Bayesian Classification

CSE634: Data Mining Concepts and Techniques
Professor Anita Wasilewska

References

Links

http://www.simafore.com/blog/3-challenges-with-naive-bayes-classifiers-and-how-to-overcome

http://ai.berkeley.edu/slides/Lecture%2017%20--%20Bayes%20Nets%20II%20Independence/

http://www3.cs.stonybrook.edu/~cse634/ch6book.pdf

https://web.stanford.edu/class/cs124/lec/naivebayes.pdf

https://classes.soe.ucsc.edu/cmps140/Winter17/slides/3.pdf

http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

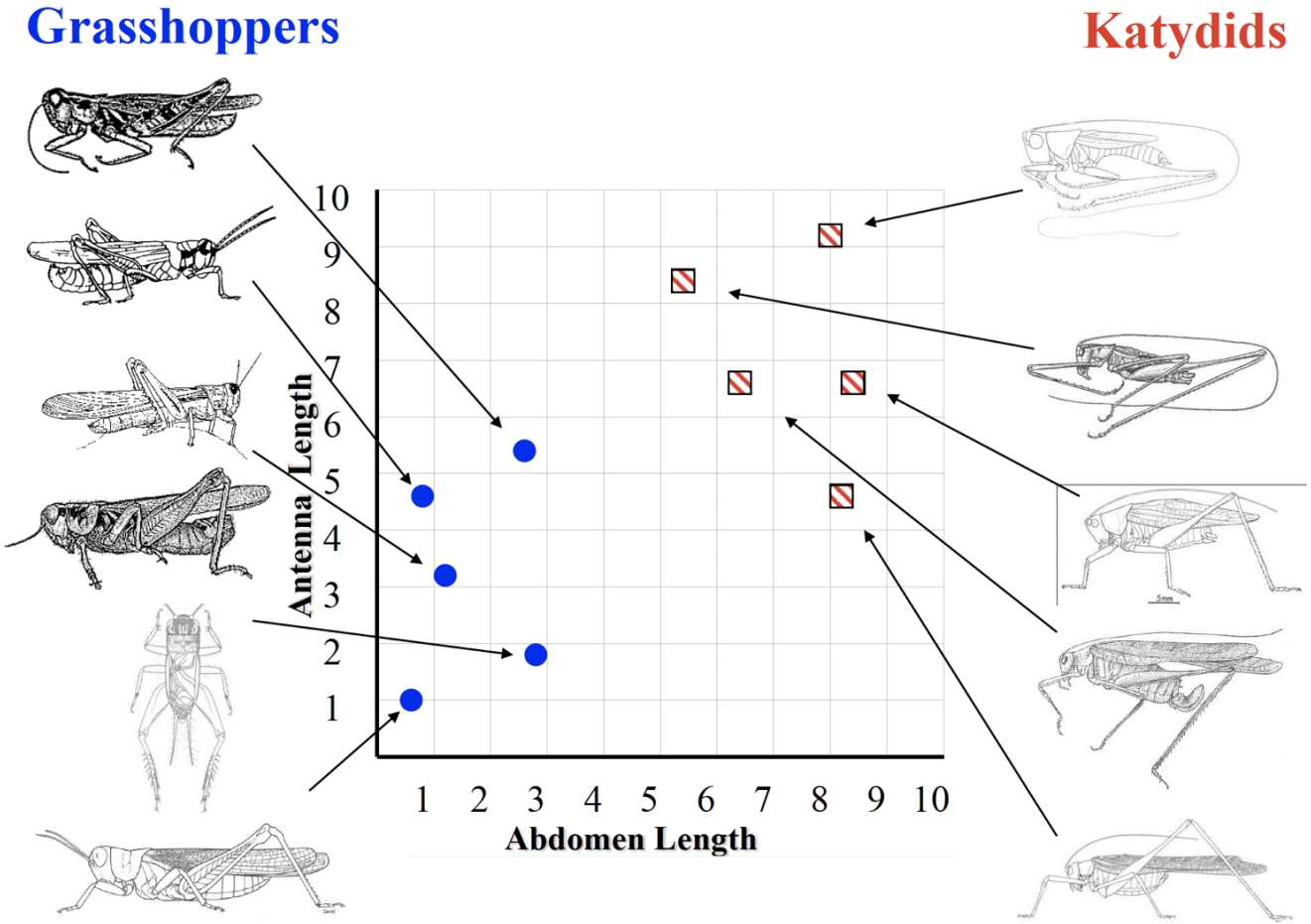
Papers

https://dl-acm-org.proxy.library.stonybrook.edu/citation.cfm?id=3025454

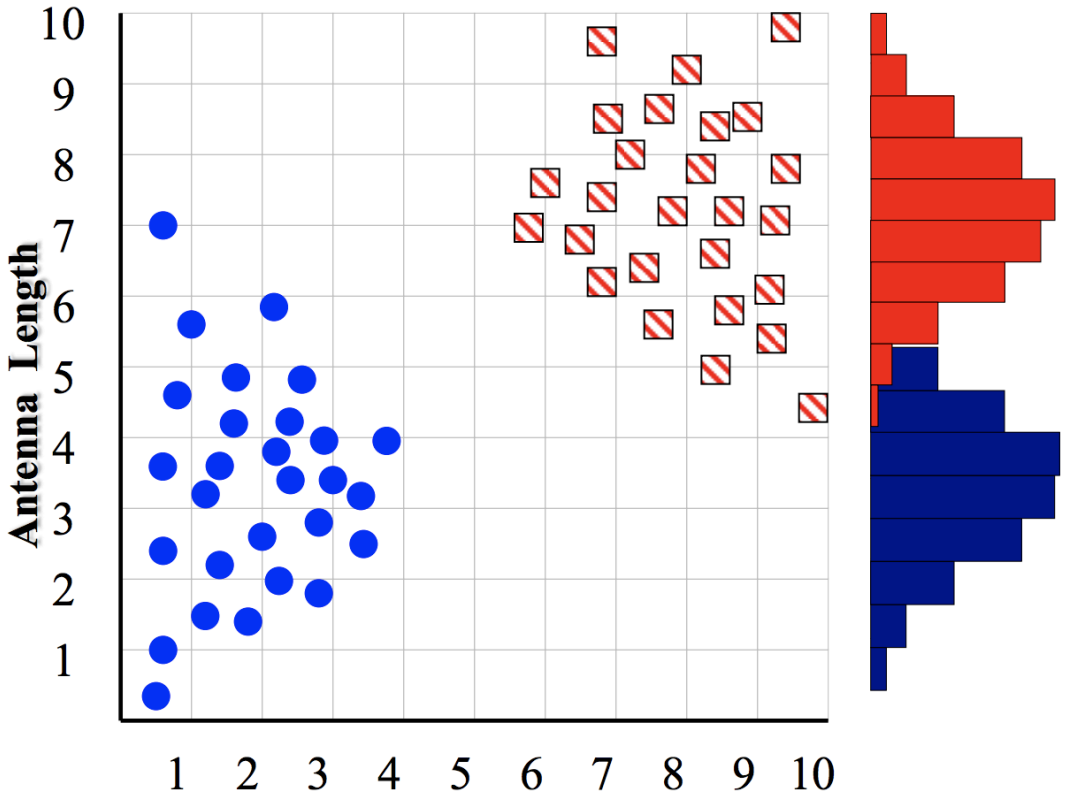
Outline

- Bayes Rule and Naive Bayes
- Text Classification using Naive Bayes
- Advantages/Disadvantages and Issues with Naive Bayes
- Bayesian Belief Networks
- COMPASS: Rotational Keyboard on Non-Touch Smartwatches

The Intuition

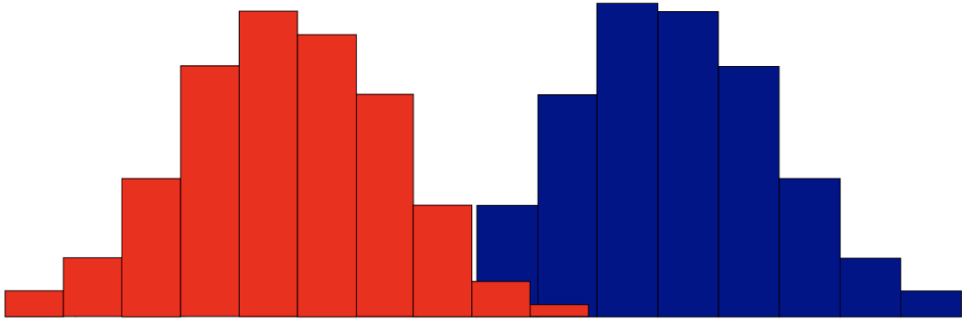


Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

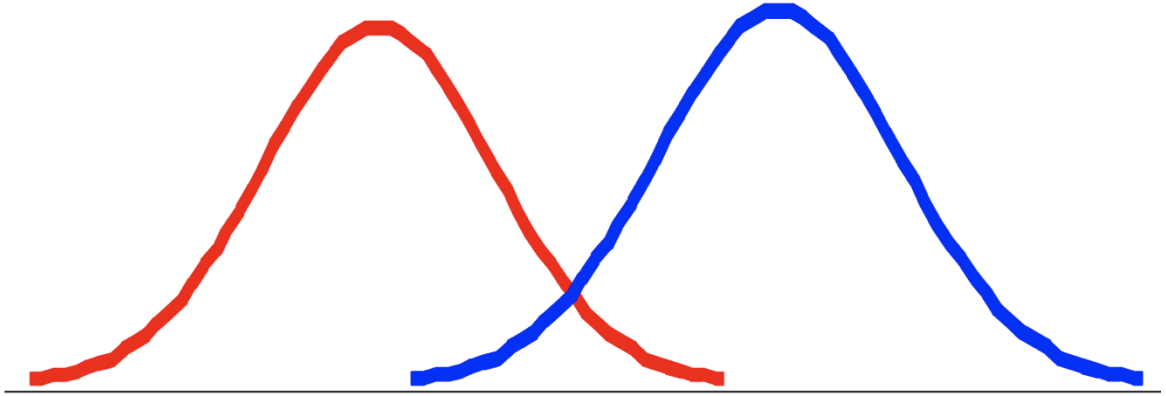


Histograms for the Antenna Lengths

Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf



Summarizing the Histograms as two Normal Distributions for the ease of visualization

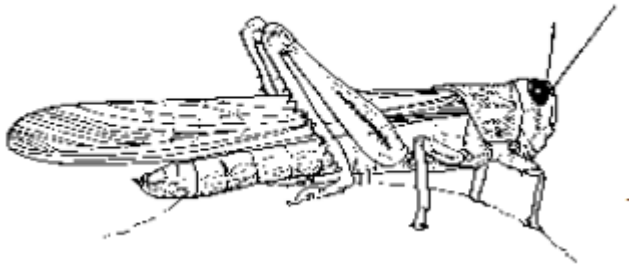


Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

Which Insect?

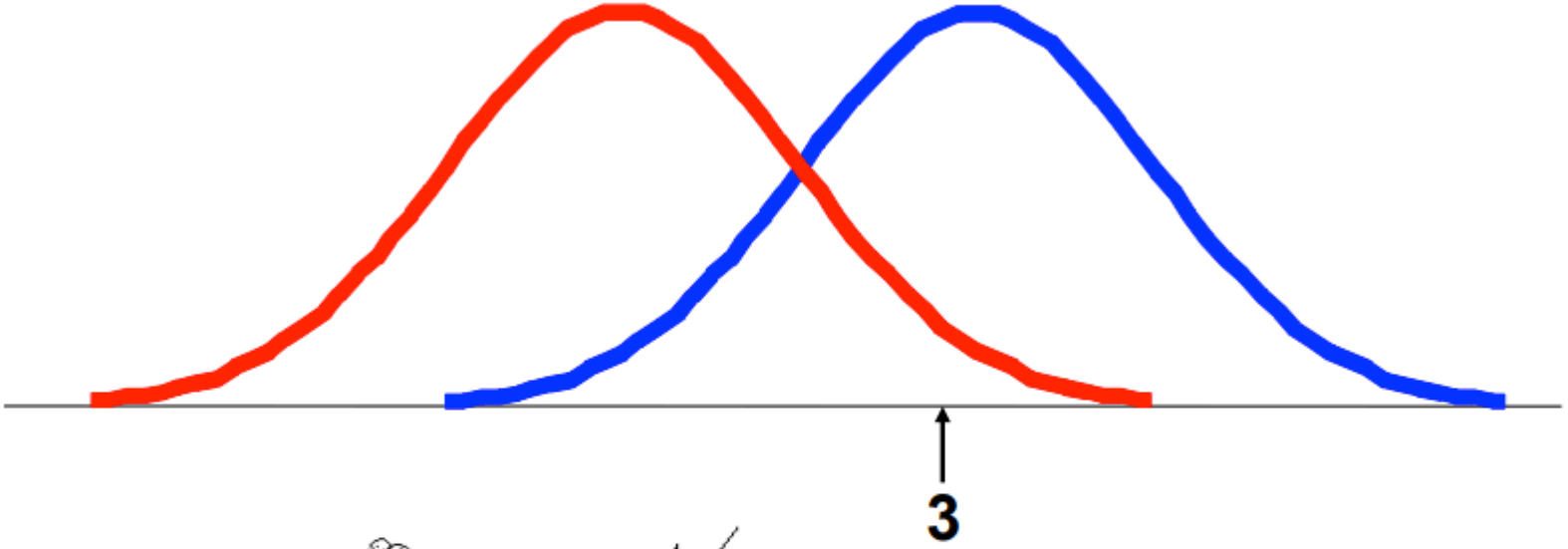
We want to classify an insect we have found. Its antennae are 3 units long. How can we classify it?

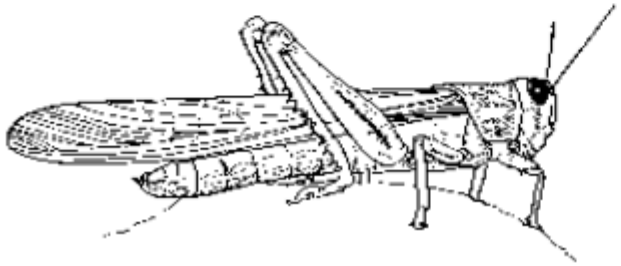
Given the distributions of antennae lengths we have seen, is it more probable that our insect is a Grasshopper or a Katydid



Formal way to discuss the most probable classification:

P(Cj | d) = Probability of class Cj , given observed/evidence d



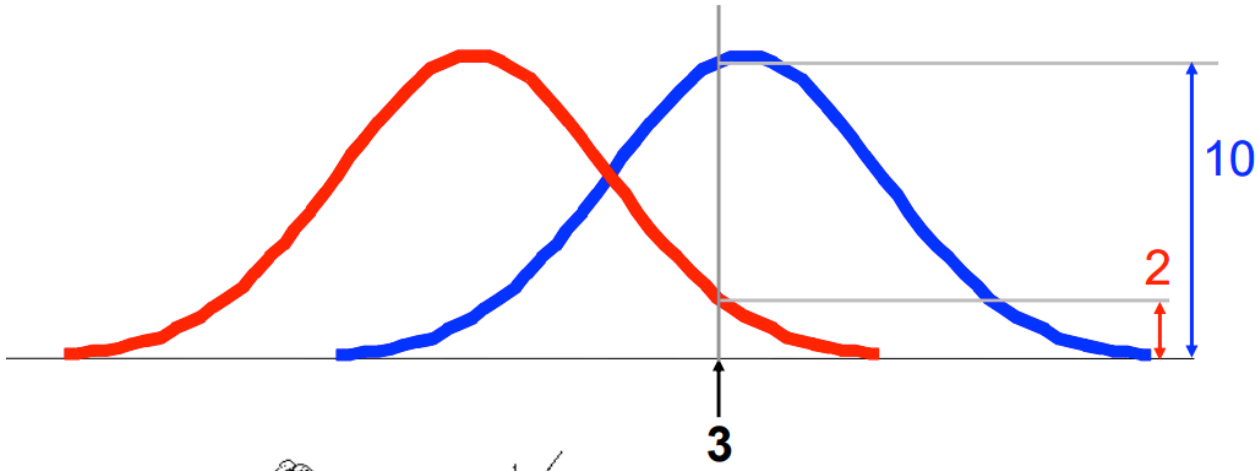


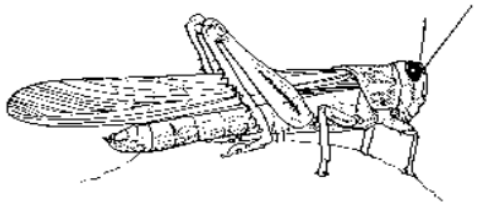
Antennae length is **3**

Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

P(Cj | d) = Probability of class Cj , given observed/evidence d

P(Grasshopper| 3) = 10 / (10 + 2) = 0.833
P(Katydid| 3) = 2 / (10 + 2) = 0.166



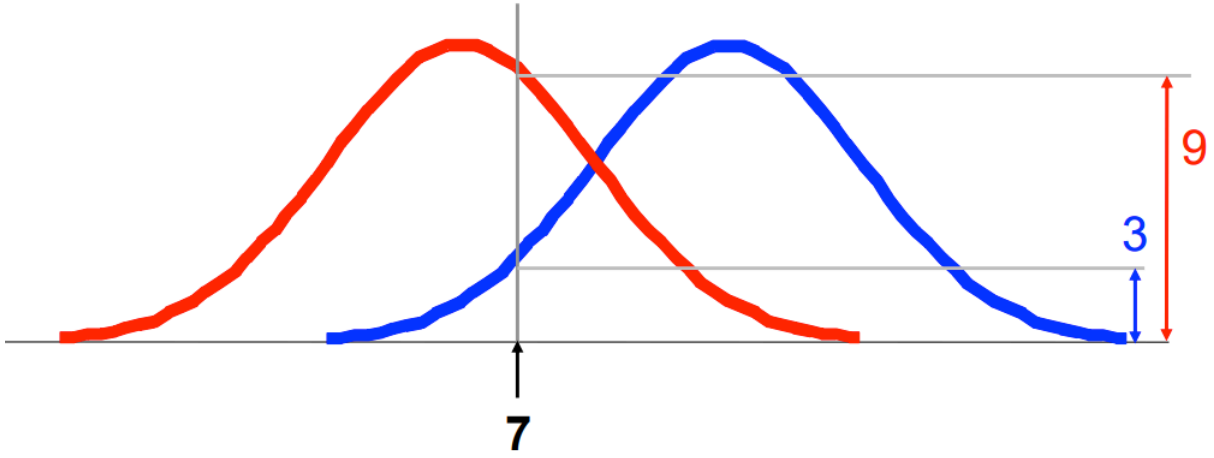


Antennae length is **3**

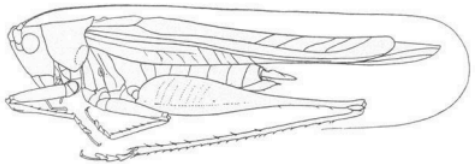
Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

P(Cj | d) = Probability of class Cj , given observed/evidence d

P(Grasshopper| 7) = 3 / (3 + 9) = 0.25
P(Katydid| 7) = 9 / (3 + 9) = 0.166



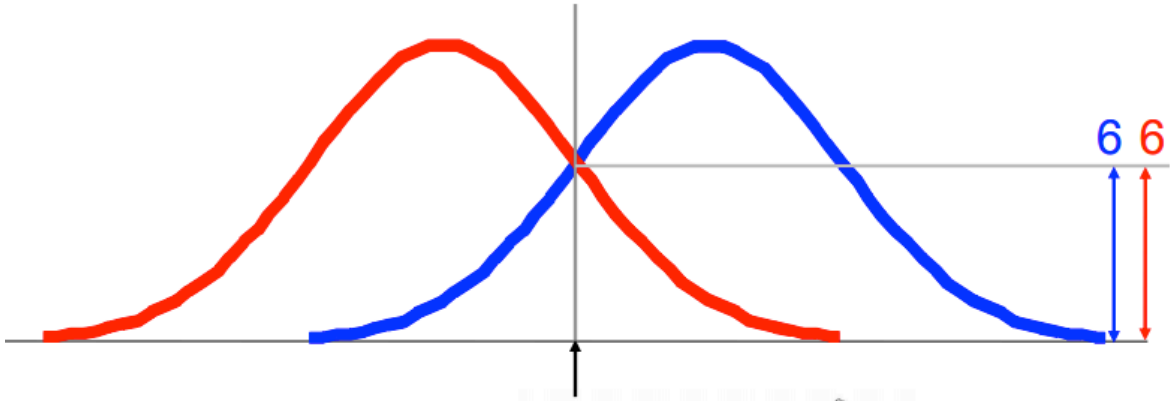
Antennae length is 7



Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

P(Cj | d) = Probability of class Cj , given observed/evidence d

P(Grasshopper| 5) = 6/ (6 + 6) = 0.5
P(Katydid| 5) = 6/ (6 + 6) = 0.5



Antennae length is **5**

Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

Bayesian Classification

That was a visual intuition for a simple case of the Bayes classifier, also called Naïve Bayes or Simple Bayes

Before we look into the mathematical representations, keep in mind the basic idea:

Find out the probability of the **previously unseen instance** belonging to each class, then simply pick the most probable class.

Bayes Theorem

Bayesian classifiers use **Bayes theorem**, which says

$$p(c_j \mid d) = \frac{p(d \mid c_j)\, p(c_j)}{p(d)}$$

$p(c_j \mid d)$ = probability of instance d being in class c_j,

This is what we are trying to compute

$p(d \mid c_j)$ = probability of generating instance d given class c_j,

We can imagine that being in class c_j, causes you to have feature d with some probability

$p(c_j)$ = probability of occurrence of class c_j,

This is just how frequent the class c_j, is in our database

$p(d)$ = probability of instance d occurring

This can actually be ignored, since it is the same for all classes

Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

Guess the gender

Assume that we have two classes

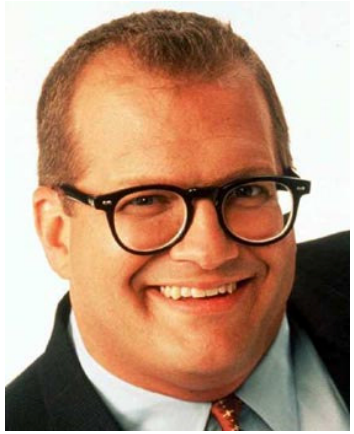
C_1= male, C_2= female

We have a person whose sex we do not know, say “drew” or d.

Drew can be a male or a female name

Classifying drew as male or female is equivalent to asking is it more probable that drew is male or female.

I.e which is greater P(male | drew) or P(female | drew)



Drew Carey

Drew Barrymore

What is the probability of being called "*drew*" given that you are a male?

What is the probability of being a male?

$$p(\text{male} \mid drew) = \frac{p(drew \mid \text{male})\, p(\text{male})}{p(drew)}$$

constant(independent of class) → $p(drew)$ ← What is the probability of being named "*drew*"?

Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

Example

Is this officer a Male or a Female?

Officer Drew

Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

The Dataset

Name	Sex
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

Officer Drew

$$p(c_j \mid d) = \frac{p(d \mid c_j)\, p(c_j)}{p(d)}$$

Name	Sex
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

P(male | drew) = ((⅓) x (⅜) / (⅜)) = 0.125/(⅜)

P(female | drew) = ((⅓) x (⅜) / (⅜)) = **0.25**/(⅜)

Hence Officer Drew is more likely to be a female

Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

Correct Classification!

Officer Drew is female

P(male | drew) = ((⅓) x (⅜) / (⅜)) = 0.125/(⅜)

P(female | drew) = ((⅓) x (⅜) / (⅜)) = **0.25**/(⅜)

Officer Drew

Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

How to deal with multiple attributes?

So far we have only considered Bayes Classification when we have one attribute (the “antennae length”, or the “name”). But we may have many features.

Ex: Height, Eye Color, Hair Length, and so on.

How do we use all the features?

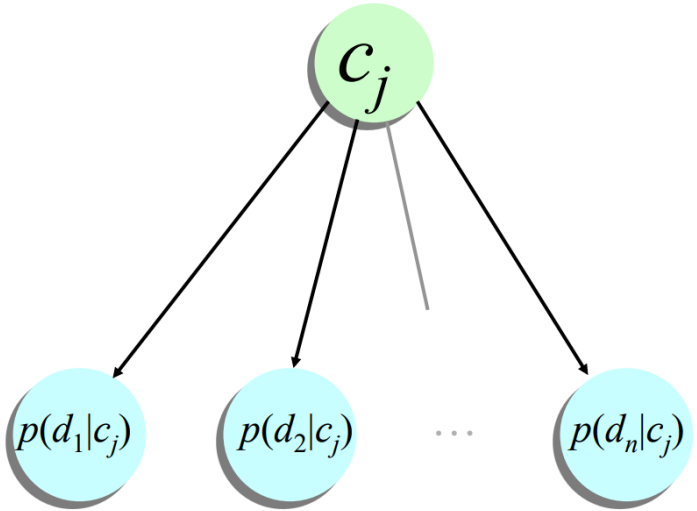
P(male| Height, Eye, Hair Length) $\propto$ P(Height, Eye, Hair Length | male) P(male)

Computing Probability P(Height, Eye, Hair Length | male) is **infeasible**!

Naïve Bayes Classification

Assume all input features are class conditionally independent!

P(male| Height, Eye, Hair Length) $\propto$ P(Height| male) P(Eye|male) P(Hair Length | male) P(male)



Note: Direction of arrow from class to feature

Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

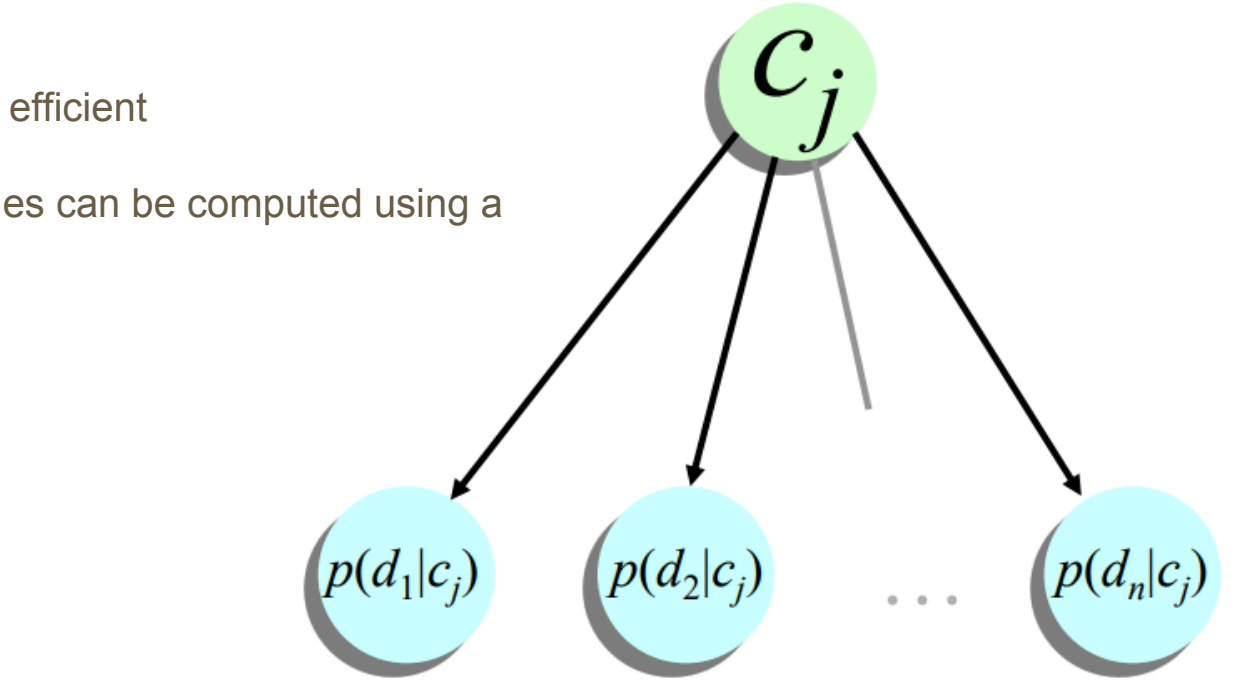
The Dataset

Name	Over 170cm	Eye	Hair length	Sex
Drew	No	Blue	Short	Male
Claudia	Yes	Brown	Long	Female
Drew	No	Blue	Long	Female
Drew	No	Blue	Long	Female
Alberto	Yes	Brown	Short	Male
Karin	No	Blue	Long	Female
Nina	Yes	Brown	Short	Female
Sergio	Yes	Blue	Long	Male

Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

Naive Bayes is fast and space efficient

The conditional probability tables can be computed using a single pass over the data



Sex	Over190$_{cm}$	
Male	Yes	0.15
	No	0.85
Female	Yes	0.01
	No	0.99

Sex	Long Hair	
Male	Yes	0.05
	No	0.95
Female	Yes	0.70
	No	0.30

Sex	
Male	
Female	

Ref: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

Quick Overview

- Bayesian Classification uses a probabilistic approach to classification
- Find out the probability of the **previously unseen instance** belonging to each class, then simply pick the most probable class.
- For instances with multiple features, assume all input features are class conditionally independent

Naive Bayes Classifier

Now the "naive" conditional independence assumptions come into play:

Assume that each feature is conditionally independent of every other feature,then

P(Xi|Xi+1,....Xn,Ck) = Product of (P(xi |Ck)) for i = 1 to n

Naive Bayes based Text Classification

- Relies on very simple representation of document
- **Bag of words**

Some Examples

- Classify email as spam/not spam
- Classify movie as favorable/unfavorable/neutral
- Classify journal as Electrical, SciTech,Travel,Meditation
- Learn to classify web pages by topic

Learning to classify text:Why Naive Bayes

East to compute and simple

Naive Bayes perform well with small amounts of training data.

Naive Bayes is among the most effective algorithms

What attributes shall we use to represent the text documents ?

$\gamma($

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.

$)=C$

$\gamma($

great	2
love	2
recommend	1
laugh	1
happy	1
...	...

$) = c$

Learning to classify Text: Movie Reviews

- Document -> (+,-)
- Represent each document by vector of words : one attribute per word position in document
- Learning :Use training example to estimate
- P(+)
- P(-)
- P(doc | +)
- P(doc | -)

$$P(doc|v_j) = \prod_{i=1}^{length(doc)} P(a_i = w_k|v_j)$$

Where $P(a_i = w_k|v_j)$ is the probability that word in position i is w_k given v_j

An example : Movie Reviews

You have set of review(documents) and classification

Doc	Text	Class
1	I loved the movie	+
2	I hated the movie	-
3	A great movie,good movie	+
4	Poor acting	_
5	Great acting good movie	+

10 Unique words<I,loved,the,movie,hated,a,great,poor,acting,good>

Document with positive outcomes

Convert the document into feature sets , where the attributes are possible words and value are the number of times word occur in the given document.

Doc	I	loved	the	movie	hated	a	great	poor	acting	good	Class
1	1	1	1	1							+
2	1		1	1	1						-
3				2		1	1			1	+
4								1	1		-
5				1		1	1		1	1	+

Document with positive outcomes

Doc	I	loved	the	movie	hated	a	great	poor	acting	good	Class
1	1	1	1	1							+
3				2		1	1			1	+
5				1		1	1		1	1	+

p(+) = ⅗ = 0.6
Compute :p(I | +); p(love | +) ; p(the | +) ; p(movie | +) ; p(a | +) ; p(great | +); p(acting | +) ;p(good | +)
Let n be the number of words in the + case
N_k number of times word k occurs in these + cases.

So $p(w_k \mid +) = (n_k + 1) / (n + |\text{Vocabulary}|)$

Document with positive outcomes

Doc	I	loved	the	movie	hated	a	great	poor	acting	good	Class
1	1	1	1	1							+
3				2		1	1			1	+
5				1		1	1		1	1	+

p(+) = ⅗ = 0.6
p(I | +) = 1+ 1/14+10 = 0.0833 ; p(loved | +) = 1 + 1/14 + 10 = 0.0833 p(the | +) = 0.0833
p(movie | +) = 4+1/14+10 = 0.208 p(a | +) = 2+1/14+10 = 0.125 p(great | +) = 0.125
Let n be the number of words in the + case
Nk number of times word k occurs in these + cases.

So $p(w_k \mid +) = (n_k + 1)/ (n + |\text{Vocabulary}|)$

Classify new sentence

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

If C_J is + ; p(+) p(I | +) p(hated | +) p(the | +) p(poor | +) p(acting | +) = 6.03*10^-7

If C_J is - ; p(-) p(I | -) p(hated | -) p(the | -) p(poor | -) p(acting | -) = 1.22*10^-5)

Multinomial Naive Bayes Independence Assumption

- **Bag of Words assumption**: Assume position doesn't matter
- **Conditional Independence**: Assume the feature probabilities $P(x_i|c_j)$ are independent given the class c.

$$P(x_1,\ldots,x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

Multinomial Naive Bayes Classifier

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

positions ← all word positions in test document

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

Naive Text Classifier : Numeric Underflow Problem

- When p(x|c) is often a very small number: the probability of observing any particular high-dimensional vector is small.
- This will lead to numerical underflow.

$$P(x_1,\ldots,x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

Naive Text Classifier : Zero Problem

- If there is a class,C_i, and X has an attribute value ,x_k,such that none of the samples in C_i has that attribute value ?
- In that case **$P(x_k|Ci) = 0$,** which results in $P(x_k|C_i) = 0$ even though **$P(x_k|Ci)$ for all the other attributes in X may be large.**

Evaluating Naive Bayes Classifier

- **Advantages**

  Fast to train (just one scan of database) and classify

  Not sensitive to irrelevant features

  Handles discrete data well

  Handles streaming data well

- **Disadvantage**

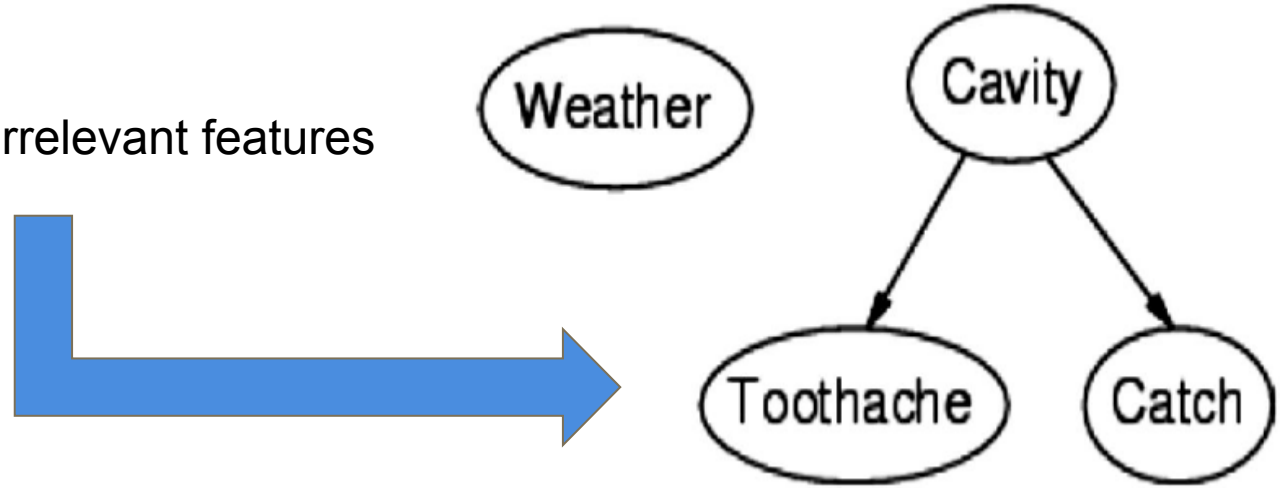
  Assumes independence of features - Losing the accuracy.

  Not really good for continuous attribute values.

Evaluating Naive Bayes Classifier

- **Advantages**

  Not sensitive to irrelevant features



http://www.aistudy.com/ai/probabilistic_aima.htm

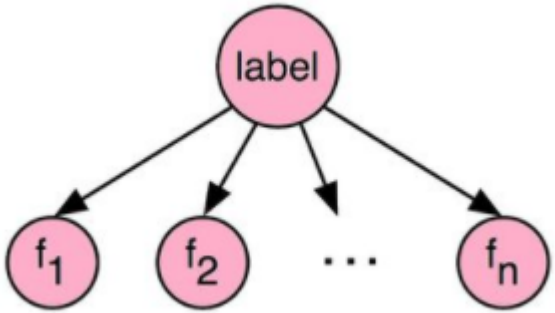
Issues with Naive Bayes Classifier - 1

ZERO CONDITIONAL PROBABILITY PROBLEM

- Incomplete training data
- A given class and feature value never occur together in the training set.
- This is problematic since it will wipe out all information in the other probabilities when they are multiplied.
- Conditional probability will be zero and the whole construction collapses!

Issues with Naive Bayes Classifier - 1

ZERO CONDITIONAL PROBABILITY PROBLEM



Example: A feature **F2** and the class **label** do not occur in training set.

Since, P(F2 | label) = 0

P(label | F1, F2, F3) = P(label) P(F1|label) P(F2|label) P(F3|label)

P(label | F1, F2, F3) = 0

https://sites.google.com/site/yorkyuhuang/home/research/machine-learning-information-retrieval/scene-classification

Correction to Zero Probability Problem

Laplace Smoothing

- To eliminate zeros joint probability, we use **add-one or Laplace smoothing**
- Adds arbitrary low probabilities in such cases so that the probability computation does not become zero.
- **Basic Idea: Pretend that you saw every feature-class outcome pair k extra times.**

Laplace Smoothing

X_i = The i-th attribute in dataset D.

x_i = A particular value of the X_i attribute in dataset D.

N = Total number of tuples in dataset D.

k = Laplace Smoothing Factor.

Count ($X_i = x_i$) = Number of tuples where the attribute X_i takes the value x_i

$| X_i |$ = Number of different values attribute X_i can take.

$$P_{Lap,k}(X_i = x_i) = \frac{count(X_i = x_i) + k}{N + k|X_i|}$$

Laplace Smoothing

Count ($X_i = x_i$, Y =y) = Joint probability of $X_i = x_i$ and Y =y appearing together in the dataset.

$|X_i|$ = Number of different values attribute X_i can take.

$$P_{Lap,k}(X_i = x_i|Y = y) = \frac{count(X_i = x_i, Y = y) + k}{count(Y = y) + k|X_i|}$$

Correction to Zero Probability Problem

TextBook Example 6.5

- Class buys_computer = yes and an attribute income = {low, medium, high} in some training database, D, containing 1000 tuples such that

  0 tuples with income = low

  990 tuples with income = medium

  10 tuples with income = high

- **The probabilities of these events, without the Laplacian correction, are**

  P(income=low | buys_computer = yes) = 0

  P(income=medium | buys_computer = yes) = 0.990 (i.e. 990/1000)

  P(income=high | buys_computer = yes) = 0.010 (i.e. 10/1000)

- Lets use Laplacian correction, using k = 1 for each of the three attribute values.

Correction to Zero Probability Problem

TextBook Example 6.5

- Class buys_computer = yes and an attribute income = {low, medium, high} in some training database, D, containing 1000 **+ 3 = 1003** tuples such that

~~**0** tuples with income = low~~ **1 tuples with income = low**

~~**990** tuples with income = medium~~ **991 tuples with income = medium**

~~**10** tuples with income = high~~ **11 tuples with income = high**

- Using Laplacian correction, using k = 1 for each of the three attribute values.
- The “corrected” probability estimates are close to their “uncorrected” counterparts.

Correction to Zero Probability Problem

TextBook Example 6.5 contd….

- **The new probabilities of these events, with the Laplacian correction, are**

  $P_{LAP,K=1}$ (income=low | buys_computer = yes) = 0.001 (i.e. 1/1003)

  $P_{LAP,K=1}$ (income=medium | buys_computer = yes) = 0.988 (i.e. 991/1003)

  $P_{LAP,K=1}$ (income=high | buys_computer = yes) = 0.0109 (i.e. 11/1003)

- **The "corrected" probability estimates are close to their "uncorrected" counterparts**

  The zero probability value is avoided!

- Note: N i.e. total number of tuples is increased to 1003 from 1000.

Issues with Naive Bayes Classifier - 2

CONTINUOUS VARIABLES

- When an attribute is continuous, computing the probabilities by the traditional method of frequency counts is not possible.

Solution ---- May lead to loss in classification accuracy

Discretization: Convert the attribute values to discrete values - Binning

Probability Density Functions: To compute probability densities instead of actual probabilities.

Concept of Probability Density Function (PDF)

Example

Even though a fast-food chain might advertise a hamburger as weighing a quarter-pound, you can well imagine that it is not exactly 0.25 pounds. One randomly selected hamburger might weigh 0.23 pounds while another might weigh 0.27 pounds. What is the probability that a randomly selected hamburger weighs between 0.20 and 0.30 pounds? That is, if we let X denote the weight of a randomly selected quarter-pound hamburger in pounds, what is $P(0.20 < X < 0.30)$?

https://onlinecourses.science.psu.edu/stat414/node/97

Concept of Probability Density Function (PDF)

- Finding P(X = x) for a continuous random variable X is not going to work.
- **Solution - Find the probability that X falls in some interval [a, b], i.e.**

  find P(a <= X <= b) ------> PDF comes to the rescue.

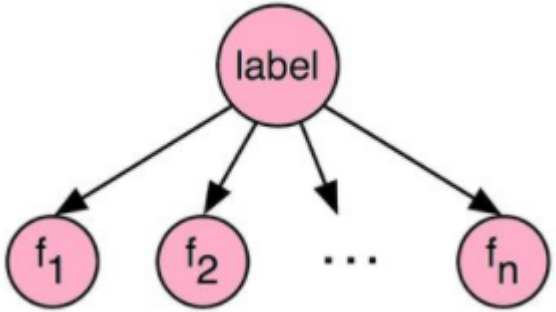
  $f_X(x)$ = Probability Density Function of attribute X.

$$\Pr[a \leq X \leq b] = \int_a^b f_X(x)\,dx.$$

Issues with Naive Bayes Classifier - 3

THE CONDITIONAL INDEPENDENCE ASSUMPTION

Naive Bayes Classifier works well here since attributes are independent of each other.



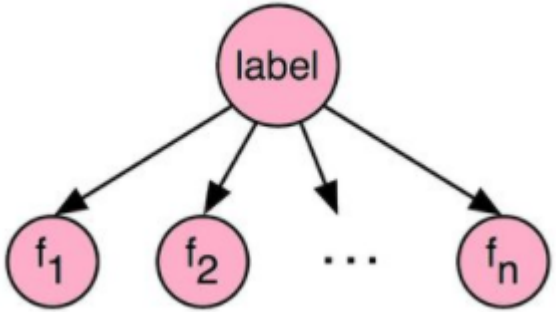
P (label | F1, F2, … , Fn) = P(label) P(F1|label) P(F2|label) …… P(Fn|label)

https://sites.google.com/site/yorkyuhuang/home/research/machine-learning-information-retrieval/scene-classification

Issues with Naive Bayes Classifier - 3

THE CONDITIONAL INDEPENDENCE ASSUMPTION

Naive Bayes Classifier works well here since attributes are independent of each other.



P (label | F1, F2, … , Fn) = P(label) P(F1|label) P(F2|label) …… P(Fn|label)

https://sites.google.com/site/yorkyuhuang/home/research/machine-learning-information-retrieval/scene-classification

Issues with Naive Bayes Classifier - 3

THE CONDITIONAL INDEPENDENCE ASSUMPTION

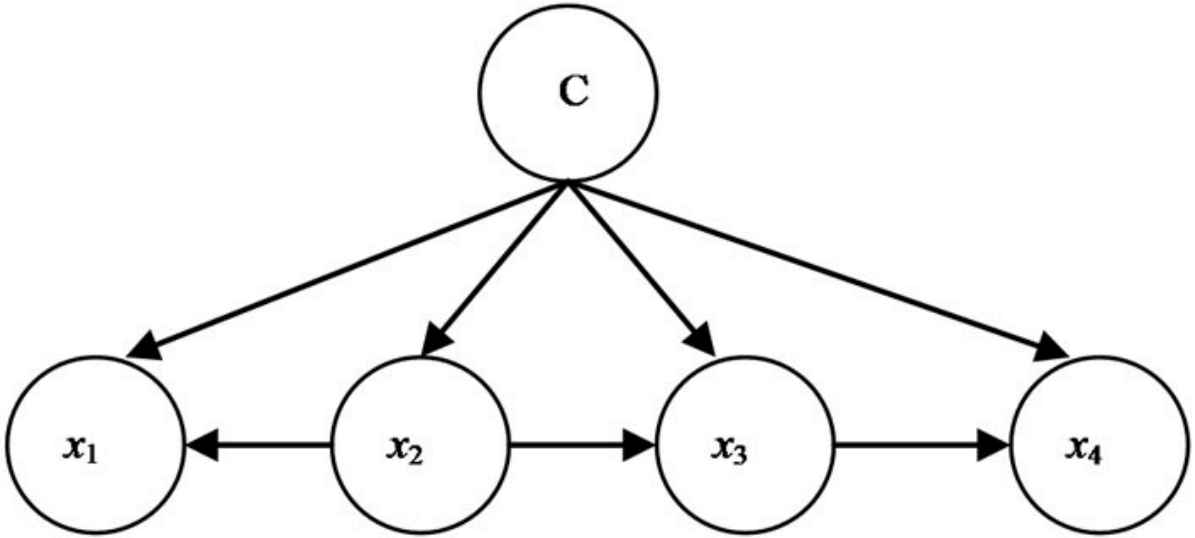
- The **NAIVETY** of Naive Bayes

- Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes.

- This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is considered “naive.”

Issues with Naive Bayes Classifier - 3

What if the attributes have some correlation among themselves?

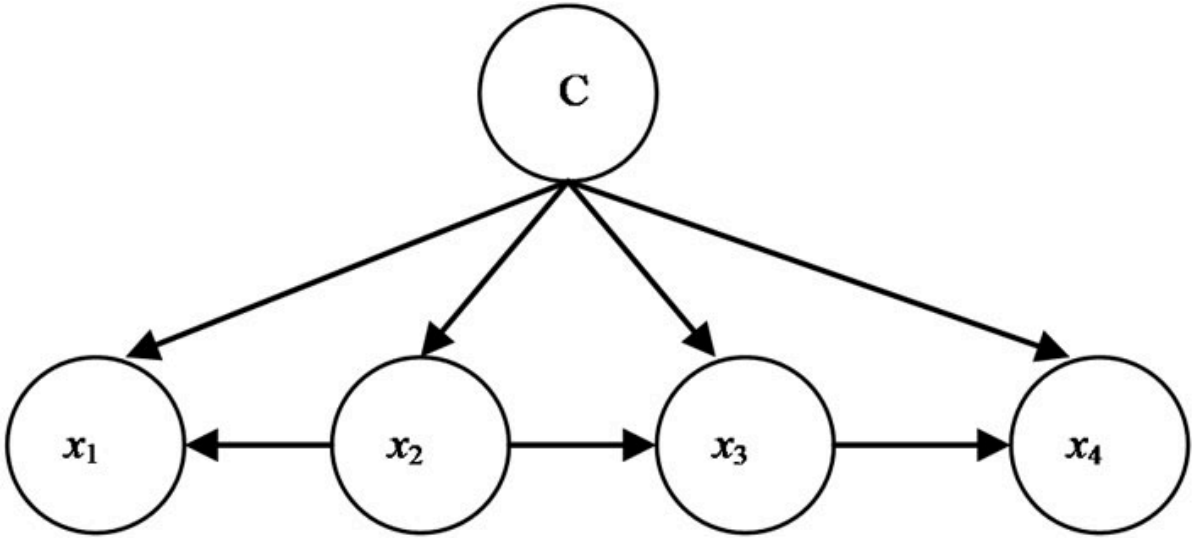


Find P (C | x1, x2, x3, x4) ???

http://file.scirp.org/Html/6-8701042/55b8702f-c8b1-46c4-9ac3-60af9a36b130.jpg

Issues with Naive Bayes Classifier - 3

What if the attributes have some correlation among themselves?



P (C | x1, x2, x3, x4) = P(C) P(x1|C) P(x2|C) P(x3|C) P(x4|C)

http://file.scirp.org/Html/6-8701042/55b8702f-c8b1-46c4-9ac3-60af9a36b130.jpg

Issues with Naive Bayes Classifier - 3

THE CONDITIONAL INDEPENDENCE ASSUMPTION - Fails to classify here

Example:

Text Classification: Consider a **bag of words**: w1, w2, w3, ... wn

Needed to be classified as either **Class-1: Spam or Class-2: Ham**

Consider a text that is classified as following:

P(spam| w1, w2, w3 ... wn) = 0.8 (80%)

P(ham | w1, w2, w3 ... wn) = 0.2 (20%)

Now we jumble the order of the words in the text (they may not make any sense now):

The probabilities of the jumbled sentence will remain the same!

P(spam| w2, w5, w3 ... wn) = 0.8 (80%)

P(ham | w2, w5, w3 ... wn) = 0.2 (20%)

Improvement - Fixing Conditional Independence Problem

1. When it is known beforehand that a few of the attributes are correlated.
   - **Ignore one of the correlated attributes if it's not giving any significant information. For Example: attributes age_group={child,youth,old_aged}, age∈ [10,60] in a dataset.**

2. When it is not known which attributes are dependent on the other.
   - **Find the correlation among attributes. For example, Pearson Correlation Test to know the correlation between two attributes.**

Pearson Correlation Test

- To investigate the relationship between two continuous variables/attributes X and Y in the dataset.
- X-bar = Mean of Attribute X, Y-bar = Mean of Attribute Y.
- **‘r’ measures the strength of the association.**
- **r ∈ [-1, 1]**

$$r = \frac{\sum(X - \overline{X})(Y - \overline{Y})}{\sqrt{\sum(X - \overline{X})^2}\sqrt{\sum(Y - \overline{Y})^2}}$$

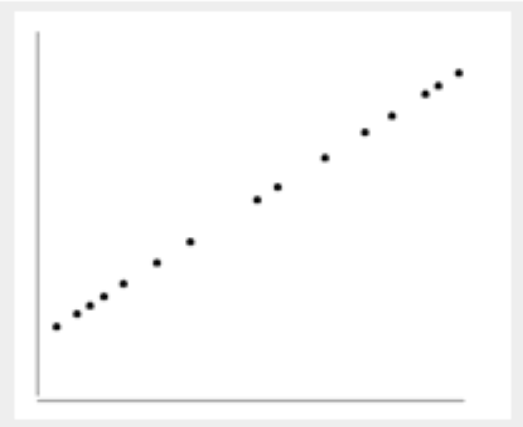
Pearson Correlation Test

Let one attribute is denoted by x-axis and second attribute is denoted by y-axis.

r = Pearson Correlation Coefficient (Range: [-1,1])

Negative Relationship (r = -1)

No Relationship (r = 0)



Positive Relationship (r = 1)

Improvement - BAYESIAN BELIEF NETWORKS

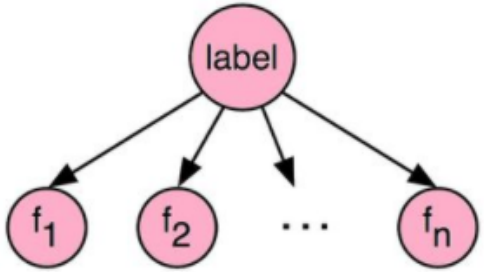
BAYESIAN BELIEF NETWORKS

- An improvement over Naive Bayes Classifiers
- Do not assume the Class Conditional Independence
- Specify joint conditional probability distribution
- Computationally intensive in comparison to Naive Bayes Classifiers

Bayesian Belief Network is a **directed acyclic graph** that specify dependencies between the **attributes (the nodes in the graph)** of the dataset. The topology of the graph exploits any conditional dependency between the various attributes.

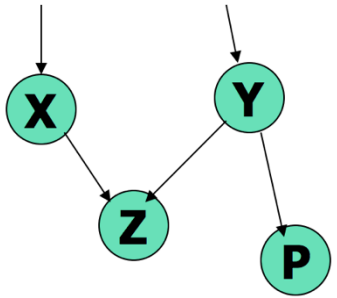
Bayesian Belief Networks

- Also known as **Belief Networks**, **Bayesian Networks**, and **Probabilistic Networks**
- Directed Acyclic Graphs
- They represent probabilistic dependencies between variables
- What we saw earlier (Naive Bayes) was a simple Bayesian Belief Network!



Bayesian Belief Networks

- **Nodes** represent random variables and their states
- **Arcs** represent the Bayesian probabilistic relationships between these variables
- They give a specification of **Joint Probability Distribution**



- ❑ Nodes: random variables
- ❑ Links: dependency
- ❑ X and Y are the parents of Z, and Y is the parent of P
- ❑ No dependency between Z and P
- ❑ Has no loops or cycles

Ref: http://www3.cs.stonybrook.edu/~cse634/ch6book.pdf

Bayesian Belief Networks

Size of Bayesian Networks

- How big is a joint distribution over N Boolean variables? **2^N**
- How big is an N-node BBN if nodes have up to k parents?**O(N * 2^(k+1))**
- Both give you the joint probability $P(X_1, X_2, \ldots X_n)$
- So, BNNs provide **huge space savings**!
- Easier to calculate local Conditional Probability Tables (CPTs)
- Also **faster to answer queries**!

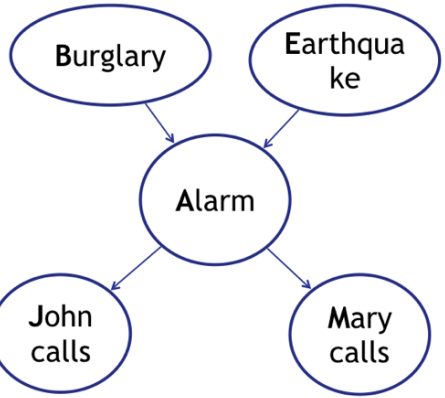
Bayesian Belief Networks

Assumptions

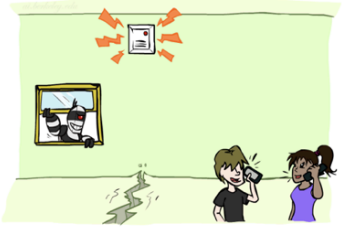
- **Chain rule** (valid for all distributions): $$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i | x_1 \ldots x_{i-1})$$
- **Main assumption** $$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$$
- Also infer conditional independence from graph
- Simplifies answering queries

Bayesian Belief Network Example

B	P(B)
+b	0.001
-b	0.999



E	P(E)
+e	0.002
-e	0.998



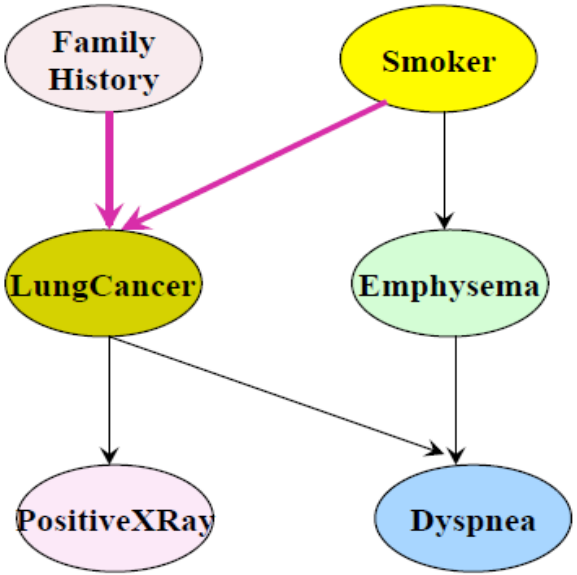
A	J	P(J|A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

A	M	P(M|A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

Ref: http://ai.berkeley.edu/slides/Lecture%2017%20--%20Bayes%20Nets%20II%20Independence/

Bayesian Belief Network Example



Example:

- Having **lung cancer** is influenced by a person's **family history** of lung cancer, as well as whether or not the person is a **smoker**
- Note that the variable **PositiveXRay** is **independent** of whether the patient has a family history of **lung cancer** or is a **smoker**, given that we know the patient has lung cancer

- Using the joint probability, **classification queries can be answered!**
- E.g., What is the probability that an individual will have LungCancer, given that they have both PositiveXRay and Dyspnea

Ref: http://www3.cs.stonybrook.edu/~cse634/ch6book.pdf

Bayesian Belief Network Example

Conditional Probability Table for the variable **Lung Cancer**

	(FH, S)	(FH, ~S)	(~FH, S)	(~FH, ~S)
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

For instance, from the upper leftmost and bottom rightmost entries, respectively, we see that:

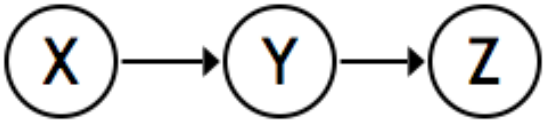
P(LungCancer = yes| FamilyHistory = yes, Smoker = yes) = **0.8**

P(LungCancer = no | FamilyHistory = no, Smoker = no) = **0.9**

Ref: http://www3.cs.stonybrook.edu/~cse634/ch6book.pdf

Independence in a Bayesian Networks

- Are two nodes independent given certain evidence?
- Example:

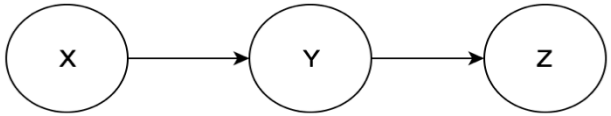


Question: are X and Z necessarily independent?

Answer: No. Example: low pressure causes rain, which causes traffic.
X can influence Z, Z can influence X (via Y)
They could be made independent: **How?**

Case 1: Causal Chains

This configuration is a "**causal chain**"



X: Low pressure Y: Rain Z: Traffic

$$P(x, y, z) = P(x)P(y|x)P(z|y)$$

- Guaranteed X independent of Z given Y?

$$P(z|x, y) = \frac{P(x, y, z)}{P(x, y)}$$

$$= \frac{P(x)P(y|x)P(z|y)}{P(x)P(y|x)}$$

$$= P(z|y)$$

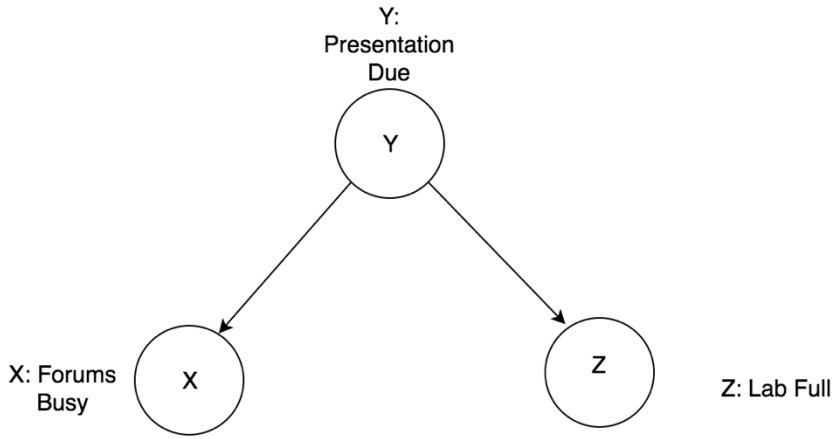
Yes!

- Evidence along the chain "blocks" the influence

Ref: http://ai.berkeley.edu/slides/Lecture%2017%20--%20Bayes%20Nets%20II%20Independence/

Case 2: Common Cause

This configuration is called "**common cause**"



- Guaranteed X and Z independent given Y?

$$P(z|x,y) = \frac{P(x,y,z)}{P(x,y)}$$

$$= \frac{P(y)P(x|y)P(z|y)}{P(y)P(x|y)}$$

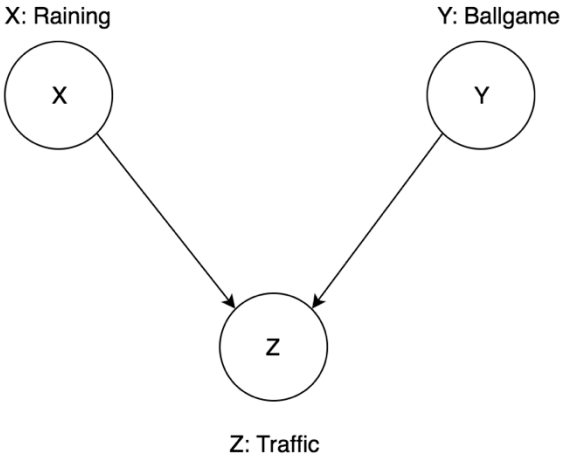
$$= P(z|y)$$

Yes!

Ref: http://ai.berkeley.edu/slides/Lecture%2017%20--%20Bayes%20Nets%20II%20Independence/

Case 3: Common Effect

This configuration is called "**common effect**"



- Are X and Y independent?
  - *Yes*: the ballgame and the rain cause traffic, but they are not correlated
  - Still need to prove they must be (try it!)
- Are X and Y independent given Z?
  - *No*: seeing traffic puts the rain and the ballgame in competition as explanation.
- This is backwards from the other cases
  - Observing an effect activates influence between possible causes.

Ref: http://ai.berkeley.edu/slides/Lecture%2017%20--%20Bayes%20Nets%20II%20Independence/

D-Separation

- Study independence properties for triples
- Analyze **3 main cases** in terms of member triples
- D-separation: a condition/algorithm for answering queries of the form:

$$X_i \perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$$

- Check all (undirected!) paths between X_ind X_j
  - If one or more active, then independence not guaranteed $X_i \not\perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$
  - Otherwise (i.e. if all paths are inactive),then independence is guaranteed $X_i \perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$

Ref: http://ai.berkeley.edu/slides/Lecture%2017%20--%20Bayes%20Nets%20II%20Independence/

Answering Queries

Answering queries like Query: $P(Q|E_1 = e_1, \ldots E_k = e_k)$

(we call this the **Posterior probability**) using a Bayesian Belief Network is termed as **Inference**

- **Probabilistic Inference**
  - Enumeration (exponential complexity)
  - Variable elimination (worst-case exponential complexity, but often better)
  - Inference is NP-complete

Result (Normalized) is a Joint Distribution capable of answering queries like above

Example:

P(Lung cancer = yes | PositiveXRay = No, Dyspnea = Yes) = 0.3

Variable Elimination: An Optimization

- Query: $P(Q|E_1 = e_1, \ldots E_k = e_k)$
- Start with initial factors:
  - Local CPTs (but instantiated by evidence)
- While there are still hidden variables (not Q or evidence):
  - Pick a hidden variable H
  - Join all factors mentioning H
  - Eliminate (sum out) H
- Join all remaining factors and normalize

Ref: http://ai.berkeley.edu/slides/Lecture%2017%20--%20Bayes%20Nets%20II%20Independence/

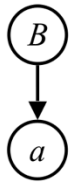
Answering Queries: A basic example

Query: $P(B|+a)$

Start / Select

$P(B)$

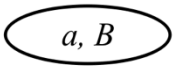
B	P
+b	0.1
¬b	0.9



$P(A|B) \rightarrow P(a|B)$

B	A	P
+b	+a	0.8
~~+b~~	~~¬a~~	~~0.2~~
¬b	+a	0.1
~~¬b~~	~~¬a~~	~~0.9~~

Join on B



$P(a, B)$

A	B	P
+a	+b	0.08
+a	¬b	0.09

Normalize

$P(B|a)$

A	B	P
+a	+b	8/17
+a	¬b	9/17

Classified Result = -b

Ref: http://ai.berkeley.edu/slides/Lecture%2017%20--%20Bayes%20Nets%20II%20Independence/

Training Bayesian Belief Networks

Scenario 1:

- The **Network Topology** is constructed by **human experts** or inferred from the data
- Human experts have a good grasp of the direct conditional dependencies in the domain.
- These probabilities can then be used to compute the remaining probability values.
- If network structure is known, and some variables are hidden (but not all), we use **gradient descent** (greedy hill-climbing) methods, analogous to neural network learning

Ref: http://www3.cs.stonybrook.edu/~cse634/ch6book.pdf

Training Bayesian Belief Networks

Scenario 2:

- The **Network Topology** is unknown
- Many algorithms exist for learning the network topology from the training data given observable variables
- We call these **Discrete Optimization Problems**
- If all variables are hidden (no specified variables): **No good algorithms** are known for this purpose

Ref: http://www3.cs.stonybrook.edu/~cse634/ch6book.pdf

Remarks

- Bayesian models are one of the **simplest and oldest classification algorithms** that are still relevant today.
- Widely used in areas such as **text classification and spam filtering**.
- Given the assumption of conditional Independence holds, a Naive Bayes classifier will **converge quicker than discriminative models** like logistic regression and requires less training time
- Due to its simplicity, this algorithm might outperform more complex models when the data set is not large

Part 2

Research Paper - COMPASS

COMPASS: Rotational Keyboard on Non-Touch Smartwatches

“ Sunggeun Ahn , Seongkook Heo , Geehyuk Lee, Xiaojun Bi , Yuanchun Shi,, October 17-20, 2017, Brighton, United Kingdom ”.

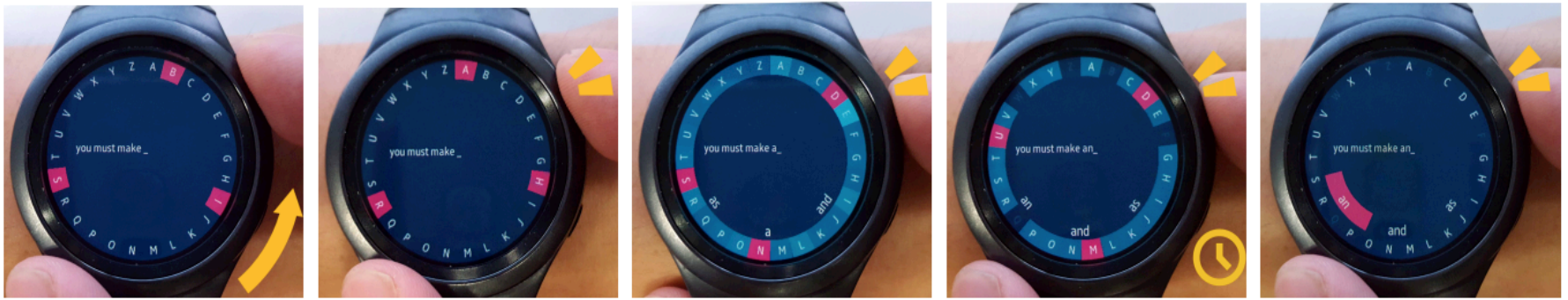
CHI '17 Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems

Pages 705-715

Rough Idea

In this paper, the authors designed and implemented COMPASS, a non-touch bezel-based text entry technique. COMPASS positions multiple cursors on a circular keyboard, with the location of each cursor dynamically optimized during typing to minimize rotational distance using the concept of Bayes theorem.

COMPASS: Rotational Keyboard on Non-Touch Smartwatches



Working

The authors employed Goodman et al.'s Bayesian model to predict the target word. Given user's input I, it calculates the probability of a word W in a predefined dictionary as:

$$P(W|I) \propto P(I|W) \times P(W)$$

As COMPASS used multiple cursors to select characters simultaneously, we have

$$I = \begin{matrix} I11\ I12 \cdots I1n \\ I21\ I22 \cdots I2n\ . \\ \cdots\cdots\cdots\cdots \\ IN1\ IN2 \cdots INn \end{matrix}$$

where N is the number of cursors, and n is the length of the input.

Working

we treat each input point independently therefore:

$$P(I|W) = \prod P(I_i|W_i) \quad \text{where i varies from 1 to n.}$$

where Ii refers to the ith column of I, and Wi is the ith character of W.

Meanwhile, we set

$$P(I_i|W_i) = 1 \text{ if } \exists\ 1 \le j \le N \quad \text{s.t. } I_{ji} = W_i$$

$$0 \text{ otherwise}$$

Corpus words

They took the top 15,000 words as well as their corresponding frequency in the American National Corpus as our corpus. According to Nation et al, this would be sufficient to cover over 95% of the common English words.

Visual Hint

When entering text using COMPASS, one of the critical factors that affect the text entry performance is the ease to visually acquire the target key. In this regard, they designed visual cues to help users find their target keys during text entry.

Visual Hint

Assuming the user has generated the input I, we denote S(I) as the set of all words W in the dictionary that P(I|W1W2...Wn) not equal to 0.

In other words, S(I) contains all words whose prefix matches the input I.

Now, for each character c in the alphabet, the probability of it being the following character can be calculated as:

$$P(c) = \sum_{W \in S(I) \wedge Wn+1=c} P(W) / \sum_{W \in S(I)} P(W)$$

Visual Hint

Based on P(c), they designed two kinds of visual cues, as shown in Figure. First, the keys whose probability was zero would be dimmed to avoid distracting the user (e.g. ‘F’ and ‘T’). Second, they adjusted the brightness of the remaining keys according to their probability.

Distance Function

Each time the user selects a character, the algorithm searches all the C N 26 possible cursor locations, and selects the one with the lowest **Expected Next Rotation Distance (ENRD)**, which was defined as:

$$\mathbf{ENRD} = \sum_{c \in \chi} \mathbf{dis(c) \times P(c)}$$

Where, χ is the set of all 26 characters,

dis(c) is the rotation distance that needed to hit key c with the closest cursor

Results

In the first experiment, they showed that dynamically adjusting the position of the cursors outperformed fixing their positions in terms of text entry speed and subjective preference, and the optimal number of cursors is three.

In the second experiment, they showed that by incorporating auto-completion algorithms, users could reach a pick-up speed of 9.3 WPM, and they could improve to 12.5 WPM after 90 minutes of practice. Some participants even reached 15.4 WPM.

Conclusion of the paper

COMPASS provides a potential solution for text entry on smartwatches without using the touchscreens, and could be implemented to other rotational interfaces.

https://www.pinterest.com/pin/471681760962616631/