Advanced Video Content Analysis and Video Compression (5LSH0), Module 8

Semantic-level content analysis and classification I

Sveta Zinger

Video Coding and Architectures Research group, TU/e
(s.zinger@tue.nl)

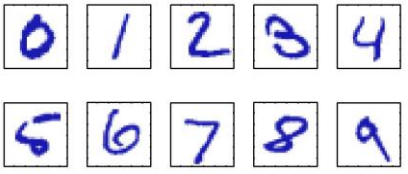
Introduction – (1)

* **Example: how to recognize handwritten digits automatically?**
  - We want to build a machine with
    - image of a digit as input
    - identity of the digit (0,…,9) as output
  - Why is it difficult?
    - Wide variability of handwriting
    - Rules or heuristics do not work

(The slides are based on "Pattern recognition and machine learning", Ch. Bishop)

Introduction – (2)

Examples of handwritten digits taken from US zip-codes



Introduction – (3)

* **Possible solution – machine learning**
  - Train the algorithm using a training set (digits) and target vector (their identities)
  - Test it with a test set (new images of digits)
  - Generalization – ability to categorize new examples correctly
* **How can we facilitate pattern recognition?**
  - Preprocess data in the training set – extract features (see module 4)

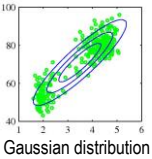
Introduction – (4)

* **Supervised learning**
  - Training data consists of input vectors and target vectors
  - Classification - assign each input vector to one of a finite number of discrete categories
* **Unsupervised learning**
  - No target vector in the input data
  - Clustering – discover groups of similar examples within the data

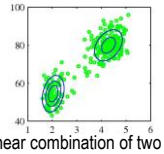
Mixture models and EM

Gaussian mixture – (1)

* **Gaussian distribution**
  - has some important analytical properties
  - but suffers from limitations when modeling real data sets



Single Gaussian distribution fails to capture the nature of data



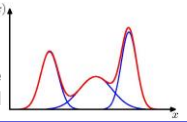
Linear combination of two Gaussians fits better

Mixture models and EM

Gaussian mixture – (2)

* **Mixture distributions**
  - linear combinations of basic distributions
* **To approximate almost any continuous density**
  - use sufficient number of Gaussians
  - adjust means, covariances, coefficients in the linear combination

Three Gaussians in blue and their sum in red



Mixture models and EM

Gaussian mixture – (3)

Superposition of K Gaussian densities

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

is called a mixture of Gaussians, where

π_k - mixing coefficients, $0 \le \pi_k \le 1$

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

Each Gaussian density has its mean μ_k and covariance Σ_k

Mixture models and EM

Gaussian mixture – (4)

* **Gaussian mixture distribution is governed by parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$**

$$\boldsymbol{\pi} = \{\pi_1, ..., \pi_K\}, \quad \boldsymbol{\mu} = \{\mu_1, ..., \mu_K\}, \quad \boldsymbol{\Sigma} = \{\Sigma_1, ..., \Sigma_K\}$$

* **How can we find these parameters?**
  - Possible solution – use maximum likelihood
  - Likelihood function expresses how probable the observed data is for a given set of parameters

Mixture models and EM

Gaussian mixture – (5)

* **Log of the likelihood function:**

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln\left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)\right)$$

$$\mathbf{X} = \{x_1, ..., x_N\}$$

  - No easy analytical solution
  - Expectation-maximization (EM) can be used

Mixture models and EM

Gaussian mixture – (6)

* **Where are Gaussian mixture models used?**
  - Data mining
  - Pattern recognition
  - Machine learning
  - Statistical analysis
* **How are their parameters determined?**
  - Maximum likelihood using the EM algorithm

K-means clustering – (1)

* **Problem: identify groups (clusters) of data points in multidimensional space**
  - we have a data set $\{x_1, ..., x_N\}$,
  - variable x - D-dimensional
  - goal: partition data into K clusters, value of K is given
* **Intuitive definition of cluster**
  - group of data points whose inter-point distances are small compared with the distances to points outside of the cluster

K-means clustering – (2)

* **Distortion measure**

$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|x_n - \mu_k\|^2$$

  - where $r_{nk} \in \{0,1\}$ - binary indicator variable: $r_{nk} = 1$ if data point x_n is assigned to cluster k and $r_{nk} = 0$ otherwise,
    x_n - data point,
    μ_k - vector assigned to cluster k (center of cluster)
  - it is sum of the squares of the distances of each data point to its assigned vector

K-means clustering – (3)

* **Goal: find values for $\{r_{nk}\}$ and $\{\mu_k\}$ that minimize J**
* **How can we find the solution?**
  - Iterative procedure
    - each iteration involves two successive steps
    - successive optimizations with respect to $\{r_{nk}\}$ and $\{\mu_k\}$
    - repeat until convergence
      - no further change in the assignments
      - or until a maximum number of iterations is exceeded

K-means clustering – (4)

* **Description of algorithm**
  - Choose some initial values for the $\{\mu_k\}$
  - First phase
    - Minimize J with respect to $\{r_{nk}\}$ keeping $\{\mu_k\}$ fixed
  - Second phase
    - Minimize J with respect to $\{\mu_k\}$ keeping $\{r_{nk}\}$ fixed
  - Repeat until convergence

K-means clustering – (5)

* **First phase of algorithm**
  - Determine $\{r_{nk}\}$ - assign data points to clusters
    - Optimize for each n separately

$$r_{nk} = \begin{cases} 1 & if \quad k = \arg\min_j \|x_n - \mu_j\|^2 \\ 0 & otherwise \end{cases}$$

K-means clustering – (6)

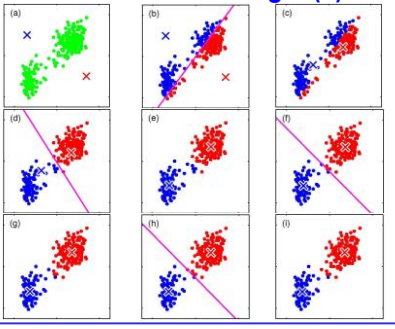
* **Second phase of algorithm**
  - Determine $\{\mu_k\}$ - compute the cluster means
    - J is a quadratic function of $\{\mu_k\}$, set its partial derivative to zero for finding its minimum

$$\frac{dJ}{d\mu_k} = 2\sum_{n=1}^{N} r_{nk}(x_n - \mu_k) = 0$$

then $$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

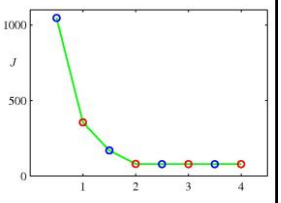
K-means clustering – (7)

Example



K-means clustering – (8)

* **Example: minimization of cost function *J***
  - Blue points – after assigning data points to clusters
  - Red points – computing cluster means
  - Algorithm converges after the third step, final cycle produces no changes



K-means clustering – (9)

* **What are the limits of this algorithm?**
  - Algorithm is based on Euclidean distance as the measure of dissimilarity between a data point and a prototype vector
    - ⇒ Data types are limited (for example, categorical labels cannot be used)
    - ⇒ determination of the cluster is not robust to outliers

K-means clustering – (10)

* ***K*-medoids algorithm**
  - Generalization of the K-means
  - Introduces a more general dissimilarity measure $\mathcal{V}$
  - The distortion measure to minimize is then

$$\tilde{J} = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\mathcal{V}(x_n, \mu_k)$$

  - But the computation of centers of clusters is more complicated

K-means clustering – (11)

* **Property of *K*-means algorithm**
  - Every data point is assigned uniquely to one of the clusters
  - but some data points lie roughly midway between cluster centers
  - and it is not clear that the hard assignment to the nearest cluster is most appropriate
* **What kind of assignment would be better?**
  - Adopt a probabilistic approach => soft assignments

K-means clustering
Image segmentation and compression – (1)

* **Some applications of *K*-means algorithm**
  - Image segmentation
  - Image compression
* **What is the goal of segmentation?**
  - Partition an image into regions each of which
    - has a reasonably homogeneous visual appearance or
    - corresponds to objects or parts of object

K-means clustering
Image segmentation and compression – (2)

* **Segmentation**
  - Each pixel is a separate {R,G,B} 3D data point
  - Apply K-means to these points
  - Redraw the image replacing each pixel vector with the {R,G,B} intensity triplet given by the center $\boldsymbol{\mu}_k$ to which this pixel is assigned

K-means clustering

Image segmentation and compression – (3)

Example: for a given value of K, the algorithm represents the image using a palette of only K colors

Smaller values of K => higher compression => poorer image quality

K-means clustering

Image segmentation and compression – (4)

Example

K-means is not a sophisticated approach to image segmentation because it takes no account of the spatial proximity of different pixels

K-means clustering

Image segmentation and compression – (5)

- **Application of the *K*-means to lossy data compression**
  - For each of the N data points, store only the identity k of the cluster to which it is assigned
  - Store the values of the K cluster centers
    - Requires less data provided that $K<N$
  - Each data point is approximated by its nearest center

It is called vector quantization; cluster centers are called code-book vectors

Mixtures of Gaussians – (1)

- **Earlier – Gaussian mixture model introduced as a simple linear superposition of Gaussian components**
  - Provides a richer class of density models than a single Gaussian
- **Now – Formulate Gaussian mixture in terms of discrete latent (hidden, unobserved) variables**
  - Provides a deeper insight in this distribution
  - Motivates the Expectation-Maximization (EM) algorithm

Mixtures of Gaussians – (2)

- **Gaussian mixture distribution**

$$p(x)=\sum_{k=1}^{K}\pi_k\mathcal{N}\left(x|\mu_\kappa,\Sigma_\kappa\right)$$

- **Introduce a *K*-dimensional binary random variable z**
  - One-of-K representation: particular element $z_k=1$ and all other elements are equal to 0
  - $z_k\in\{0,1\},\quad \sum_k z_k=1$

Mixtures of Gaussians – (3)

- **Marginal distribution over z**

$$p(z_k=1)=\pi_k,\quad 0\le\pi_k\le 1,\quad \sum_{k=1}^{K}\pi_k=1$$

- **Conditional distribution of x given a particular value for z is a Gaussian**

$$p\left(x|z_k=1\right)=\mathcal{N}\left(x|\mu_k,\Sigma_k\right)$$

Mixtures of Gaussians – (4)

* **Joint distribution is** $p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$
* **Marginal distribution of x is a sum of the joint distribution over all possible states of z**

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(x|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k \Sigma_k)$$

  - It is a Gaussian mixture
  - For every observed data point there is a latent variable

Mixtures of Gaussians – (5)

* **Another quantity – conditional probability of *z* given *x***
  - Using Bayes' theorem $p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$
  - we find

$$\gamma(z_k) \equiv p(z_k = 1|x) = \frac{p(z_k = 1)p(x|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(x|z_j = 1)} = \frac{\pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x|\mu_j, \Sigma_j)}$$

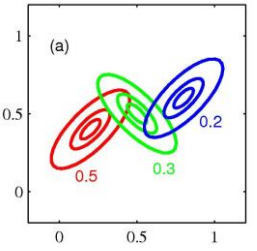
Mixtures of Gaussians – (6)

* **We view** π_k **as the posterior probability of** z_k=1
* **and** $\gamma(z_k)$ **– as the corresponding posterior probability once we have observed** x
* $\gamma(z_k)$ **will also be viewed as the responsibility that the component** k **takes for "explaining" the observation** x

Mixtures of Gaussians – (7)

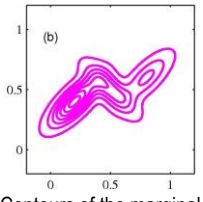
Example: mixture of 3 Gaussians in a two-dimensional space

Contours of constant density for each of the mixture components;
the 3 components are red, blue and green, and the values of the mixing coefficients are shown below each component

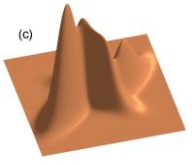


Mixtures of Gaussians – (8)

Example: mixture of 3 Gaussians in a two-dimensional space



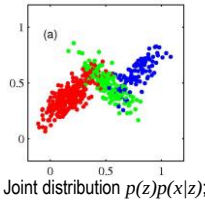
Contours of the marginal probability density $p(x)$



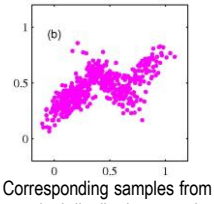
Surface plot of the distribution $p(x)$

Mixtures of Gaussians – (9)

Example: 500 points drawn from the shown above mixture of 3 Gaussians



Joint distribution $p(z)p(x|z)$; the three states of z are red, green and blue

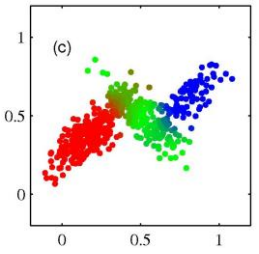


Corresponding samples from the marginal distribution $p(x)$; just x values are plotted

Mixtures of Gaussians – (10)

Example: 500 points drawn from the shown above mixture of 3 Gaussians

Colors represent the responsibilities $\gamma(z_{nk})$ associated with data point x_n, obtained by plotting the corresponding point using proportions of red, blue and green given by responsibilities



Mixtures of Gaussians
Maximum likelihood – (1)

Log of the likelihood function

$$\ln p(\mathbf{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln\left\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)\right\}$$

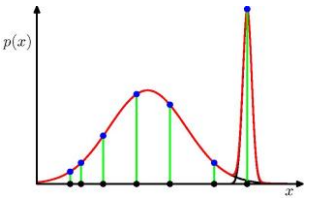
* **Problem with maximum likelihood framework applied to Gaussian models**
  - It is not a well posed problem
  - Singularities may occur

Mixtures of Gaussians
Maximum likelihood – (2)

* **What is a well posed problem?**
* **Problem is well posed according to Hadamard when**
  - A solution exists
  - The solution is unique
  - The solution depends continuously on the data (a small change in the data causes only a small change in the solution)

Mixtures of Gaussians
Maximum likelihood – (3)

* **Singularities**
  - occur when one of the Gaussian components "collapses" onto a specific data point
  - are an example of a severe over-fitting that can occur in a maximum likelihood approach



Mixtures of Gaussians
Maximum likelihood – (4)

* **Maximizing the log likelihood function for a Gaussian mixture model**
  - more complex problem than for a single Gaussian
    - no easy analytical solution
  - although gradient-based techniques are feasible
  - we now consider an alternative approach – EM algorithm

Mixtures of Gaussians
EM for Gaussian mixtures – (1)

* **Expectation-Maximization (EM) algorithm**

  (Dempster et al., 1977; McLachlan and Krishnan, 1997)
  - method for finding maximum likelihood solutions for models with latent variables
  - has broad applicability
    - medical image reconstruction
    - natural language processing
    - …

Mixtures of Gaussians
EM for Gaussian mixtures – (2)

* **Find the conditions for optimum of the likelihood functions**
  - Set the partial derivatives of $\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}$ to zero
  - For the means we get

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \quad \text{where} \quad N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

Mixtures of Gaussians
EM for Gaussian mixtures – (3)

  - For covariance

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$

  - For the mixing coefficients $\pi_k = \frac{N_k}{N}$

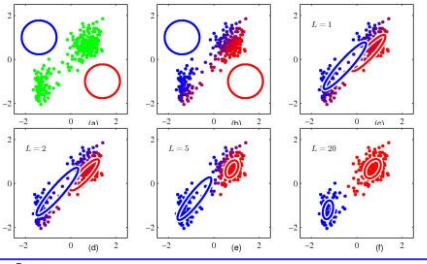
* **These results constitute no closed-form solution for the parameters of the mixture model**
  - Responsibilities $\gamma(z_{nk})$ depend on these parameters in a complex way

Mixtures of Gaussians
EM for Gaussian mixtures – (4)

* **EM algorithm – iterative scheme for finding a solution to the maximum likelihood problem**
  - Choose some initial values for the means, covariances and mixing coefficients
  - Expectation step (E step): use the current parameter values to evaluate posterior probabilities (responsibilities)
  - Maximization step (M step): estimate the means, covariances and mixing coefficients using the formulas above
  - Repeat until convergence

Mixtures of Gaussians
EM for Gaussian mixtures – (5)

Example: EM algorithm



Mixtures of Gaussians
EM for Gaussian mixtures – (6)

* **How can the results of K-means clustering be used in the EM algorithm?**
  - Run the K-means algorithm in order to find a suitable initialization of a Gaussian mixture model
  - Covariance matrices can be initialized as covariances of the clusters found by the K-means algorithm
  - Mixing coefficients can be set as fractions of data points assigned to the respective clusters

Mixtures of Gaussians
EM for Gaussian mixtures – (7)

* **EM algorithm: summary**
  - Given a Gaussian mixture model
  - The goal is to maximize the likelihood function with respect to the parameters (comprising the means and covariances of the components and the mixing coefficients)

Mixtures of Gaussians
EM for Gaussian mixtures – (8)

* **EM algorithm: main steps**
  - **1.** Initialize the means μ_k, covariances Σ_k and mixing coefficients π_k, and evaluate the initial value of the log likelihood
  - **2.** E step: evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j,\Sigma_j)}$$

Mixtures of Gaussians
EM for Gaussian mixtures – (9)

* **EM algorithm: main steps**
  - **3.** M step: Re-estimate the parameters using the current responsibilities

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$$

$$\pi_k^{new} = \frac{N_k}{N} \quad \text{where} \quad N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

Mixtures of Gaussians
EM for Gaussian mixtures – (10)

* **EM algorithm: main steps**
  - **4.** Evaluate the log likelihood

$$\ln p(\mathbf{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k) \right\}$$

  - and check the convergence of either the parameters or the log likelihood
  - If the convergence criterion is not satisfied, return to step 2

An Alternative View of EM
Relation to *K*-means – (1)

* **There is a close similarity between the *K*-means and EM algorithms**
  - *K*-means performs a hard assignment of data points to clusters => each data point is associated uniquely with one cluster
  - EM makes a soft assignment
  - *K*-means algorithm can be derived as a particular limit of EM for Gaussian mixtures

An Alternative View of EM
Relation to *K*-means – (2)

* **Consider a Gaussian mixture model with**
  - Covariance matrices of the mixture components are $\varepsilon \mathbf{I}$
  - ε – variance parameter that is shared by all the components, $\mathbf{I}$ – the identity matrix
  - Consider ε as a constant, and $\varepsilon \to 0$
  - EM algorithm for the mixture of Gaussians will lead to the following conclusions

An Alternative View of EM
Relation to *K*-means – (3)

* **Under the conditions mentioned above**
  - we obtain a hard assignment of data points to clusters, as in the *K*-means algorithm, $\gamma(z_{nk}) \to r_{nk}$
  - Each data point is assigned to the cluster with the closest mean
  - *K*-means algorithm does not estimate the covariances of the cluster but only the cluster means
  - A hard assignment version of the Gaussian mixture model with general covariance matrices is called elliptical *K*-means

An Alternative View of EM
Mixtures of Bernoulli distributions – (1)

* **So far – we focused on distributions of continuous variables described by mixtures of Gaussians**
* **Now – consider mixture of discrete binary variables**
  - described by Bernoulli distributions
  - This model is called latent class analysis

An Alternative View of EM
Mixtures of Bernoulli distributions – (2)

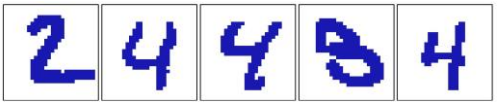
* **Bernoulli distribution**
  - Consider a set of D binary variables x_i, where $i = 1,\ldots,D$, each of which is governed by a Bernoulli distribution with parameter μ_i:

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{i=1}^{D} \mu_i^{x_i} (1-\mu_i)^{(1-x_i)}$$

where $\mathbf{x} = (x_1,\ldots, x_D)^T$ and $\boldsymbol{\mu} = (\mu_1,\ldots, \mu_D)^T$

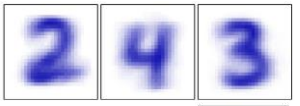
An Alternative View of EM
Mixtures of Bernoulli distributions – (3)

Example:
binary images of handwritten digits "2", "3" and "4"
The complete data set is 600 images

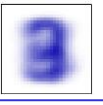


An Alternative View of EM
Mixtures of Bernoulli distributions – (4)

Example:
EM results for a three components mixture model;
μ parameters for each component of the mixture model



For comparison: single distribution



Example of EM
Image segmentation – (1)

- Model the joint distribution of color and texture features with a mixture of Gaussians
- Use EM to estimate the parameters of this model
- The resulting clusters provide the segmentation of the image

Example of EM
Image segmentation – (2)

Original and smoothed images

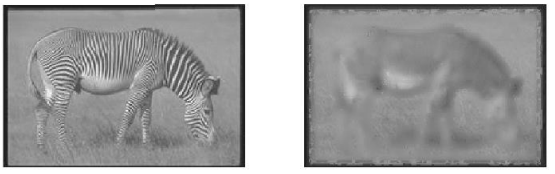


Image from Ch. Carson et al, Blobworld: image segmentation using expectation-maximization and its application to image querying, *IEEE Transactions on PAMI*, Vol. 24, Num. 8, August 2002

Example of EM
Image segmentation – (3)

Color (top row of images) and texture (bottom row) features

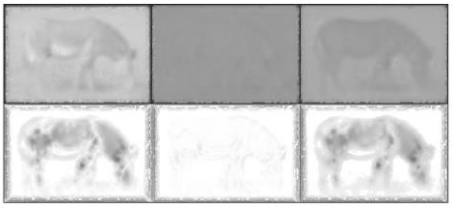


Image from Ch. Carson et al, Blobworld: image segmentation using expectation-maximization and its application to image querying, *IEEE Transactions on PAMI*, Vol. 24, Num. 8, August 2002

Example of EM
Image segmentation – (4)

The result of clustering the feature vectors into 2, 3, 4, 5 Gaussian clusters using EM

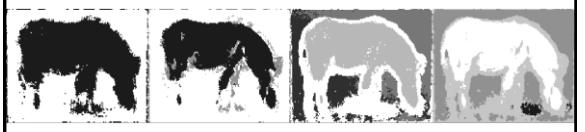


Image from Ch. Carson et al, Blobworld: image segmentation using expectation-maximization and its application to image querying, *IEEE Transactions on PAMI*, Vol. 24, Num. 8, August 2002

Principal Component Analysis
Introduction – (1)

* **So far – we discussed probabilistic models having discrete latent variables, such as mixture of Gaussians**
* **Now – explore models in which some or all of the latent variables are continuous**
  - Motivation: property of many data sets – data points can be represented by fewer dimensions than those in the original data space

Principal Component Analysis
Introduction – (2)

* **Consider an artificial data set**
  - constructed by taking images of digits, represented by 64x64 pixel grey-scale image,
  - and embedding them in larger images of size 100x100 by padding with pixels having the value 0
  - Location and orientation of digits is varied at random

Principal Component Analysis
Introduction – (3)

* **Synthetic data set**
  - multiple copies of digit images where the digit is randomly displaced and rotated within some larger image field

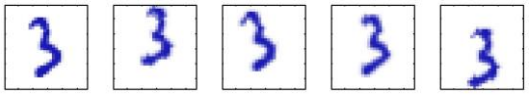


Image size: 100x100 = 10000 pixels

Principal Component Analysis
Introduction – (4)

* **Resulting images**
  - Represented by a point in the 10000-dimensional data space
  - However, across a data set of these images, there are only three degrees of freedom of variability
    - vertical translation
    - horizontal translation
    - rotation
  - Intrinsic dimensionality is three

Principal Component Analysis
Introduction – (5)

* **Translation and rotation parameters in this example**
  - latent variables
    - because we observe only the images and are not told which values of the translation or rotation variables where used to create them
  - Real digit image data => more degrees of freedom
    - Scaling, handwriting variability
  - but still smaller amount than the data set dimensionality

Principal Component Analysis
Introduction – (6)

* **Such latent variables can be used for**
  - data compression
  - density modeling
  - data modeling
    - select a point according to some latent variable distribution
    - generate an observed data point by adding noise, drawn from some conditional distribution of the data variables given the latent variables

Principal Component Analysis – (1)

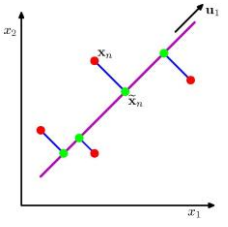
* **PCA (Principal Component Analysis) is widely used for**
  - dimensionality reduction
  - lossy data compression
  - feature extraction
  - data visualization
* **Also known as Karhunen-Loève transform**

Principal Component Analysis – (2)

* **Two definitions of PCA (give rise to the same algorithm)**
  - Orthogonal projection of the data onto a lower dimensional space (principal subspace), such that the variance of the projected data is maximized
  - Linear projection that minimizes the average projection cost, defined as the mean squared distance between the data points and their projections

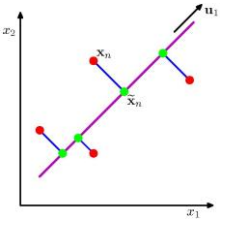
Principal Component Analysis – (3)

PCA seeks a space of lower dimensionality, denoted by the magenta line, such that the orthogonal projection of the data points (red dots) onto this subspace maximizes the variance of the projected points (green dots)



Principal Component Analysis – (4)

Alternative definition of PCA: minimize the sum of squares of the projection errors (blue lines)



Principal Component Analysis – (5)

* **PCA steps**
  - Evaluate the mean
  - and the covariance matrix of the data set
  - Find M eigenvectors of the covariance matrix that correspond to M largest eigenvalues
* **What are the eigenvalues and eigenvectors?**

Principal Component Analysis – (6)

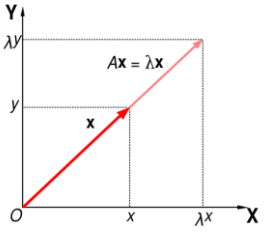
* **Eigenvalue and eigenvector**
  - Given a linear transformation $\boldsymbol{A}$, a non-zero vector $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{A}$ if it satisfies the eigenvalue equation $\mathbf{Ax} = \lambda \boldsymbol{x}$ for some scalar λ
  - The scalar λ is called eigenvalue of $\boldsymbol{A}$ corresponding to the eigenvector $\boldsymbol{x}$

(source: http://en.wikipedia.org/wiki/Eigenvector)

Principal Component Analysis – (7)

Geometrically the eigenvalue equation means that under the transformation $\boldsymbol{A}$ eigenvectors do not change their direction.

The eigenvalue λ is simply the amount of "stretch" or "shrink" to which a vector is subjected when transformed by $\boldsymbol{A}$.



(source: http://en.wikipedia.org/wiki/Eigenvector)

Principal Component Analysis
Applications – (1)

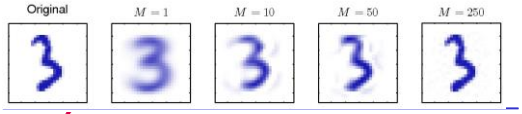
* **PCA approximation to a data vector x_n**

$$\tilde{x}_n = \bar{x} + \sum_{i=1}^{M} \left(x_n^T u_i - \bar{x}^T u_i \right) u_i$$

  - where $\bar{x}$ – mean of the data set,
  - u_i – eigenvectors of the covariance matrix for the original data set $\{x_n\}$
  - M – number of principal components

Principal Component Analysis
Applications – (2)

* **Example: PCA reconstruction obtained by retaining M principal components**
  - As M increases, the reconstruction becomes more accurate
  - It becomes perfect when $M = D = 28\text{x}28 = 784$



Principal Component Analysis
Applications – (3)

* **Example: PCA for human face recognition**
  - Eigenfaces – eigenvectors used for human face recognition
  - Obtained by PCA applied to a set of images of human faces

(source: http://en.wikipedia.org/wiki/Eigenvector)

Principal Component Analysis
Autoassociative neural networks – (1)

* **Neural networks for unsupervised learning**
  - can be used for dimensionality reduction
  - use the same number of inputs and outputs D and M hidden layers, with $M < D$
  - Input vectors and targets for training are the same => the network learns to map each vector onto itself by minimizing the error (w – network parameters, weights)

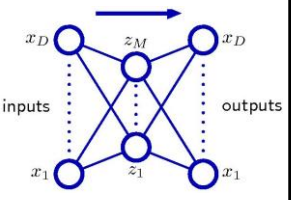
$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\|y(x_n, w) - x_n\|^2$$

Principal Component Analysis
Autoassociative neural networks – (2)

Autoassociative multilayer perceptron with two layers of weights

Network is trained to map input vectors onto themselves by minimization of sum-of-squares error

It is equivalent to PCA



Summary and conclusions – (1)

* **K-means clustering**
  - can be used for segmentation, unsupervised classification
  - simple, easy to implement
  - assigns one data point to one cluster, no soft assignment

Summary and conclusions – (2)

* **Mixture of Gaussians**
  - useful for modeling probability densities, allows flexibility necessary for it
  - parameters are estimated using EM algorithm
* **EM algorithm**
  - iterative, does not require a closed-form solution
  - estimates parameters of mixture models
  - can be used for segmentation (see slides 59-62)

Summary and conclusions – (3)

* **PCA**
  - provides principal components for a data set
  - successfully used for dimensionality reduction (see eigenfaces)
  - assumes high signal-to-noise ratio
    (large variance => important dynamics)

References

- Christopher M. Bishop, “Pattern Recognition and Machine Learning”, Springer, 2002
  - Chapter 9
  - Chapter 12