Artificial Intelligence

CS 165A

Jan 29, 2019

Instructor: Prof. Yu-Xiang Wang

Today

→ Finish NaiveBayes

→ Unsupervised Learning

Announcements

- Homework 1 due today. Submit to TA Lei Xu now.
- The TAs have sent out an announcement to clarify things regarding MP1 on Piazza.
- You are allowed to collaborate on homework and machine problems, but you need to declare who you got help from about which question.
- Go to office hours! We are there to help you!

Notetaker wanted for DSP student

- We need someone to volunteer taking notes
- This is a paid job.
- Ask me more after the class.

Some questions on Piazza

- Textbook reading:
  - The AIMA textbook is our reference.
  - The lectures are not entirely based on AIMA and will sometimes go deeper / broader / more relevant to 2019.
  - Reading the textbook helps you. Use Wikipedia and Google too.

- Homework questions
  - They can be solved by the "principles" covered in the lecture.
  - Do not post your solution!
  - TA can give hints but not answers.

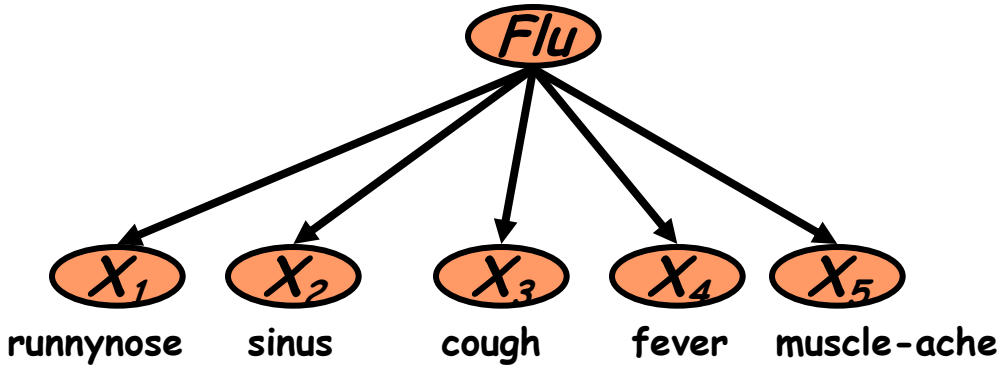
Today

- Finish off Naïve Bayes classifier
- Feature extraction
- Unsupervised Learning
  - k-means clustering and Gaussian mixture models
  - PCA and probabilistic PCA
  - Subspace clustering and Mixture of Probabilistic PCA
  - Latent Dirichlet Allocation for Topic Models

Notation (unless otherwise specified)

- d for dimension
- n for number of data points
- k for the number of classes / number of latent variables

The Naïve Bayes Classifier, revisited



- **Conditional Independence Assumption:** features are independent of each other given the class:

$$P(X_1, \cdots X_5 \mid C) = P(X_1 \mid C) \bullet P(X_2 \mid C) \bullet \cdots \bullet P(X_5 \mid C)$$

- For binary: # of parameters reduce from

Naïve Bayes Classifier for Text Classification

- Problems
  - different documents have different length.
  - Words come in a sequence.

- Let dictionary size be d and (max length =: H)
  - # of parameters = O(dkH).

- This is still too large.
  - Let's ignore the ordering information of the words.
  - # of parameters = O(dk).

- "Stan loves Eminem." ⇔ "Eminem loves Stan."

Bag of Words feature

Art display in CMU MLD

Excitement on Twitter:

"Profound artistic piece representing the struggle for representational power... "

"Dropped stop words is a nice touch QT @stanfordnlp CMU has a marvelous art installation of a bag of words #nlproc"

An example of extracting bag-of-word feature

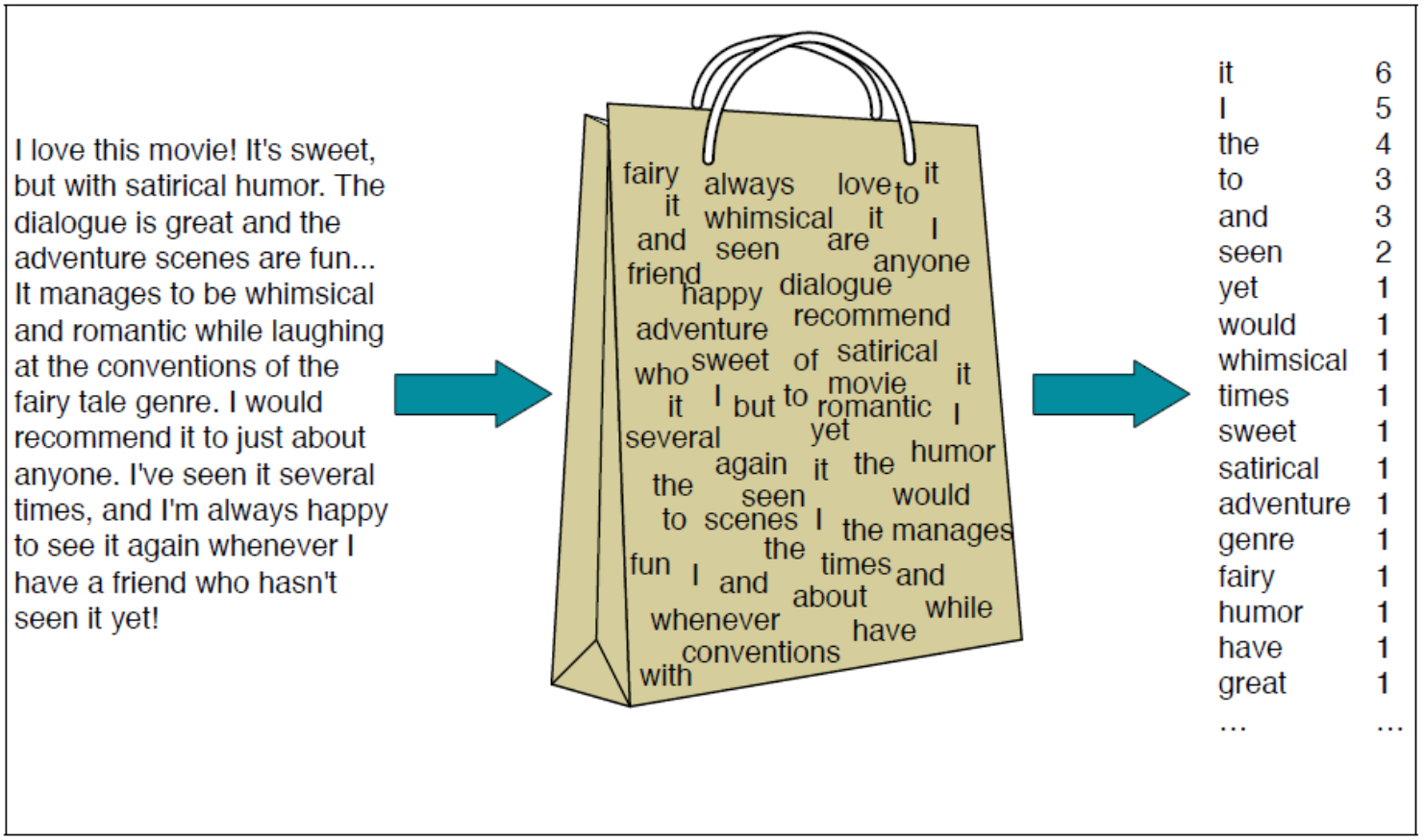


Figure 7.1 Intuition of the multinomial naive Bayes classifier applied to a movie review. The position of the words is ignored (the *bag of words* assumption) and we make use of the frequency of each word.

Figure extracted from Section 7.1 of Jurafsky and Martin, 3rd Edition

Multinomial Naïve Bayes

- Condition on the label, each document is drawn from a Multinomial distribution
  - $P(c) \sim Categorical\ (\tau)$
  - $P(x_1,x_2,\ldots,x_{n(i)}|c) = P(\text{BoW}(x_1,x_2,\ldots,x_{n(i)})|c) \sim \text{Multinomial}(\theta_c, n(i))$
  - n(i) denotes the length of ith document.

- Learning task:
  - Learn model parameters τ *and* θ_c for all c = 1,2,…,k from data
  - But how?

- By solving an optimization problem:
  - Maximum likelihood
  - Maximum A Posteriori (with a Dirichlet prior / Laplace smoothing)
  - (You will derive these in HW2!)

Violation of NB Assumptions

- Conditional independence
- “Positional independence”

Naïve Bayes Posterior Probabilities

- Classification results of naïve Bayes (the class with maximum a posteriori probability) are usually fairly accurate.
- However, due to the inadequacy of the conditional independence assumption, the actual posterior-probability numerical estimates are not accurate.
  - Output probabilities are generally very close to 0 or 1.

When does Naive Bayes work?

Sometimes NB performs well even if the Conditional Independence assumptions are badly violated.

Classification is about predicting the correct class label and NOT about accurately estimating probabilities.

Assume two classes c_1 and c_2. A new case A arrives.

NB will classify A to c_1 if:

$P(A, c_1) > P(A, c_2)$

	$P(A,c_1)$	$P(A,c_2)$	Class of A
Actual Probability	0.1	0.01	c_1
Estimated Probability by NB	0.08	0.07	c_1

Besides the big error in estimating the probabilities the classification is still correct.

Correct estimation ⇒ accurate prediction

but NOT

~~accurate prediction ⇒ Correct estimation~~

Naive Bayes is Not So Naive

- Naïve Bayes: First and Second place in KDD-CUP 97 competition, among 16 (then) state of the art algorithms

  Goal: Financial services industry direct mail response prediction model: Predict if the recipient of mail will actually respond to the advertisement – 750,000 records.

- Robust to Irrelevant Features

  Irrelevant Features cancel each other without affecting results

  Instead Decision Trees can heavily suffer from this.

- Very good in Domains with many equally important features

  Decision Trees suffer from *fragmentation* in such cases – especially if little data

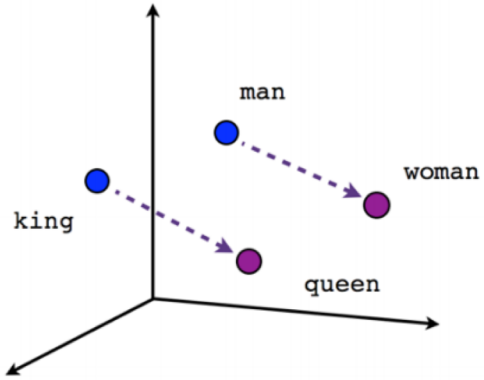
- A good dependable baseline for text classification (but not the best)!
- Optimal if the Independence Assumptions hold: If assumed independence is correct, then it is the Bayes Optimal Classifier for problem
- Very Fast: Learning with one pass over the data; testing linear in the number of attributes, and document collection size
- Low Storage requirements
- Warning: There are other much advanced classifiers (Machine Learning Course)

Other popular features for text: TF-IDF

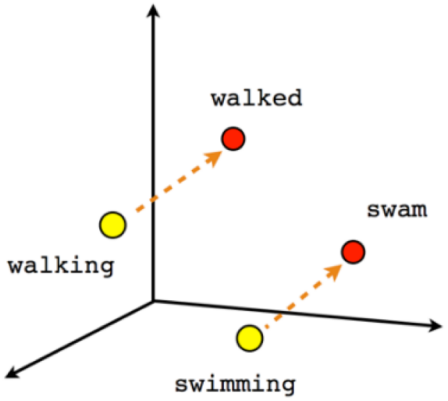
- Term Frequency (TF)
  - TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).
- Inverse Document Frequency (IDF)
  - IDF(t) = log_e(Total number of documents / Number of documents with term t in it).
- TF-IDF = TF(t) * IDF
- Really popular in Information Retrieval!
  - Especially in the 90s.
  - Deployed commercially everywhere.
  - Still used as a very strong baseline.

Word2Vec: Learned Representation

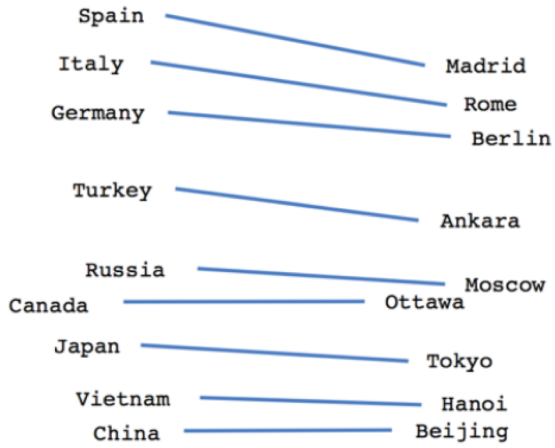
- Feature extractor φ: "Bayes" → [0.2,-0.34,5.3]



Male-Female



Verb tense



Country-Capital

Often learned through unsupervised learning (topic of this lecture).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.

Gaussian Naïve Bayes Classifier

- $x_1, x_2, \ldots, x_n$ are continuous variables.
  - TF-IDF, Word2Vec are continuous variables!
- $P(x_i \mid c)$ is drawn from a Gaussian distribution.
- You will need to:
  - Deriving MLE, MAP for Gaussian naïve Bayes in HW2!
  - Implement them and try them out in MP1.

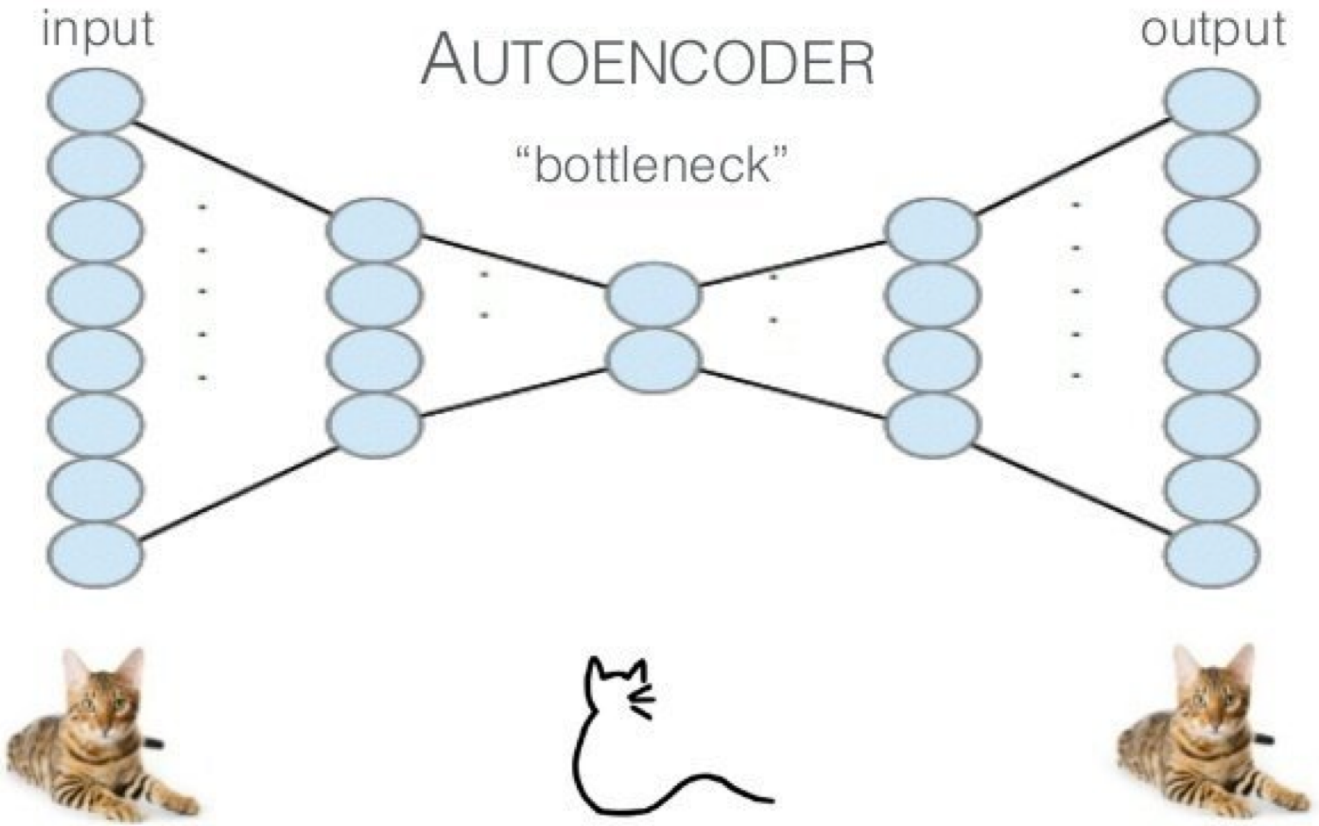
Unsupervised Learning

- Supervised learning is about finding f such that y=f(x)
  - Important to observe labels y

- Unsupervised learning is about
  - Learning to reduce dimension
  - Learning compact representation of x
  - Learning a feature representation

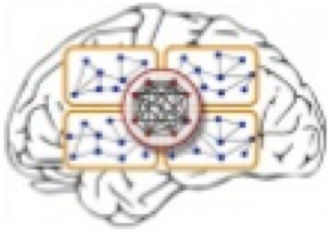
- How to learn without labels?
  - By learning x = f(x)
  - Restricting f to an “easy” class

A deep learning example: Autoencoder



Convolutional NN based Autoencoder

Artificial Neural Network



ML watches YouTube for three straight days!
(and learns to recognize cats?!)

http://www.npr.org/2012/06/26/155792609/a-massive-google-network-learns-to-identify

Building High-level Features Using Large Scale Unsupervised Learning

Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeffrey Dean, and Andrew Y. Ng

Recall: Generative vs. Discriminative Model

- A generative model starts with a probabilistic description of how the data are generated.
  - This gives rise to an objective function to optimize via MLE or MAP.

- A discriminative model ignores probabilities and directly specifies the objective function to optimize.

- Questions to think about:
  - Do I need to fully describe the world?
  - Do I need to have a probabilistic framework at all?
  - Optimal solution in an approximation of the world vs. Approximal solution to the actual problem

Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems* (pp. 841-848).

Often they do very similar things.

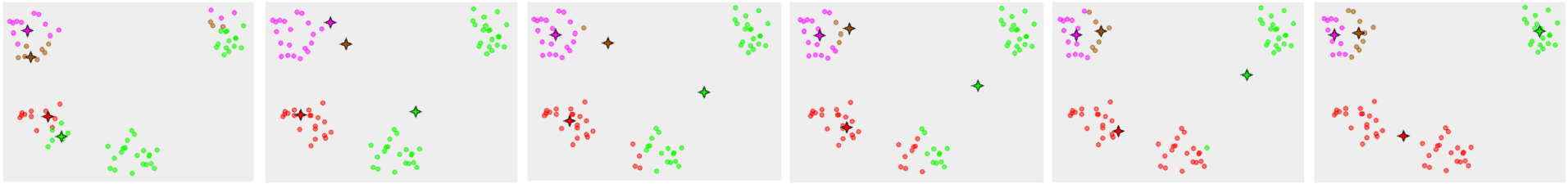
- Types of unsupervised learning
  - Clustering: k-means clustering --- Gaussian Mixture Models
  - Dimension reduction: PCA --- Probabilistic PCA
  - Clustering/Dimension reduction at the same time

    Subspace Clustering --- Mixture of Prob. PCA

K-Means clustering

- Objective function $\min_{c_1,\ldots,c_k} \sum_i \min_{j\in[k]} \|x_i - c_j\|^2$
- Algorithm: Lloyd's algorithm.
  - Randomly initialize the centers at data points.
  - Assign data points to closest center
  - Update centers to be the mean of the data points assigned to it

- A typical run of Lloyd's algorithm:

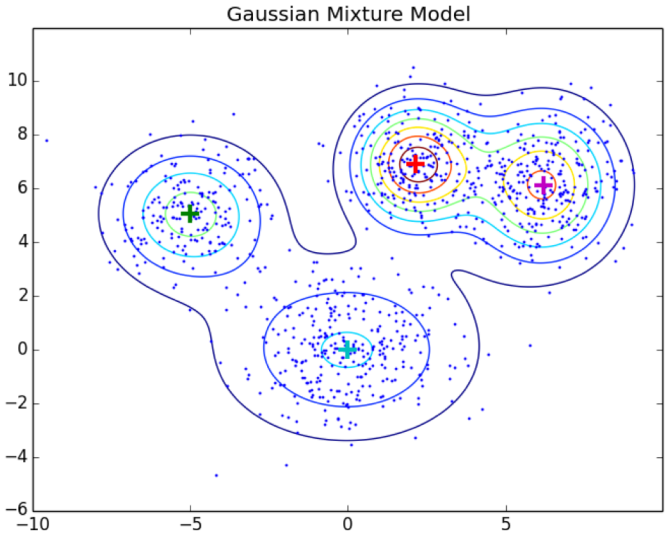


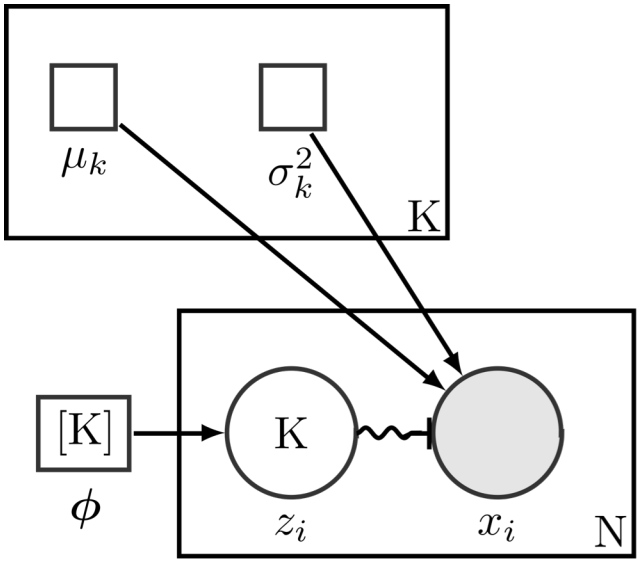
Gaussian Mixture Model

- Generative process:

$$
\begin{aligned}
\theta_{i=1\ldots K} &= \{\mu_{i=1\ldots K}, \sigma^2_{i=1\ldots K}\} \\
\mu_{i=1\ldots K} &= \text{mean of component } i \\
\sigma^2_{i=1\ldots K} &= \text{variance of component } i \\
z_{i=1\ldots N} &\sim \text{Categorical}(\phi) \\
x_{i=1\ldots N} &\sim \mathcal{N}(\mu_{z_i}, \sigma^2_{z_i})
\end{aligned}
$$

- Graphical model:



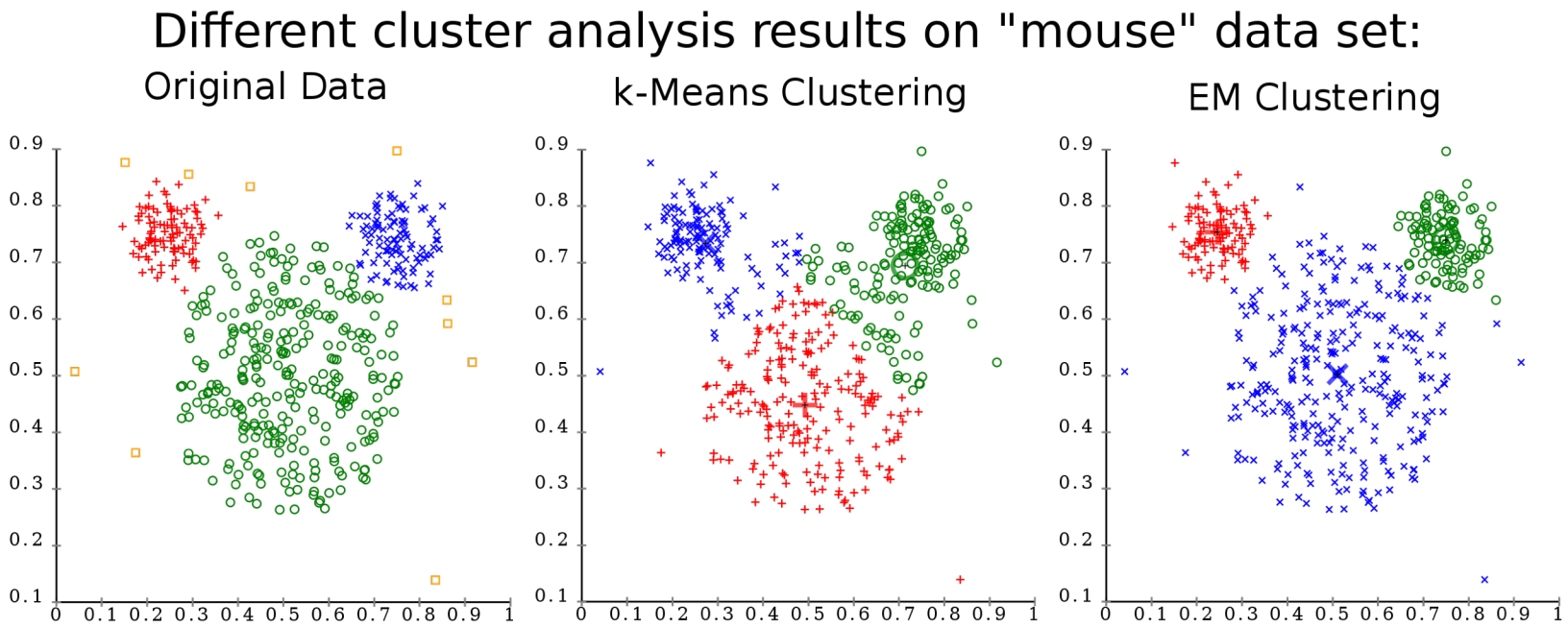


The EM algorithm of Gaussian mixture model

- E-step: Soft assignments of data points to each mixture components via posterior distribution.

- M-step: Update the parameters of each Gaussian by maximizing the likelihood.
  - When variance goes to 0. This converges to the Lloyd's algorithm!

The probabilistic framework helps the algorithm to be more robust



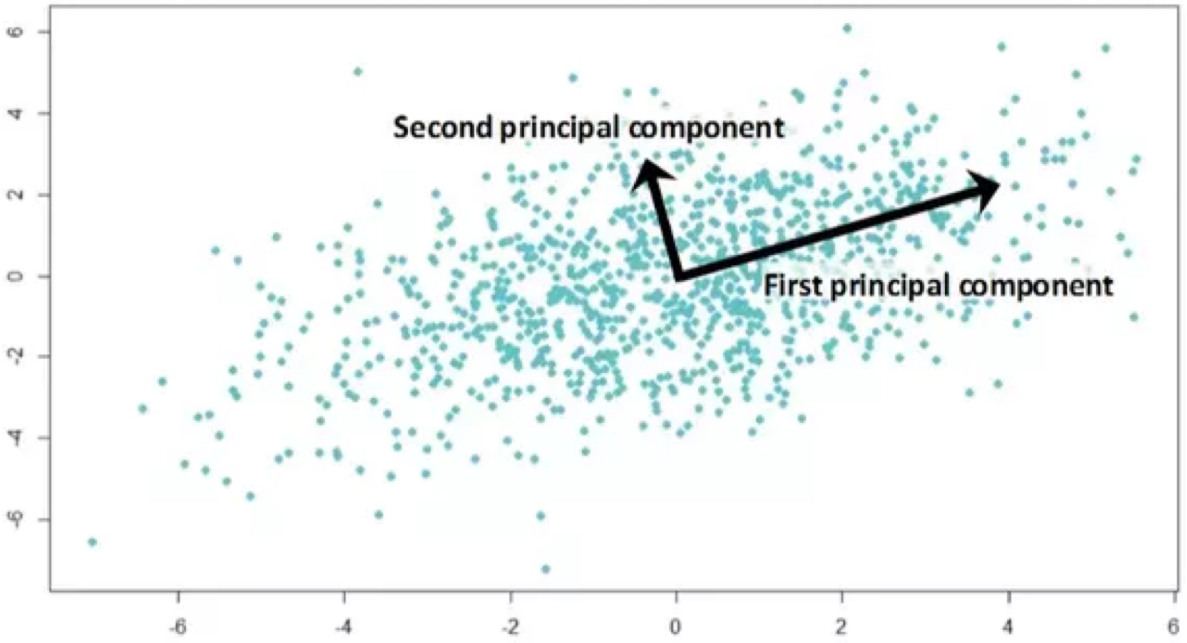
*When the assumptions are approximately true.

Principle Components Analysis

- Given data x in R^d, find best k-dimensional approximations that explains the variance of the data.

$$\min_{W \in \mathbb{R}^{d \times k}} \|WW^T - \mathrm{Cov}(X)\|_F^2$$

$$\mathrm{Cov}(X) = \frac{1}{n}(x_i - \bar{x})(x_i - \bar{x})^T$$



Probabilistic Principle Component Analysis

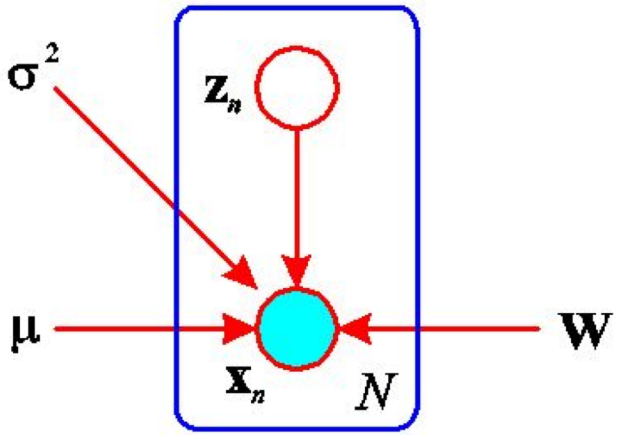
- Generative process:

$$z_i \sim N(0, I_k)$$
$$x_i \sim N(\mu + W z_i, \sigma^2 I_d)$$

Or equivalently:

$$x_i \sim N(\mu, WW^T + \sigma^2 I_d)$$

- BayesNet of PPCA: (from Chris Bishop)



PCA vs. MLE of Prob. PCA

- Sample covariance matrix:

$$\mathrm{Cov}(X) = \frac{1}{n}(x_i - \bar{x})(x_i - \bar{x})^T$$

- Singular Value decomposition of Cov(X) = U Λ U'

- PCA: Output U[:, 1:k] Λ[:, 1:k]$^{1/2}$

- PPCA's MLE: $W_{\mathrm{MLE}} = U[:, 1:k](\Lambda[:, 1:k] - \sigma^2 I_k)^{1/2}$

$$\sigma^2_{\mathrm{MLE}} = \frac{1}{d-k}\sum_{j=k+1}^{d} \lambda_j$$

Eigenface: a nice application of PCA!

- Matthew Turk (UCSB CS) and Pentland (MIT)