CS 335: Clustering

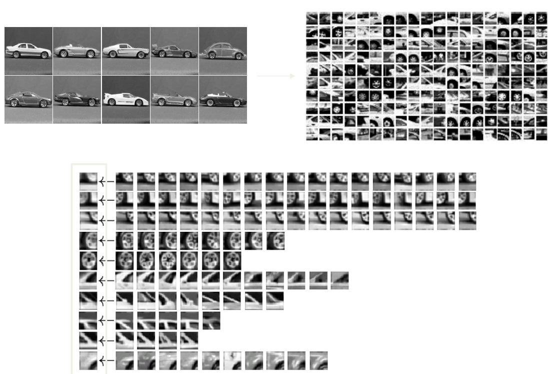
Dan Sheldon

November 25, 2014

Machine Learning Problems

	Supervised Learning	*Unsupervised Learning*
Discrete	classification or categorization	$\boxed{\text{clustering}}$
Continuous	regression	dimensionality reduction

Visual "Dictionaries"



Appearance dictionary

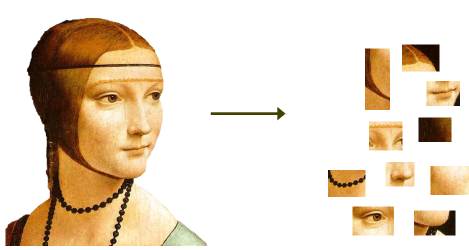
Source: B. Leibe

Visual "Dictionaries"

Appearance dictionary

Source: B. Leibe

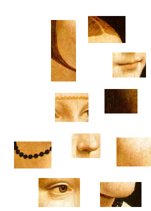
Histogram Features for Computer Vision



Step 1: Divide image into (many) patches

Svetlana Lazebnik

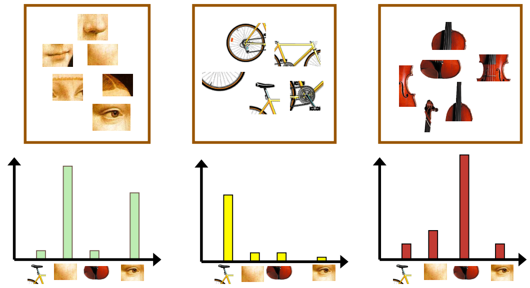
Histogram Features for Computer Vision



Step 2: Look up each patch in dictionary
(Find closest "visual word")

Svetlana Lazebnik

Histogram Features for Computer Vision

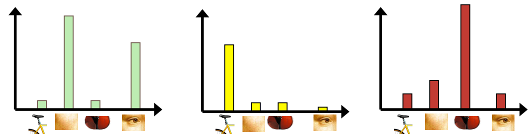


Step 3: Represent images by frequencies of “visual words” (histograms)

Svetlana Lazebnik

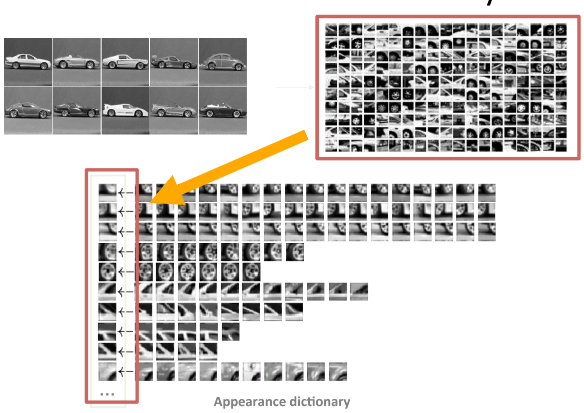
Histogram Features for Computer Vision

Step 4: Use histograms as feature vectors in supervised ML algorithm



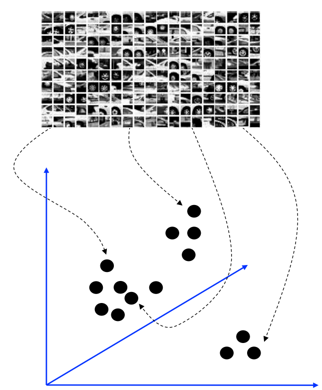
$\mathbf{x}^{(1)}$ $\mathbf{x}^{(2)}$ $\mathbf{x}^{(3)}$

How Do We Create The Dictionary?



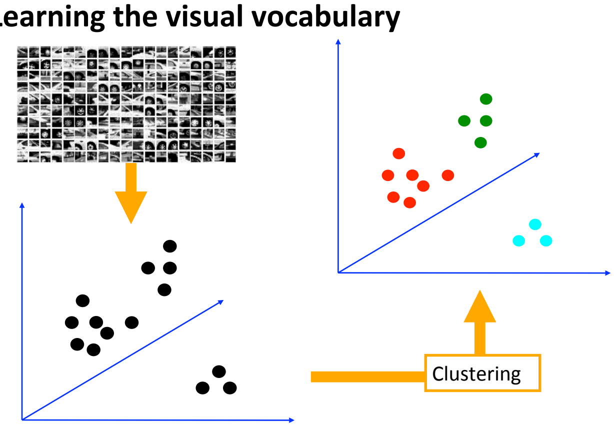
Source: B. Leibe

Learning the visual vocabulary



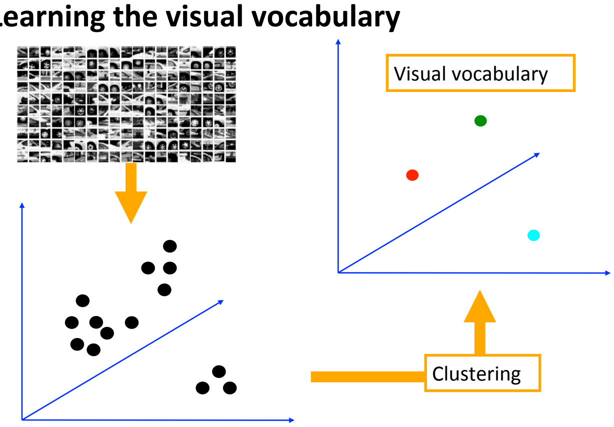
Slide credit: Josef Sivic

Learning the visual vocabulary



Slide credit: Josef Sivic

Learning the visual vocabulary



Slide credit: Josef Sivic

K-Means Problem

Given

- Feature vectors $\mathbf{x}^{(1)}, \dots \mathbf{x}^{(m)} \in \mathbb{R}^n$
- Desired number of clusters k

Find

- Cluster centers $\mu_1, \dots, \mu_k \in \mathbb{R}^n$
- Cluster labels $c^{(i)} \in \{1, 2, \dots, k\}$

Minimize

$$J(c, \mu) = \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

K-Means Algorithm

1. Initialize $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$ randomly
2. Repeat until convergence
   - For all points i, assign $\mathbf{x}^{(i)}$ to closest cluster center

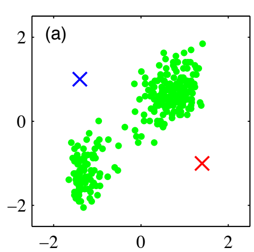
$$c^{(i)} \leftarrow \text{argmin}_j \, ||\mathbf{x}^{(i)} - \mu_j||^2$$

   - For all clusters j, set $\boldsymbol{\mu}_j$ = average of currently assigned points

$$\boldsymbol{\mu}_j \leftarrow \frac{\sum_{i=1}^{m} \mathbf{1}\{c^{(i)} = j\}\mathbf{x}^{(i)}}{\sum_{i=1}^{m} \mathbf{1}\{c^{(i)} = j\}}$$

K-Means Example

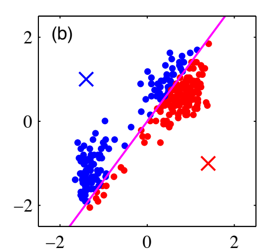
Initialize cluster centers arbitrarily:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

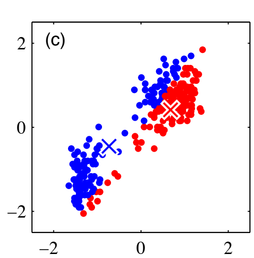
Assign points:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

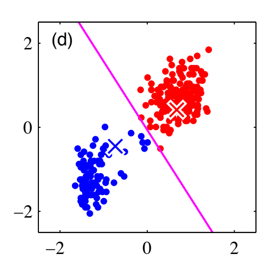
Update centers:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

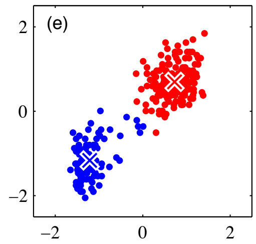
Assign points:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

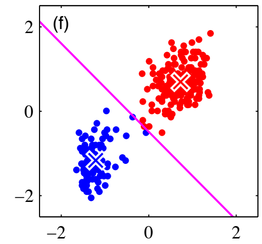
Update centers:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

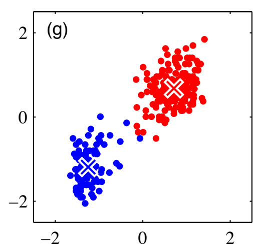
Assign points:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

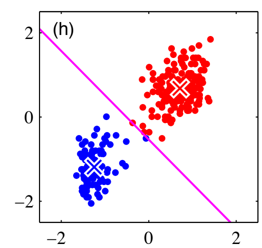
Update centers:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

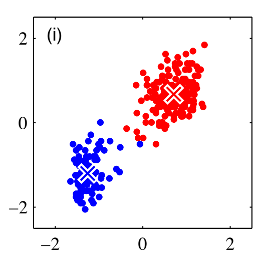
Assign points:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Example

Update centers:



[Bishop *Pattern Recognition and Machine Learning*]

K-Means Convergence

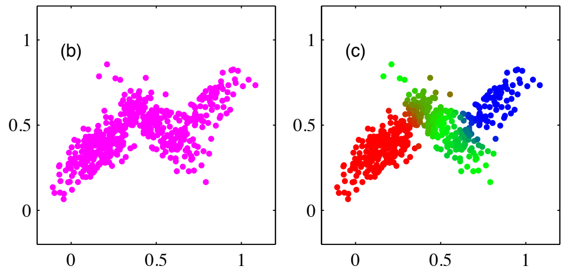
$$J(c,\mu) = \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

Not hard to show that

1. μ updates minimize J while holding c fixed
2. c updates minimize J while holding μ fixed
3. The algorithm converges

"Soft" clustering

Often desirable to fractionally assign points to clusters



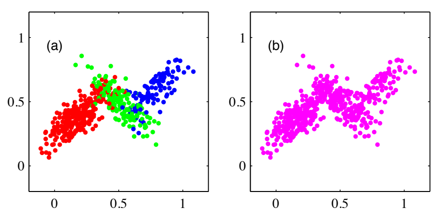
[Bishop *Pattern Recognition and Machine Learning*]

"Soft" clustering

There's something Bayesian happening here...

Generative Model: Mixture of Gaussians

- First choose cluster: $p(c^{(i)} = j) = \phi_j$
- Then generate $\mathbf{x}^{(i)}$ from conditional distribution $p(\mathbf{x}^{(i)} \mid c^{(i)})$
- Hide cluster assignments



$p(\mathbf{x}^{(i)} \mid c^{(i)} = k)$ follows a **Gaussian distribution** with parameters μ_k and Σ_k

Aside: Multivariate Gaussian Distribution

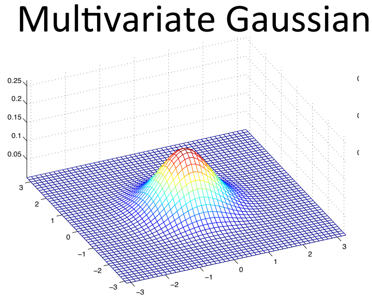
$$p(\mathbf{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

Describes random vector $\mathbf{x} \in \mathbb{R}^n$ with

- Mean vector $\boldsymbol{\mu} \in \mathbb{R}^n$
- Covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$
- $P(\mathbf{x} \in A) = \int_A p(\mathbf{x}; \boldsymbol{\mu}, \Sigma) d\mathbf{x}$

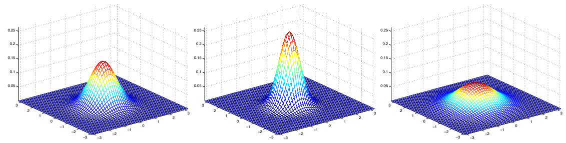
Examples

Multivariate Gaussian



$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right).$$

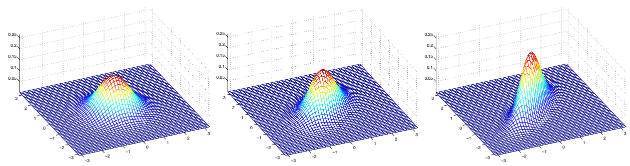
Examples: Symmetric



$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\quad \Sigma = 0.6I;$ $\quad \Sigma = 2I.$

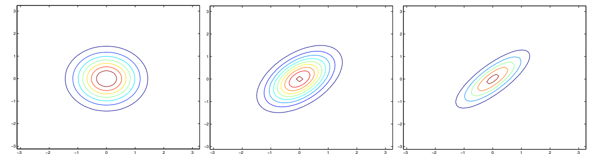
$\Sigma = I$

Examples: Non-Symmetric



$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \ \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}; \ .\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}.$$

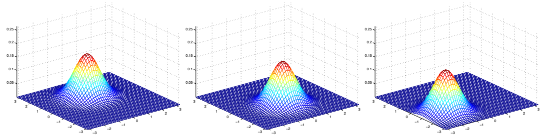
Contours



$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \ \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}; \ .\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}.$$

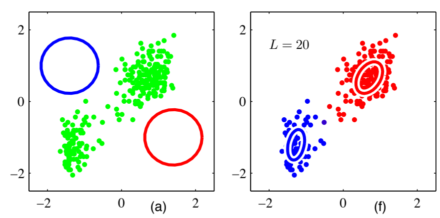
Mean

- Change mu: move mean of density around



$$\mu = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \ \mu = \begin{bmatrix} \text{-0.5} \\ 0 \end{bmatrix}; \ \mu = \begin{bmatrix} \text{-1} \\ \text{-1.5} \end{bmatrix}.$$

Mixture of Gaussians Problem



Given feature vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$, number of "clusters" k, find:

- Cluster priors $\phi_j = p(c = j)$
- Gaussian parameters $\boldsymbol{\mu}_j$ and Σ_j for each cluster
- Soft cluster assignments $p(c^{(i)} = j | \mathbf{x}^{(i)})$

Mixture of Gaussians Algorithm

Repeat until convergence

1. Compute posterior probability that $\mathbf{x}^{(i)}$ comes from cluster j

$$\begin{aligned} w_j^{(i)} &= p(y^{(i)} = j \,|\, \mathbf{x}^{(i)}) \\ &= \frac{\phi_j \cdot p(\mathbf{x}^{(i)} \,|\, y^{(i)} = j)}{\sum_{l=1}^{k} \phi_l \cdot p(\mathbf{x}^{(i)} \,|\, y^{(i)} = l)} \quad \text{(Bayes rule)} \end{aligned}$$

2. Update parameters ϕ_j, μ_j, Σ_j using $w_j^{(i)}$ values as weights

Update Parameters

ϕ_j = average weight assigned to class j
$\boldsymbol{\mu}_j$ = weighted mean for class j
Σ_j = weighted covariance for class j

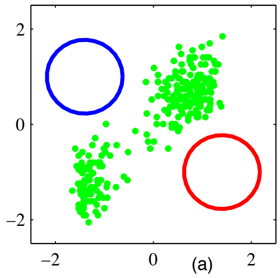
$$\phi_j = \frac{1}{m} \sum_{i=1}^{m} w_j^{(i)}$$

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{m} w_j^{(i)} \mathbf{x}^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{m} w_j^{(i)} \sum_{i=1}^{m} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)(\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)^T}{\sum_{i=1}^{m} w_j^{(i)}}$$

Mixture of Gaussians

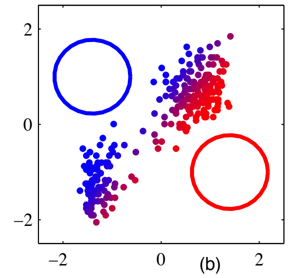
Initialize cluster parameters:



[Bishop *Pattern Recognition and Machine Learning*]

Mixture of Gaussians

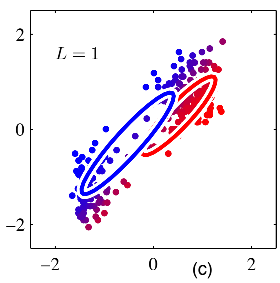
Update soft assignments:



[Bishop *Pattern Recognition and Machine Learning*]

Mixture of Gaussians

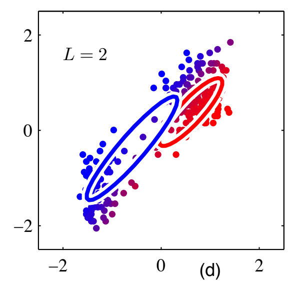
Update cluster parameters:



[Bishop *Pattern Recognition and Machine Learning*]

Mixture of Gaussians

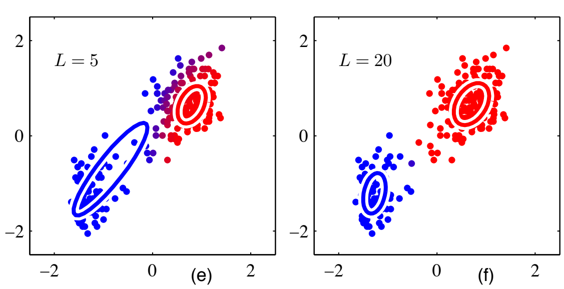
Next iteration:



[Bishop *Pattern Recognition and Machine Learning*]

Mixture of Gaussians

And so on:



[Bishop *Pattern Recognition and Machine Learning*]

Mixture of Gaussians Convergence

- This algorithm converges
- Can be formally justified as an instance of the Expectation Maximization (EM) algorithm
- For your next ML class!

The End

That's it for new material!

Quiz Topics

- SVMs
  - margin, functional margin, support vectors
  - Kernel trick
  - Gaussian kernel SVMs
  - Role of C and γ
- Neural nets
  - Idea of back-prop (chain rule)
  - Application of back-prop at level of movie recommendations and logistic regression

Quiz Topics

- Movie recommendations
- PCA
  - $X \approx ZW^T$
  - Model interpretation and use
- Bayes
  - Bayes rule
  - Application of Bayes rule (e.g. biased coin calculation)
  - Naive Bayes
  - Conceptual understanding of discriminiative vs. generative
- K-means

Extra Slides

...

Gaussian Distribution in 1D

$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(-\frac{(x-\mu)^2}{\sigma^2}\Big)$$

Describes continuous random variable X with

- Mean μ
- Standard deviation σ
- $P(X \in [a, b]) = \int_a^b p(x; \mu, \sigma^2) dx$

Illustrate μ, σ, area under curve

1D Gaussian Estimation

Given scalars $x^{(1)}, \dots x^{(m)}$

Find best-fitting 1D Gaussian density

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m} (x^{(i)} - \mu)^2$$

Multivariate Gaussian Estimation

Given feature vectors $\mathbf{x}^{(1)}, \ldots \mathbf{x}^{(m)} \in \mathbb{R}^n$

Find best-fitting Gaussian density

$$\boldsymbol{\mu} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}^{(i)}$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m} (\mathbf{x}^{(i)} - \boldsymbol{\mu})(\mathbf{x}^{(i)} - \boldsymbol{\mu})^T$$

Gaussian Discriminant Analysis

Generative model where $p(\mathbf{x} \,|\, y)$ is Gaussian

picture

Gaussian Discriminant Analysis Estimation

Given labeled training examples $(\mathbf{x}^{(1)}, y^{(1)}), \ldots (\mathbf{x}^{(m)}, y^{(m)}) \in \mathbb{R}^n$

Find

For each class $j \in \{1, \ldots, k\}$

- Class prior $\phi_j = p(y = j)$
- Class-conditional Gaussian density $p(\mathbf{x} \,|\, y = j)$
  (find $\boldsymbol{\mu}_j$ and Σ_j)

Gaussian Discriminant Analysis Estimation

ϕ_j = fraction of examples with label j
$\boldsymbol{\mu}_j$ = mean of examples with label j
Σ_j = covariance of examples with label j

$$\phi_j = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\{y^{(i)} = j\}$$

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{m} \mathbf{1}\{y^{(i)} = j\}\mathbf{x}^{(i)}}{\sum_{i=1}^{m} \mathbf{1}\{y^{(i)} = j\}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{m} \mathbf{1}\{y^{(i)} = j\} \sum_{i=1}^{m} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)(\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)^T}{\sum_{i=1}^{m} \mathbf{1}\{y^{(i)} = j\}}$$