# kNN & Naïve Bayes

School of Electrical and Computer Engineering

University of Tehran

Erfan Darzi

erfandarzi@ut.ac.ir

Today’s lecture

* Instance-based classifiers
  + k nearest neighbors
  + Non-parametric learning algorithm
* Model-based classifiers
  + Naïve Bayes classifier
* A generative model
  + Parametric learning algorithm

How to classify this document?

*Documents by vector*

*space representation*

**?**

Sports

Politics

# Let’s check the nearest neighbor

*Are you confident about this?*

**?**

Sports

Politics

# Let’s check more nearest neighbors

• Ask k nearest neighbors – Let them vote

**?**

Sports

Politics

# Probabilistic interpretation of kNN

* Approximate Bayes decision rule in a subset of data around the testing point
* Let 𝑉𝑉 be the volume of the 𝑚𝑚 dimensional ball around 𝑥𝑥 containing the 𝑘𝑘 nearest neighbors for 𝑥𝑥, we have *Nearest neighbors from class 1*

𝑝𝑝

𝑥𝑥

|

𝑦𝑦

=

1

=

𝑘𝑘

1

𝑁𝑁

1

𝑉𝑉

=

𝑁𝑁

1

𝑁𝑁

×

𝑘𝑘

1

𝑁𝑁

1

𝑉𝑉

𝑘𝑘

𝑁𝑁

𝑉𝑉

=

𝑘𝑘

1

𝑘𝑘

𝑘𝑘 𝑘𝑘𝑁𝑁1

𝑝𝑝 𝑥𝑥 𝑉𝑉 = 𝑁𝑁 => 𝑝𝑝 𝑥𝑥 = 𝑁𝑁𝑉𝑉 𝑝𝑝 𝑦𝑦 = 1 = 𝑁𝑁

*Total number of instances*

With Bayes rule: *Total number of*

*instances in class 1*

𝑝𝑝𝑦𝑦 = 1|𝑥𝑥

*Counting the nearest neighbors from class 1*

# kNN is close to optimal

* Asymptotically, the error rate of 1-nearestneighbor classification is less than twice of the Bayes error rate
* Decision boundary *A non-parametric estimation*

–

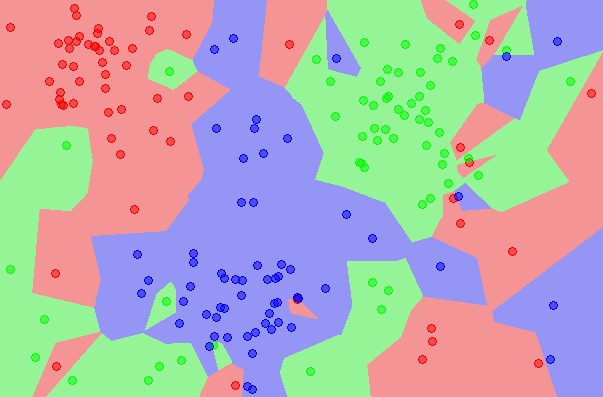
1

NN

-

Voronoi

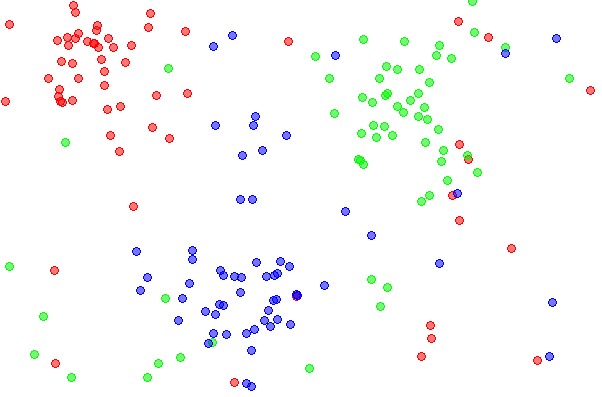
tessellation



*of posterior distribution*

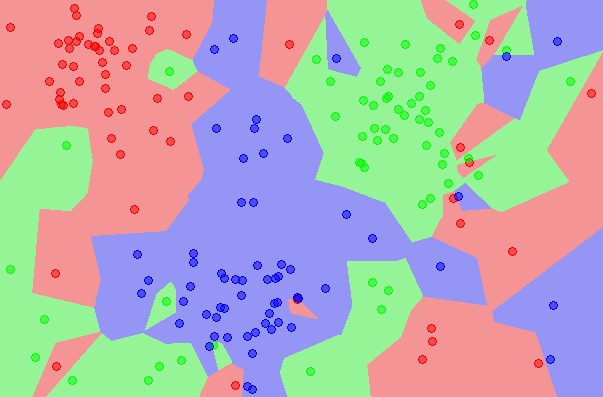
# Components in kNN

* A distance metric
  + Euclidean distance/cosine similarity
* How many nearby neighbors to look at
  + k
* Instance look up
  + Efficiently search nearby points
* Choice of k influences the “smoothness” of the resulting classifier



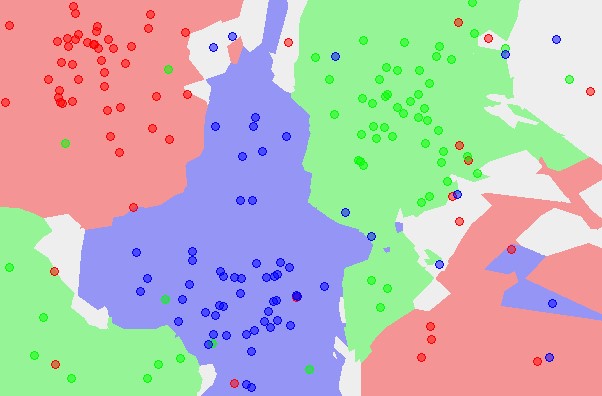
* Choice of k influences the “smoothness” of the resulting classifier

k=1



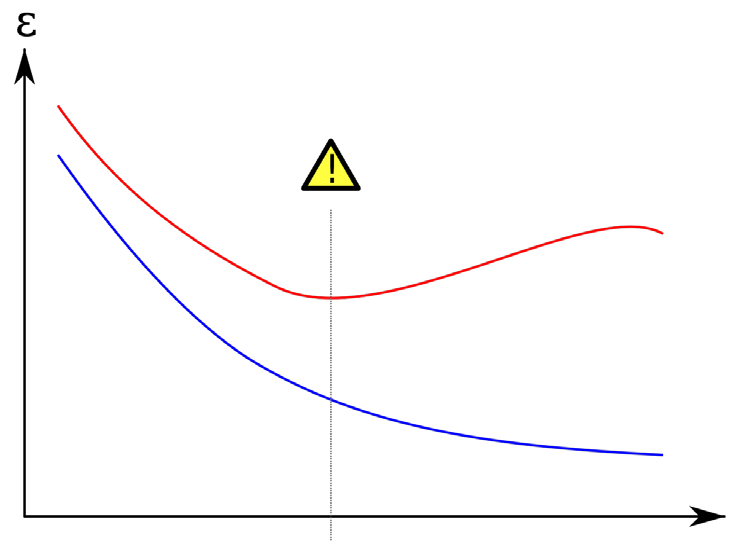
* Choice of k influences the “smoothness” of the resulting classifier

k=5



* Large k -> smooth shape for decision boundary
* Small k -> complicated decision boundary

Error

Error on testing set

Error on training set

**Larger kSmaller k**

Model complexity

* Recall MP1
  + In Yelp\_small data set, there are 629K reviews for training and 174K reviews for testing
  + Assume we have a vocabulary of 15K
  + Complexity of kNN
* 𝑂𝑂(𝑁𝑁𝑁𝑁𝑉𝑉)

Feature size

Training corpus size Testing corpus size

* Exact solutions
  + Build inverted index for text documents
* Special mapping: word -> document list
* Speed-up is limited when average document length is large

information

retrieval

retrieved

is

helpful

Doc1

Doc2

Doc1

Doc2

Doc1

Doc2

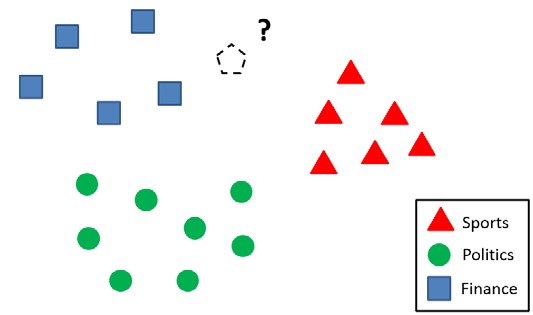
Doc1

Doc2

***Dictionary***

***Postings***

* Exact solutions
  + Build inverted index for text documents
* Special mapping: word -> document list
* Speed-up is limited when average document length is large
  + Parallelize the computation
* Map-Reduce
  + - Map training/testing data onto different reducers
    - Merge the nearest k neighbors from the reducers
* Approximate solution
  + Locality sensitive hashing
* Similar documents -> (likely) same hash values



h(x)

* Approximate solution
* Locality sensitive hashing
  + Similar documents -> (likely) same hash values
  + Construct the hash function such that similar items map to the same “buckets” with a high probability – Learning-based: learn the hash function with annotated examples, e.g., must-link, cannot-link
* Random projection
* Approximate the cosine similarity between vectors
  + ℎ𝑟𝑟 𝑥𝑥 = 𝑠𝑠𝑠𝑠𝑠𝑠(𝑥𝑥 ⋅ 𝑟𝑟), 𝑟𝑟 is a **random** unit vector

|  |  |  |  |
| --- | --- | --- | --- |
|  | 𝒓𝒓𝟏𝟏 | 𝒓𝒓𝟐𝟐 | 𝒓𝒓𝟑𝟑 |
| 𝐷𝐷𝑥𝑥 | 1 | 1 | 0 |
| 𝐷𝐷𝑦𝑦 | 1 | 0 | 1 |

* + Each 𝑟𝑟 defines one hash function, i.e., one bit in the hash value

𝐷𝐷𝑥𝑥 𝐷𝐷𝑥𝑥 𝐷𝐷𝑥𝑥

𝐷𝐷

𝑦𝑦

𝜃𝜃

𝒓𝒓

𝟏𝟏

𝐷𝐷

𝑦𝑦

𝜃𝜃

𝒓𝒓

𝟐𝟐

𝐷𝐷

𝑦𝑦

𝜃𝜃

𝒓𝒓𝟑𝟑

* Approximate the cosine similarity between vectors
  + ℎ𝑟𝑟 𝑥𝑥 = 𝑠𝑠𝑠𝑠𝑠𝑠(𝑥𝑥 ⋅ 𝑟𝑟), 𝑟𝑟 is a random unit vector

|  |  |  |  |
| --- | --- | --- | --- |
|  | 𝒓𝒓𝟏𝟏 | 𝒓𝒓𝟐𝟐 | 𝒓𝒓𝟑𝟑 |
| 𝐷𝐷𝑥𝑥 | 1 | 0 | 1 |
| 𝐷𝐷𝑦𝑦 | 1 | 0 | 1 |

* + Each 𝑟𝑟 defines one hash function, i.e., one bit in the hash value

𝐷𝐷

𝑥𝑥

𝜃𝜃

𝒓𝒓

𝟏𝟏 𝐷𝐷𝑥𝑥 𝐷𝐷𝑥𝑥

𝐷𝐷𝑦𝑦 𝐷𝐷𝑦𝑦

𝒓𝒓

𝟐𝟐

𝜃𝜃

𝐷𝐷

𝑦𝑦

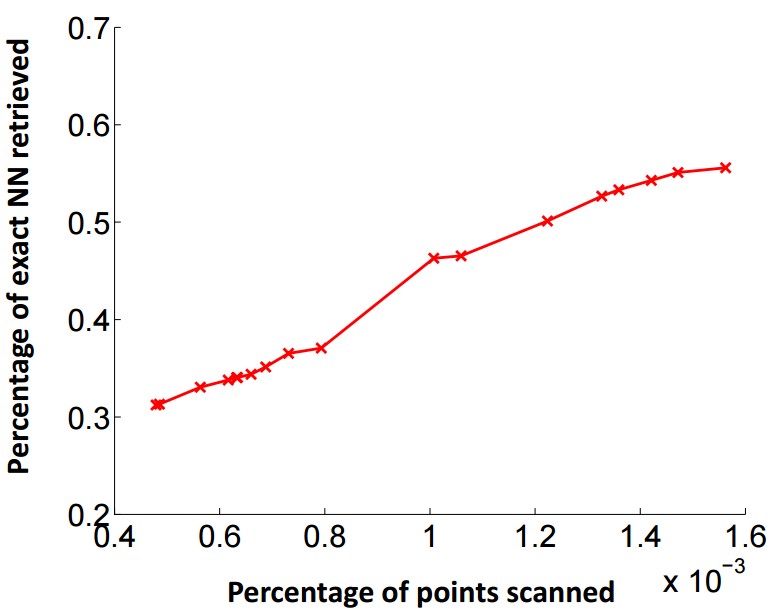
𝜃𝜃

𝒓𝒓𝟑𝟑

* Approximate the cosine similarity between vectors
  + ℎ𝑟𝑟 𝑥𝑥 = 𝑠𝑠𝑠𝑠𝑠𝑠(𝑥𝑥 ⋅ 𝑟𝑟), 𝑟𝑟 is a random unit vector
  + Each 𝑟𝑟 defines one hash function, i.e., one bit in the hash value
  + Provable approximation error
* 𝑃𝑃 ℎ 𝑥𝑥 = ℎ 𝑦𝑦 = 1 − 𝜃𝜃(𝑥𝑥𝜋𝜋,𝑦𝑦)

Efficient instance look-up

* Effectiveness of random projection – 1.2M images + 1000 dimensions



1000

x speed-up

# Weight the nearby instances

• When the data distribution is highly skewed, frequent classes might dominate majority vote – They occur more often in the k nearest neighbors just because they have large volume

**?**

Sports

Politics

Finance

# Weight the nearby instances

* When the data distribution is highly skewed, frequent classes might dominate majority vote – They occur more often in the k nearest neighbors just because they have large volume
* Solution

– Weight the neighbors in voting

* 𝑤𝑤𝑥𝑥, 𝑥𝑥𝑖𝑖 = |𝑥𝑥−𝑥𝑥1 𝑖𝑖| or 𝑤𝑤𝑥𝑥, 𝑥𝑥𝑖𝑖 = cos(𝑥𝑥, 𝑥𝑥𝑖𝑖)

# Summary of kNN

* Instance-based learning
  + No training phase
  + Assign label to a testing case by its nearest neighbors
  + Non-parametric
  + Approximate Bayes decision boundary in a local region
* Efficient computation – Locality sensitive hashing
* Random projection

Recall optimal Bayes decision boundary

* 𝑓𝑓 𝑋𝑋 = 𝑎𝑎𝑟𝑟𝑠𝑠𝑚𝑚𝑎𝑎𝑥𝑥𝑦𝑦𝑃𝑃(𝑦𝑦|𝑋𝑋)

*\*Optimal Bayes decision boundary*

𝑋𝑋

𝑝𝑝

(

𝑋𝑋

,

𝑦𝑦

)

𝑝𝑝

𝑋𝑋

𝑦𝑦

=

1

𝑝𝑝

(

𝑦𝑦

=

1)

𝑝𝑝

𝑋𝑋

𝑦𝑦

=

0

𝑝𝑝

(

𝑦𝑦

=

0)

𝑦𝑦

=

0

𝑦𝑦

=

1

***False negative False positive***

# Estimating the optimal classifier

• 𝑓𝑓 𝑋𝑋 = 𝑎𝑎𝑟𝑟𝑠𝑠𝑚𝑚𝑎𝑎𝑥𝑥𝑦𝑦𝑃𝑃 𝑦𝑦𝑋𝑋 ***Requirement:***

## = 𝑎𝑎𝑟𝑟𝑠𝑠𝑚𝑚𝑎𝑎𝑥𝑥𝑦𝑦𝑃𝑃 𝑋𝑋 𝑦𝑦 𝑃𝑃(𝑦𝑦) **|D|>>**𝒀𝒀 ×(𝟐𝟐𝑽𝑽−𝟏𝟏)

Class conditional density Class prior

#parameters: 𝑌𝑌 × (2𝑉𝑉 − 1) 𝑌𝑌 − 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | text | information | identify | mining | mined | is | useful | to | from | apple | delicious | **Y** |
| D1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | **1** |
| D2 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | **1** |
| D3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | **0** |

V binary features

# We need to simplify this

* Features are conditionally independent given class labels
  + 𝑝𝑝𝑥𝑥1, 𝑥𝑥2 𝑦𝑦 = 𝑝𝑝 𝑥𝑥2 𝑥𝑥1, 𝑦𝑦𝑝𝑝(𝑥𝑥1|𝑦𝑦)

= 𝑝𝑝𝑥𝑥2 𝑦𝑦 𝑝𝑝(𝑥𝑥1|𝑦𝑦)

* + E.g.,

𝑝𝑝 ‘𝑤𝑤ℎ𝑤𝑤𝑤𝑤𝑤𝑤 ℎ𝑜𝑜𝑜𝑜𝑠𝑠𝑤𝑤𝑜, ‘𝑜𝑜𝑜𝑜𝑎𝑎𝑚𝑚𝑎𝑎𝑜𝑝𝑝𝑜𝑜𝑝𝑝𝑤𝑤𝑤𝑤𝑤𝑤𝑝𝑝𝑎𝑎𝑝𝑝 𝑠𝑠𝑤𝑤𝑤𝑤𝑠𝑠 =

𝑝𝑝 ‘𝑤𝑤ℎ𝑤𝑤𝑤𝑤𝑤𝑤 ℎ𝑜𝑜𝑜𝑜𝑠𝑠𝑤𝑤𝑜𝑝𝑝𝑜𝑜𝑝𝑝𝑤𝑤𝑤𝑤𝑤𝑤𝑝𝑝𝑎𝑎𝑝𝑝 𝑠𝑠𝑤𝑤𝑤𝑤𝑠𝑠 ×

𝑝𝑝(‘𝑜𝑜𝑜𝑜𝑎𝑎𝑚𝑚𝑎𝑎𝑜|𝑝𝑝𝑜𝑜𝑝𝑝𝑤𝑤𝑤𝑤𝑤𝑤𝑝𝑝𝑎𝑎𝑝𝑝 𝑠𝑠𝑤𝑤𝑤𝑤𝑠𝑠)

*This does not mean ‘white house’ is independent of ‘obama’!*

Conditional v.s. marginal independence • Features are not necessarily marginally independent from each other

* 𝑝𝑝‘𝑤𝑤ℎ𝑤𝑤𝑤𝑤𝑤𝑤 ℎ𝑜𝑜𝑜𝑜𝑠𝑠𝑤𝑤𝑜‘𝑜𝑜𝑜𝑜𝑎𝑎𝑚𝑚𝑎𝑎𝑜 > 𝑝𝑝(‘𝑤𝑤ℎ𝑤𝑤𝑤𝑤𝑤𝑤 ℎ𝑜𝑜𝑜𝑜𝑠𝑠𝑤𝑤𝑜)
* However, once we know the class label, features become independent from each other
  + Knowing it is already political news, observing

‘obama’ contributes little about occurrence of ‘while house’

# Naïve Bayes classifier

• 𝑓𝑓 𝑋𝑋 = 𝑎𝑎𝑟𝑟𝑠𝑠𝑚𝑚𝑎𝑎𝑥𝑥𝑦𝑦𝑃𝑃 𝑦𝑦𝑋𝑋

= 𝑎𝑎𝑟𝑟𝑠𝑠𝑚𝑚𝑎𝑎𝑥𝑥𝑦𝑦𝑃𝑃 𝑋𝑋 𝑦𝑦 𝑃𝑃(𝑦𝑦)

𝑉𝑉

= 𝑎𝑎𝑟𝑟𝑠𝑠𝑚𝑚𝑎𝑎𝑥𝑥𝑦𝑦 𝑃𝑃(𝑥𝑥𝑖𝑖|𝑦𝑦)𝑃𝑃 𝑦𝑦

𝑖𝑖=1

|  |  |
| --- | --- |
| Class conditional density | Class prior |
| #parameters: 𝑌𝑌 × (𝑉𝑉− 1)  v.s. | 𝑌𝑌 − 1 |

***Computationally feasible*** 𝑌𝑌 × (2𝑉𝑉 − 1)

# Naïve Bayes classifier

* 𝑓𝑓 𝑋𝑋 = 𝑎𝑎𝑟𝑟𝑠𝑠𝑚𝑚𝑎𝑎𝑥𝑥𝑦𝑦𝑃𝑃 𝑦𝑦𝑋𝑋

= 𝑎𝑎𝑟𝑟𝑠𝑠𝑚𝑚𝑎𝑎𝑥𝑥𝑦𝑦𝑃𝑃 𝑋𝑋 𝑦𝑦 𝑃𝑃(𝑦𝑦) ***By Bayes rule***

𝑉𝑉

= 𝑎𝑎𝑟𝑟𝑠𝑠𝑚𝑚𝑎𝑎𝑥𝑥𝑦𝑦 𝑃𝑃(𝑥𝑥𝑖𝑖|𝑦𝑦)𝑃𝑃 𝑦𝑦

𝑖𝑖=1

***By conditional independence assumption***

y

x

2

x

3

x

v

x

1

…

Estimating parameters

* Maximial likelihood estimator

– 𝑃𝑃

–

𝑃𝑃

𝑥𝑥

𝑖𝑖

𝑦𝑦

=

∑

𝑑𝑑

∑

𝑗𝑗

𝛿𝛿

(

𝑥𝑥

𝑑𝑑

𝑗𝑗

=

𝑤𝑤

𝑖𝑖

,

𝑦𝑦

𝑑𝑑

=

𝑦𝑦

)

∑

𝑑𝑑

𝛿𝛿

(

𝑦𝑦

𝑑𝑑

𝑦𝑦

=

)

(

𝑦𝑦

)

=

∑

𝑑𝑑

𝛿𝛿

(

𝑦𝑦

𝑑𝑑

=

𝑦𝑦

)

∑

𝑑𝑑

1



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | text | information | identify | mining | mined | is | useful | to | from | apple | delicious | **Y** |
| D1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | **1** |
| D2 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | **1** |
| D3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | **0** |

## classification I

* The frequency of words in a document matters
  + 𝑃𝑃 𝑋𝑋 𝑦𝑦 = ∏𝑖𝑖=1|𝑑𝑑| 𝑃𝑃 𝑥𝑥𝑖𝑖 𝑦𝑦 𝑐𝑐(𝑥𝑥𝑖𝑖,𝑑𝑑)
  + In log space ***Essentially, estimating*** |𝒀𝒀| ***different language models!***
* 𝑓𝑓𝑦𝑦, 𝑋𝑋 = 𝑎𝑎𝑟𝑟𝑠𝑠𝑚𝑚𝑎𝑎𝑥𝑥𝑦𝑦 log 𝑃𝑃 𝑦𝑦 𝑋𝑋 |𝑑𝑑|

= 𝑎𝑎𝑟𝑟𝑠𝑠𝑚𝑚𝑎𝑎𝑥𝑥𝑦𝑦 log 𝑃𝑃(𝑦𝑦) + 𝑝𝑝(𝑥𝑥𝑖𝑖, 𝑑𝑑) log 𝑃𝑃(𝑥𝑥𝑖𝑖|𝑦𝑦)

𝑖𝑖=1

Class bias Feature vector Model parameter

## classification

• For binary case

𝑃𝑃 𝑦𝑦 = 1 𝑋𝑋

– 𝑓𝑓 𝑋𝑋 = 𝑠𝑠𝑠𝑠𝑠𝑠log

𝑃𝑃 𝑦𝑦 = 0 𝑋𝑋

𝑑𝑑

𝑃𝑃 𝑦𝑦 = 1𝑃𝑃 𝑥𝑥𝑖𝑖 𝑦𝑦 = 1

= 𝑠𝑠𝑠𝑠𝑠𝑠 log + 𝑝𝑝 𝑥𝑥𝑖𝑖, 𝑑𝑑 log

𝑃𝑃 𝑦𝑦 = 0𝑃𝑃 𝑥𝑥𝑖𝑖 𝑦𝑦 = 0

𝑖𝑖=1

= 𝑠𝑠𝑠𝑠𝑠𝑠(𝑤𝑤𝑇𝑇𝑥𝑥̅)

log

𝑃𝑃

𝑦𝑦

=

1

𝑃𝑃

𝑦𝑦

=

0

,log

a linear model with vector space representation? where

𝑃𝑃 𝑥𝑥

𝑤𝑤 =1 𝑦𝑦 = 1 , … , log 𝑃𝑃 𝑥𝑥𝑣𝑣 𝑦𝑦 = 1

𝑃𝑃 𝑥𝑥1 𝑦𝑦 = 0 𝑃𝑃 𝑥𝑥𝑣𝑣 𝑦𝑦 = 0

𝑥𝑥̅ = (1, 𝑝𝑝(𝑥𝑥1, 𝑑𝑑), … , 𝑝𝑝(𝑥𝑥𝑣𝑣, 𝑑𝑑))

We will come back to this topic later.

## classification II

* Usually, features are not conditionally independent
  + 𝑝𝑝 𝑋𝑋 𝑦𝑦 ≠ ∏|𝑖𝑖=1𝑑𝑑| 𝑃𝑃(𝑥𝑥𝑖𝑖|𝑦𝑦)
* Enhance the conditional independence assumptions by N-gram language models
  + 𝑝𝑝 𝑋𝑋 𝑦𝑦 = ∏|𝑖𝑖=1𝑑𝑑| 𝑃𝑃(𝑥𝑥𝑖𝑖|𝑥𝑥𝑖𝑖−1, … , 𝑥𝑥𝑖𝑖−𝑁𝑁+1, 𝑦𝑦)

## classification III

* Sparse observation
  + 𝛿𝛿 𝑥𝑥𝑑𝑑𝑗𝑗 = 𝑤𝑤𝑖𝑖, 𝑦𝑦𝑑𝑑 = 𝑦𝑦 = 0 ⇒ 𝑝𝑝𝑥𝑥𝑖𝑖|𝑦𝑦 = 0
  + Then, no matter what values the other features take, 𝑝𝑝𝑥𝑥1, … , 𝑥𝑥𝑖𝑖, … , 𝑥𝑥𝑉𝑉|𝑦𝑦 = 0
* Smoothing class conditional density – All smoothing techniques we have discussed in language models are applicable here

# Maximum a Posterior estimator

* Adding pseudo instances *Can be estimated from a related* 
  + Priors: 𝑞𝑞(𝑦𝑦) and 𝑞𝑞(𝑥𝑥, 𝑦𝑦)*corpus or manually tuned*
  + MAP estimator for Naïve Bayes
* 𝑃𝑃 𝑥𝑥𝑖𝑖 𝑦𝑦 = ∑𝑑𝑑 ∑𝑗𝑗 𝛿𝛿∑(𝑥𝑥𝛿𝛿𝑑𝑑𝑗𝑗(=𝑦𝑦𝑤𝑤𝑖𝑖=,𝑦𝑦𝑦𝑦𝑑𝑑)=+𝑦𝑦𝑀𝑀𝑀𝑀)+(𝑀𝑀𝑀𝑀𝑦𝑦) (𝑥𝑥𝑖𝑖,𝑦𝑦)

𝑑𝑑

𝑑𝑑

#pseudo instances

# Summary of Naïve Bayes

* Optimal Bayes classifier
  + Naïve Bayes with independence assumptions
* Parameter estimation in Naïve Bayes
  + Maximum likelihood estimator
  + Smoothing is necessary

# Today’s reading

* Introduction to Information Retrieval – Chapter 13: Text classification and Naive Bayes
* 13.2 – Naive Bayes text classification
* 13.4 – Properties of Naive Bayes – Chapter 14: Vector space classification
* 14.3 k nearest neighbor
* 14.4 Linear versus nonlinear classifiers