    diagnoses = {'flu' : 1/3, 'cold' : 1/3, 'healthy' : 1/3}
    if 'fever' in symptoms:
        # Patient has a fever, very unlikely to be healthy
        diagnoses['healthy'] /= 5
        diagnoses.normalize()
    if 'upset stomach' and 'sore throat' not in symptoms:
        # Patient doesn't have an upset stomach or a sore
        # throat, unlikely to have the flu
        diagnoses['flu'] /= 2
        diagnoses.normalize()
    if 'runny nose' and 'cough' not in symptoms:
        # Patient doesn't have a runny nose or a cough
        # unlikely to have a cold
        diagnoses['cold'] /= 2
        diagnoses.normalize()
    return diagnoses
```

Let's enhance our model to check all possible combinations of symptoms. How many if-statements do we have to write?

Intractable

Welp... We're screwed.

Intractable

Welp... We're screwed.

5 symptoms means $2^5 = 32$ if-statements to check all combinations. 10 symptoms would be 1024 if-statements.

Intractable

Welp... We're screwed.

5 symptoms means $2^5 = 32$ if-statements to check all combinations. 10 symptoms would be 1024 if-statements.

There isn't even an upper-bound for the number of words we could have in an email...

Two major problems in statistical modeling

1. We assumed that $\boxed{\text{Sickness} \rightarrow \text{Symptoms}}$, so our model predicts the probability certain symptoms manifest *given* we know what the patient has.

What we have	What we want

Two major problems in statistical modeling

1. We assumed that $\boxed{\text{Sickness} \rightarrow \text{Symptoms}}$, so our model predicts the probability certain symptoms manifest *given* we know what the patient has.

What we have	What we want
Pr(symptoms|diagnosis)	

Two major problems in statistical modeling

1. We assumed that $\boxed{\text{Sickness} \to \text{Symptoms}}$, so our model predicts the probability certain symptoms manifest *given* we know what the patient has.

What we have	What we want
Pr(symptoms|diagnosis)	

We want to choose the highest probability diagnosis, so what we want is to compute Pr(diagnosis|symptoms) for each diagnosis.

Two major problems in statistical modeling

1. We assumed that $\boxed{\text{Sickness} \rightarrow \text{Symptoms}}$, so our model predicts the probability certain symptoms manifest *given* we know what the patient has.

What we have	What we want
Pr(symptoms|diagnosis)	Pr(diagnosis|symptoms)

We want to choose the highest probability diagnosis, so what we want is to compute Pr(diagnosis|symptoms) for each diagnosis.

Two major problems in statistical modeling

1. We assumed that $\boxed{\text{Sickness} \rightarrow \text{Symptoms}}$, so our model predicts the probability certain symptoms manifest *given* we know what the patient has.

What we have	What we want
$\Pr(\text{symptoms}\|\text{diagnosis})$	$\Pr(\text{diagnosis}\|\text{symptoms})$

We want to choose the highest probability diagnosis, so what we want is to compute $\Pr(\text{diagnosis}|\text{symptoms})$ for each diagnosis.

We have the opposite of what we need!

Two major problems in statistical modeling

1. We assumed that $\boxed{\text{Sickness} \rightarrow \text{Symptoms}}$, so our model predicts the probability certain symptoms manifest *given* we know what the patient has.

What we have	What we want
$\Pr(\text{symptoms}\vert\text{diagnosis})$	$\Pr(\text{diagnosis}\vert\text{symptoms})$

   We want to choose the highest probability diagnosis, so what we want is to compute $\Pr(\text{diagnosis}\vert\text{symptoms})$ for each diagnosis.

   We have the opposite of what we need!

2. We need to somehow address all combinations of symptoms and how they affect each diagnosis.

$\Pr(S|D)$ vs. $\Pr(D|S)$

Maximize!

$$\max_{\text{diagnosis}} \Pr(\text{diagnosis}|\text{symptoms})$$

$\Pr(S|D)$ vs. $\Pr(D|S)$

Maximize!

$$\max_{\text{diagnosis}} \Pr(\text{diagnosis}|\text{symptoms})$$

Apply Bayes Rule!

$$\max_{\text{diag}} \Pr(\text{diag}|\text{symp}) = \max_{\text{diag}} \frac{\Pr(\text{symp}|\text{diag}) \cdot \Pr(\text{diag})}{\Pr(\text{symp})}.$$

$\Pr(S|D)$ vs. $\Pr(D|S)$

Maximize!

$$\max_{\text{diagnosis}} \Pr(\text{diagnosis}|\text{symptoms})$$

Apply Bayes Rule!

$$\max_{\text{diag}} \Pr(\text{diag}|\text{symp}) = \max_{\text{diag}} \frac{\Pr(\text{symp}|\text{diag}) \cdot \Pr(\text{diag})}{\Pr(\text{symp})}.$$

Notice that if we want to find the most likely diagnosis for a set of symptoms, we vary the diagnosis and compute probability. Since we don't vary the symptoms, $\Pr(\text{symptoms}) = c$

Pr(S|D) vs. Pr(D|S) continued

Thus, we see that

$$\begin{aligned}\max_{\text{diag}} \Pr(\text{diag}|\text{symp}) &= \max_{\text{diag}} \frac{1}{c} \cdot \Pr(\text{symp}|\text{diag}) \cdot \Pr(\text{diag}) \\ &= \frac{1}{c} \max_{\text{diag}} \Pr(\text{symp}|\text{diag}) \cdot \Pr(\text{diag}).\end{aligned}$$

Pr(D|S) vs. Pr(S|D) continued

What we've derived:

$$\max_{\text{diag}} \Pr(\text{diag}|\text{symp}) = \frac{1}{c} \cdot \max_{\text{diag}} \Pr(\text{symp}|\text{diag}) \cdot \Pr(\text{diag}).$$

In English: To maximize the probability of the diagnosis given the symptoms, choose a diagnosis which is likely to cause those symptoms and isn't too unlikely itself.

Combinations of symptoms

Clearly some symptoms are not actually related to each other given what we are trying to diagnose, e.g. runny nose and swelling of the feet and toes.

Combinations of symptoms

Clearly some symptoms are not actually related to each other given what we are trying to diagnose, e.g. runny nose and swelling of the feet and toes.

It makes our problem **tractable** [read: a lot easier] if we assume that unrelated symptoms are independent given a diagnosis!

Combinations of symptoms

Clearly some symptoms are not actually related to each other given what we are trying to diagnose, e.g. runny nose and swelling of the feet and toes.

It makes our problem **tractable** [read: a lot easier] if we assume that unrelated symptoms are independent given a diagnosis!

Let's assume that **all** the symptoms are independent of one another given the diagnosis.

Combinations of symptoms

Clearly some symptoms are not actually related to each other given what we are trying to diagnose, e.g. runny nose and swelling of the feet and toes.

It makes our problem **tractable** [read: a lot easier] if we assume that unrelated symptoms are independent given a diagnosis!

Let's assume that **all** the symptoms are independent of one another given the diagnosis.

That's pretty naive, but now we don't need domain knowledge about which symptoms commonly occur together.

Combinations of symptoms continued

Claim

Conditional independence of all the symptoms allows us to say

$$\Pr(s_1 \cap s_2 \cap \ldots \cap s_n | \text{diagnosis}) = \prod_{i=1}^{n} \Pr(s_i | \text{diagnosis}).$$

Combinations of symptoms continued

Claim

Conditional independence of all the symptoms allows us to say

$$\Pr(s_1 \cap s_2 \cap \ldots \cap s_n | \text{diagnosis}) = \prod_{i=1}^{n} \Pr(s_i | \text{diagnosis}).$$

See the Naive Bayes notes for a full derivation using the product [chain] rule!

Table of Contents

Our first classifier

Apply Bayes Rule!

Our first classifier

Apply Bayes Rule!

$$\max_{\text{diag}} \Pr(\text{diag}|\text{symp}) = \max_{\text{diag}} \Pr(\text{symp}|\text{diag}) \cdot \Pr(\text{diag}).$$

Our first classifier

Apply Bayes Rule!

$$\max_{\text{diag}} \Pr(\text{diag}|\text{symp}) = \max_{\text{diag}} \Pr(\text{symp}|\text{diag}) \cdot \Pr(\text{diag}).$$

Apply conditional independence!

Our first classifier

Apply Bayes Rule!

$$\max_{\mathrm{diag}} \Pr(\mathrm{diag}|\mathrm{symp}) = \max_{\mathrm{diag}} \Pr(\mathrm{symp}|\mathrm{diag}) \cdot \Pr(\mathrm{diag}).$$

Apply conditional independence!

$$\max_{\mathrm{diag}} \Pr(\mathrm{diag}|\mathrm{symp}) = \max_{\mathrm{diag}} \left[\Pr(\mathrm{diag}) \cdot \prod_{s \in \mathrm{symp}} \Pr(s|\mathrm{diag}) \right].$$

Our first classifier

Apply Bayes Rule!

$$\max_{\text{diag}} \Pr(\text{diag} \mid \text{symp}) = \max_{\text{diag}} \Pr(\text{symp} \mid \text{diag}) \cdot \Pr(\text{diag}).$$

Apply conditional independence!

$$\max_{\text{diag}} \Pr(\text{diag} \mid \text{symp}) = \max_{\text{diag}} \left[\Pr(\text{diag}) \cdot \prod_{s \in \text{symp}} \Pr(s \mid \text{diag}) \right].$$

That's it!

Our second classifier

Applying the exact same steps

Our second classifier

Applying the exact same steps

$$\max_{\text{spam}} \Pr(\text{spam}|\text{email})$$

Our second classifier

Applying the exact same steps

$$\max_{\text{spam}} \Pr(\text{spam}|\text{email}) = \max_{\text{spam}} \Pr(\text{email}|\text{spam}) \cdot \Pr(\text{spam})$$

Our second classifier

Applying the exact same steps

$$\begin{aligned}\max_{\text{spam}} \Pr(\text{spam}|\text{email}) &= \max_{\text{spam}} \Pr(\text{email}|\text{spam}) \cdot \Pr(\text{spam}) \\ &= \max_{\text{spam}} \left[\Pr(\text{spam}) \cdot \prod_{\text{word}\in\text{email}} \Pr(\text{word}|\text{spam}) \right].\end{aligned}$$

Training a Naive Bayes Model

What we've derived so far:

$$\max_{\text{diag}} \Pr(\text{diag}|\text{symp}) = \max_{\text{diag}} \left[\Pr(\text{diag}) \cdot \prod_{s \in \text{symp}} \Pr(s|\text{diag}) \right].$$

Suppose we have a database of patients with symptoms, and a doctor has professionally diagnosed all of them.

Our model needs to learn two things in order to make predictions [by computing Pr(diagnosis|symptoms)]:

Training a Naive Bayes Model

What we've derived so far:

$$\max_{\text{diag}} \Pr(\text{diag}|\text{symp}) = \max_{\text{diag}} \left[\Pr(\text{diag}) \cdot \prod_{s \in \text{symp}} \Pr(s|\text{diag}) \right].$$

Suppose we have a database of patients with symptoms, and a doctor has professionally diagnosed all of them.

Our model needs to learn two things in order to make predictions [by computing Pr(diagnosis|symptoms)]:

- Pr(diagnosis) for each diagnosis.

Training a Naive Bayes Model

What we've derived so far:

$$\max_{\text{diag}} \Pr(\text{diag}|\text{symp}) = \max_{\text{diag}} \left[\Pr(\text{diag}) \cdot \prod_{s \in \text{symp}} \Pr(s|\text{diag}) \right].$$

Suppose we have a database of patients with symptoms, and a doctor has professionally diagnosed all of them.

Our model needs to learn two things in order to make predictions [by computing Pr(diagnosis|symptoms)]:

- Pr(diagnosis) for each diagnosis.
- Pr(symptom|diagnosis) for each symptom, for each diagnosis.

Training a Naive Bayes Model continued

Example

Consider the training set:

Name	Cough	Sore Throat	Fever	Diagnosis
A	Y	N	Y	Flu
B	N	Y	Y	Flu
C	Y	N	N	Cold
D	Y	Y	N	Cold
E	N	N	N	Healthy

$|\text{Cough} \cap \text{Flu}| =$
$|\text{Flu}| =$

$\Pr(\text{Cough}|\text{Flu}) =$

$\Pr(\text{Flu}) =$

Training a Naive Bayes Model continued

Example

Consider the training set:

Name	Cough	Sore Throat	Fever	Diagnosis
A	Y	N	Y	Flu
B	N	Y	Y	Flu
C	Y	N	N	Cold
D	Y	Y	N	Cold
E	N	N	N	Healthy

$|\text{Cough} \cap \text{Flu}| = 1$
$|\text{Flu}| =$

$\Pr(\text{Cough}|\text{Flu}) =$

$\Pr(\text{Flu}) =$

Training a Naive Bayes Model continued

Example

Consider the training set:

Name	Cough	Sore Throat	Fever	Diagnosis
A	Y	N	Y	Flu
B	N	Y	Y	Flu
C	Y	N	N	Cold
D	Y	Y	N	Cold
E	N	N	N	Healthy

$|\text{Cough} \cap \text{Flu}| = 1$
$|\text{Flu}| = 2$

$\Pr(\text{Cough}|\text{Flu}) =$

$\Pr(\text{Flu}) =$

Training a Naive Bayes Model continued

Example

Consider the training set:

Name	Cough	Sore Throat	Fever	Diagnosis
A	Y	N	Y	Flu
B	N	Y	Y	Flu
C	Y	N	N	Cold
D	Y	Y	N	Cold
E	N	N	N	Healthy

$|\text{Cough} \cap \text{Flu}| = 1$
$|\text{Flu}| = 2$

$$\Pr(\text{Cough}|\text{Flu}) = \frac{|\text{Cough} \cap \text{Flu}|}{|\text{Flu}|} = \frac{1}{2}$$

$\Pr(\text{Flu}) =$

Training a Naive Bayes Model continued

Example

Consider the training set:

Name	Cough	Sore Throat	Fever	Diagnosis
A	Y	N	Y	Flu
B	N	Y	Y	Flu
C	Y	N	N	Cold
D	Y	Y	N	Cold
E	N	N	N	Healthy

$|\text{Cough} \cap \text{Flu}| = 1$
$|\text{Flu}| = 2$
$\Pr(\text{Cough}|\text{Flu}) = \dfrac{|\text{Cough} \cap \text{Flu}|}{|\text{Flu}|} = \dfrac{1}{2}$
$\Pr(\text{Flu}) = \dfrac{|\text{Flu}|}{|\text{Patients}|} = \dfrac{2}{5}$

Training a Naive Bayes Model continued

Example

Name	Cough	Sore Throat	Fever	Diagnosis
A	Y	N	Y	Flu
B	N	Y	Y	Flu
C	Y	N	N	Cold
D	Y	Y	N	Cold
E	N	N	N	Healthy
F	N	Y	N	???

What is the probability new patient F is healthy?

$\Pr(\text{Cough}|\text{Healthy}) =$ $\Pr(\text{Sore Throat}|\text{Healthy}) =$

$\Pr(\text{Fever}|\text{Healthy}) =$ $\Pr(\text{Healthy}) =$

Training a Naive Bayes Model continued

Example

Name	Cough	Sore Throat	Fever	Diagnosis
A	Y	N	Y	Flu
B	N	Y	Y	Flu
C	Y	N	N	Cold
D	Y	Y	N	Cold
E	N	N	N	Healthy
F	N	Y	N	???

What is the probability new patient F is healthy?

$\Pr(\text{Cough}|\text{Healthy}) = 0$ $\Pr(\text{Sore Throat}|\text{Healthy}) =$

$\Pr(\text{Fever}|\text{Healthy}) =$ $\Pr(\text{Healthy}) =$

Problem: This model thinks that it's impossible for a healthy person to have a sore throat.

Training a Naive Bayes Model continued

Example

Name	Cough	Sore Throat	Fever	Diagnosis
A	Y	N	Y	Flu
B	N	Y	Y	Flu
C	Y	N	N	Cold
D	Y	Y	N	Cold
E	N	N	N	Healthy
F	N	Y	N	???

What is the probability new patient F is healthy?

$\Pr(\text{Cough}|\text{Healthy}) = 0 \quad \Pr(\text{Sore Throat}|\text{Healthy}) = 0$

$\Pr(\text{Fever}|\text{Healthy}) = \qquad \Pr(\text{Healthy}) =$

Problem: This model thinks that it's impossible for a healthy person to have a sore throat.

Training a Naive Bayes Model continued

Example

Name	Cough	Sore Throat	Fever	Diagnosis
A	Y	N	Y	Flu
B	N	Y	Y	Flu
C	Y	N	N	Cold
D	Y	Y	N	Cold
E	N	N	N	Healthy
F	N	Y	N	???

What is the probability new patient F is healthy?

$\Pr(\text{Cough}|\text{Healthy}) = 0 \quad \Pr(\text{Sore Throat}|\text{Healthy}) = 0$

$\Pr(\text{Fever}|\text{Healthy}) = 0 \quad \Pr(\text{Healthy}) =$

Problem: This model thinks that it's impossible for a healthy person to have a sore throat.

Training a Naive Bayes Model continued

Example

Name	Cough	Sore Throat	Fever	Diagnosis
A	Y	N	Y	Flu
B	N	Y	Y	Flu
C	Y	N	N	Cold
D	Y	Y	N	Cold
E	N	N	N	Healthy
F	N	Y	N	???

What is the probability new patient F is healthy?

$\Pr(\text{Cough}|\text{Healthy}) = 0$ $\Pr(\text{Sore Throat}|\text{Healthy}) = 0$

$\Pr(\text{Fever}|\text{Healthy}) = 0$ $\Pr(\text{Healthy}) = \frac{1}{5}$

Problem: This model thinks that it's impossible for a healthy person to have a sore throat.

Smooth

Solution: Let's add an extra occurrence of everything when computing conditional probabilities! That way we won't have anything with 0 probability.

Example

$|\text{Sore Throat} \cap \text{Healthy}| = 0$

$|\text{Healthy}| = 1$

$$\Pr(\text{Sore Throat}|\text{Healthy}) = \frac{|\text{Sore Throat} \cap \text{Healthy}|}{|\text{Healthy}|} = \frac{0}{1}$$

Smooth

Solution: Let's add an extra occurrence of everything when computing conditional probabilities! That way we won't have anything with 0 probability.

Example

$|\text{Sore Throat} \cap \text{Healthy}| = 0$
$|\text{Healthy}| = 1$
$\Pr(\text{Sore Throat}|\text{Healthy}) = \dfrac{|\text{Sore Throat} \cap \text{Healthy}| + 1}{|\text{Healthy}| + 2} = \dfrac{0+1}{1+2} = \dfrac{1}{3}$

Smooth

Solution: Let's add an extra occurrence of everything when computing conditional probabilities! That way we won't have anything with 0 probability.

Example

$|\text{Sore Throat} \cap \text{Healthy}| = 0$

$|\text{Healthy}| = 1$

$$\Pr(\text{Sore Throat}|\text{Healthy}) = \frac{|\text{Sore Throat} \cap \text{Healthy}| + 1}{|\text{Healthy}| + 2} = \frac{0+1}{1+2} = \frac{1}{3}$$

We add 2 to the denominator [not 1!] because we added one extra patient who was healthy with a sore throat and another extra patient who was healthy and didn't have a sore throat. That's 2 total extra healthy people we've seen!

Go forth!

Time to try it yourself on the next homework! Make sure to read the Naive Bayes notes for some other computation tricks we have to worry about during implementation!

Have fun!