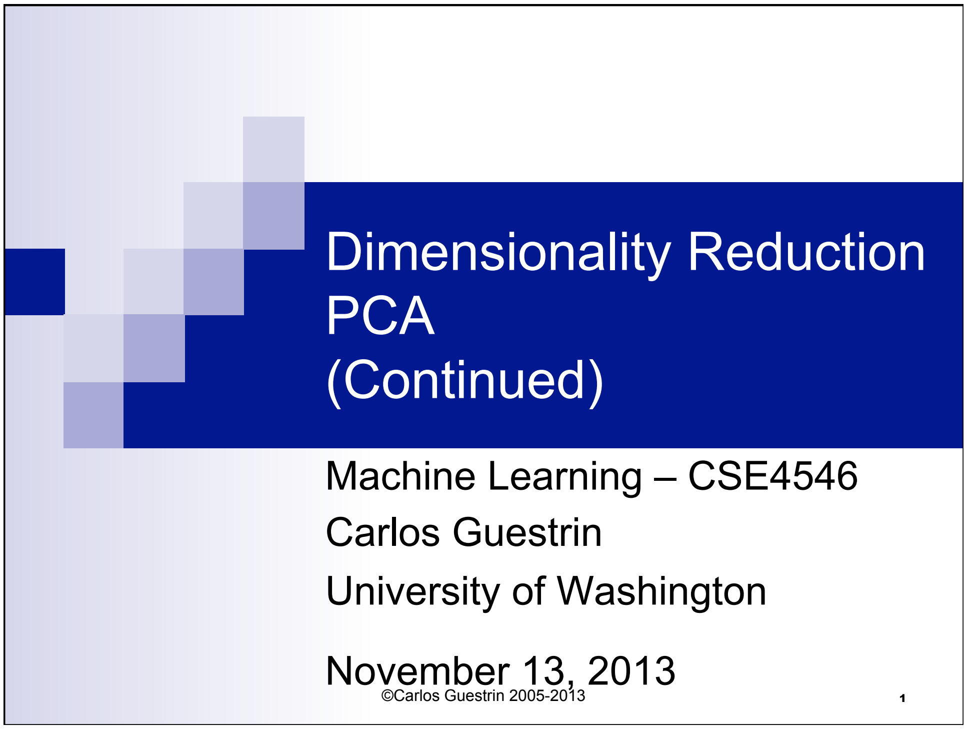


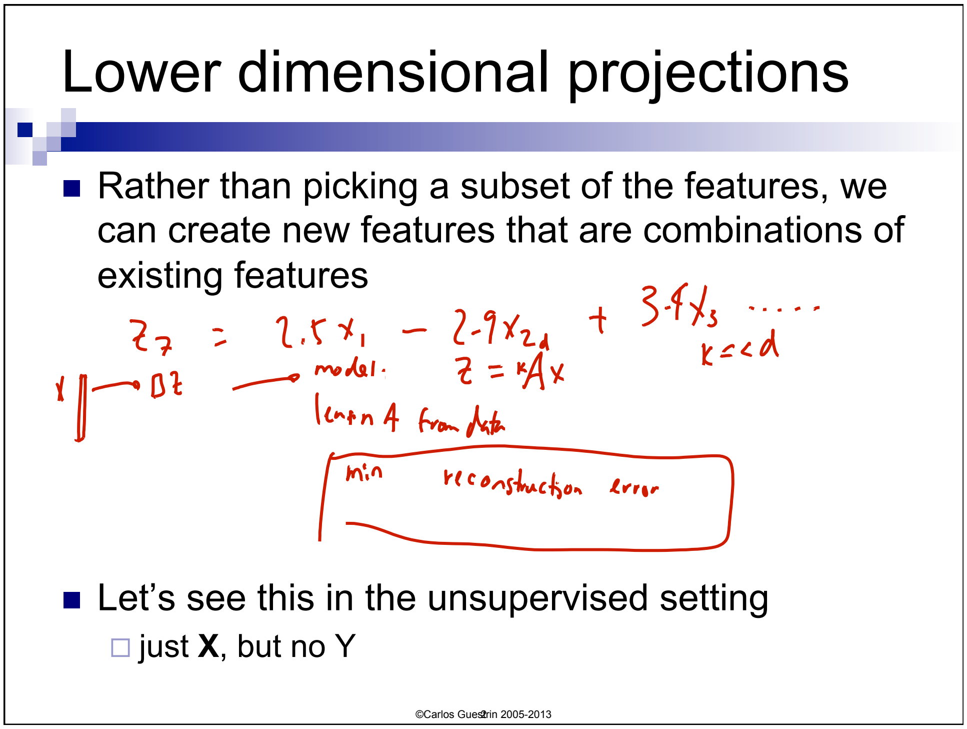
Dimensionality Reduction PCA (Continued)

Machine Learning – CSE4546

Carlos Guestrin

University of Washington

November 13, 2013



Lower dimensional projections

- Rather than picking a subset of the features, we can create new features that are combinations of existing features

$z_7 = 2.5 x_1 - 2.9 x_{2d} + 3.4 x_3 \ldots\ldots$

$k << d$

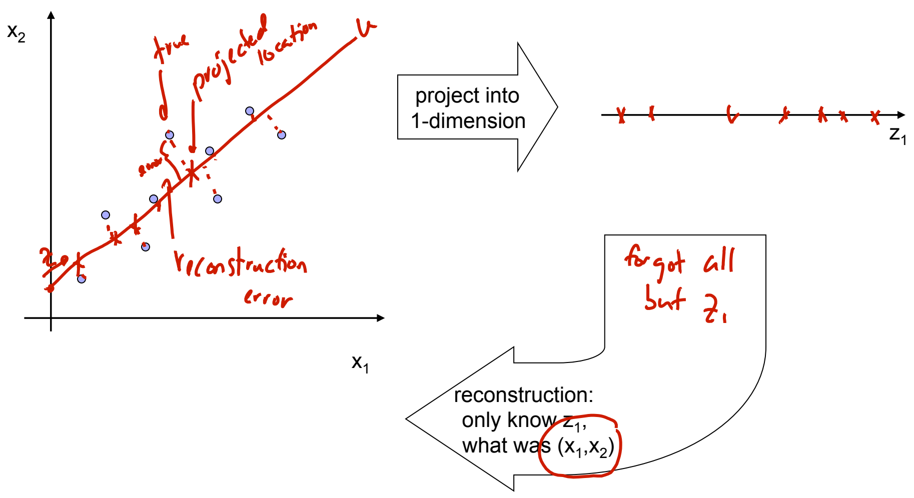
$x \, \| \rightarrow \square z \rightarrow$ model. $z = {}^{k}A x$

learn A from data

$\boxed{\text{min reconstruction error}}$

- Let's see this in the unsupervised setting
  - □ just **X**, but no Y

Linear projection and reconstruction



project into 1-dimension

reconstruction: only know z_1, what was (x_1,x_2)

PCA finds projection that minimizes reconstruction error

- Given N data points: $\mathbf{x}^i = (x_1^i,\ldots,x_d^i)$, i=1…N
- Will represent each point as a projection:

  coeffs of projection

  mean

  □ $\hat{\mathbf{x}}^i = \bar{\mathbf{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$ where: $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^i$ and $z_j^i = (\mathbf{x}^i - \bar{\mathbf{x}}) \cdot \mathbf{u}_j$

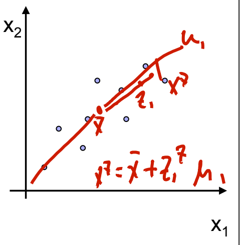
- PCA: 3 1000000
  - □ Given k<<d, find $(\mathbf{u}_1,\ldots,\mathbf{u}_k)$ minimizing reconstruction error:

    Sum over points

    $$error_k = \sum_{i=1}^{N} (\mathbf{x}^i - \hat{\mathbf{x}}^i)^2 \leftarrow \text{squared}$$

    true

    projection onto lower dimension



Reconstruction error and covariance matrix

$$error_k = \sum_{i=1}^{N} \sum_{j=k+1}^{d} [\underbrace{\mathbf{u}_j \cdot (\mathbf{x}^i - \bar{\mathbf{x}})}_{z_j^i}]^2$$

$$= \sum_{i=1}^{N} \sum_{j=k+1}^{d} u_j^T (x^i - \bar{x}) (x^i - \bar{x})^T u_j$$

$$= \sum_{j=k+1}^{d} u_j^T \underbrace{\left[\sum_{i=1}^{N} (x^i - \bar{x}) (x^i - \bar{x})^T \right]}_{N \Sigma} u_j$$

min error:

$$\min_{u} N \sum_{j=k+1}^{d} u_j^T \Sigma u_j$$

↑ find u_j that minimize this error

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}^i - \bar{\mathbf{x}})(\mathbf{x}^i - \bar{\mathbf{x}})^T$$

$$\Sigma = \begin{pmatrix} \sigma_i^2 & & \sigma_{ij} \\ & & \\ & & \end{pmatrix}$$

$$\sigma_{uv} \overset{MLE}{=} \frac{1}{N} \sum_{i=1}^{N} (x_u^i - \bar{x}_u)(x_v^i - \bar{x}_v)$$

in vector format

$$\Sigma \overset{MLE}{=} \frac{1}{N} \sum_{i=1}^{N} (x^i - \bar{x})(x^i - \bar{x})^T$$

Minimizing reconstruction error and eigen vectors

- Minimizing reconstruction error equivalent to picking orthonormal basis ($\mathbf{u}_1$,…,$\mathbf{u}_d$) minimizing:

$$error_k = N \sum_{j=k+1}^{d} \mathbf{u}_j^T \Sigma \mathbf{u}_j$$

- Eigen vector:
- Minimizing reconstruction error equivalent to picking ($\mathbf{u}_{k+1}$,…,$\mathbf{u}_d$) to be eigen vectors with smallest eigen values

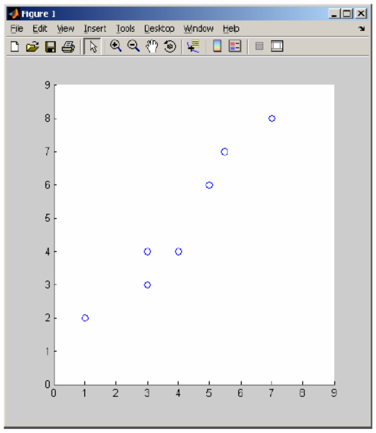
Basic PCA algoritm

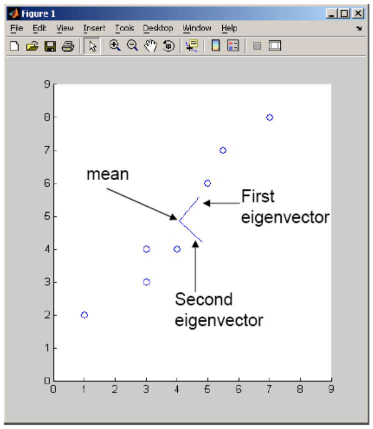
- Start from N by d data matrix **X**
- **Recenter**: subtract mean from each row of **X**
  - □ $\mathbf{X_c} \leftarrow \mathbf{X} - \overline{\mathbf{X}}$
- **Compute covariance matrix**:
  - □ $\Sigma \leftarrow 1/N\ \mathbf{X_c}^T \mathbf{X_c}$
- Find **eigen vectors and values** of Σ
- **Principal components:** k eigen vectors with highest eigen values

©Carlos Guestrin 2005-2013

PCA example

$$\hat{\mathbf{x}}^i = \bar{\mathbf{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$$



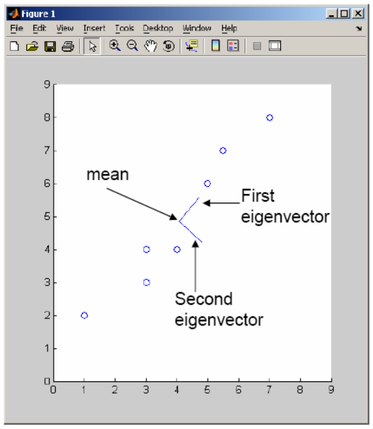


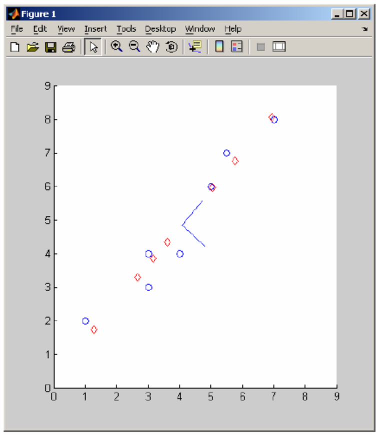
©Carlos Guestrin 2005-2013

PCA example – reconstruction

$$\widehat{\mathbf{x}}^i = \bar{\mathbf{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$$

only used first principal component

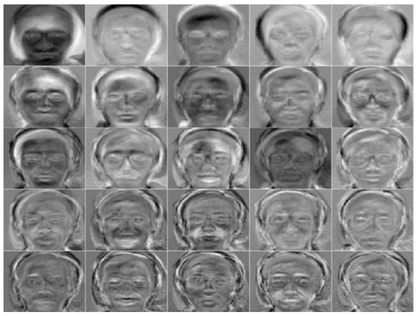




Eigenfaces [Turk, Pentland '91]

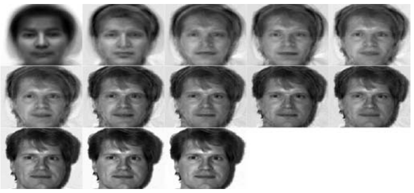
- Input images:

- Principal components:



Eigenfaces reconstruction

- Each image corresponds to adding k principal components:



Scaling up

- Covariance matrix can be really big!
  - □ Σ is d by d
  - □ Say, only 10000 features
  - □ finding eigenvectors is very slow…
- Use singular value decomposition (SVD)
  - □ finds k eigenvectors
  - □ great implementations available, e.g., python, R, Matlab svd
  - □ Never have to form Σ explicitly!

SVD

- Write **X = W S V**^{T}
  - □ **X** ← data matrix, one row per datapoint
  - □ **W** ← weight matrix, one row per datapoint – coordinate of $\mathbf{x}^i$ in eigenspace
  - □ **S** ← singular value matrix, diagonal matrix
    - in our setting each entry is eigenvalue λ_j
  - □ **V**^{T} ← singular vector matrix
    - in our setting each row is eigenvector $\mathbf{v}_j$

PCA using SVD algoritm

- Start from m by n data matrix **X**
- **Recenter**: subtract mean from each row of **X**
  - □ $\mathbf{X_c} \leftarrow \mathbf{X} - \overline{\mathbf{X}}$
- Call SVD algorithm on $\mathbf{X_c}$ – ask for top k singular vectors
- **Principal components:** k singular vectors with highest singular values (rows of **V**^{T})
  - □ **Coefficients** become:

What you need to know

- Dimensionality reduction
  - □ why and when it's important
- Simple feature selection
- Principal component analysis
  - □ minimizing reconstruction error
  - □ relationship to covariance matrix and eigenvectors
  - □ using SVD

Naïve Bayes

Machine Learning – CSE546
Emily Fox
University of Washington

November 18, 2013

Classification

- **Learn**: h:**X** $\mapsto$ Y
  - □ **X** – features
  - □ Y – target classes

- Suppose you know P(Y|**X**) exactly, how should you classify?
  - □ Bayes optimal classifier:

Bayes Rule

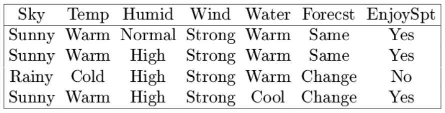
$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Which is shorthand for:

$$(\forall i, j) P(Y = y_i | X = x_j) = \frac{P(X = x_j | Y = y_i) P(Y = y_i)}{P(X = x_j)}$$

How hard is it to learn the optimal classifier?

- Data =



Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

- How do we represent these? How many parameters?
  - □ Prior, P(Y):
    - Suppose Y is composed of *k* classes
  - □ Likelihood, P(**X**|Y):
    - Suppose **X** is composed of *d* binary features
- Complex model ! High variance with limited data!!!

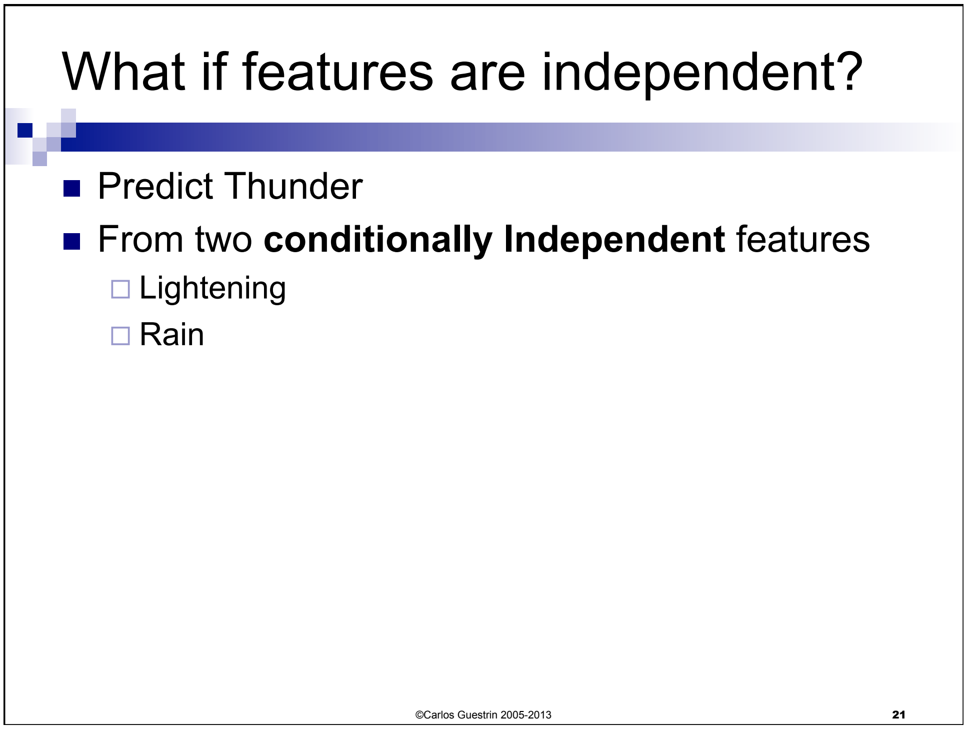
Conditional Independence

- X is **conditionally independent** of Y given Z, if the probability distribution governing X is independent of the value of Y, given the value of Z

$$(\forall i,j,k) P(X=i|Y=j,Z=k) = P(X=i|Z=k)$$

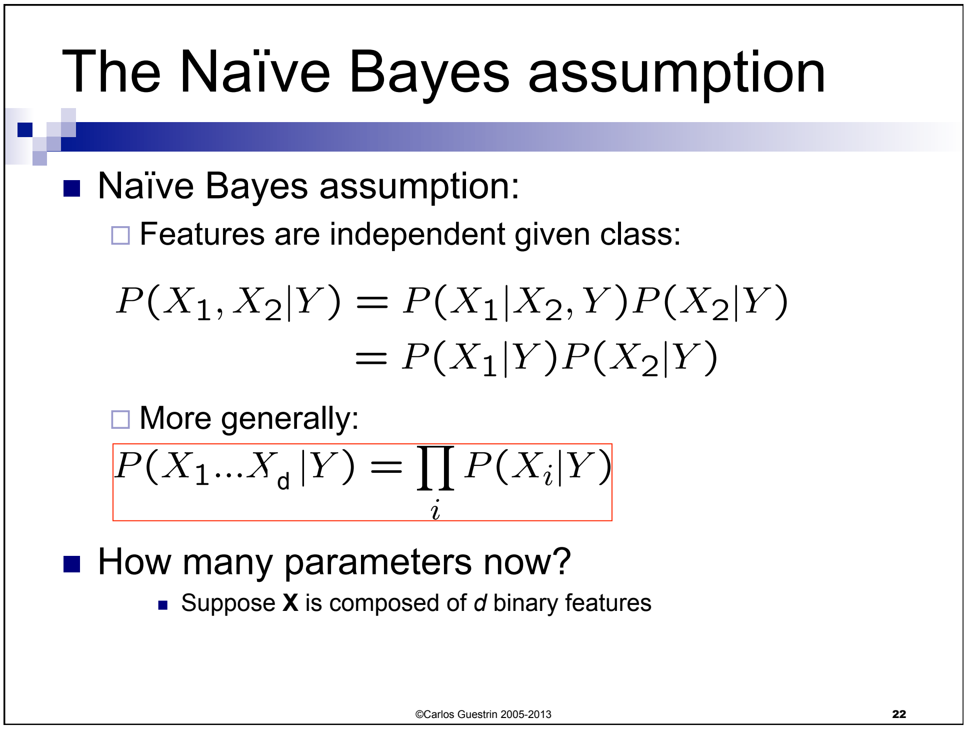
- e.g., $P(Thunder|Rain, Lightning) = P(Thunder|Lightning)$
- Equivalent to:

$$P(X,Y \mid Z) = P(X \mid Z)P(Y \mid Z)$$



What if features are independent?

- Predict Thunder
- From two **conditionally Independent** features
  - □ Lightening
  - □ Rain



The Naïve Bayes assumption

- Naïve Bayes assumption:
  - □ Features are independent given class:

$$
\begin{aligned}
P(X_1, X_2|Y) &= P(X_1|X_2, Y)P(X_2|Y) \\
&= P(X_1|Y)P(X_2|Y)
\end{aligned}
$$

  - □ More generally:

$$\boxed{P(X_1...X_{\mathrm{d}}|Y) = \prod_i P(X_i|Y)}$$

- How many parameters now?
  - Suppose **X** is composed of *d* binary features

The Naïve Bayes Classifier

- Given:
  - □ Prior P(Y)
  - □ *d* conditionally independent features **X** given the class Y
  - □ For each X_i, we have likelihood $P(X_i|Y)$

- Decision rule:

$$\begin{aligned} y^* = h_{NB}(\mathbf{x}) &= \arg\max_y P(y)P(x_1,\ldots,x_d \mid y) \\ &= \arg\max_y P(y)\prod_i P(x_i|y) \end{aligned}$$

- If assumption holds, NB is optimal classifier!

MLE for the parameters of NB

- Given dataset
  - □ Count(A=a,B=b) == number of examples where A=a and B=b

- MLE for NB, simply:
  - □ Prior: P(Y=y) =
  - □ Likelihood: $P(X_i=x_i|Y=y)$ =

Subtleties of NB classifier 1 – Violating the NB assumption

- Usually, features are not conditionally independent:

$$P(X_1 ... X_{\text{d}}|Y) \neq \prod_i P(X_i|Y)$$

- Actual probabilities P(Y|**X**) often biased towards 0 or 1
- Nonetheless, NB is the single most used classifier out there
  - □ NB often performs well, even when assumption is violated
  - □ [Domingos & Pazzani ’96] discuss some conditions for good performance

Subtleties of NB classifier 2 – Insufficient training data

- What if you never see a training instance where X_1=a when Y=b?
  - □ e.g., Y={SpamEmail}, X_1={‘CSE546’}
  - □ P(X_1=a | Y=b) = 0
- Thus, no matter what the values X_2,…,X_d take:
  - □ P(Y=b | X_1=a,X_2,…,X_d) = 0

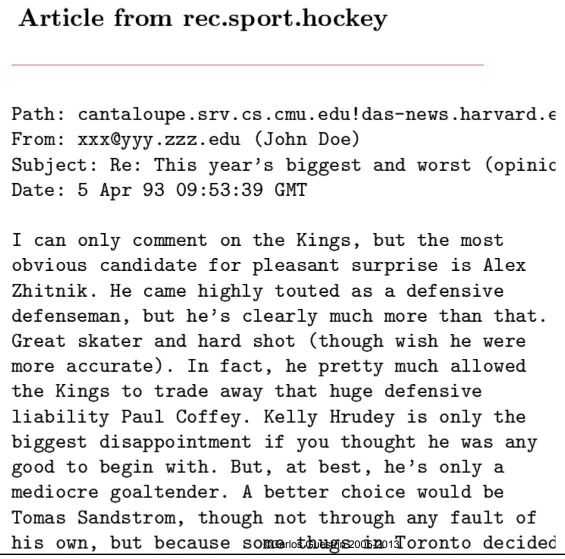
- “Solution”: smoothing
  - □ Add “fake” counts, usually uniformly distributed
  - □ Equivalent to Bayesian Learning

Text classification

- Classify e-mails
  - □ Y = {Spam,NotSpam}
- Classify news articles
  - □ Y = {what is the topic of the article?}
- Classify webpages
  - □ Y = {student, professor, project, …}

- What about the features **X**?
  - □ The text!

Features **X** are entire document – X_i for i^{th} word in article



Article from rec.sport.hockey

```
Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.e
From: xxx@yyy.zzz.edu (John Doe)
Subject: Re: This year's biggest and worst (opinic
Date: 5 Apr 93 09:53:39 GMT

I can only comment on the Kings, but the most
obvious candidate for pleasant surprise is Alex
Zhitnik. He came highly touted as a defensive
defenseman, but he's clearly much more than that.
Great skater and hard shot (though wish he were
more accurate). In fact, he pretty much allowed
the Kings to trade away that huge defensive
liability Paul Coffey. Kelly Hrudey is only the
biggest disappointment if you thought he was any
good to begin with. But, at best, he's only a
mediocre goaltender. A better choice would be
Tomas Sandstrom, though not through any fault of
his own, but because some thugs in Toronto decided
```

NB for Text classification

- P(**X**|Y) is huge!!!
  - □ Article at least 1000 words, **X**={X_1,…,X_{1000}}
  - □ X_i represents i^{th} word in document, i.e., the domain of X_i is entire vocabulary, e.g., Webster Dictionary (or more), 10,000 words, etc.

- NB assumption helps a lot!!!
  - □ P(X_i=x_i|Y=y) is just the probability of observing word x_i in a document on topic y

$$h_{NB}(\mathbf{x}) \;=\; \arg\max_y P(y) \prod_{i=1}^{LengthDoc} P(x_i|y)$$

Bag of words model

- Typical additional assumption – **Position in document doesn't matter**: P(X_i=x_i|Y=y) = P(X_k=x_i|Y=y)
  - □ "Bag of words" model – order of words on the page ignored
  - □ Sounds really silly, but often works very well!

$$P(y) \prod_{i=1}^{LengthDoc} P(x_i|y)$$

When the lecture is over, remember to wake up the person sitting next to you in the lecture room.

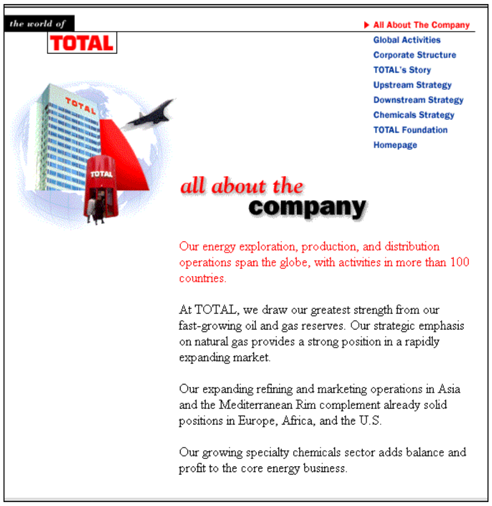
Bag of words model

- Typical additional assumption – **Position in document doesn't matter**: $P(X_i=x_i|Y=y) = P(X_k=x_i|Y=y)$
  - □ "Bag of words" model – order of words on the page ignored
  - □ Sounds really silly, but often works very well!

$$P(y) \prod_{i=1}^{LengthDoc} P(x_i|y)$$

in is lecture lecture next over person remember room sitting the the the to to up wake when you

Bag of Words Approach



the world of

TOTAL

▶ All About The Company

Global Activities

Corporate Structure

TOTAL's Story

Upstream Strategy

Downstream Strategy

Chemicals Strategy

TOTAL Foundation

Homepage

all about the **company**

Our energy exploration, production, and distribution operations span the globe, with activities in more than 100 countries.

At TOTAL, we draw our greatest strength from our fast-growing oil and gas reserves. Our strategic emphasis on natural gas provides a strong position in a rapidly expanding market.

Our expanding refining and marketing operations in Asia and the Mediterranean Rim complement already solid positions in Europe, Africa, and the U.S.

Our growing specialty chemicals sector adds balance and profit to the core energy business.

→

aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
…	
Zaire	0

NB with Bag of Words for text classification

- Learning phase:
  - □ Prior P(Y)
    - Count how many documents you have from each topic (+ prior)
  - □ $P(X_i|Y)$
    - For each topic, count how many times you saw word in documents of this topic (+ prior)
- Test phase:
  - □ For each document
    - Use naïve Bayes decision rule

$$h_{NB}(\mathbf{x}) = \arg\max_y P(y) \prod_{i=1}^{LengthDoc} P(x_i|y)$$

Twenty News Groups results

Given 1000 training documents from each group
Learn to classify new documents into
which newsgroup it came from

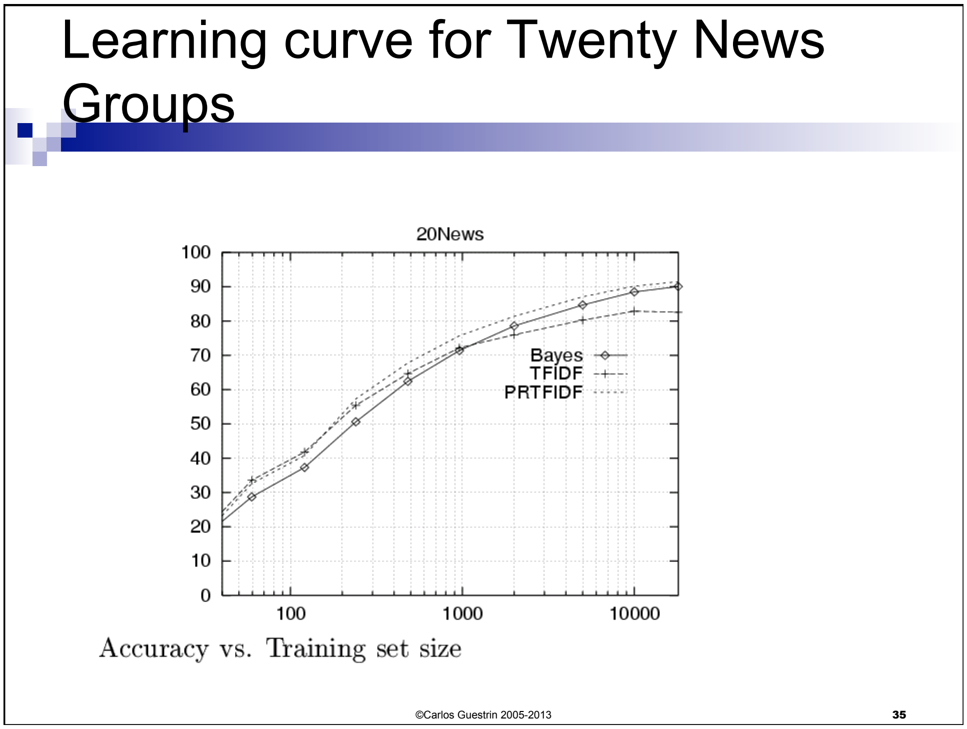
comp.graphics
comp.os.ms-windows.misc
comp.sys.ibm.pc.hardware
comp.sys.mac.hardware
comp.windows.x

misc.forsale
rec.autos
rec.motorcycles
rec.sport.baseball
rec.sport.hockey

alt.atheism
soc.religion.christian
talk.religion.misc
talk.politics.mideast
talk.politics.misc
talk.politics.guns

sci.space
sci.crypt
sci.electronics
sci.med

Naive Bayes: 89% classification accuracy



Learning curve for Twenty News Groups

Accuracy vs. Training set size