CS598:Visual information Retrieval

Lecture IV: Image Representation:
Feature Coding and Pooling

Recap of Lecture III

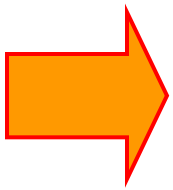
- Blob detection
  - Brief of Gaussian filter
  - Scale selection
  - Lapacian of Gaussian (LoG) detector
  - Difference of Gaussian (DoG) detector
  - Affine co-variant region
- Learning local image descriptors (optional reading)

Outline

- Histogram of local features
- Bag of words model
- Soft quantization and sparse coding
- Supervector with Gaussian mixture model

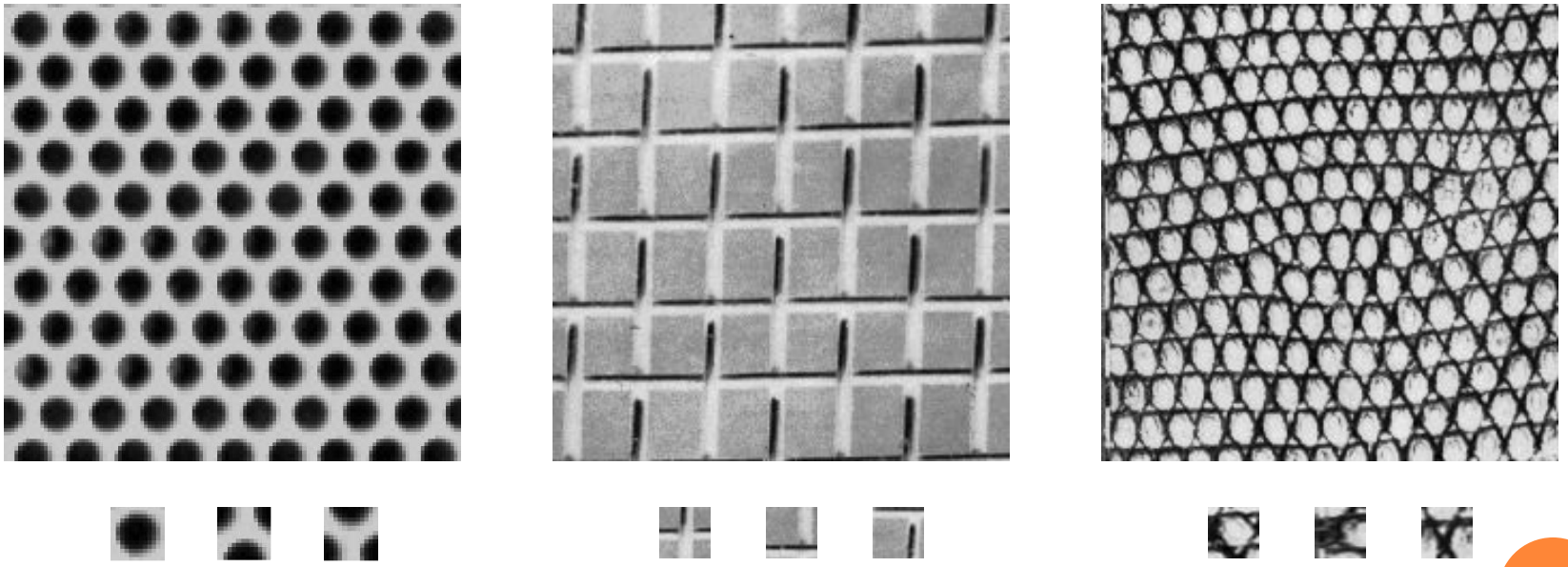
Lecture IV: Part I

Bag-of-features models

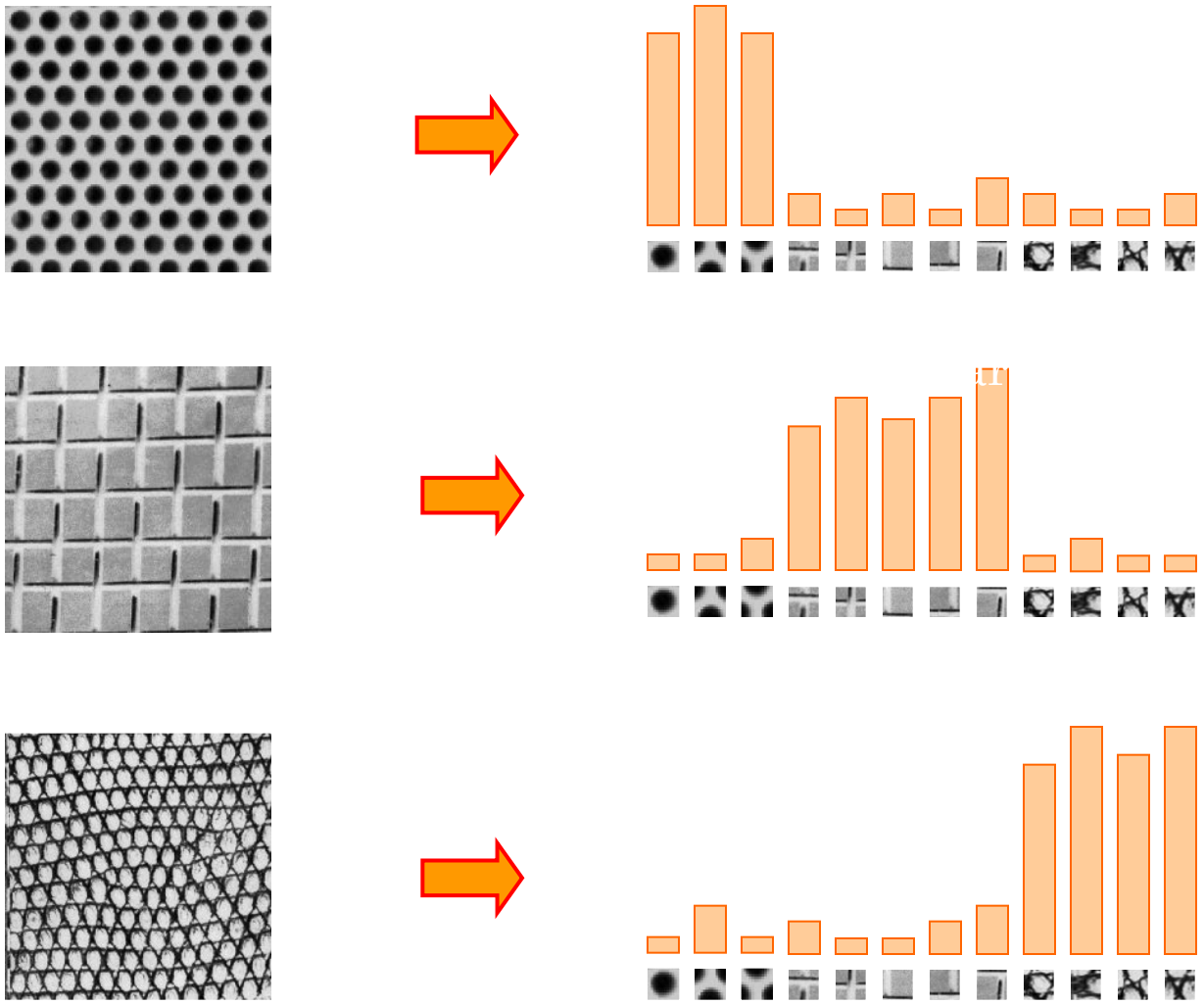


Origin 1: Texture recognition

- Texture is characterized by the repetition of basic elements or *textons*
- For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters



Origin 1: Texture recognition



Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

Origin 2: Bag-of-words models

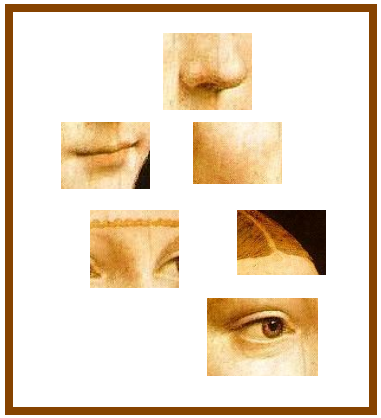
- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

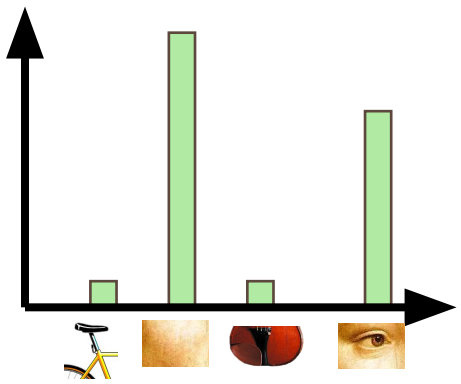
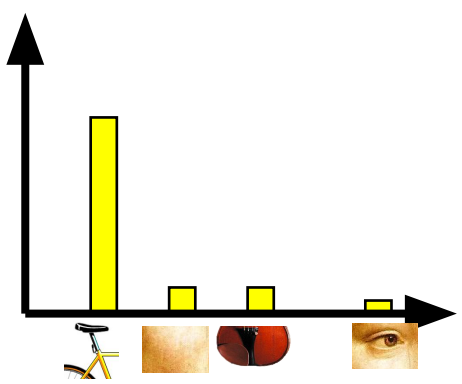
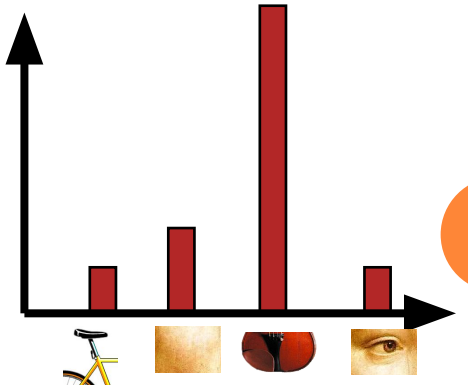
Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

Bag-of-features steps

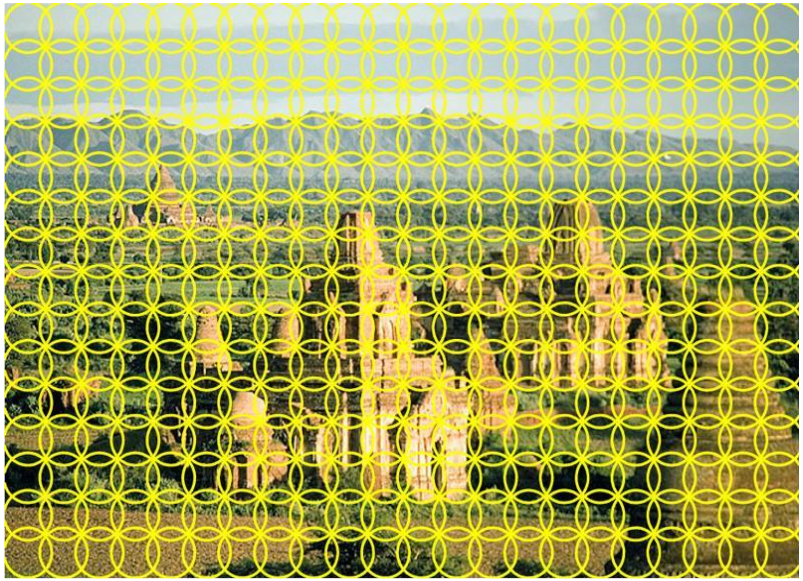
1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
4. Represent images by frequencies of “visual words”



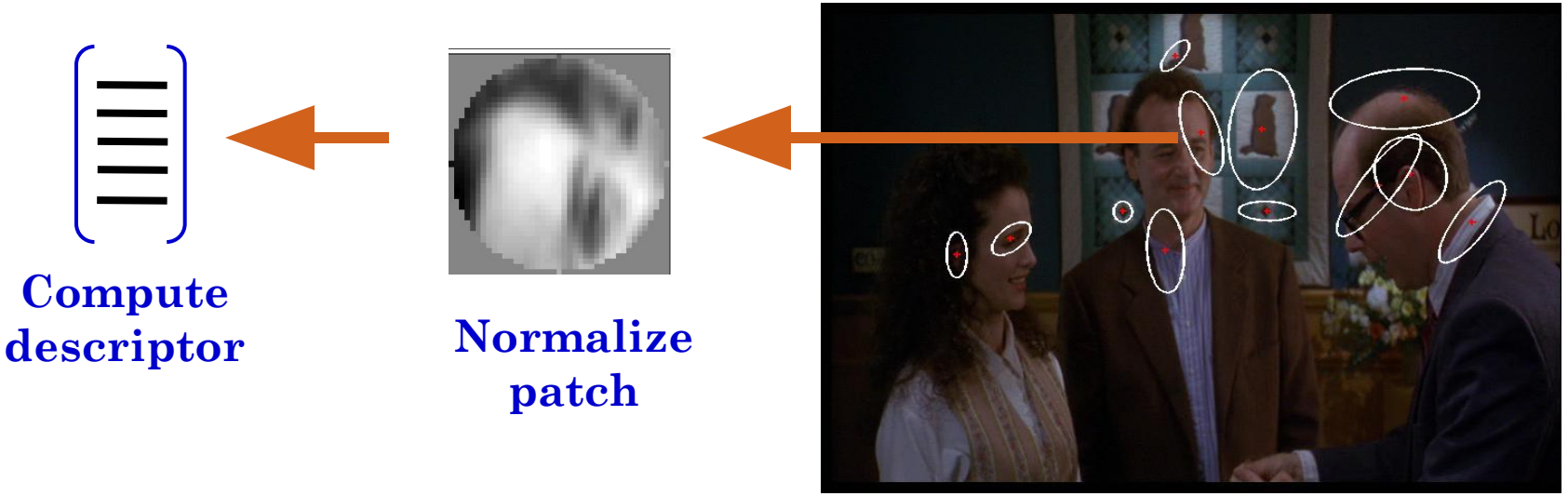





1. Feature extraction

- Regular grid or interest regions

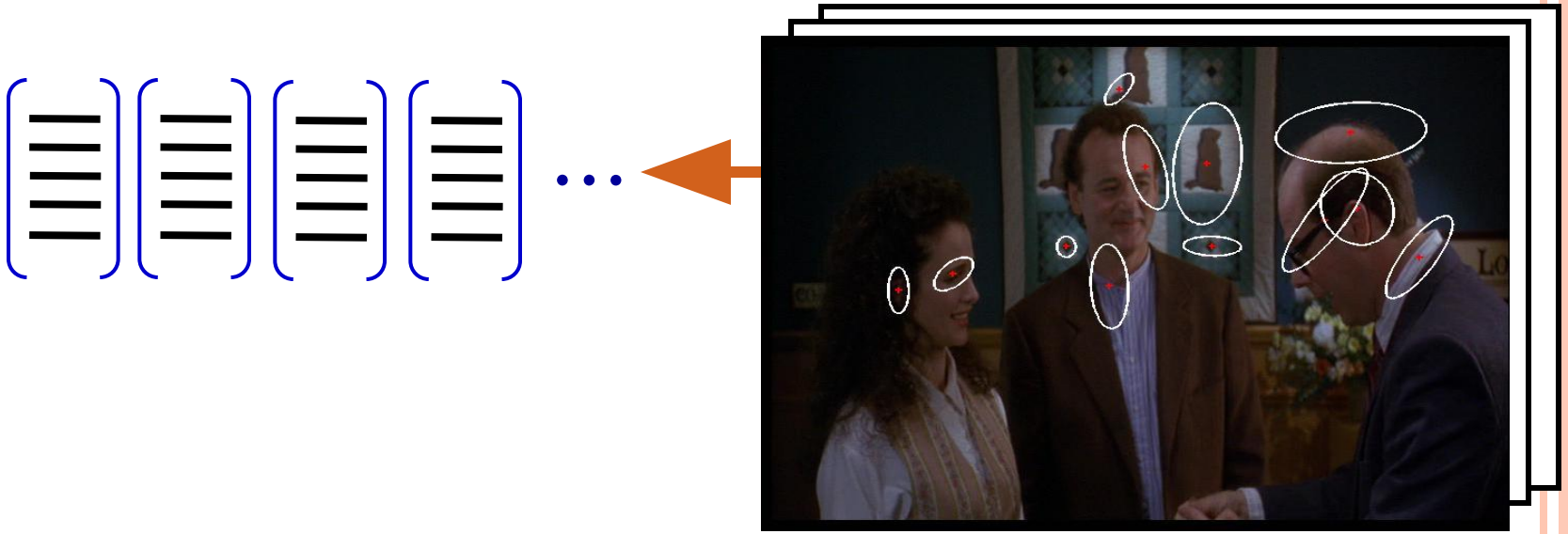


1. Feature extraction



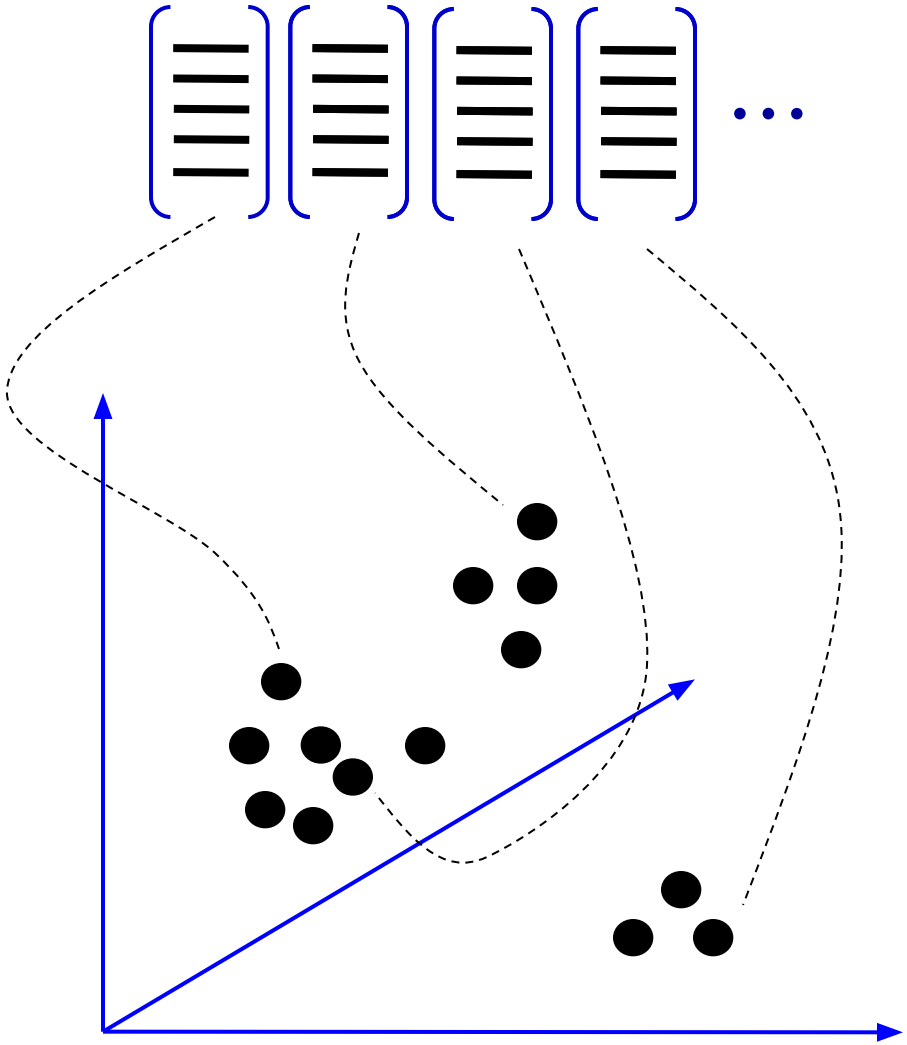
Slide credit: Josef

1. Feature extraction



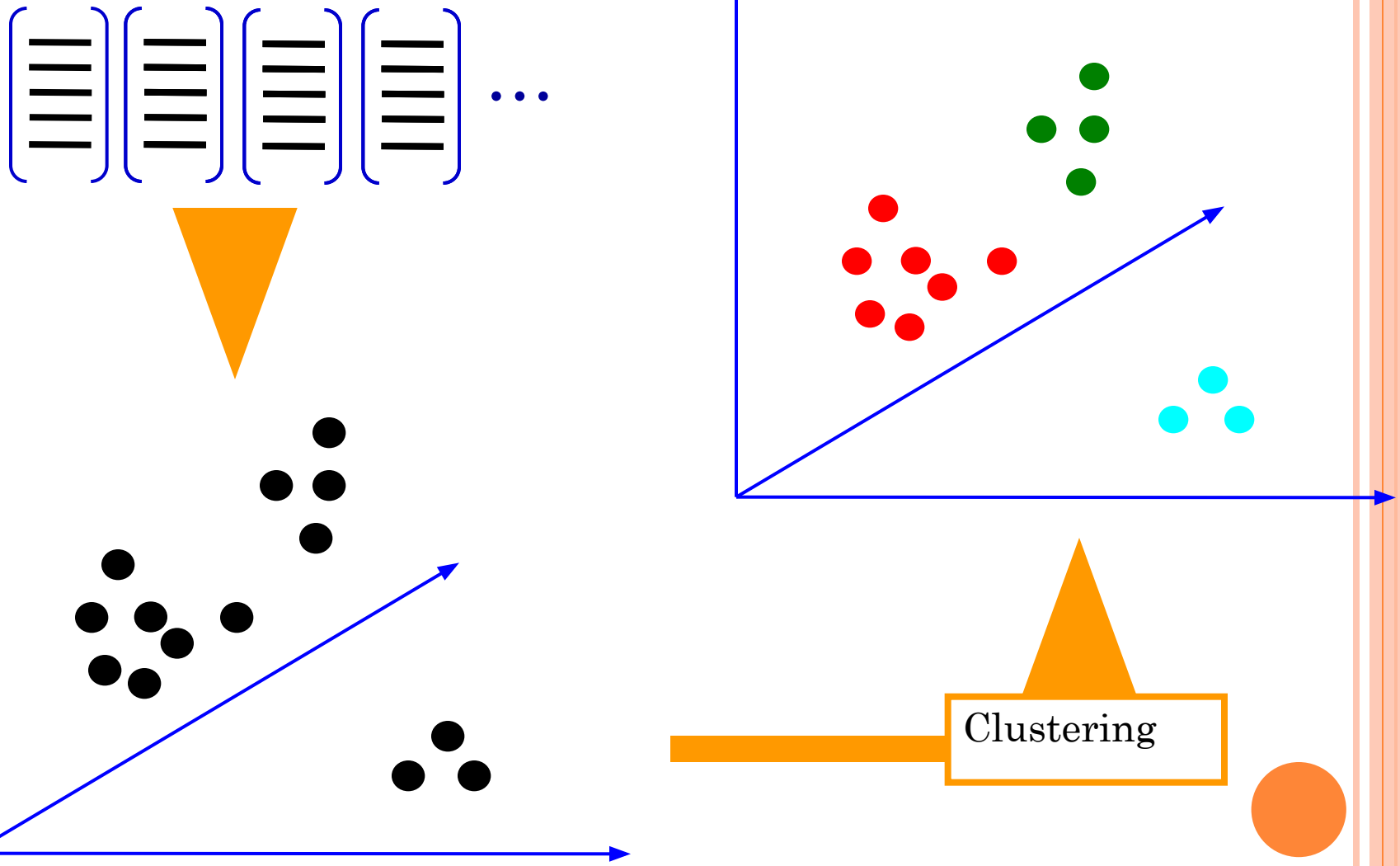
Slide credit: Josef

2. Learning the visual vocabulary



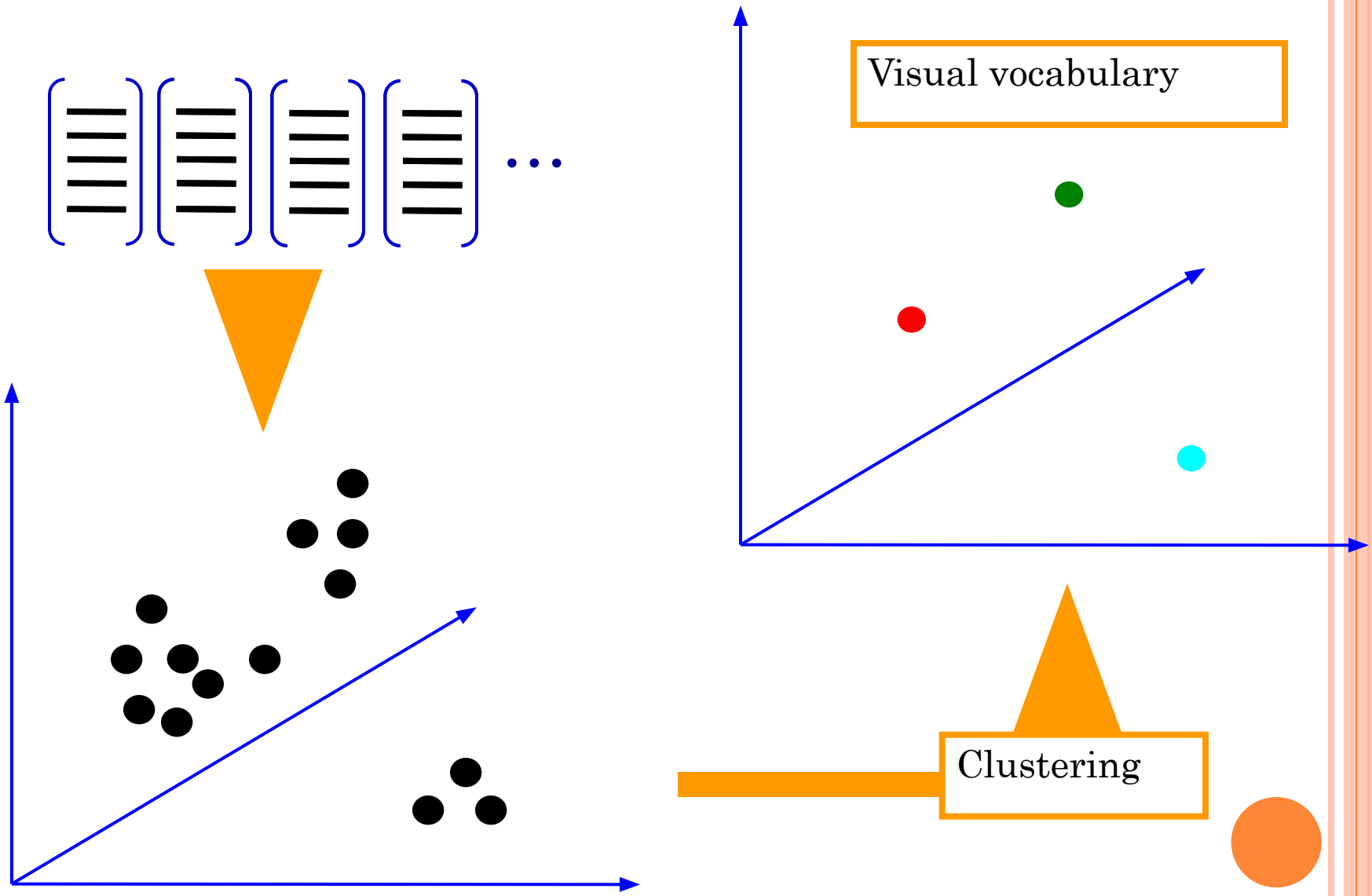
Slide credit: Josef

2. Learning the visual vocabulary



Slide credit: Josef

2. Learning the visual vocabulary



Slide credit: Josef

K-means clustering

- Want to minimize sum of squared Euclidean distances between points x_i and their nearest cluster centers m_k

$$D(X,M) = \sum_{\text{cluster } k} \sum_{\substack{\text{point } i \text{ in} \\ \text{cluster } k}} (x_i - m_k)^2$$

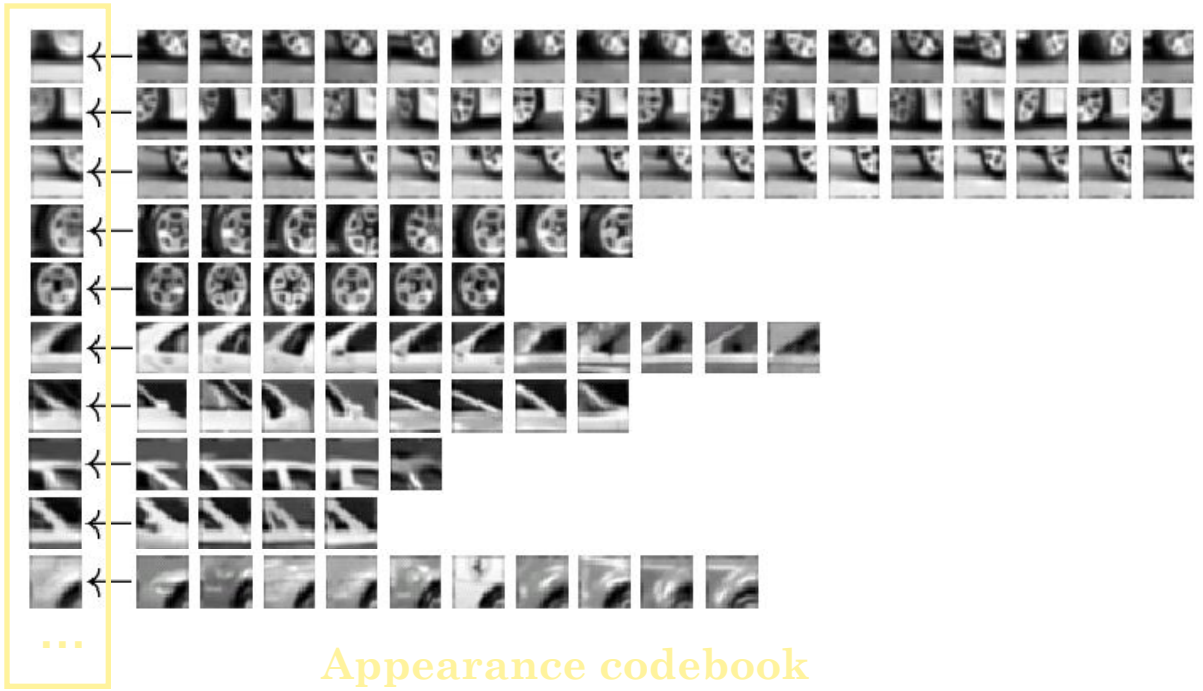
- Algorithm:
- Randomly initialize K cluster centers
- Iterate until convergence:
  - Assign each data point to the nearest center
  - Recompute each cluster center as the mean of all points assigned to it

Clustering and vector quantization

- Clustering is a common method for learning a visual vocabulary or codebook
  - Unsupervised learning process
  - Each cluster center produced by k-means becomes a codevector
  - Codebook can be learned on separate training set
  - Provided the training set is sufficiently representative, the codebook will be "universal"

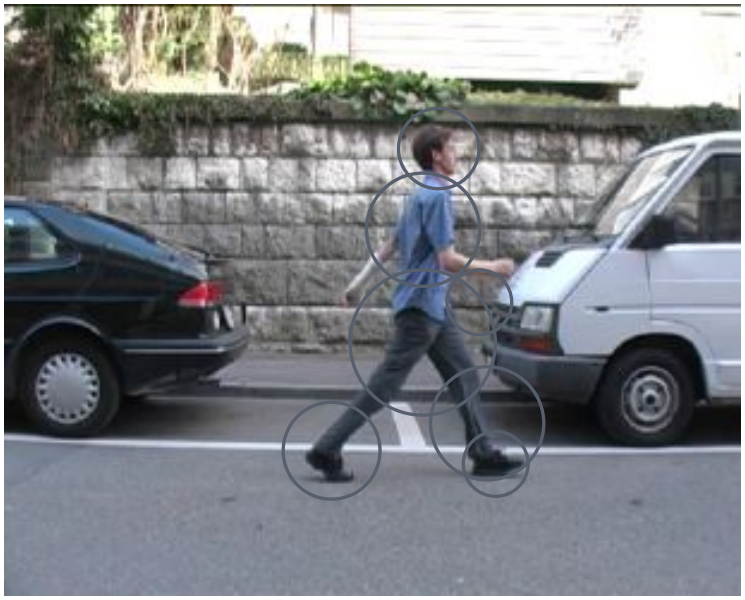
- The codebook is used for quantizing features
  - A *vector quantizer* takes a feature vector and maps it to the index of the nearest codevector in a codebook
  - Codebook = visual vocabulary
  - Codevector = visual word

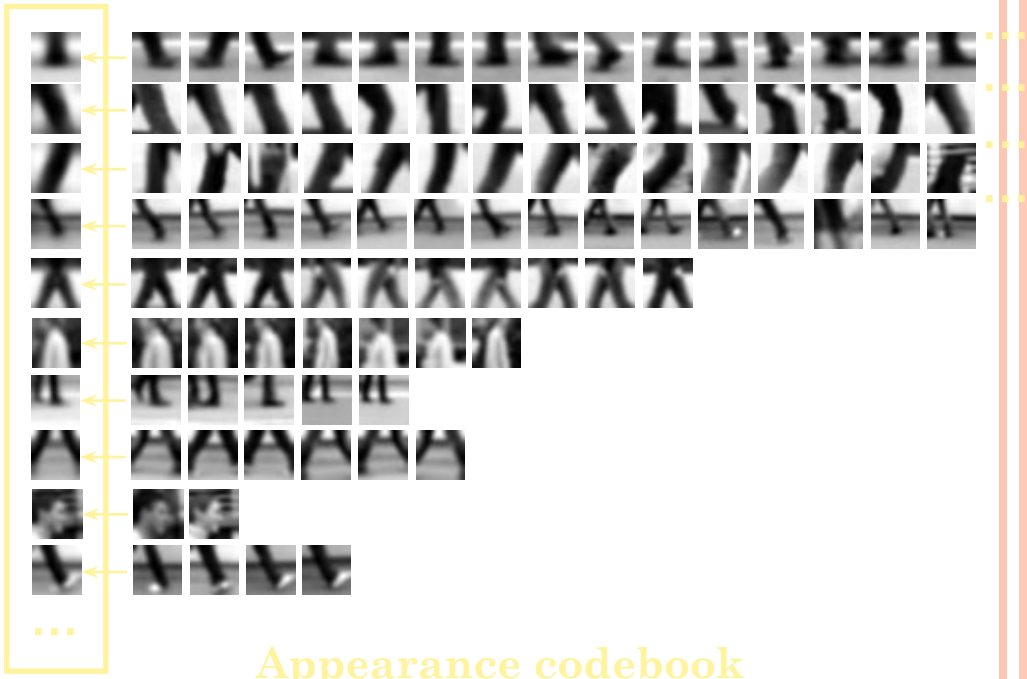
Example codebook



Source: B. Leibe

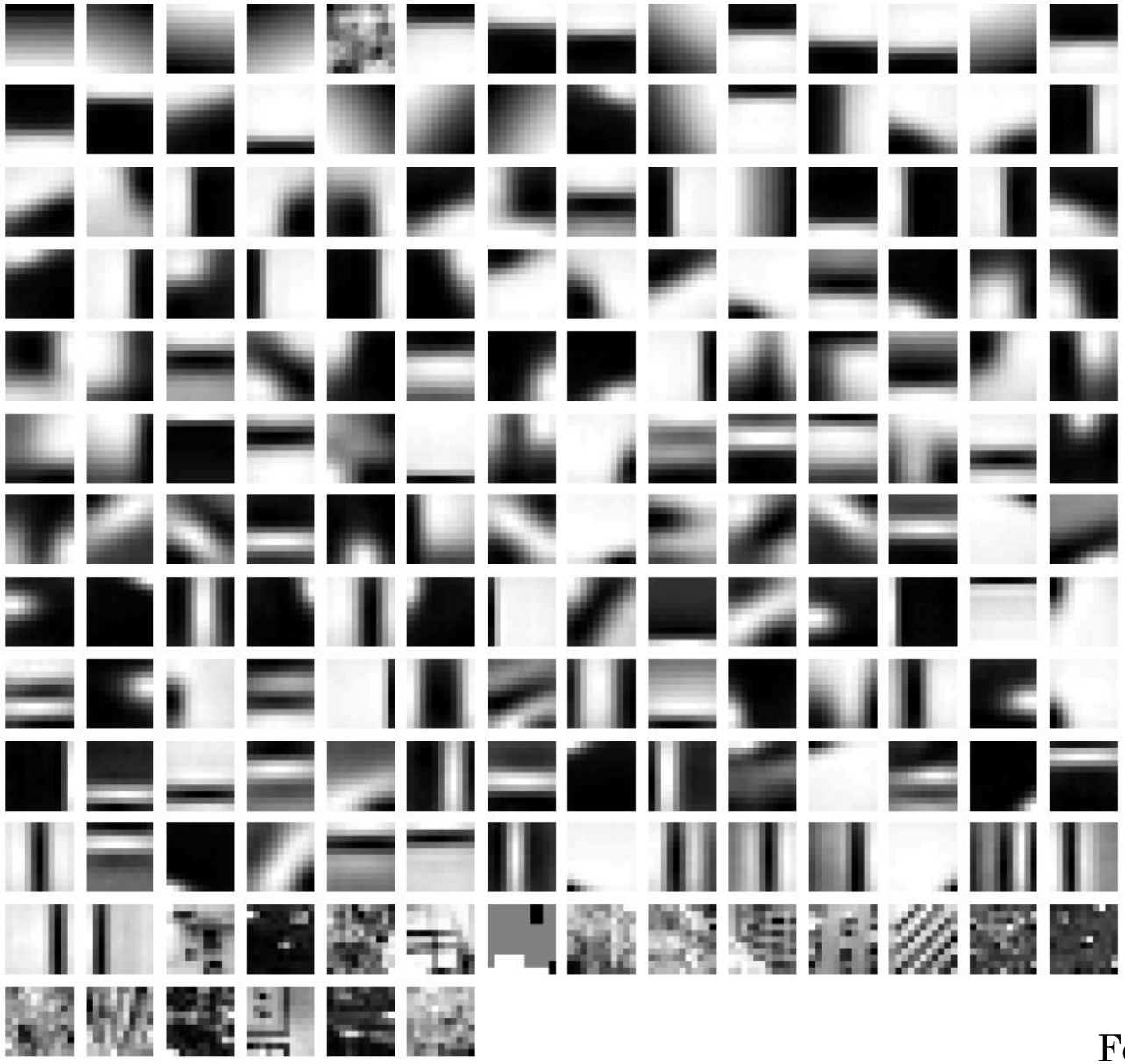
Another codebook





Source: B. Leibe

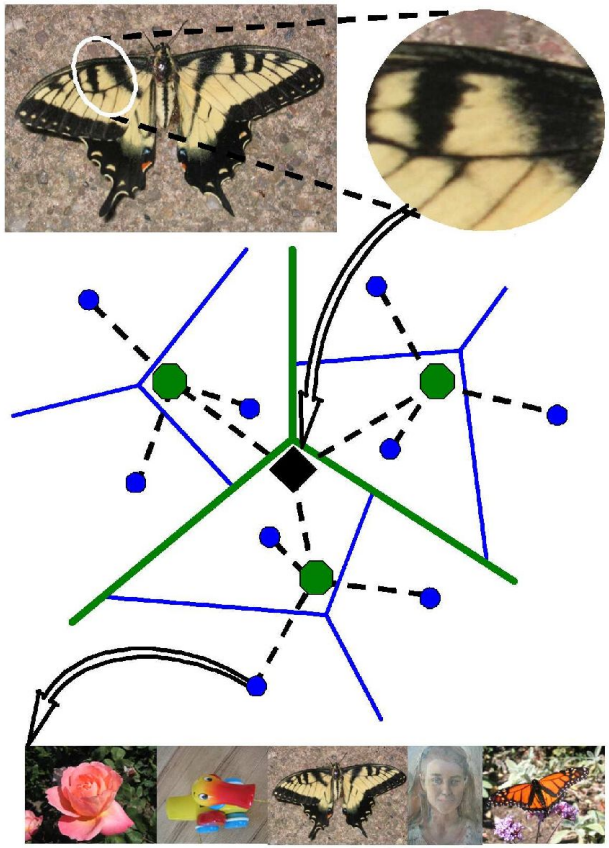
Yet another codebook



Fei-Fei et al.

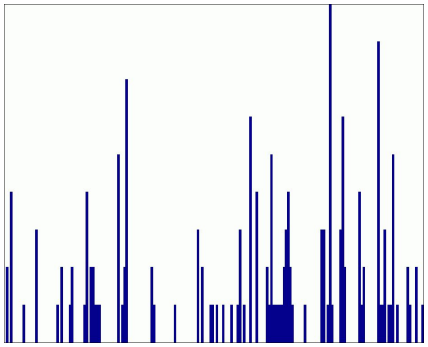
Visual vocabularies: Issues

- How to choose vocabulary size?
  - Too small: visual words not representative of all patches
  - Too large: quantization artifacts, overfitting
- Computational efficiency
  - Vocabulary trees (Nister & Stewenius, 2006)



Spatial pyramid representation

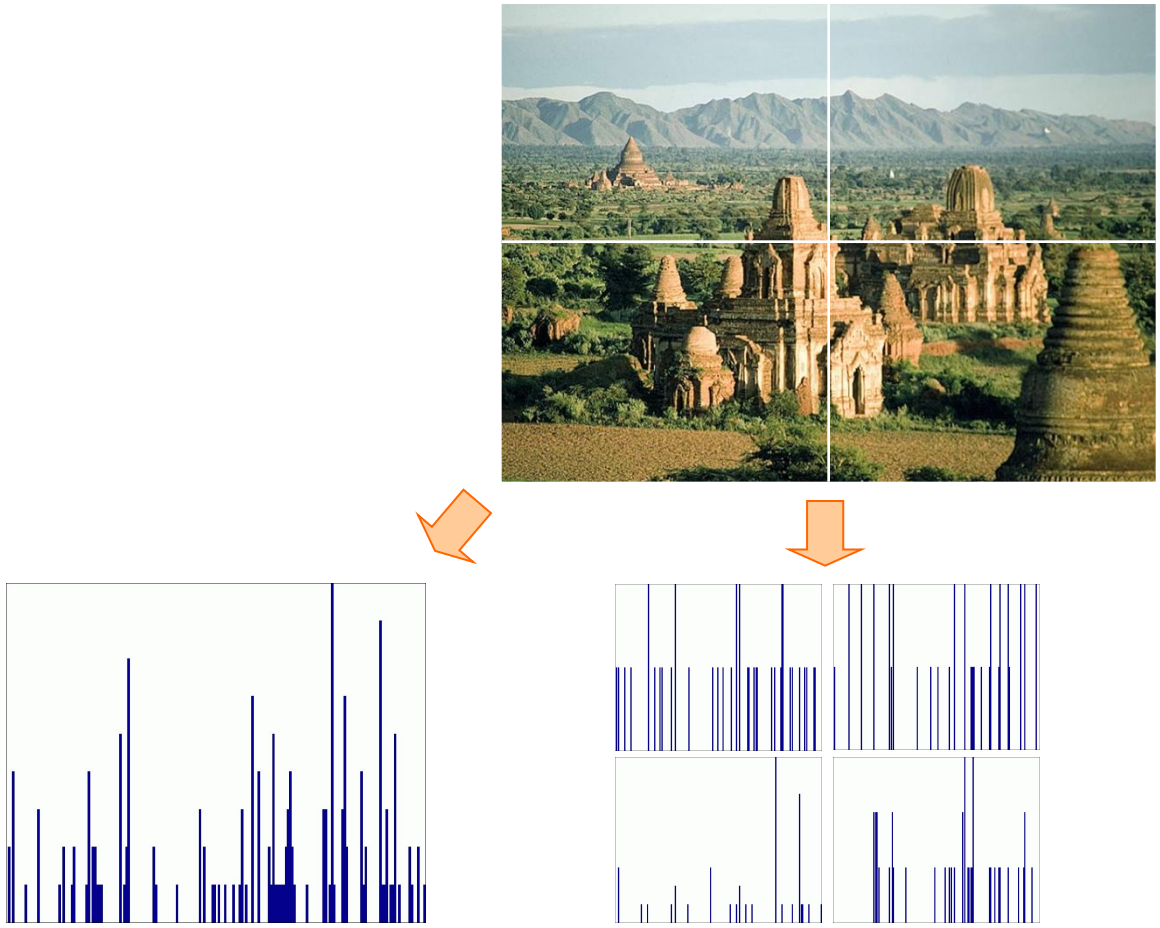
- Extension of a bag of features
- Locally orderless representation at several levels of resolution



Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution



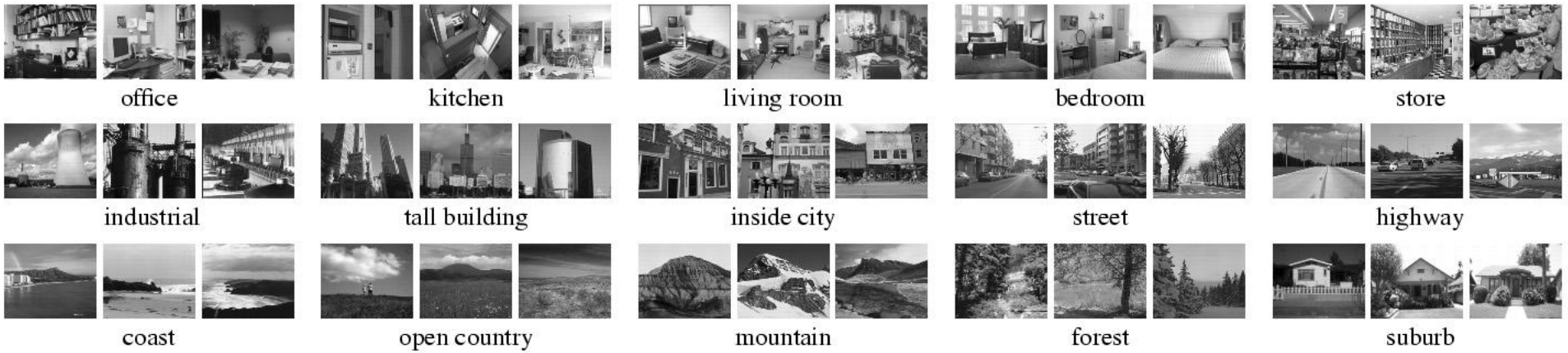
Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution

Lazebnik, Schmid & Ponce (CVPR 2006)

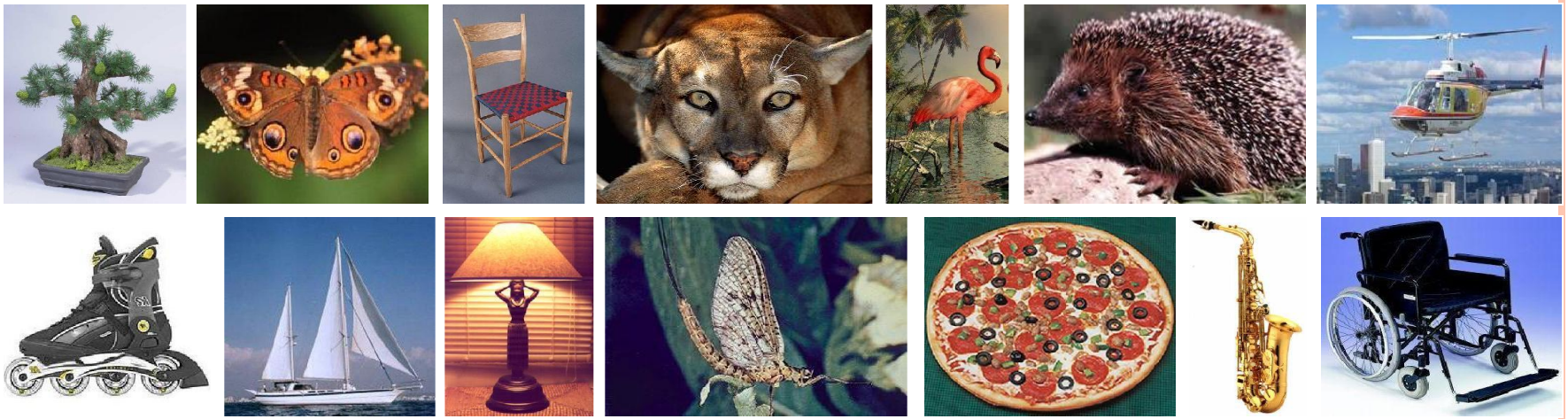
Scene category dataset



Multi-class classification results

	Weak features (vocabulary size: 16)		Strong features (vocabulary size: 200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0 (1 × 1)	45.3 ±0.5		72.2 ±0.6	
1 (2 × 2)	53.6 ±0.3	56.2 ±0.6	77.9 ±0.6	79.0 ±0.5
2 (4 × 4)	61.7 ±0.6	64.7 ±0.7	79.4 ±0.3	**81.1** ±0.3
3 (8 × 8)	63.3 ±0.8	**66.8** ±0.6	77.2 ±0.4	80.7 ±0.3

Caltech101 dataset



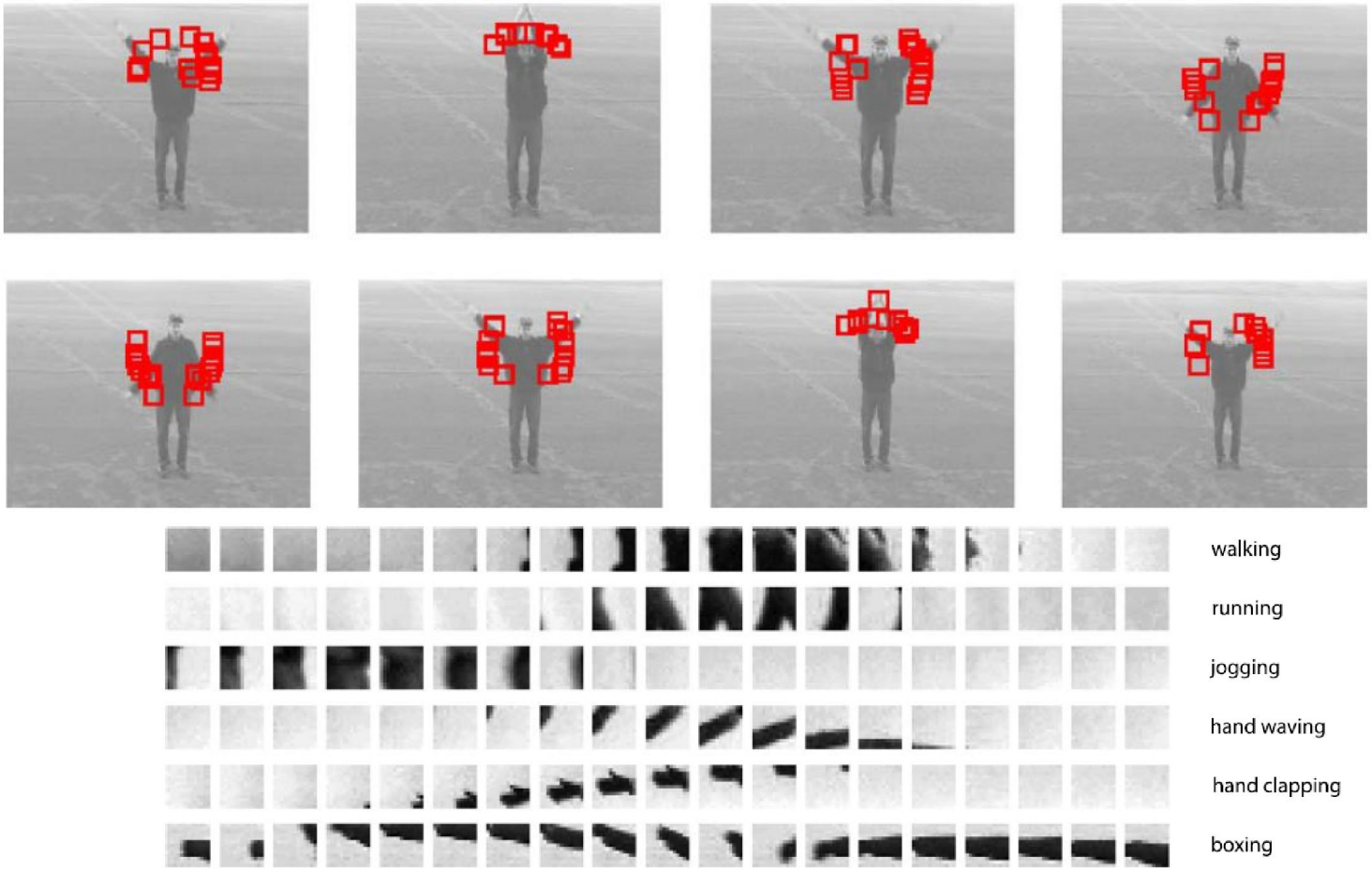
http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.htm

Multi-class classification results (30 training images per [1]

	Weak features (16)		Strong features (200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0	15.5 ±0.9		41.2 ±1.2	
1	31.4 ±1.2	32.8 ±1.3	55.9 ±0.9	57.0 ±0.8
2	47.2 ±1.1	49.3 ±1.4	63.6 ±0.9	**64.6** ±0.8
3	52.2 ±0.8	**54.0** ±1.1	60.3 ±0.9	64.6 ±0.7

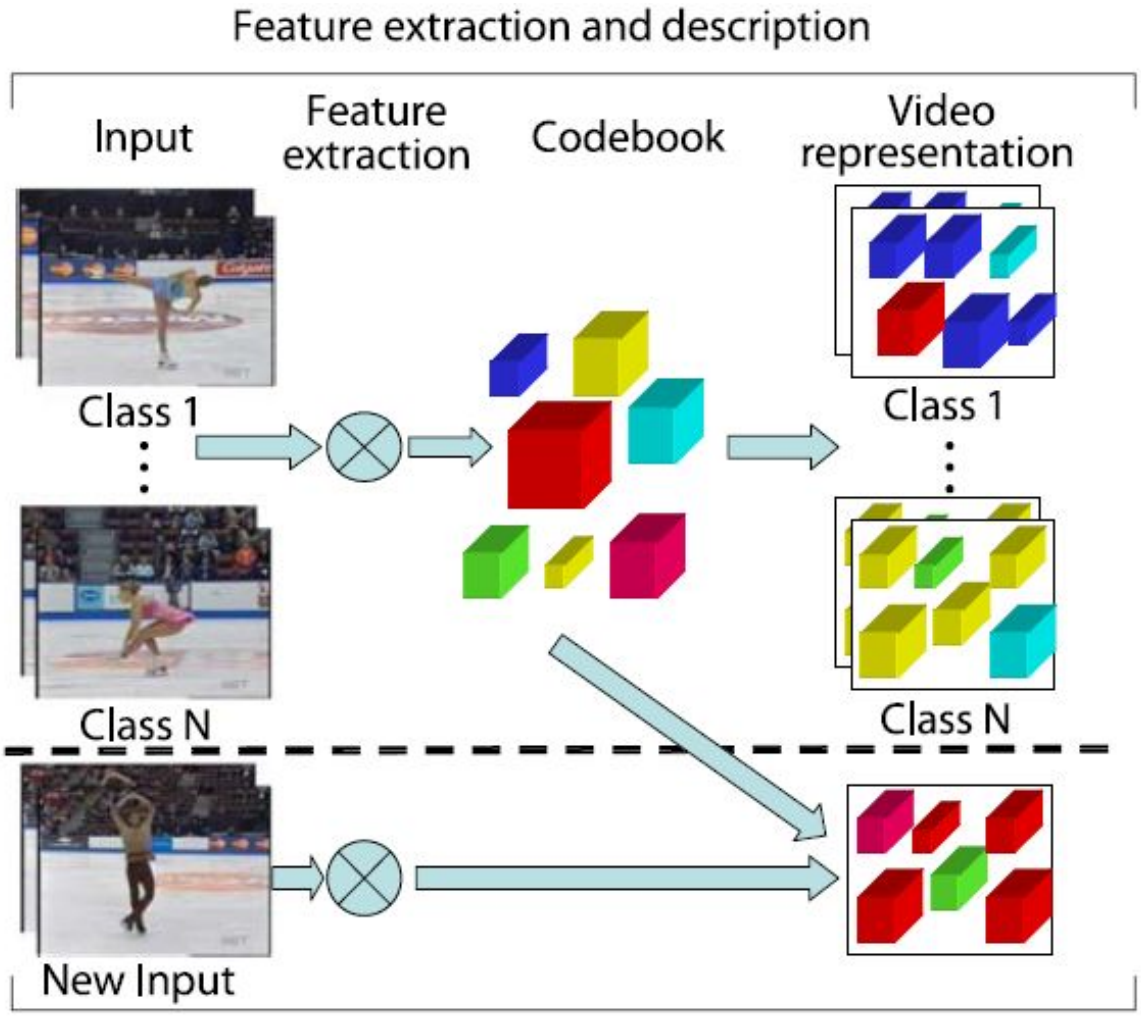
Bags of features for action recognition

Space-time interest points



Juan Carlos Niebles, Hongcheng Wang and Li Fei-Fei, **Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words**, IJCV 2008.

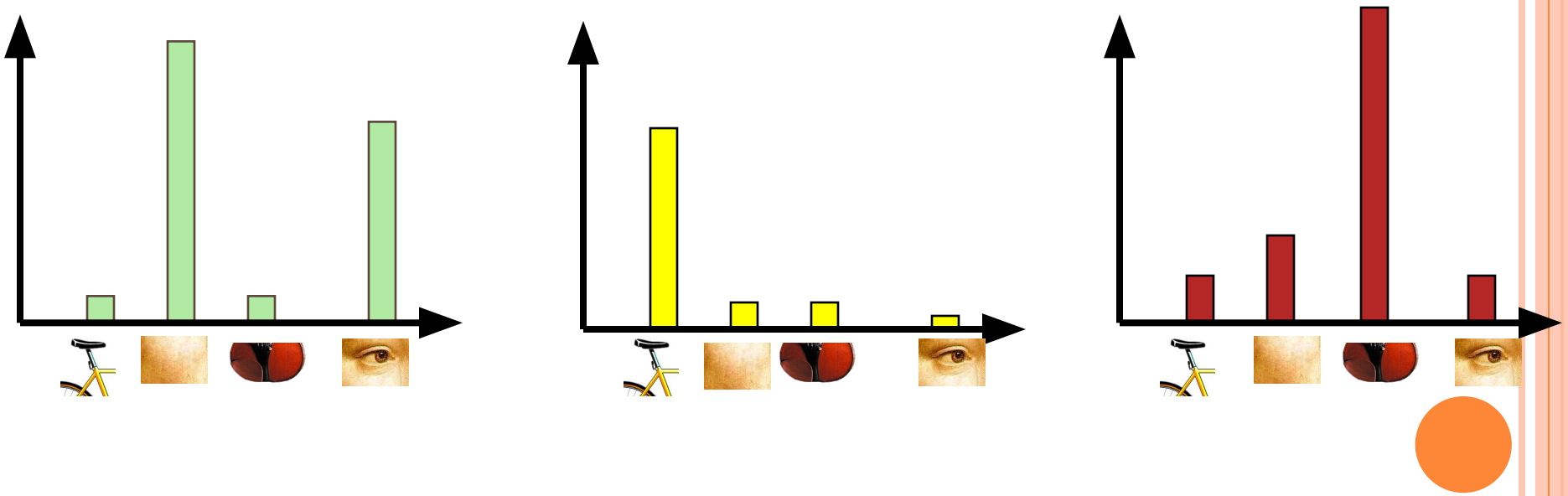
Bags of features for action recognition



Juan Carlos Niebles, Hongcheng Wang and Li Fei-Fei, **Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words**, IJCV 2008.

Image classification

- Given the bag-of-features representations of images from different classes, how do we learn a model for distinguishing them?



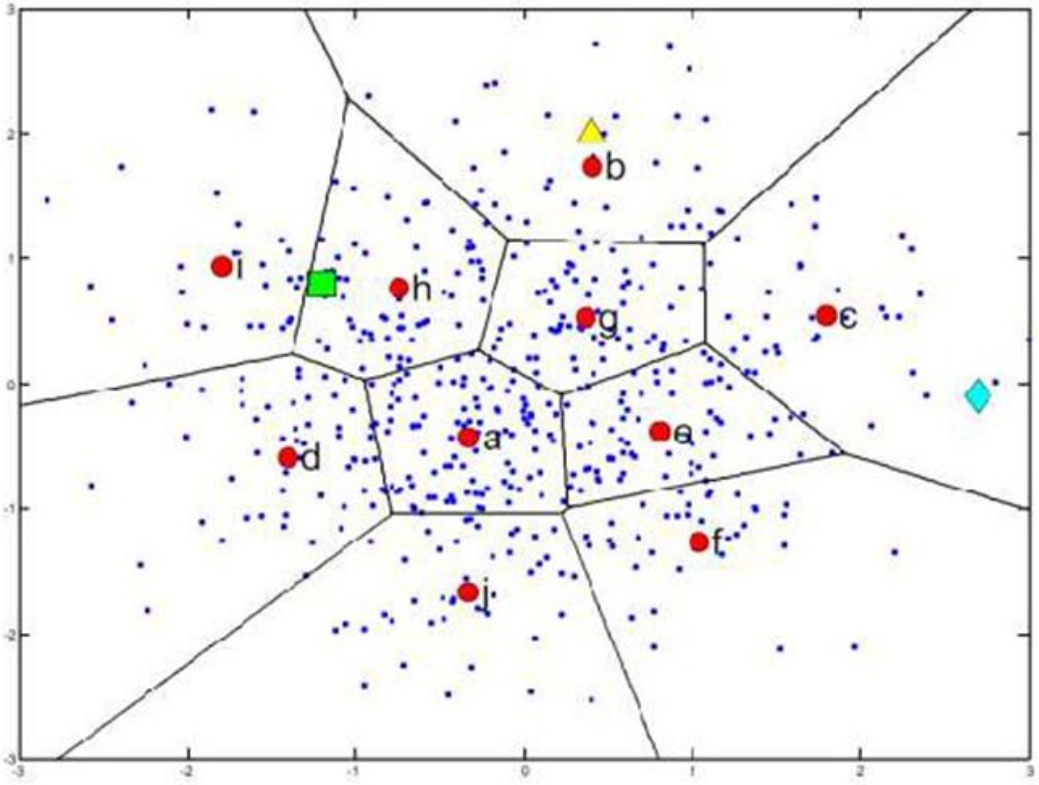
Lecture IV: Part II

Outline

- Histogram of local features
- Bag of words model
- Soft quantization and sparse coding
- Supervector with Gaussian mixture model

Hard quantization

$$H(w) = \frac{1}{n}\sum_{i=1}^{n} \begin{cases} 1, & if\ w = argmin_{v \in V}(D(v, r_i)) \\ 0, & otherwise \end{cases}$$



Slides credit: Cao & Feris

Soft quantization based on uncertainty

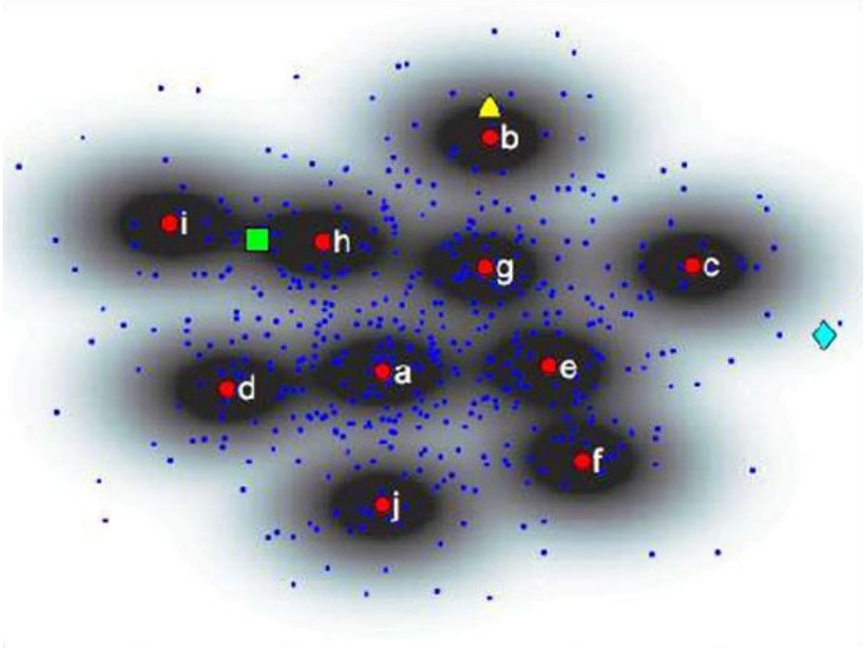
- Quantize each local feature into multiple code-words that are closest to it by appropriately splitting its weight

$$K_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

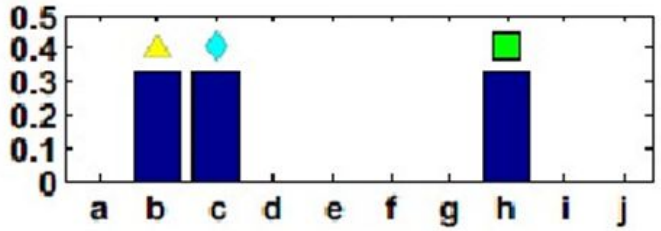
$$UNC(w) = \frac{1}{n}\sum_{i=1}^{n} \frac{K_\sigma(D(w,f))}{\sum_{j=1}^{|V|} K_\sigma(D(v_j, r_i))}$$

Gemert et at., Visual Word Ambiguity, PAMI 2009

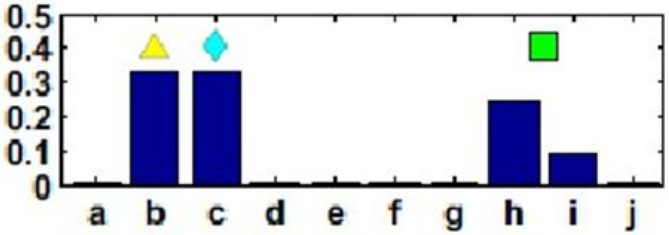
Soft Quantization



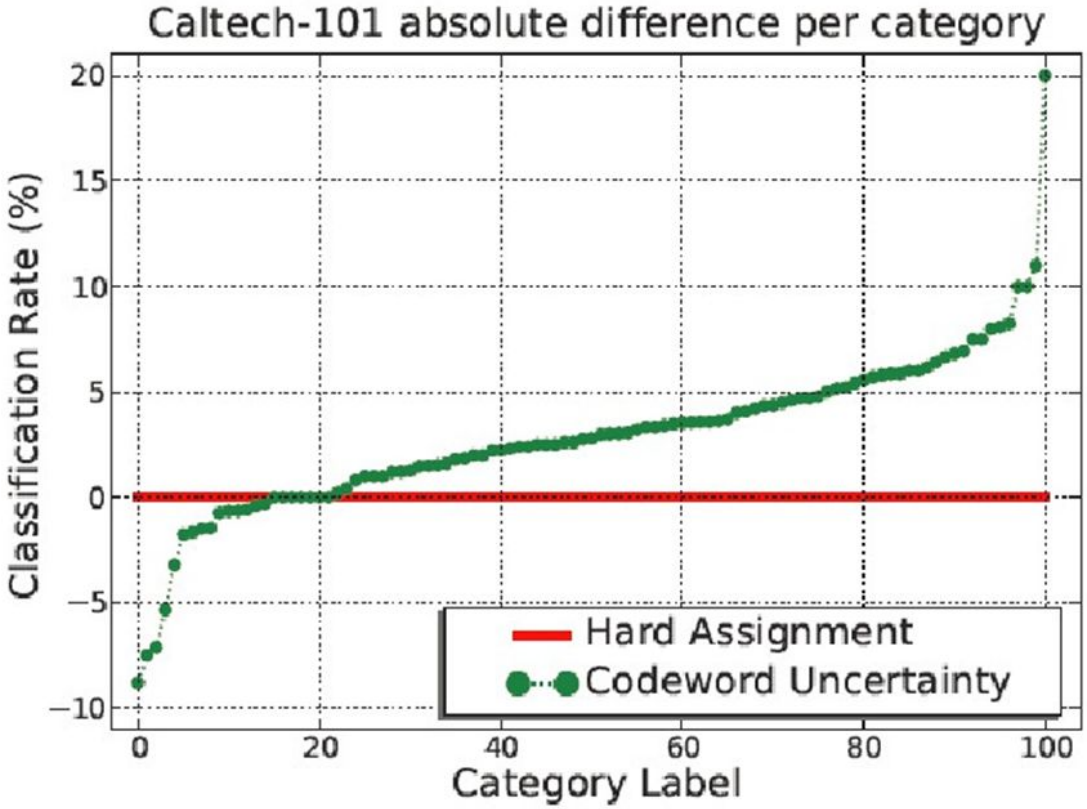
Hard quantization



Soft quantization



Some experiments on soft quantization



- Improvement on classification rate from soft quantization

Sparse coding

- Hard quantization is an "extremely sparse representation

$$min\|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{z}\|_0 = 1$$

- Generalizing from it, we may consider to solve

$$min\|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \lambda\|\mathbf{z}\|_0$$

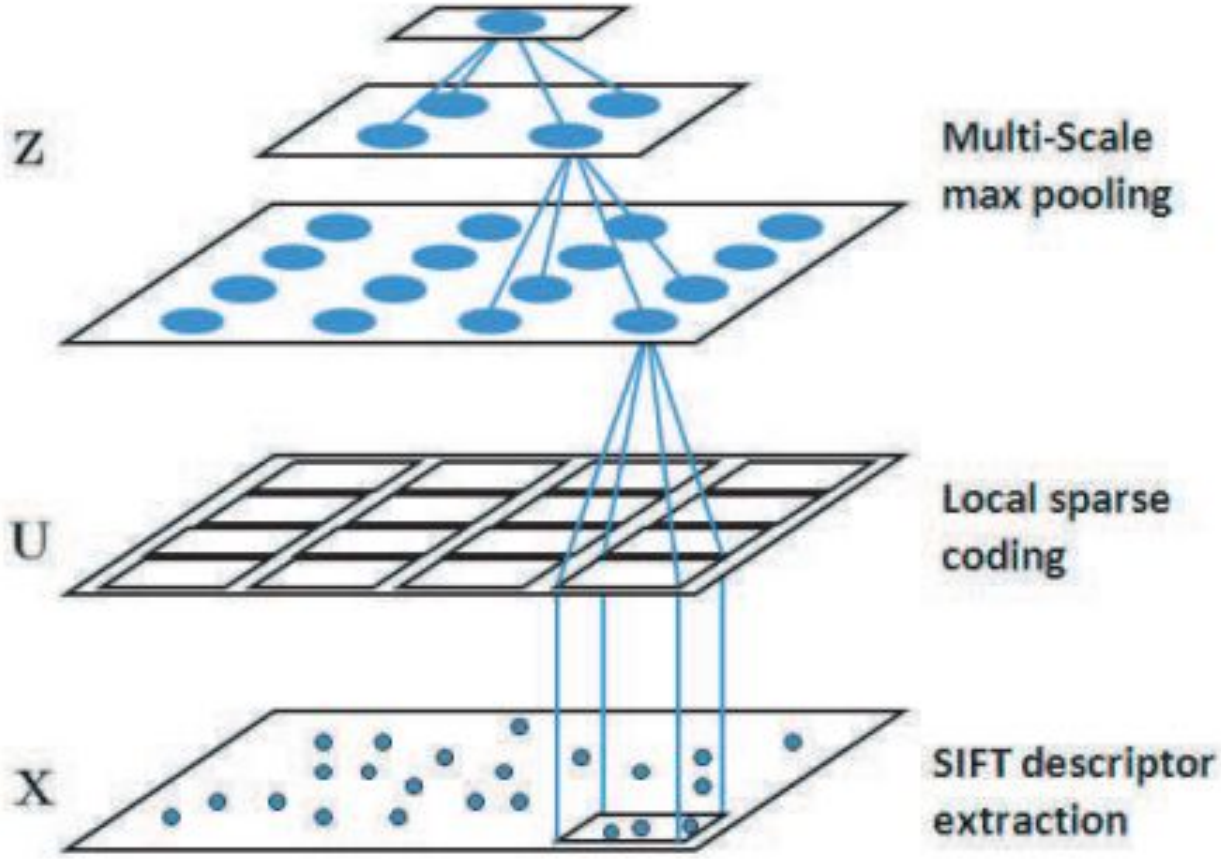
for soft quantization, but it is hard to solve

- In practice, we consider solving the sparse coding as

$$min\|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \lambda\|\mathbf{z}\|_1$$

for quantization

Soft quantization and pooling



[Yang et al, 2009]:
Linear Spatial Pyramid Matching using Sparse Coding for Image Classification

Outline

- Histogram of local features
- Bag of words model
- Soft quantization and sparse coding
- Supervector with Gaussian mixture model

Quiz

- What is the potential shortcoming of bag-of-feature representation based on quantization and pooling?

Modeling the feature distribution

- The bag-of-feature histogram represents the distribution of a set of features
  - The quantization steps introduces information loss!

- How about modeling the distribution without quantization?
  - Gaussian mixture model

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Quiz

- Given a set of features $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, how can we fit the GMM distribution?

The Gaussian Distribution

- Multivariate Gaussian

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi|\boldsymbol{\Sigma}|)^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

mean

covariance

- Define precision to be the inverse of the covariance

$$\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$$

- In 1-dimension

$$\tau = \frac{1}{\sigma^2}$$

Slides credit: C. M. Bishop

Likelihood Function

- Data set

$$D = \{\mathbf{x}_n\} \qquad n = 1, \ldots, N$$

- Assume observed data points generated independently

$$p(D|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- Viewed as a function of the parameters, this is known as the *likelihood function*

Slides credit: C. M. Bishop

Maximum Likelihood

- Set the parameters by maximizing the likelihood function
- Equivalently maximize the log likelihood

$$\ln p(D|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{N}{2}\ln(2\pi) - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu})$$

Slides credit: C. M. Bishop

Maximum Likelihood Solution

- Maximizing w.r.t. the mean gives the *sample mean*

$$\boldsymbol{\mu}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

- Maximizing w.r.t covariance gives the *sample covariance*

$$\boldsymbol{\Sigma}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})^{\mathsf{T}}$$

Slides credit: C. M. Bishop

Bias of Maximum Likelihood

- Consider the expectations of the maximum likelihood estimates under the Gaussian distribution

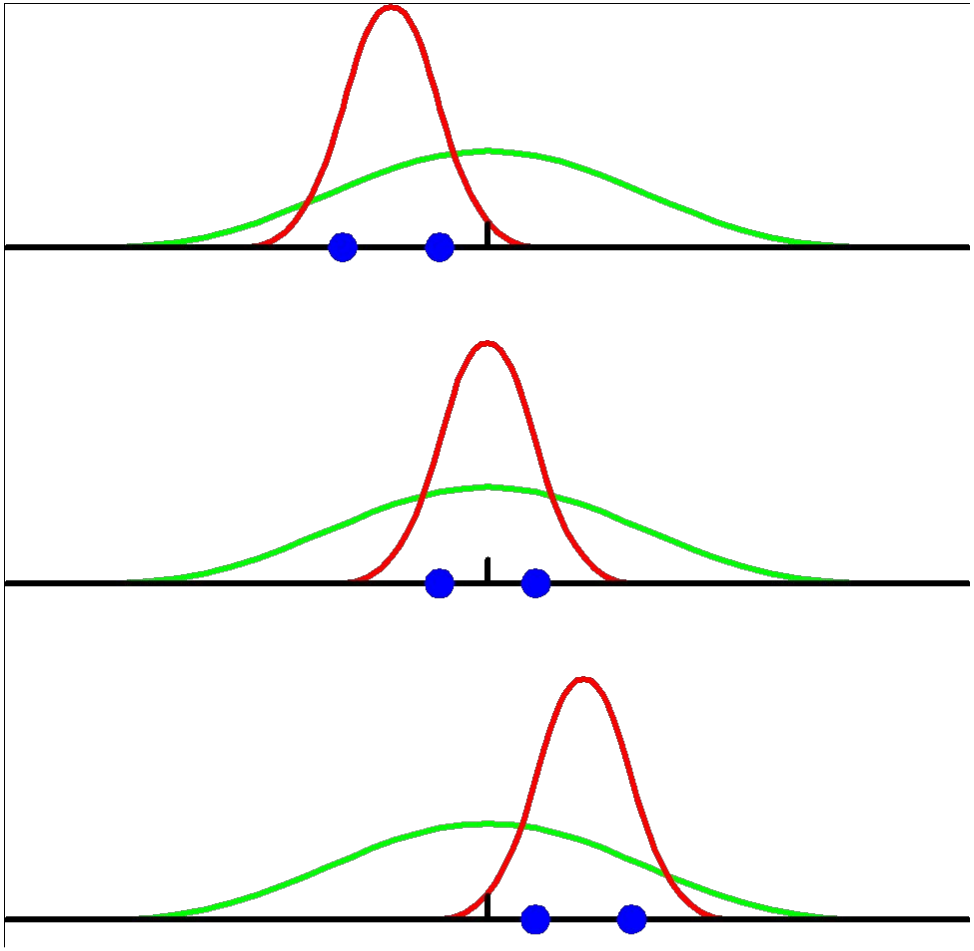
$$\mathrm{E}[\boldsymbol{\mu}_{\mathrm{ML}}] = \boldsymbol{\mu}$$

$$\mathrm{E}[\boldsymbol{\Sigma}_{\mathrm{ML}}] = \left(\frac{N-1}{N}\right)\boldsymbol{\Sigma}$$

- The maximum likelihood solution systematically under-estimates the covariance
- This is an example of *over-fitting*

Slides credit: C. M. Bishop

Intuitive Explanation of Over-fitting



Unbiased Variance Estimate

- Clearly we can remove the bias by using

$$\widetilde{\Sigma} = \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})^{\top}$$

  since this gives

$$\mathsf{E}\left[\widetilde{\Sigma}\right] = \Sigma$$

- For an infinite data set the two expressions are equal

Slides credit: C. M. Bishop

Gaussian Mixtures

- Linear super-position of Gaussians

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

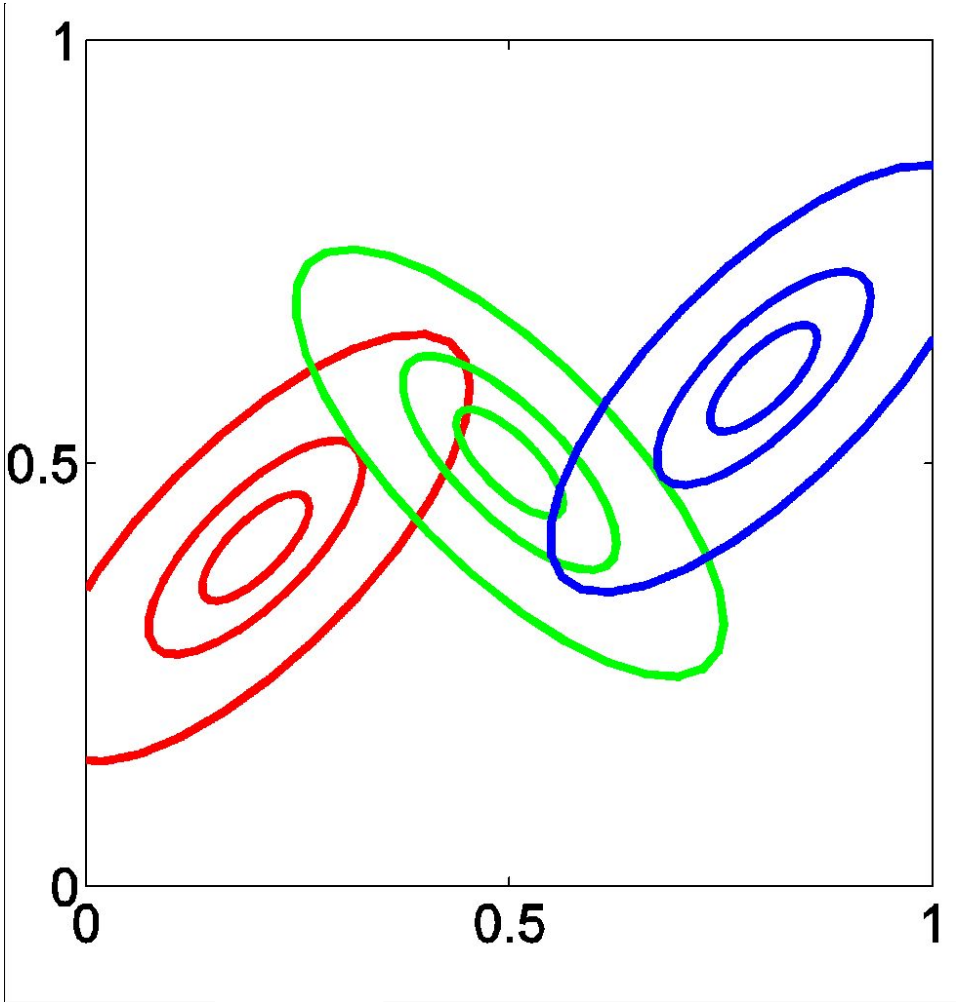
- Normalization and positivity require

$$\sum_{k=1}^{K} \pi_k = 1 \qquad 0 \leqslant \pi_k \leqslant 1$$

- Can interpret the mixing coefficients as prior probabilities

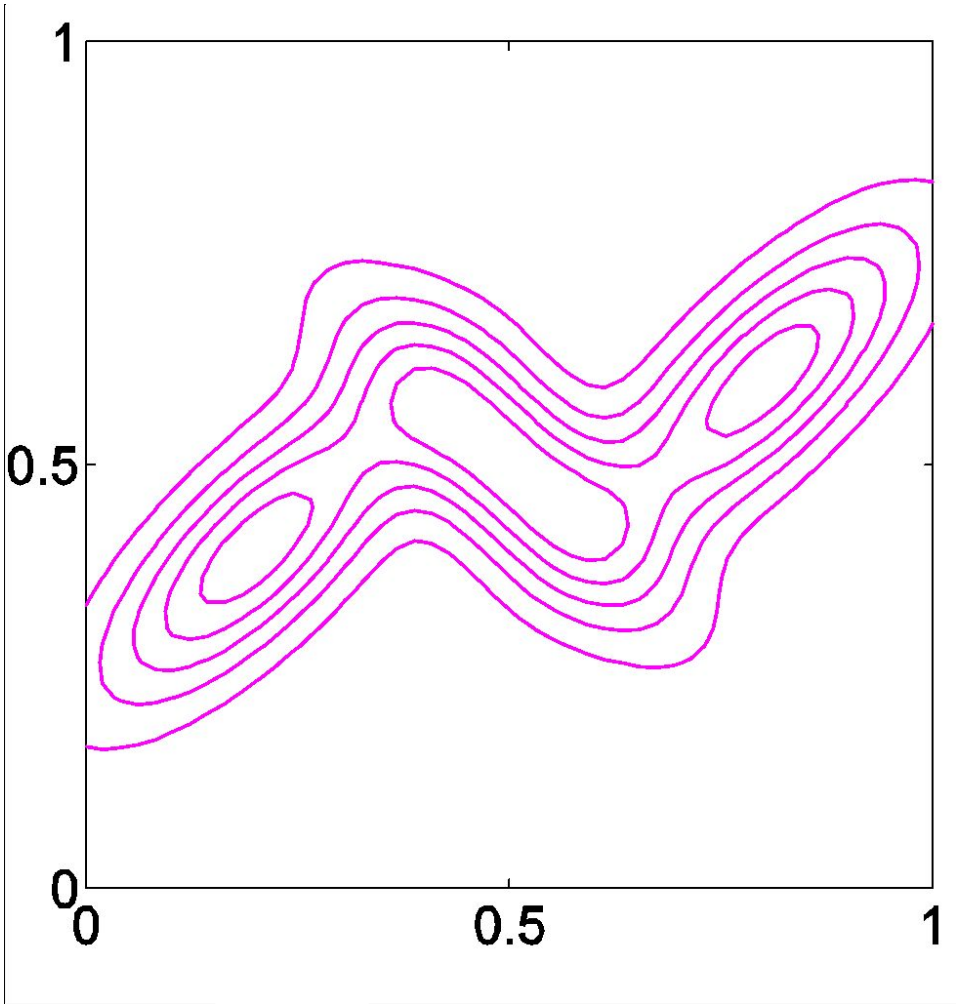
$$p(\mathbf{x}) = \sum_{k=1}^{K} p(k) p(\mathbf{x}|k)$$

Example: Mixture of 3 Gaussians

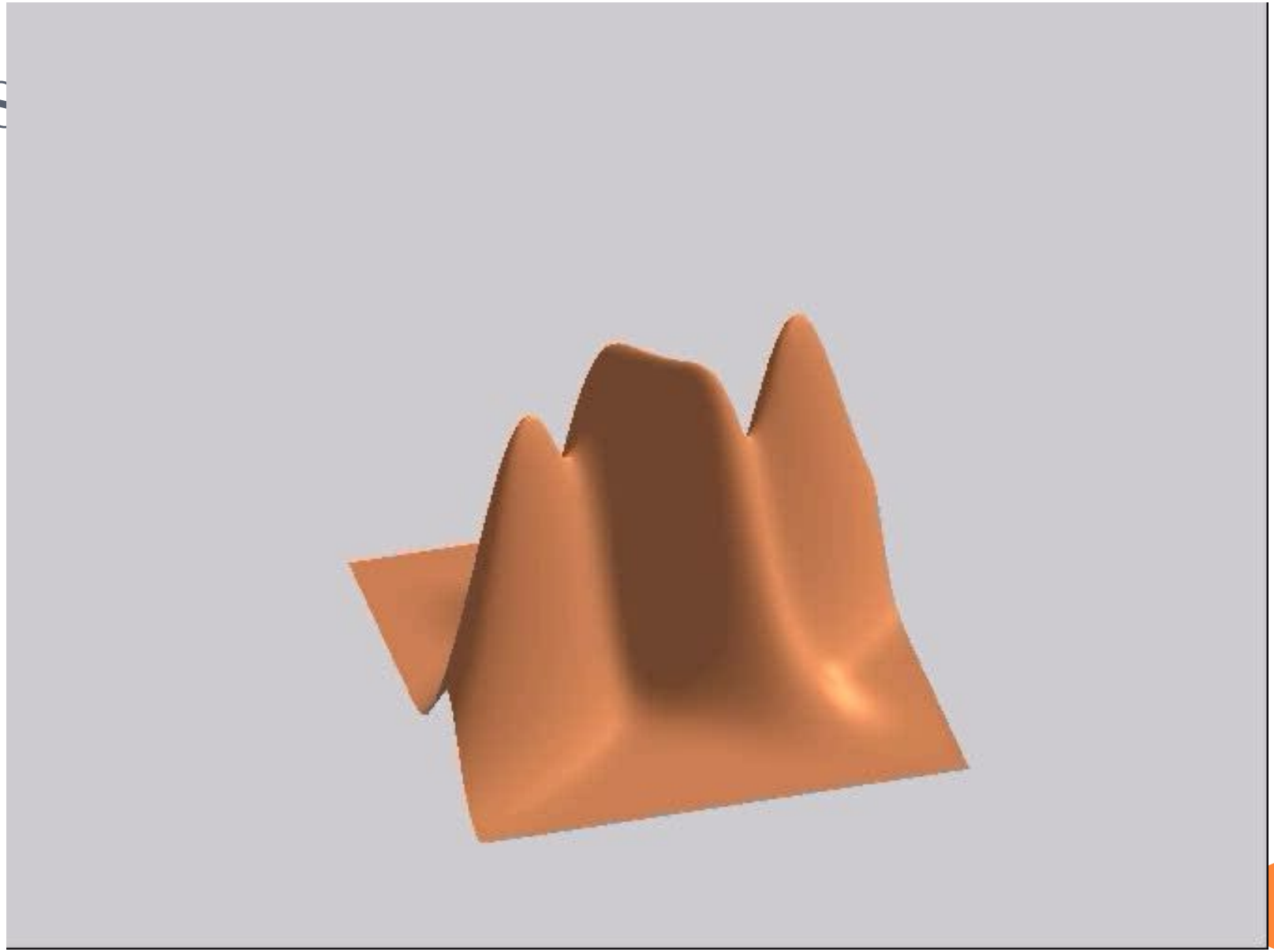


Slides credit: C. M. Bishop

Contours of Probability Distribution



Slides credit: C. M. Bishop



Slides credit: C. M. Bishop

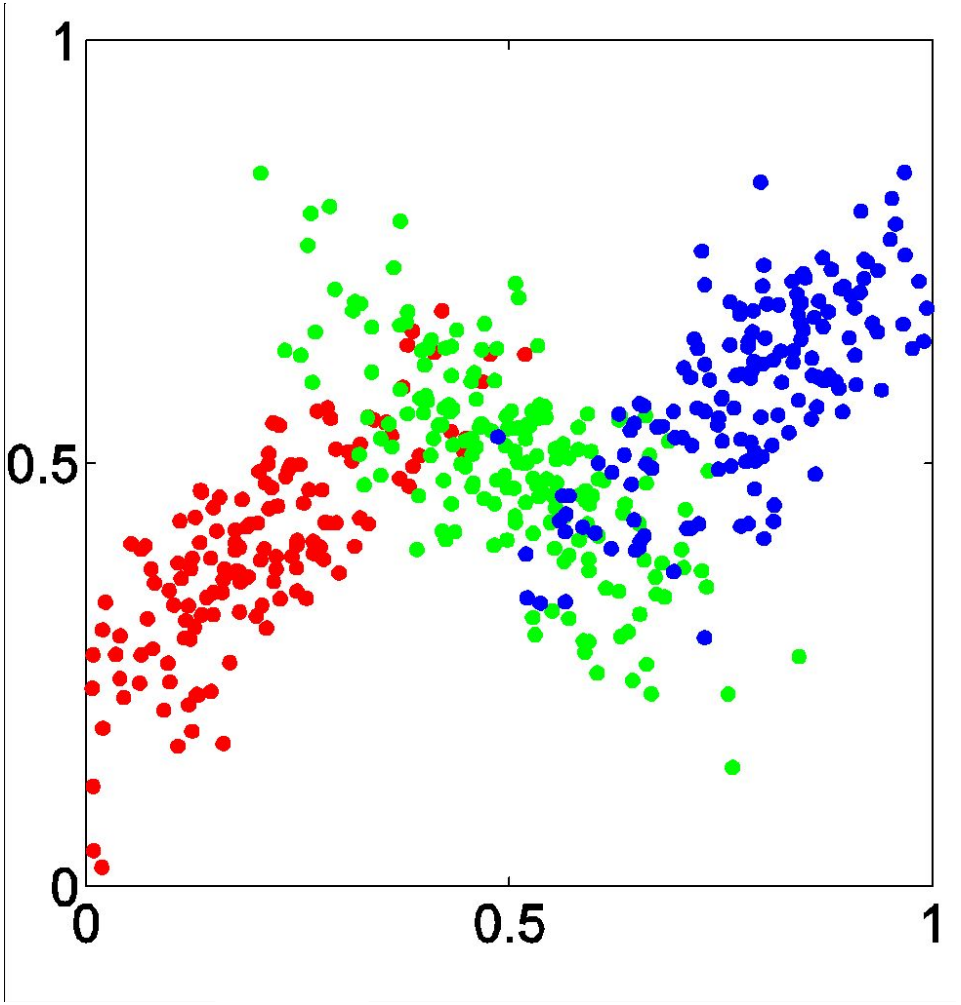
Sampling from the Gaussian

- To generate a data point:
  - first pick one of the components with probability π_k
  - then draw a sample $\mathbf{x}_n$ from that component

- Repeat these two steps for each new data point

Slides credit: C. M. Bishop

Synthetic Data Set



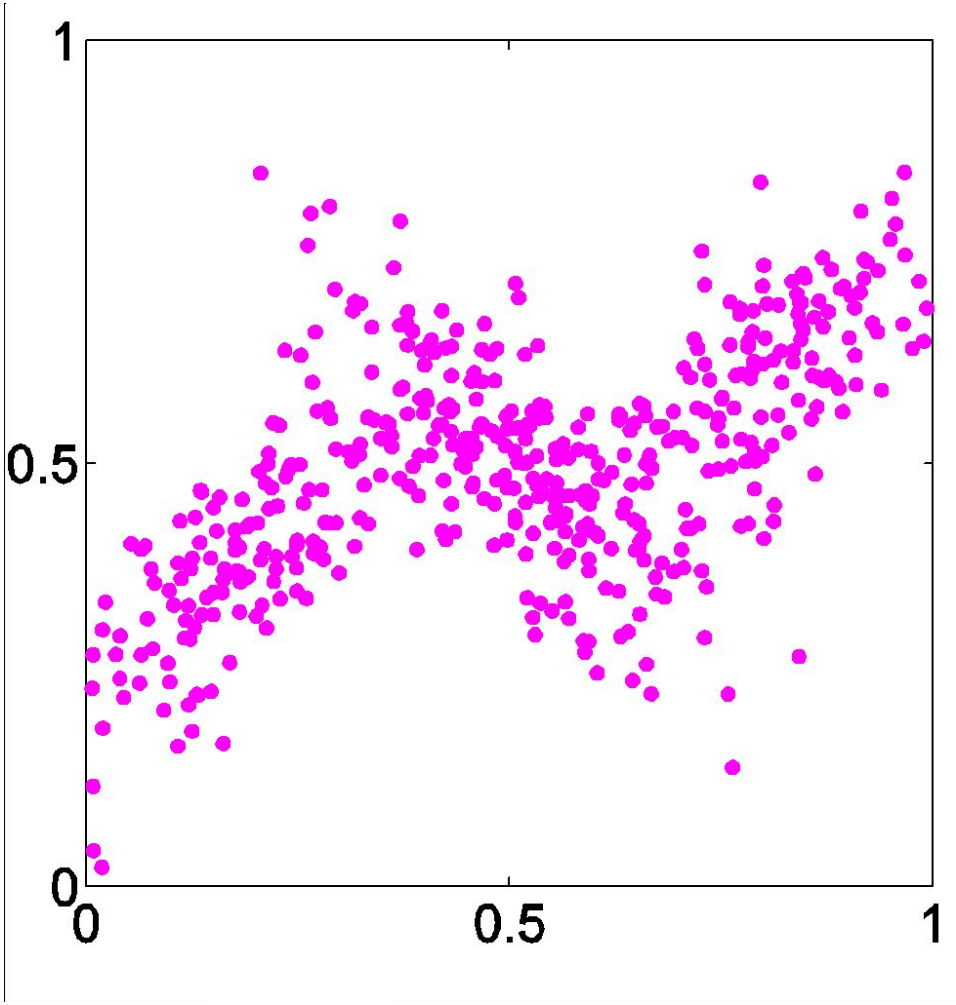
Slides credit: C. M. Bishop

Fitting the Gaussian Mixture

- We wish to invert this process – given the data set, find the corresponding parameters:
  - mixing coefficients, means, and covariances
- If we knew which component generated each data point, the maximum likelihood solution would involve fitting each component to the corresponding cluster
- Problem: the data set is unlabelled
- We shall refer to the labels as *latent* (= hidden) variables

Slides credit: C. M. Bishop

Synthetic Data Set Without Labels



Slides credit: C. M. Bishop

Posterior Probabilities

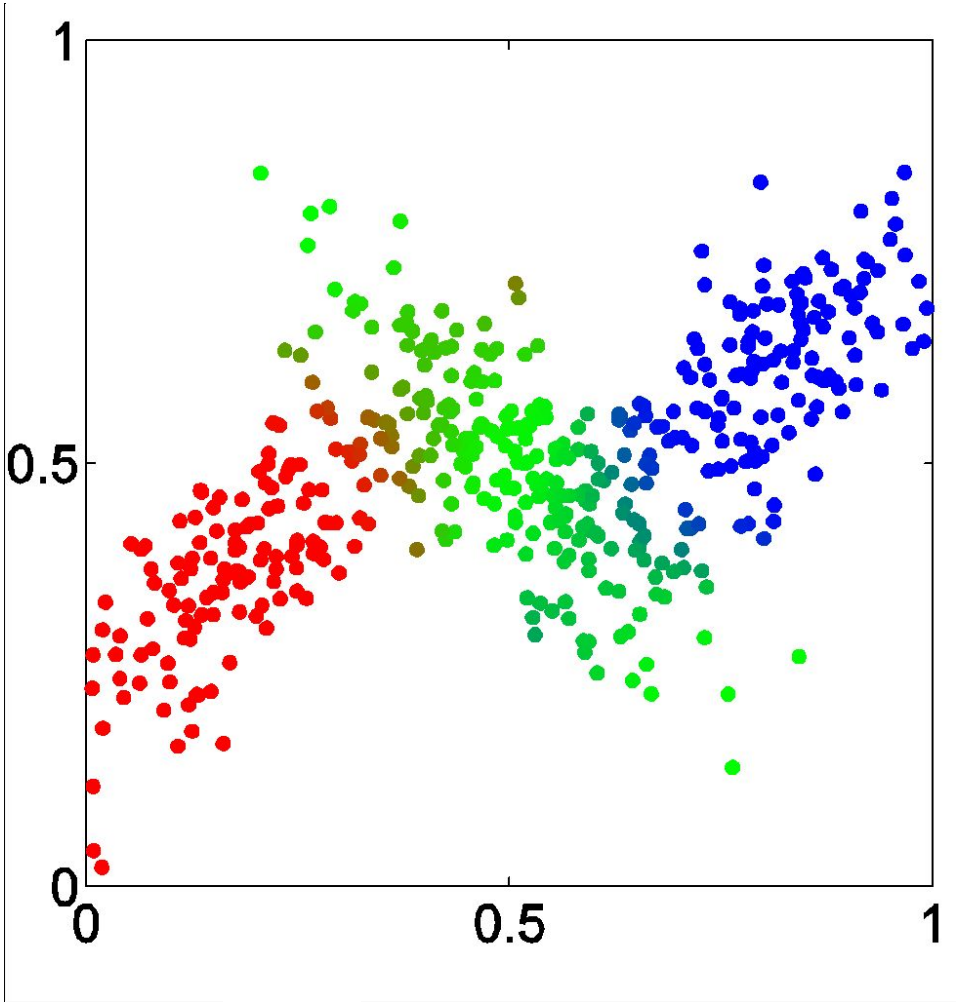
- We can think of the mixing coefficients as prior probabilities for the components
- For a given value of $\mathbf{x}$ we can evaluate the corresponding posterior probabilities, called *responsibilities*
- These are given from Bayes' theorem by

$$\gamma_k(\mathbf{x}) \equiv p(k|\mathbf{x}) = \frac{p(k)p(\mathbf{x}|k)}{p(\mathbf{x})}$$

$$= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

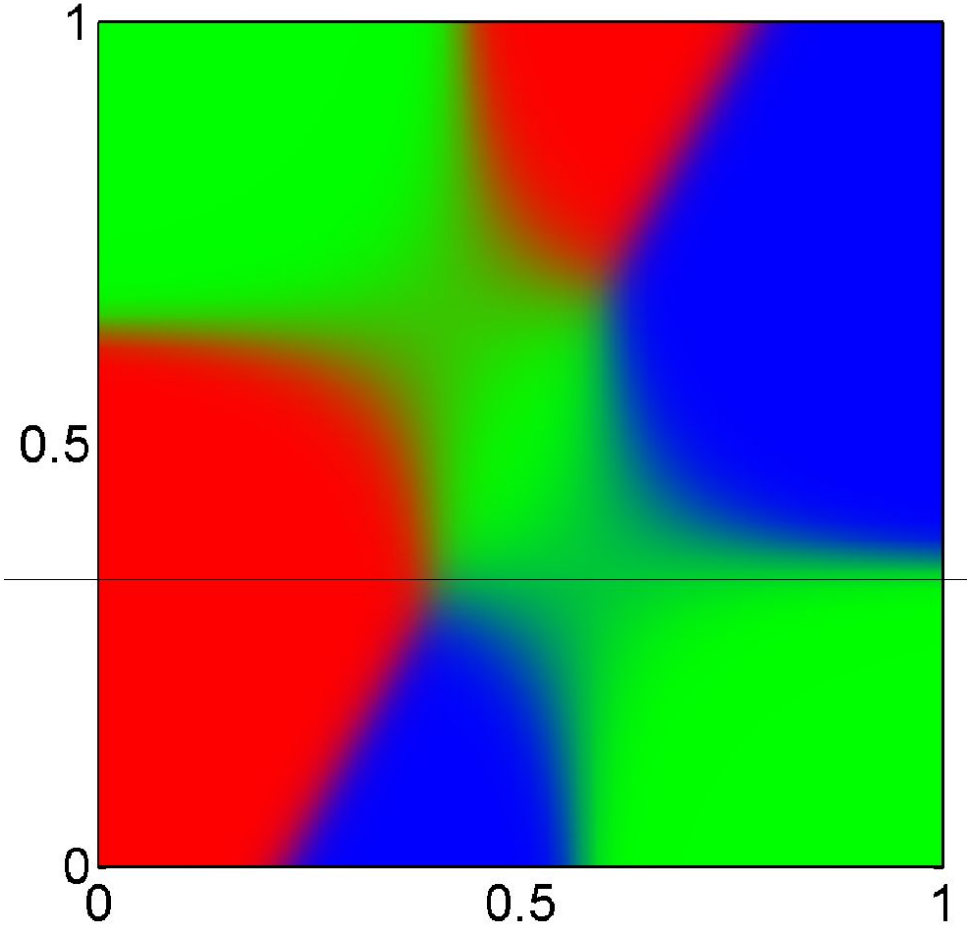
Slides credit: C. M. Bishop

Posterior Probabilities (colour coded)



Slides credit: C. M. Bishop

Posterior Probability Map



Slides credit: C. M. Bishop

Maximum Likelihood for the GMM

- The log likelihood function takes the form

$$\ln p(D|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

- Note: sum over components appears *inside* the log

- There is no closed form solution for maximum likelihood!

Slides credit: C. M. Bishop

Over-fitting in Gaussian Mixture Models

- Singularities in likelihood function when a component 'collapses' onto a data point:

$$\mathcal{N}(\mathbf{x}_n|\mathbf{x}_n, \sigma_j^2\mathbf{I}) = \frac{1}{(2\pi)^{1/2}}\frac{1}{\sigma_j}$$

  then consider $\sigma_j \to 0$

- Likelihood function gets larger as we add more components (and hence parameters) to the model
  - not clear how to choose the number K of components

Slides credit: C. M. Bishop

Problems and Solutions

- How to maximize the log likelihood
  - solved by expectation-maximization (EM) algorithm
- How to avoid singularities in the likelihood function
  - solved by a Bayesian treatment
- How to choose number K of components
  - also solved by a Bayesian treatment

Slides credit: C. M. Bishop

EM Algorithm – Informal Derivation

- Let us proceed by simply differentiating the log likelihood
- Setting derivative with respect to $\boldsymbol{\mu}_j$ equal to zero gives

$$-\sum_{n=1}^{N} \underbrace{\frac{\pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_k \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}}_{\gamma_j(\mathbf{x}_n)} \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_j) = 0$$

giving

$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)}$$

- which is simply the weighted mean of the data

EM Algorithm – Informal Derivation

- Similarly for the co-variances

$$\boldsymbol{\Sigma}_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)(\mathbf{x}_n - \boldsymbol{\mu}_j)(\mathbf{x}_n - \boldsymbol{\mu}_j)^{\mathsf{T}}}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)}$$

- For mixing coefficients use a Lagrange multiplier to give

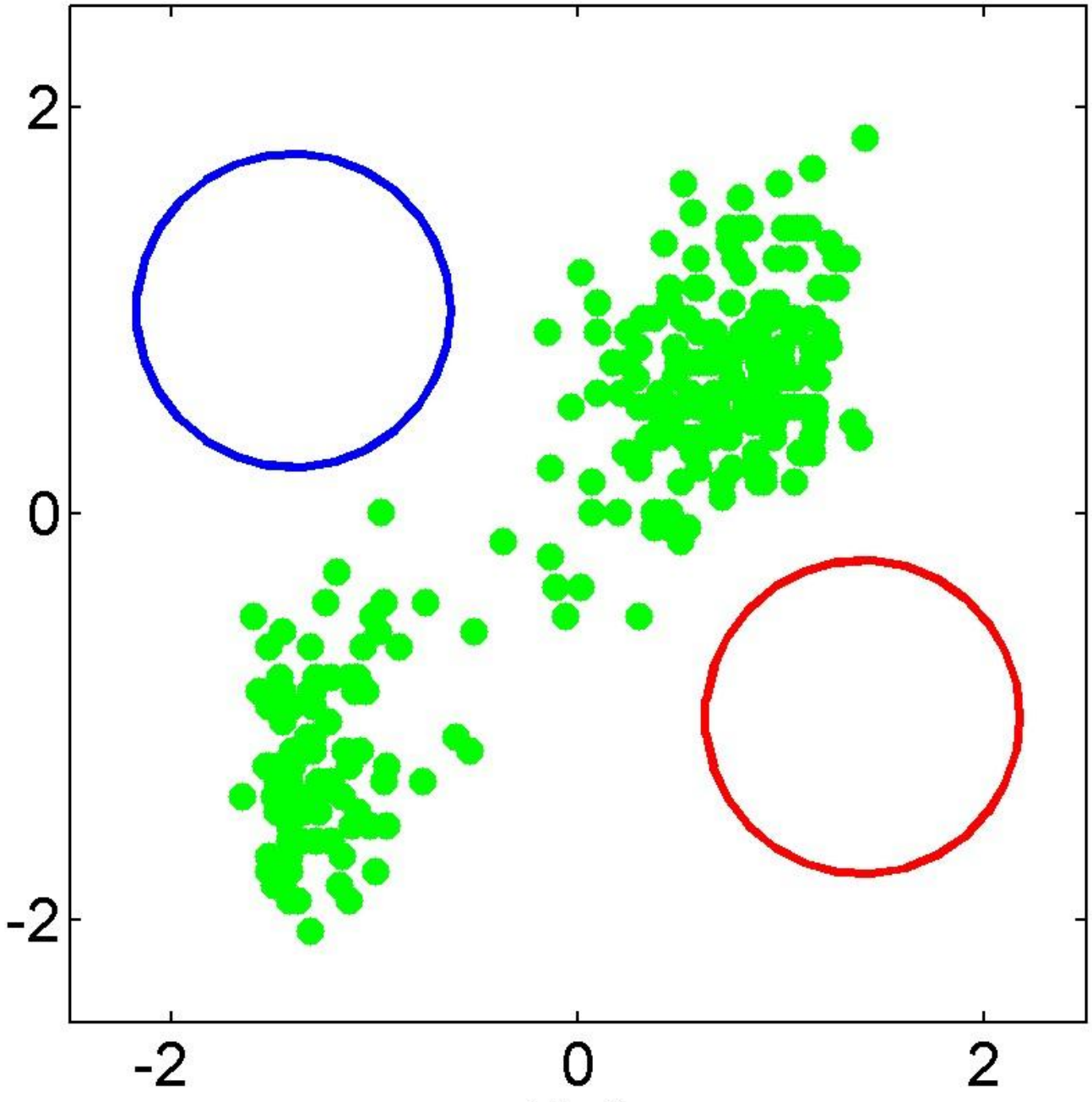
$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)$$

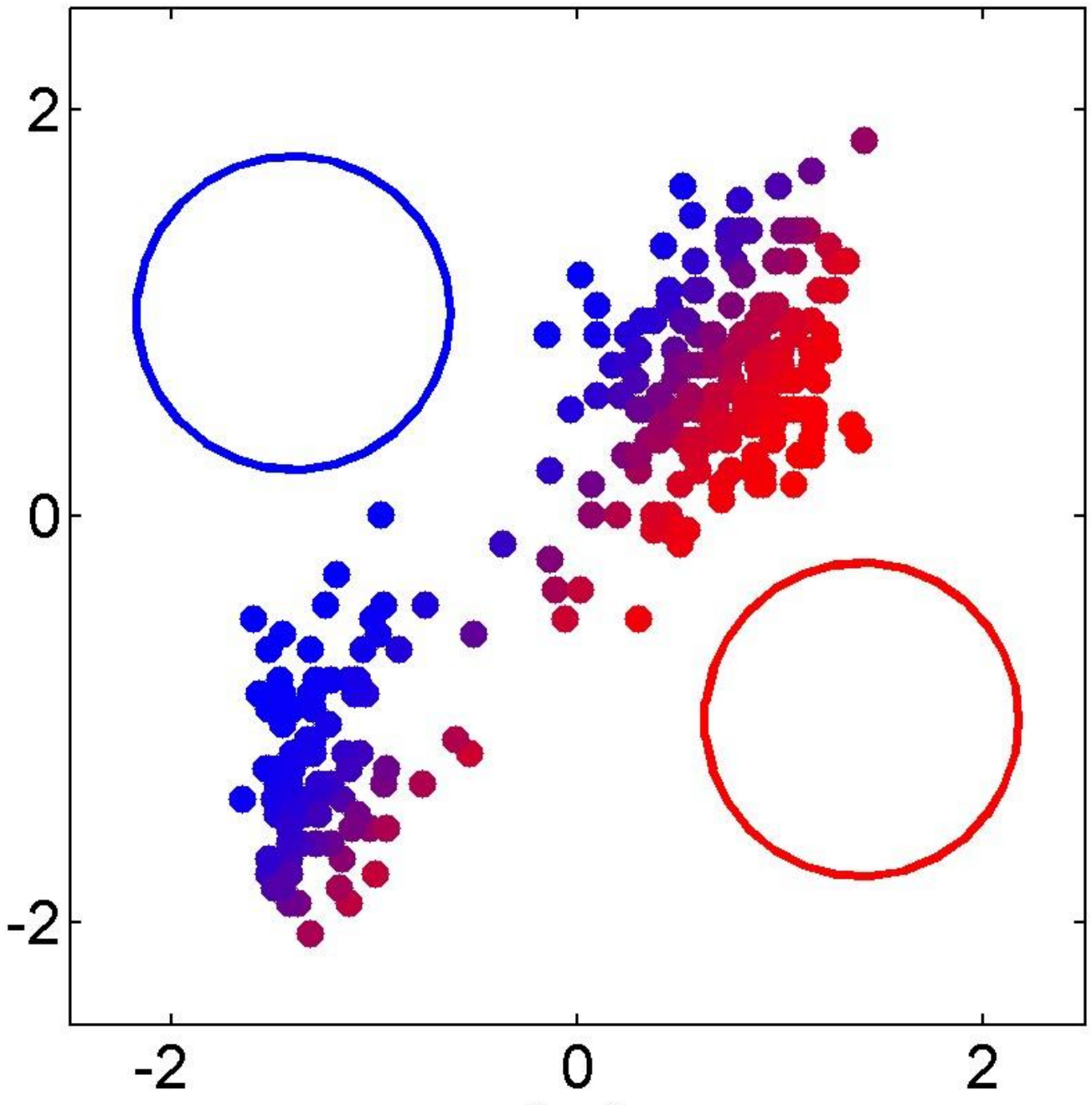
Slides credit: C. M. Bishop

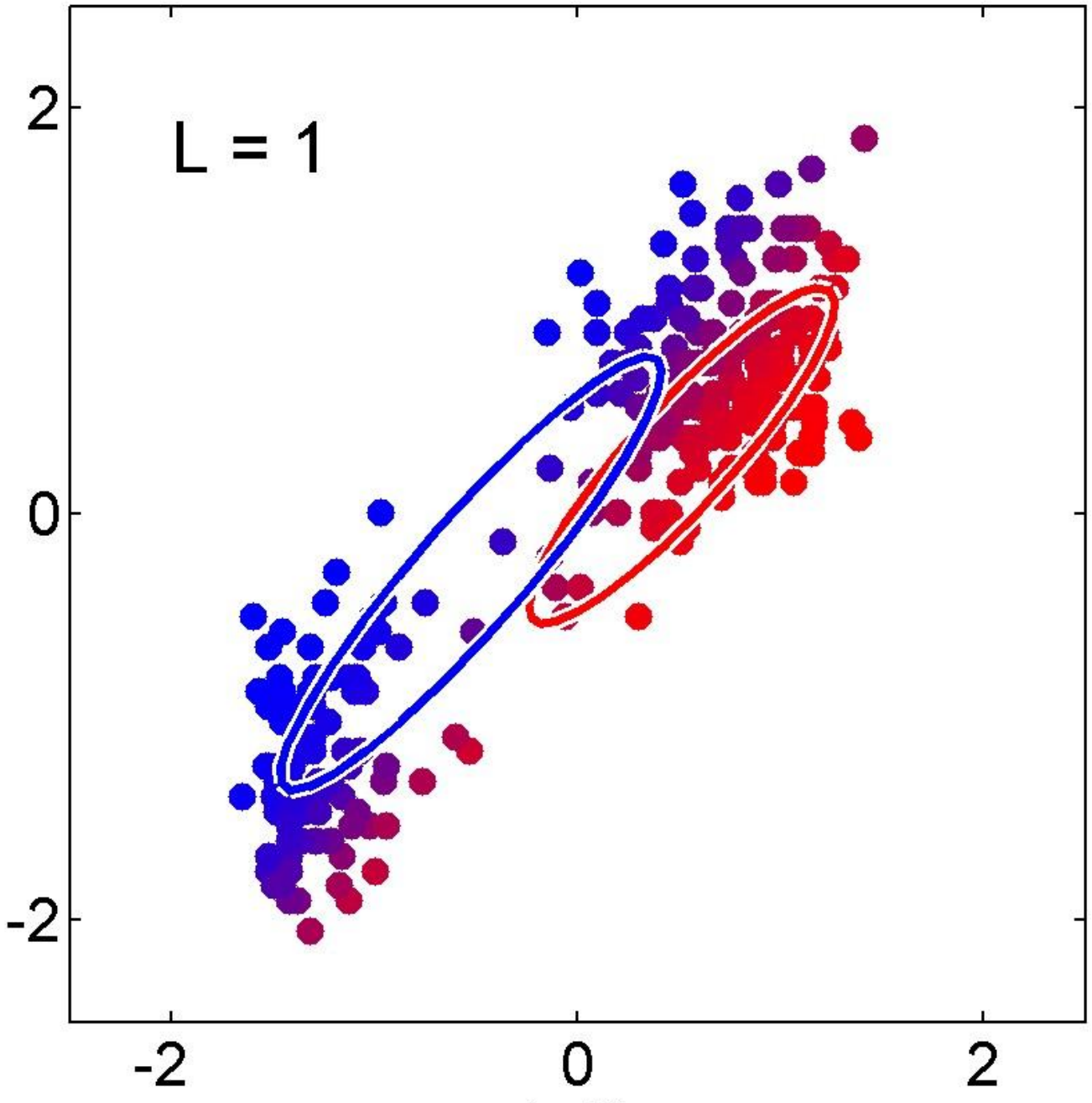
EM Algorithm – Informal Derivation

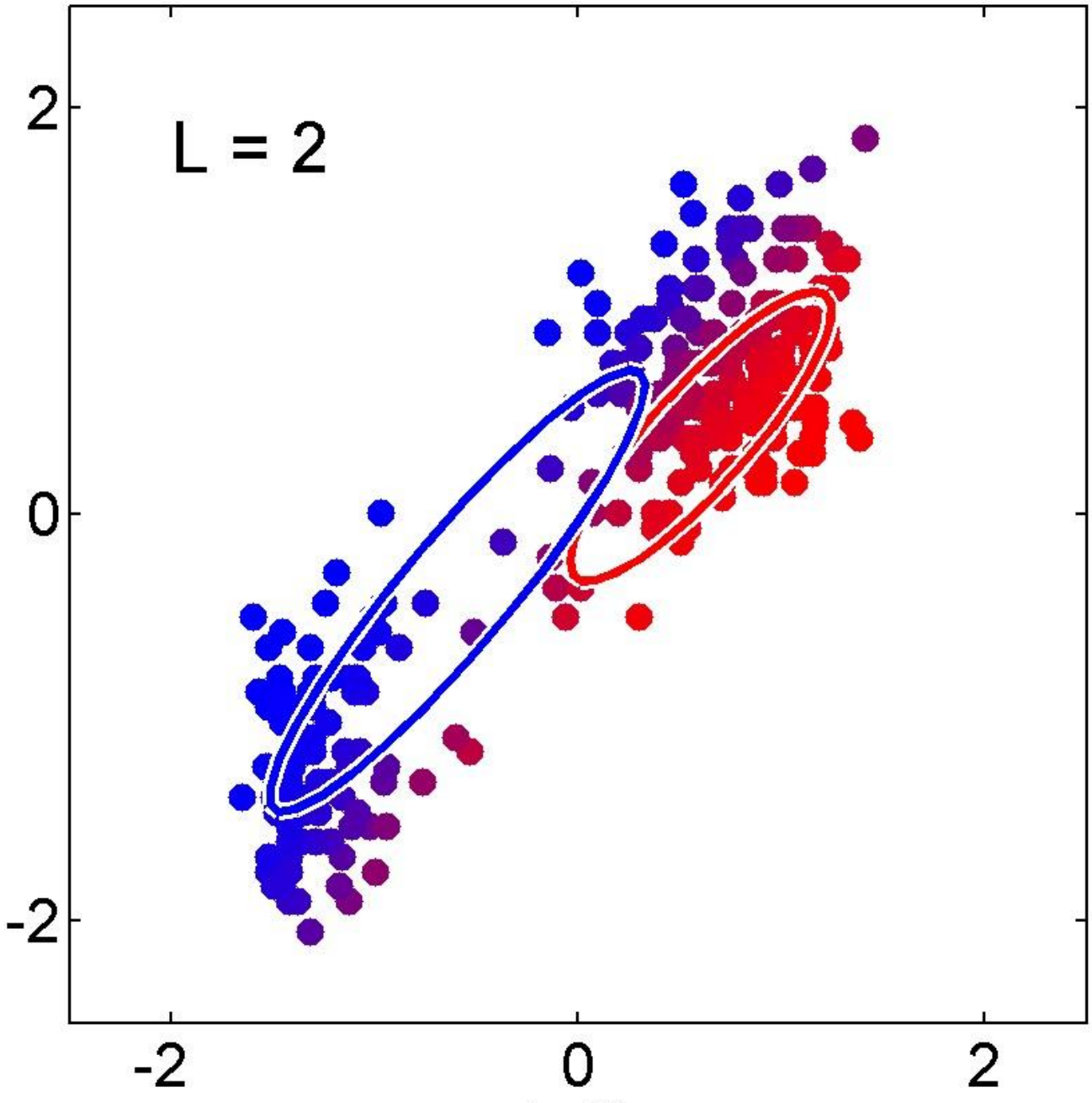
- The solutions are not closed form since they are coupled
- Suggests an iterative scheme for solving them:
  - Make initial guesses for the parameters
  - Alternate between the following two stages:
    1. E-step: evaluate responsibilities
    2. M-step: update parameters using ML results

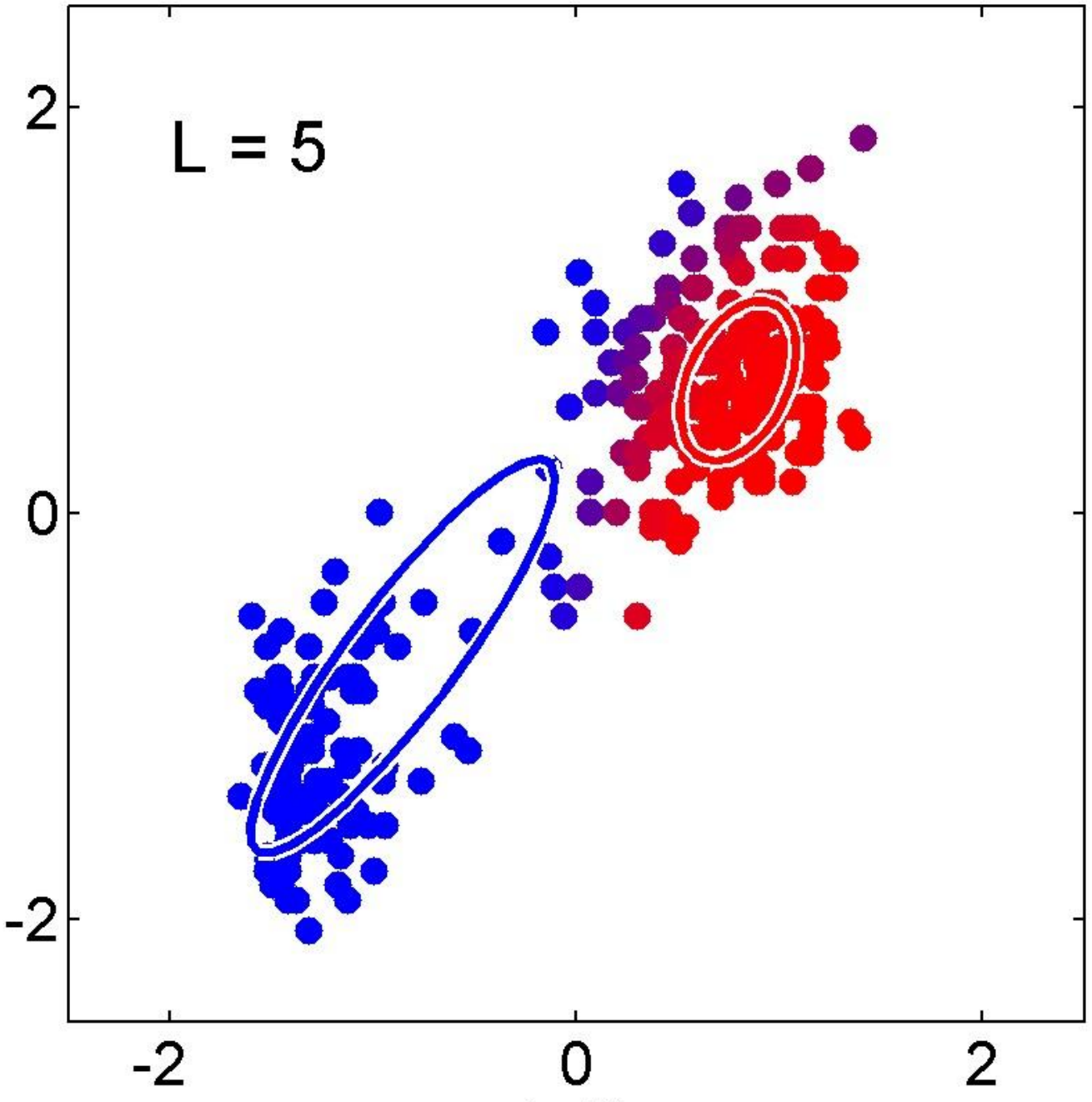
Slides credit: C. M. Bishop

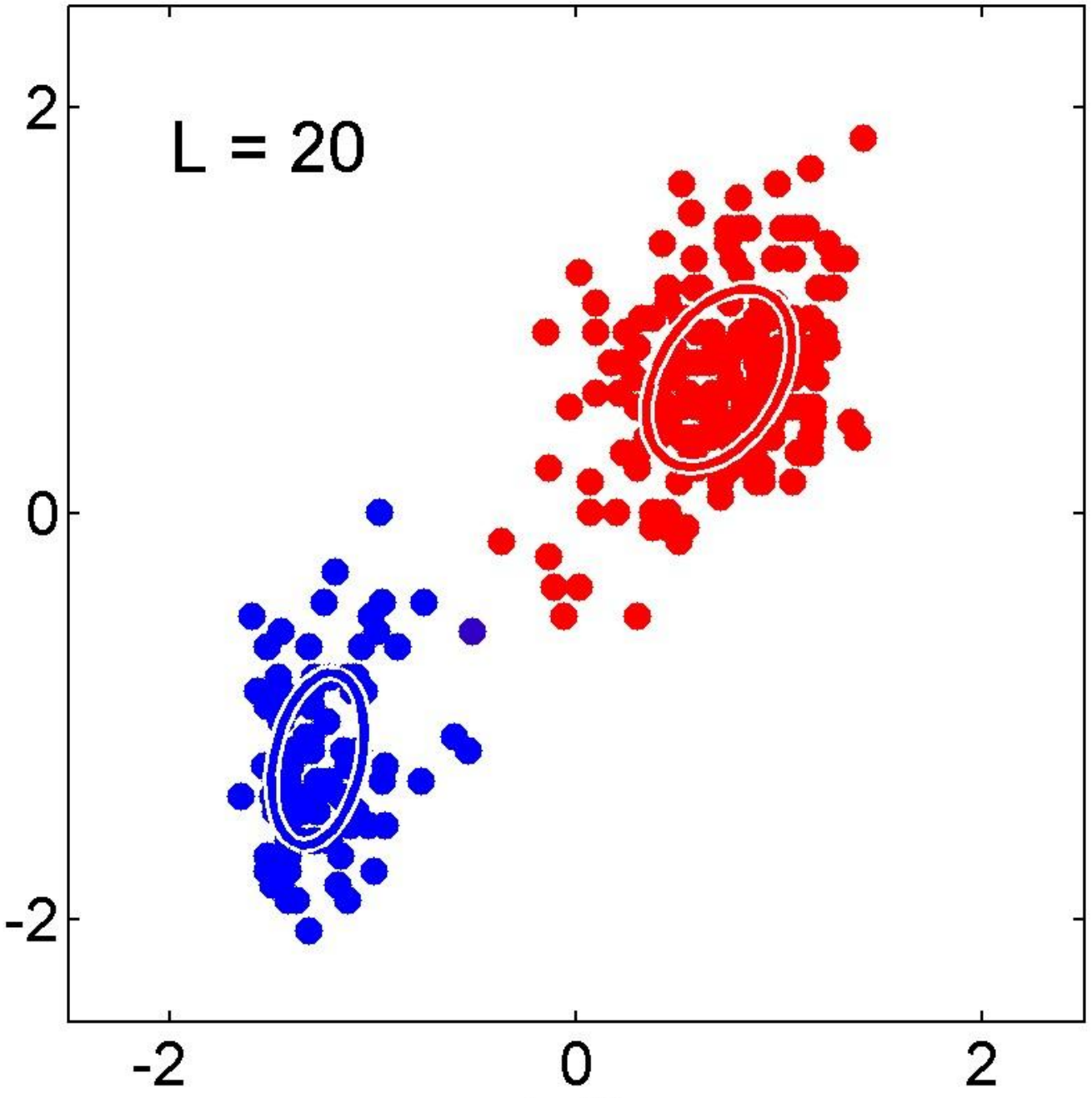

2
0
-2
-2
0
2


2
0
-2
-2
0
2


L = 1
2
0
-2
-2
0
2

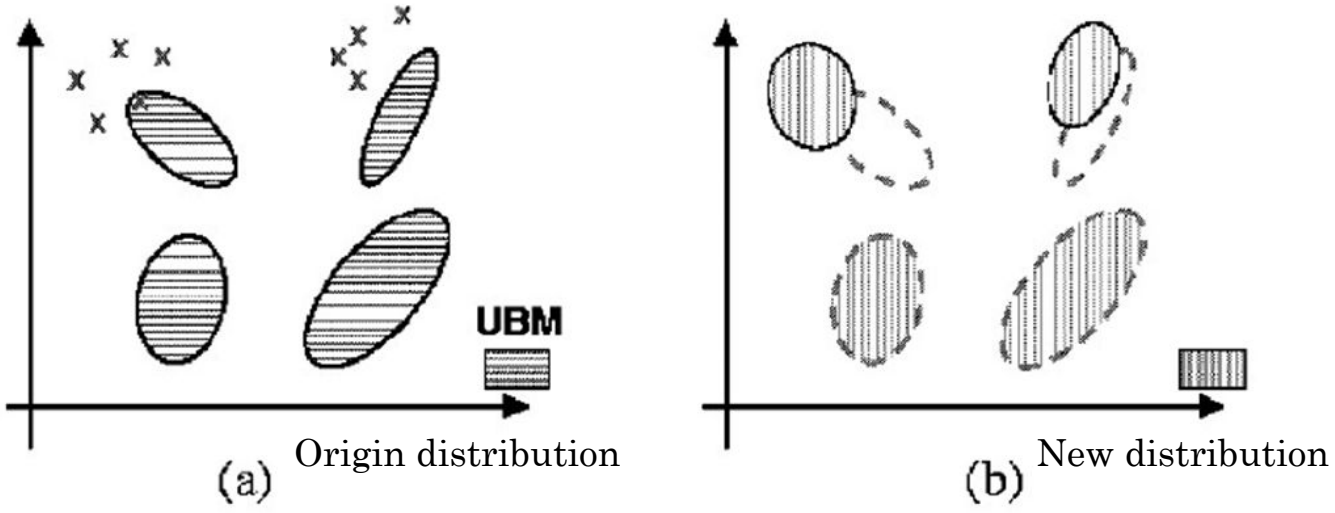

L = 2
2
0
-2
-2
0
2


L = 5
2
0
-2
-2
0
2


L = 20
2
0
-2
-2
0
2

The supervector representation (1)

- Given a set of features from a set of images, train a Gaussian mixture model. This is called a Universal Background Model.
- Given the UBM and a set of features from a single image, adapt the UBM to the image feature set by Bayesian EM (check equations in paper below).



Zhou et al, "**A novel Gaussianized vector representation for natural scene categorization**", **ICPR2008**

The supervector representation (2)

- Supervector:

$$[\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_K]$$

- Normalized supervector:

$$[\sqrt{\omega_1}\Sigma_1^{-1/2}\mathbf{m}_1, \sqrt{\omega_2}\Sigma_2^{-1/2}\mathbf{m}_2, \ldots, \sqrt{\omega_K}\Sigma_K^{-1/2}\mathbf{m}_K]$$

- Centralized and normalized supervector

$$\left[\sqrt{\omega_1}\Sigma_1^{-\frac{1}{2}}(\mathbf{m}_1 - \boldsymbol{\mu}_1), \sqrt{\omega_2}\Sigma_2^{-1/2}(\mathbf{m}_2 - \boldsymbol{\mu}_2), \ldots, \sqrt{\omega_K}\Sigma_K^{-\frac{1}{2}}(\mathbf{m}_K - -\boldsymbol{\mu}_K)\right]$$

Zhou et al, "**A novel Gaussianized vector representation for natural scene categorization**", **ICPR2008**