Probabilistic Latent Semantic Analysis (pLSA)

SS 2008 – Bayesian Networks

Multimedia Computing, Universität Augsburg
Rainer.Lienhart@informatik.uni-augsburg.de
www.multimedia-computing.{de,org}

References

- Thomas Hoffmann. **Unsupervised Learning by Probabilistic Latent Semantic Analysis**. *Machine Learning*, Vol. 42, Issue 1-2, pp. 177-196, 2001.

- D. Lowe. **Distinctive image features from scale invariant keypoints**. In IJCV 60(2):91-110, 2004
- K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. Van Gool, **A comparison of affine region detectors**. In IJCV 65(1/2):43-72, 2005.
- K. Mikolajczyk, C. Schmid, **A performance evaluation of local descriptors**. In PAMI 27(10):1615-1630.

- A. Bosch , A. Zisserman and X. Munoz. **Scene Classification via pLSA.** Proceedings of the European Conference on Computer Vision (2006).

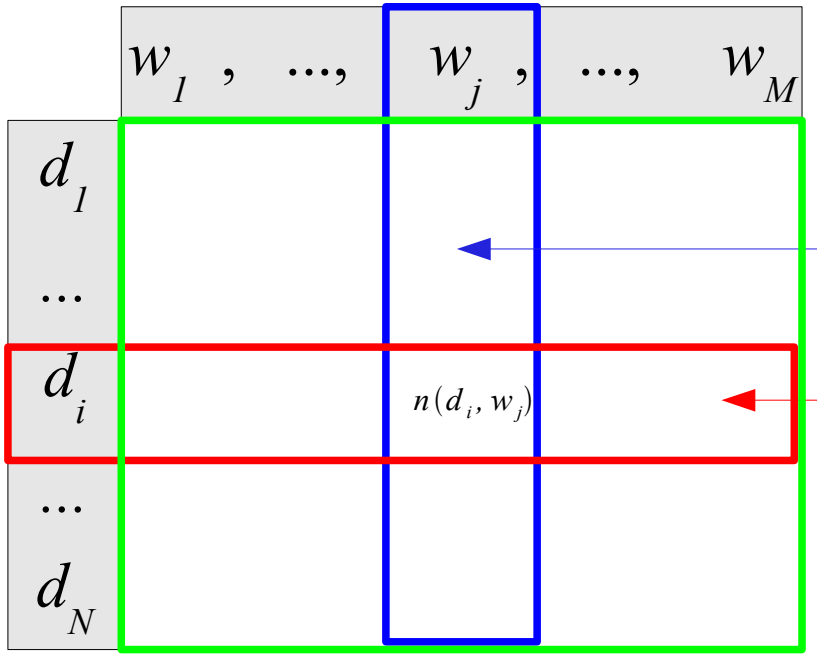
Definitions

Given

- a collection of text ~~documents~~ (users) $D = \{d_1, \ldots, d_N\}$
- with ~~terms from a vocabulary~~ $W = \{w_1, \ldots, w_M\}$ (movie ratings)
- in which we ignore the sequential order the words occur

==> data can be summarized by

- N x M ***co-occurrence* table** of counts $\boldsymbol{N} = (n(d_i, w_j))_{ij}$ (ratings: R = (r(u_i, m_j))_ij)
- where $n(d_i, w_i)$ == # of times term w_j occurend in document d_i



User-Item matrix

~~**Term-document matrix:**~~

Item vector

~~Term vector~~

User vector

~~Document vector~~

Probabilistic LSA (pLSA)

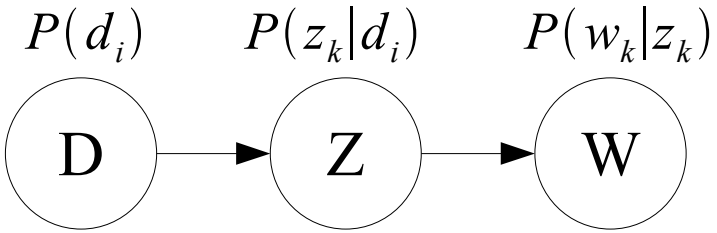
Assume we have a collection of ~~text documents~~ (user ratings) $D=\{d_1,\dots,d_N\}$ from which we have determined the ~~co-occurrence table of counts~~ (table of ratings) $\boldsymbol{N}=(n(d_i,w_j))_{ij}$ using the vocabulary $W=\{w_1,\dots,w_M\}$. Further assume that with each observation $\langle d_i,w_j\rangle$ an unobserved class variable $z_k\in\{z_1,\dots,z_K\}$ is associated. A class z_k can be regarded as a ~~concept a document talks about~~ (community a user belongs to). Every ~~document~~ (user) can ~~talk about~~ (belong to) multiple ~~concepts~~ (communities) to different extend.

Assume we have an urn consisting of 3-tupels $\langle d_i,z_k,w_j\rangle$ (u_i, m_j, z_k). Note that z_k is regarded as being unobservable.

Using these definitions, one may define a **generative model** (generative models approximate joint probabilities) for the **observation pair** $\langle d_i,w_j\rangle$ by the following scheme:

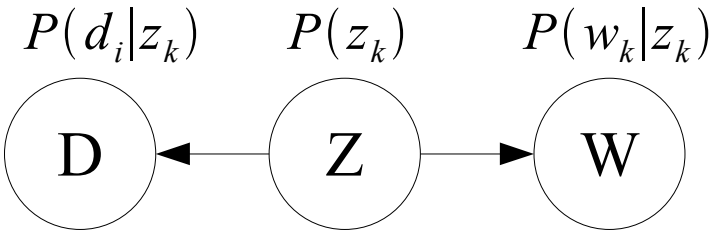
1. select a document d_i with probability $P(d_i)$ (user)
2. pick a latent class z_k with probability $P(z_k|d_i)$ P(z_k, u_i)
3. generate a ~~word~~ (movie) w_j with probability $P(w_j|z_k)$

BN - Model



$$P(d_i, w_j) = \sum_{k=1}^{K} P(z_k) P(d_i|z_k) P(w_j|z_k)$$

Markow equivalent



$$P(d_i, w_j) = \sum_{k=1}^{K} P(d_i) P(z_k|d_i) P(w_j|z_k)$$

pLSA (2)

$$P(d_i, w_j) = \sum_{k=1}^{K} P(d_i) P(z_k|d_i) P(w_j|z_k)$$

or

$$P(d_i, w_j) = P(d_i) P(w_j|d_i) \quad \text{with}$$

user -> latent variable -> movie

$$P(w_j|d_i) = \sum_{k=1}^{K} P(w_j, z_k|d_i) = \sum_{k=1}^{K} P(w_j|z_k) P(z_k|d_i)$$

Implicit conditional independence assumption: d_i and w_j are independent conditioned on the state of the associated latent variable z_k (sometimes called aspect z_k).

pLSA (3)

The goal of probabilistic latent semantic analysis (pLSA) is to learn the unobservable probabilities in a maximum likelihood framework and thus to infer the possible hidden "aspect".

$$L = \prod_{i=1}^{N} \prod_{j=0}^{M} P(d_i, w_j)^{n(d_i, w_j)}$$

$$\log L = \sum_{i=1}^{N} \sum_{j=0}^{M} n(d_i, w_j) \log P(d_i, w_j)$$

with

$$\log x^n = n \log x$$
$$\log(x\,y) = \log x + \log y$$
$$\log\left(\frac{x}{y}\right) = \log x - \log y$$

pSLA (4)

$$\log L = \sum_{i=1}^{N}\sum_{j=0}^{M} n(d_i, w_j)\log P(d_i, w_j)$$

$$= \sum_{i=1}^{N}\sum_{j=0}^{M} n(d_i, w_j)\log\Big[\sum_{k=1}^{K} P(d_i)P(z_k|d_i)P(w_j|z_k)\Big]$$

$$= \sum_{i=1}^{N}\sum_{j=0}^{M} \cancel{n(d_i, w_j)}\log\Big[P(d_i)\sum_{k=1}^{K} P(z_k|d_i)P(w_j|z_k)\Big]$$

$$= \sum_{i=1}^{N}\sum_{j=0}^{M} \cancel{n(d_i, w_j)}\Big[\log P(d_i)+\log\sum_{k=1}^{K} P(z_k|d_i)P(w_j|z_k)\Big]$$

$$= \sum_{i=1}^{N}\sum_{j=0}^{M} n(d_i, w_j)\log P(d_i)+n(d_i, w_j)\log\Big[\sum_{k=1}^{K} P(z_k|d_i)P(w_j|z_k)\Big]$$

$$= \sum_{i=1}^{N} n(d_i)\log P(d_i)+\frac{n(d_i)}{n(d_i)}\sum_{j=0}^{M} n(d_i, w_j)\log\Big[\sum_{k=1}^{K} P(z_k|d_i)P(w_j|z_k)\Big]$$

$$= \sum_{i=1}^{N} n(d_i)\Big[\log P(d_i)+\sum_{j=0}^{M}\frac{n(d_i, w_j)}{n(d_i)}\log\Big[\sum_{k=1}^{K} P(z_k|d_i)P(w_j|z_k)\Big]\Big]$$

Expectation Maximization (EM)

EM consists of two steps:

1. **E-step**: Calculate expectation (= posterior probabilities) for latent variables given the observations by using the current estimates of the parameters
2. **M-step**: Update parameters such that the data log-likelihood (log L) increases using the posterior probabilities in the E-step

E-step:

- $$P(z_k|d_i, w_j) = \frac{P(w_j, z_k|d_i)}{P(w_j|d_i)} = \frac{P(w_j|z_k, d_i)P(z_k|d_i)}{P(w_j|d_i)} = \frac{P(w_j|z_k)P(z_k|d_i)}{\sum_{l=1}^{K} P(w_j|z_l)P(z_l|d_i)}$$

M-step:

- $P(d_i), n(d_i), n(d_i, w_j)$ can directly be estimated from the data
- In order to maximize log L:
  - $$P(w_j|z_k) = \frac{\sum_{i=1}^{N} n(d_i, w_j)P(z_k|d_i, w_j)}{\sum_{m=1}^{M}\sum_{i=1}^{N} n(d_i, w_m)P(z_k|d_i, w_m)}, \quad P(z_k|d_i) = \frac{\sum_{j=1}^{M} n(d_i, w_j)P(z_k|d_i, w_j)}{n(d_i)}$$
- The E-step and M-step must be applied until convergence condition is met.

use log L to determine termination

Example: Word Usage Analysis (1)

- Topic Detection and Tracking corpus (TDT1)
  - http://www.ldc.upenn.edu/TDT
  - Approx. 7 million words
  - 15863 documents
- K = 128

- Display 2 most probable aspects that generate the term 'flight' and 'love'

Example: Word Usage Analysis (2)

Aspect 1	Aspect 2	Aspect 3	Aspect 4
plane	space	home	film
airport	shuttle	family	movie
crash	mission	like	music
flight	astronauts	love	new
safety	launch	kids	best
aircraft	station	mother	hollywood
air	crew	life	love
passenger	nasa	happy	actor
board	satellite	friends	entertainment
airline	earth	cnn	star

Figure 5. The 2 aspects to most likely generate the word 'flight' (left) and 'love' (right), derived from a $K = 128$ aspect model of the TDT1 document collection. The displayed terms are the most probable words in the class-conditional distribution $P(w_j \mid z_k)$, from top to bottom in descending order.

Example: Word Usage Analysis (3)

Aspect 1	Aspect 2	Aspect 3	Aspect 4
imag	video	region	speaker
SEGMENT	sequenc	contour	speech
textur	motion	boundari	recogni
color	frame	descrip	signal
tissu	scene	imag	train
brain	SEGMENT	SEGMENT	hmm
slice	shot	precis	sourc
cluster	imag	estim	speakerindepend
mri	cluster	pixel	SEGMENT
algorithm	visual	paramet	sound

Figure 3. The 4 aspects to most likely generate the word 'segment', derived from a $K = 128$ aspect model of the CLUSTER document collection. The displayed word stems are the most probable words in the class-conditional distribution $P(w_j \mid z_k)$, from top to bottom in descending order.

1586 documents on clustering

Ex: Image Classification Model

Metaphor:

- Image ≡ documents
- Object categories ≡ topics
  - (e.g, human, grass, houses, etc.)
- Visual patches ≡ visual words
  - Visual words are local descriptors --> vectors quantizing color, texture and SIFT features like region descriptors

=> an image with multiple objects is modeled as a mixture of topics

Excursion: Sparse Salient Points and Their Local Invariant Features

Most parts are taken from
David Lowe's CVPR'03 Tutorial

Objective

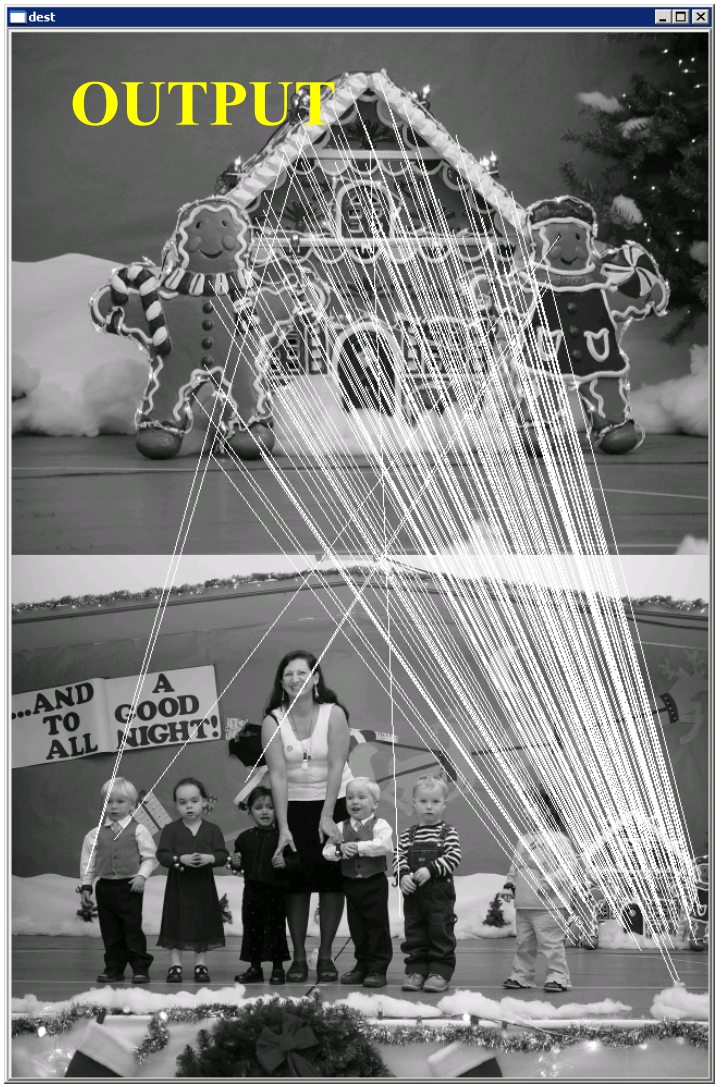
- Determine highly distinctive points in a scene such that they can be repeatedly found under a wide range of circumstances. Commonly required properties are:

- Invariant to
  - image scaling and rotation
  - Additive and multiplicative illumination changes

- Partially invariant to change in
  - illumination (in general)
  - 3D camera viewpoint
  - Additive noise

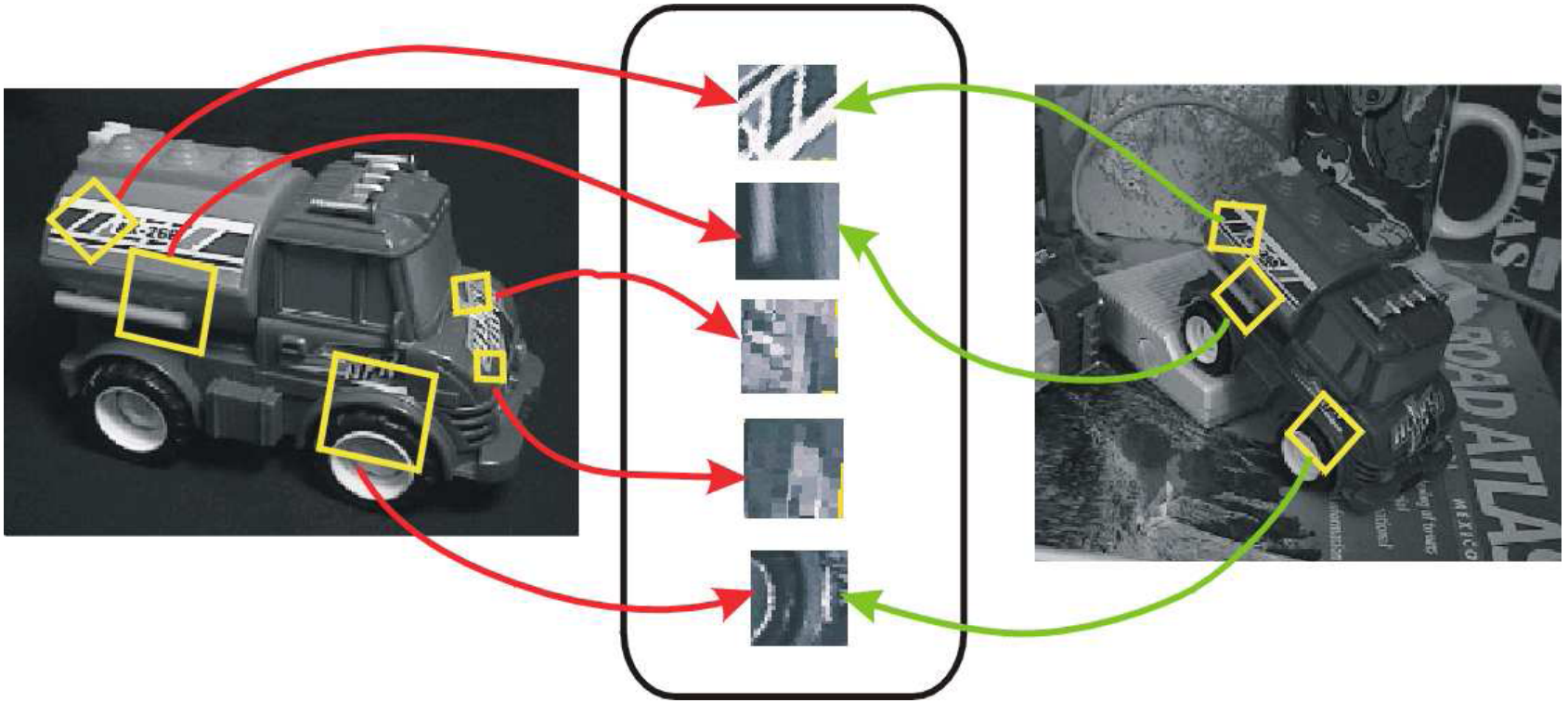
- Robust towards clutter, noise, or partial occlusion

Example



Invariant Local Features

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



Advantages of Invariant Local Features

- **Locality**: features are local → robust to occlusion and clutter (no prior segmentation)
- **Distinctiveness**: individual features can be matched to a large database of objects
- **Quantity**: many features can be generated for even small objects
- **Efficiency**: close to real-time performance
- **Extensibility**: can easily be extended to wide range of differing feature types, with each adding robustness

Scale Invariance

Requires a method to repeatably select points in locations and scale:

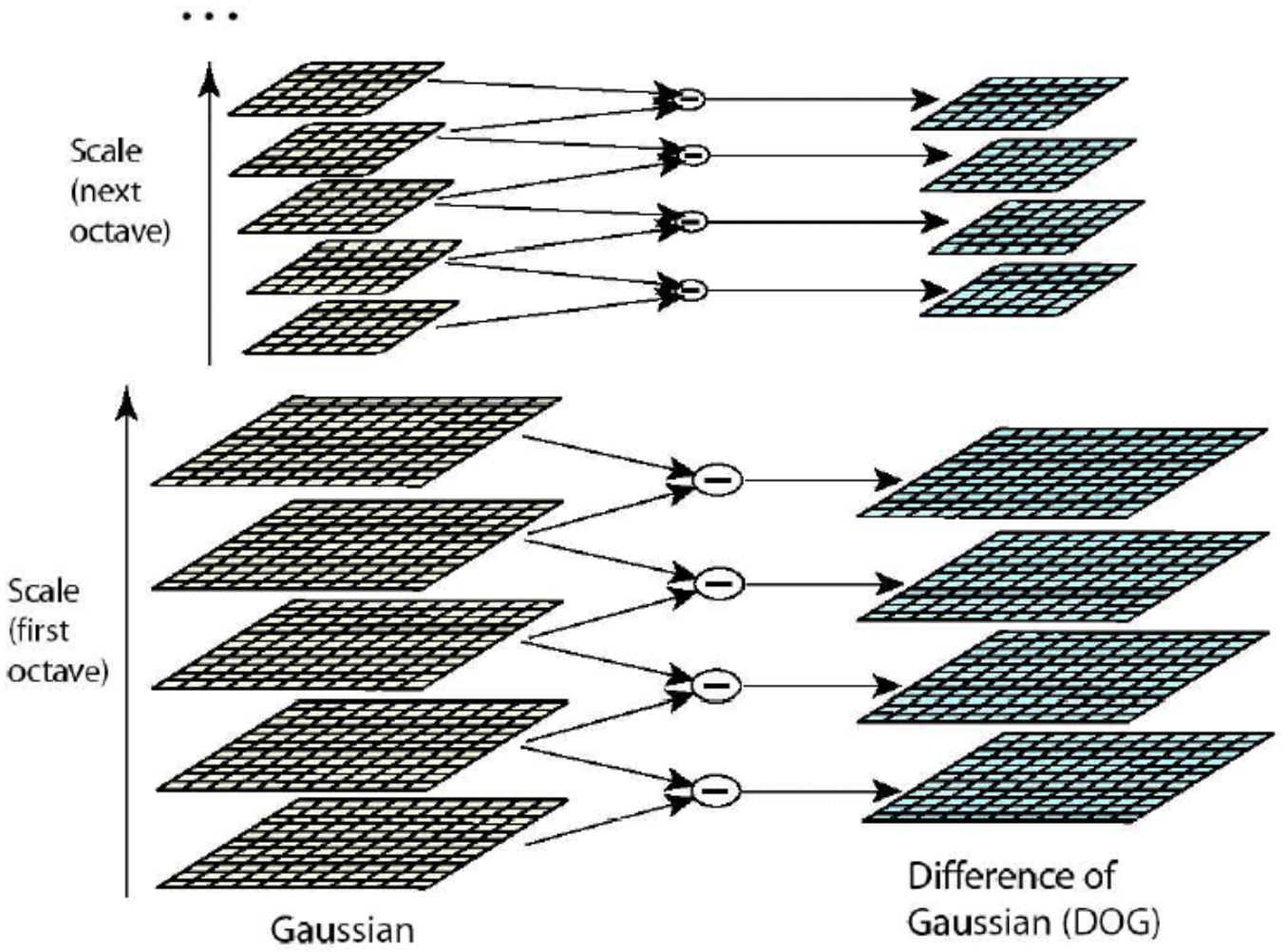
- Use scale-space with Gaussian kernel (Koenderink, 1984; Lindeberg, 1994)

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$

- An efficient choice is to detect peaks in the difference of Gaussian pyramid (Burt & Adelson, 1983; Crowley & Parker, 1984)

$$D(x,y,\sigma)=L(x,y,k\sigma)-L(x,y,\sigma)$$

Scale Space: One Octave at a Time



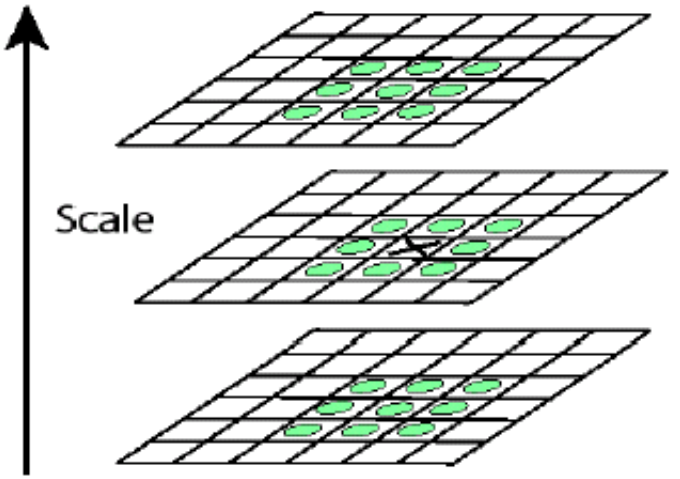
Key Point Localization

- Detect maxima and minima of difference-of-Gaussian in scale space
- Fit a quadratic to surrounding values for sub-pixel and sub-scale interpolation (Brown & Lowe, 2002)
- Taylor expansion around point:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^\mathbf{T}\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$$

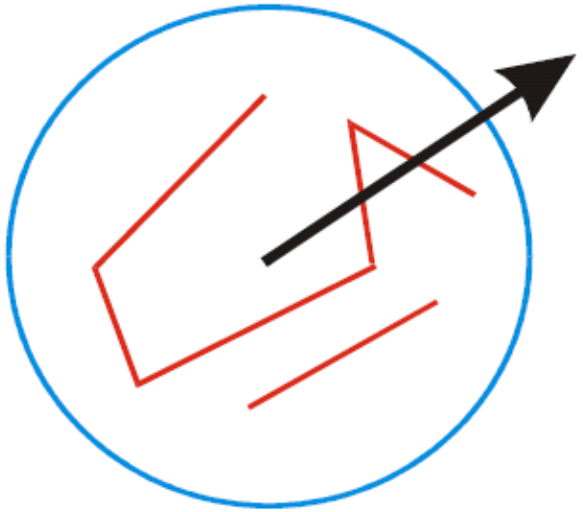
- Offset of extremum (use finite differences for derivatives):

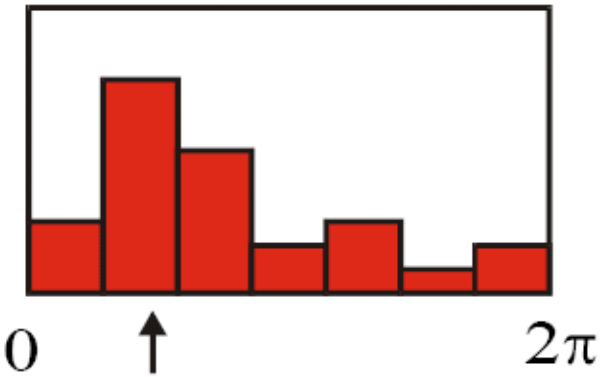
$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1}\frac{\partial D}{\partial \mathbf{x}}$$



Select Canonical Orientation

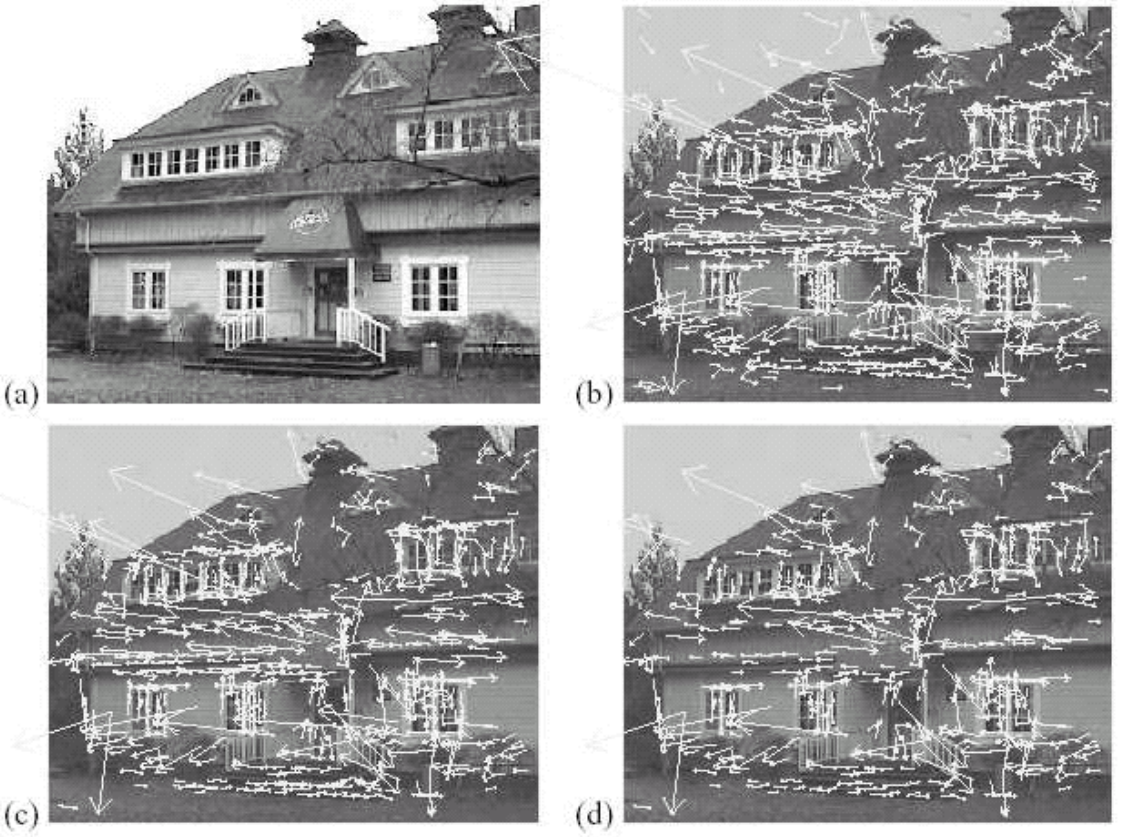
- Create histogram of local gradient directions computed at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale, orientation)





Example of Keypoint Detection

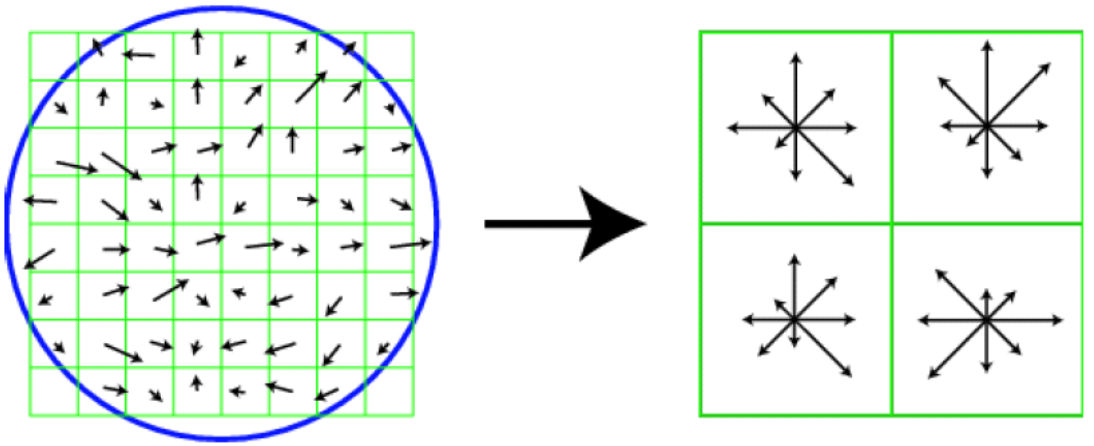
Threshold on value at DOG peak and on ratio of principle curvatures (Harris approach)



(a) 233x189 image
(b) 832 DOG extrema
(c) 729 left after peak value threshold
(d) 536 left after testing ratio of principle curvatures

SIFT Vector Formation

- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions



Excursion:
K-Means Clustering

Problem Statement

- Given:
  - A set of *n* data points $X=\{x_1, ..., x_n\}$ in *d*-dimensional space R^d and an integer *k*
- Goal
  - Find set of *k* points $C=\{c_1, ..., c_k\}$ (called center points) in *d*-dimensional space R^d that minimizes the mean squared distance from each data point to its nearest center point

$$(c_1, c_2, \ldots, c_k) \leftarrow arg \min_{(c_1, \ldots, c_k)} \sum_{i=1}^{n} [\min_{k} \|x_i, c_k\|]$$

- No exact polynomial-time algorithms are known for this problem
- Demo at http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html

k-Means Algorithm

1. Choose *k* initial center points randomly in the *d*-dimensional space
2. Cluster data using Euclidean distance (or other distance metric)
3. Calculate new center points for each cluster using only points within the cluster
4. Re-Cluster all data using the new center points

   This step could cause data points to be placed in a different cluster
5. Repeat steps 3 & 4 until the center points have moved such that in step 4 no data points are moved from one cluster to another or some other convergence criteria is met

Resume:
Latent Semantic Analysis

SS 2006 – Media Mining 2

Multimedia Computing, Universität Augsburg
Rainer.Lienhart@informatik.uni-augsburg.de
www.multimedia-computing.{de,org}

Ex: Image Classification

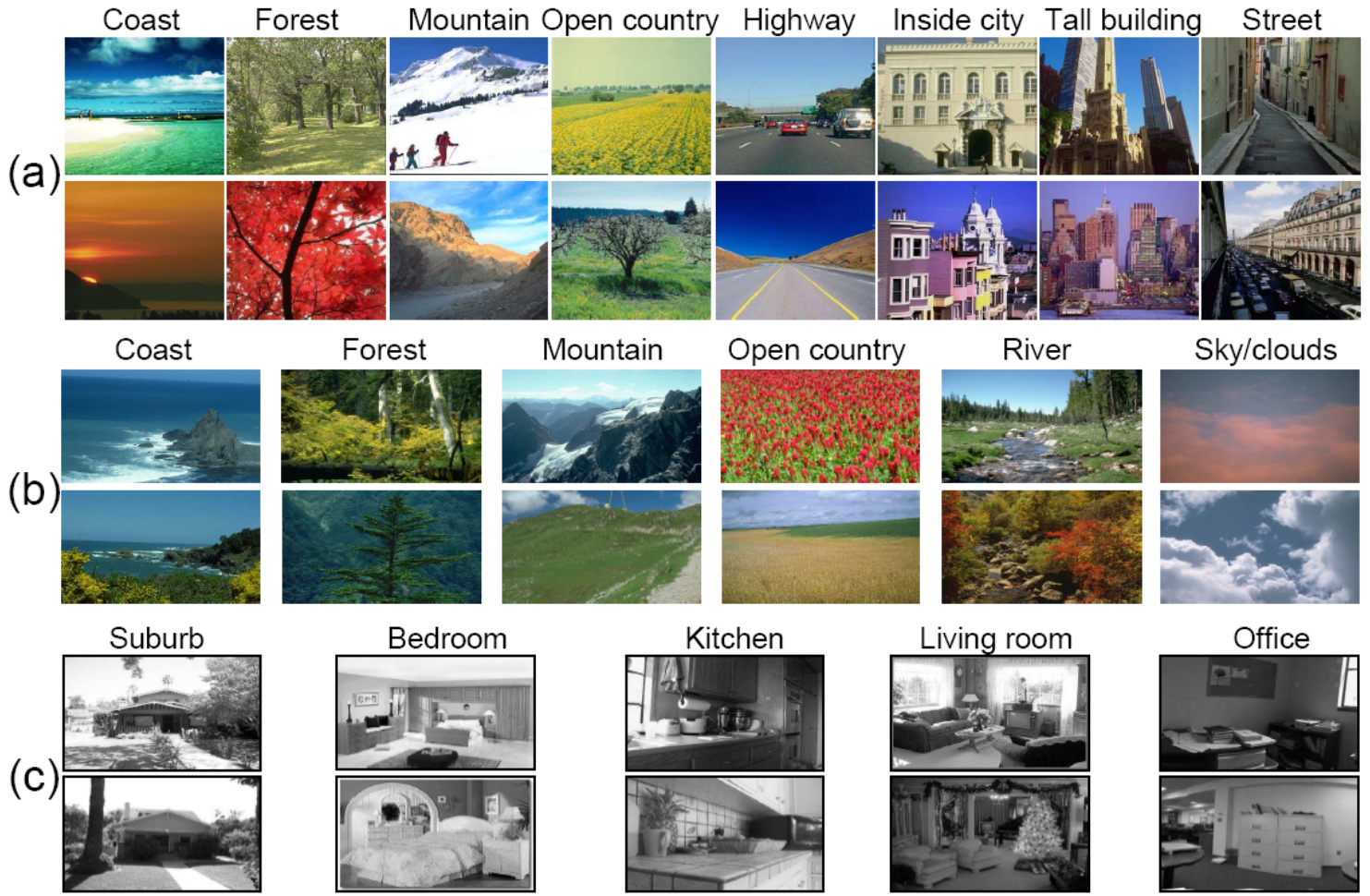


Fig. 2. Example images from the three different datasets used. (a) from dataset OT [11], (b) from dataset VS [19], and (c) from the dataset FP [3]. The remaining images of this dataset are the same as in OT but in greyscale.

Sparse Features

- For sparse and dense features:
  - visual vocabulary is obtained by vector quantizing descriptors computed from the training images using k-means

- Grey SIFT: As described in excursion
  - each region represented by 128-dim SIFT descriptor

Four Dense Descriptors (1)

- Important parameters:
  - size of patches (*N*)
  - spacing (*M*) between patches

  --> controls the degree of overlap

Grey patches: descriptor = N x N square neighborhood around a pixel.

- Patch sizes *N* = 5, 7 and 11 spaced by *M* pixels on a regular grid
- Not overlap when *M* = *N*
- Overlap when *M* = 3 (for *N* = 5, 7) and *M* = 7 (for *N* = 11).

Color patches: As above, but the HSV color information is used for each pixel ==> N^2 x 3 descriptor

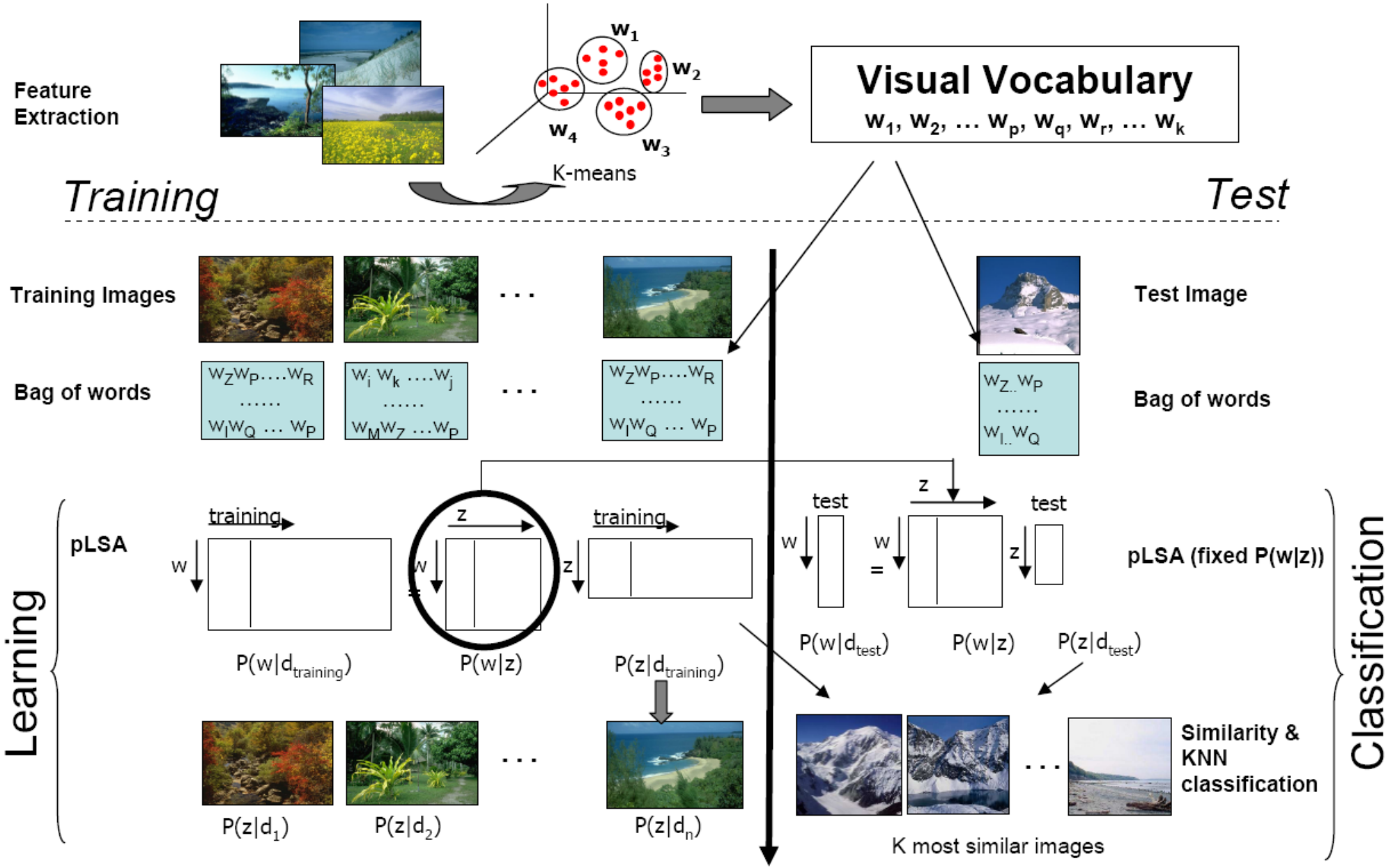
Four Dense Descriptors (2)

Grey SIFT: (rotation invariant) SIFT descriptors computed at points on a regular grid with spacing M pixels

- *M* = 5, 10, 15.
- circular support patches with radii r = 4, 8, 12, 16

==> each point is represented by *4* SIFT descriptors, each is 128-dim.

Color SIFT: As above, but now computed for each HSV component. ==>128 x 3 dim-SIFT descriptor for each point.



Classification

2 stages in classifying of an unseen test image

1. Document specific mixing coefficients $P(z|d_{test})$ are computed
2. Used them to classify the test images using a K nearest neighbor (KNN) scheme.

At 1:

- Determine mixing coefficients $P(z_k|d_{test})$ such that the Kullback-Leibler divergence between the measured empirical distribution and $P(w|d_{test}) = \sum_{z \in Z} P(w|z) P(z|d_{test})$ is minimized.
- This is achieved by running EM in a similar manner to that used in learning, but now only the coefficients $P(z_k|d_{test})$ are updated in each M-step with the learnt $P(w|z)$ kept fixed.

==> test image is represented by a Z-vector.
==> classified using a KNN on Z-vectors of the training images (L2 norm).

At 2:

- KNN selects the K nearest neighbors of training database and assigns the label of the category which is most represented within the K nearest neighbors.

Kullback-Leibler Divergence

From Wikipedia, the free encyclopedia.

:= ***relative entropy*** := quantity measuring difference between two probability distributions

→ Not a metric (violates triangle inequality) & not symmetric

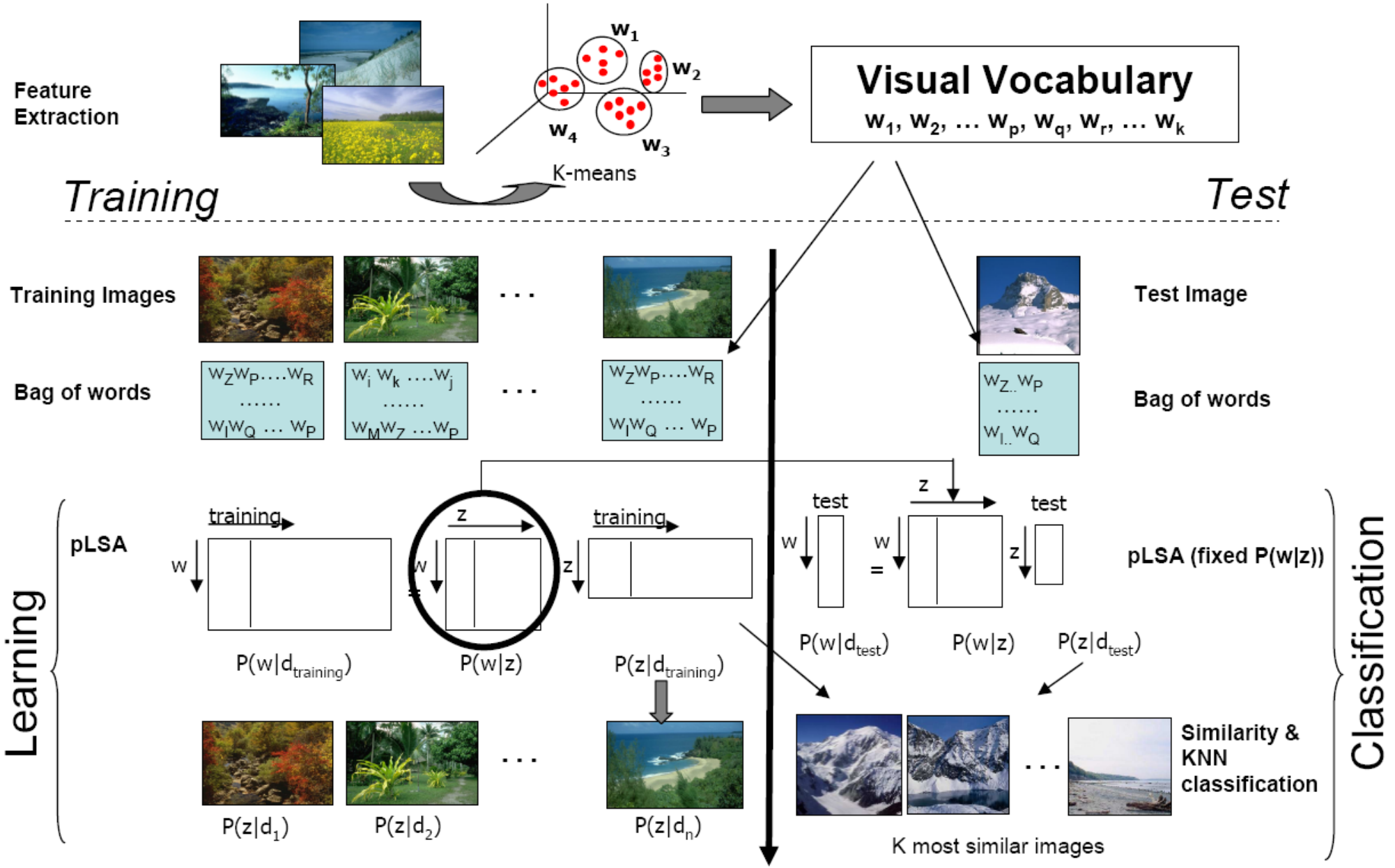
The KL divergence between two probability distributions p and q is defined as

$$KL(p,q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \qquad KL(p,q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)}\, dx$$

for distributions of a discrete/ continuous variable:

It can be seen $KL(p,q) = -\sum_x p(x) \log q(x) + \sum_x p(x) \log p(x) = H(p,q) - H(p)$

denoting by $H(p,q)$ the cross entropy of p and q, and by $H(p)$ the entropy of p. As the cross-entropy is always greater than or equal to the entropy, this shows that the Kullback-Leibler divergence is nonnegative, and furthermore $KL(p,q)$ is zero iff $p = q$.

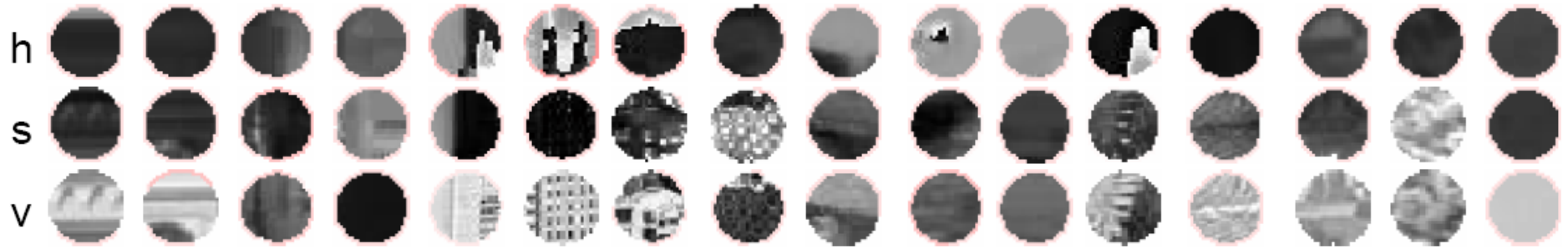


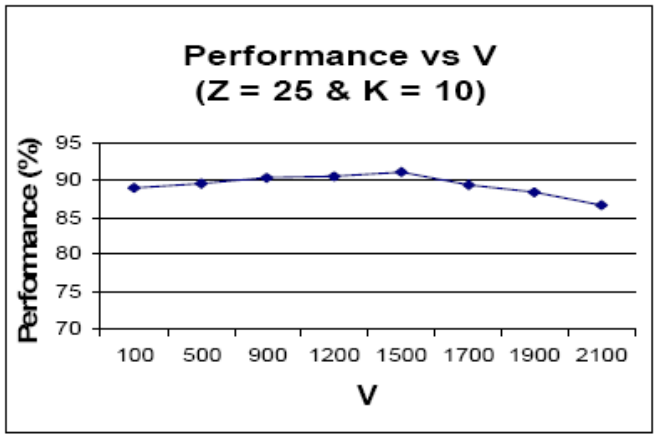
3 Parameters

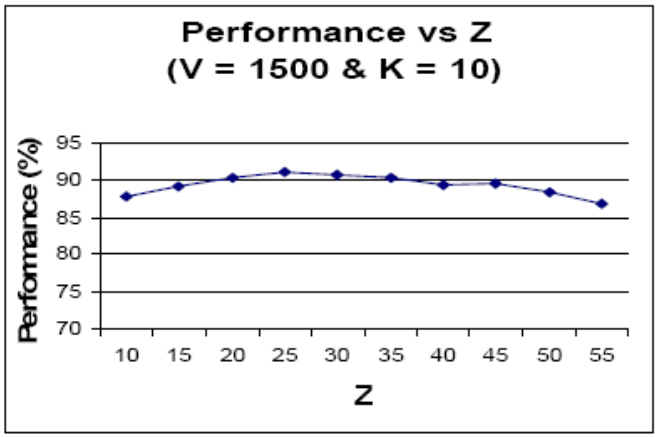
- # of visual words: *V*
- # of topics: *Z*
- # of neighbors in KNN: k

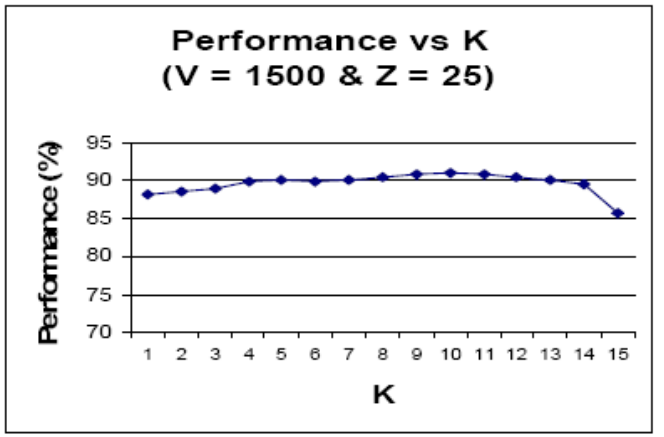
Next slide:

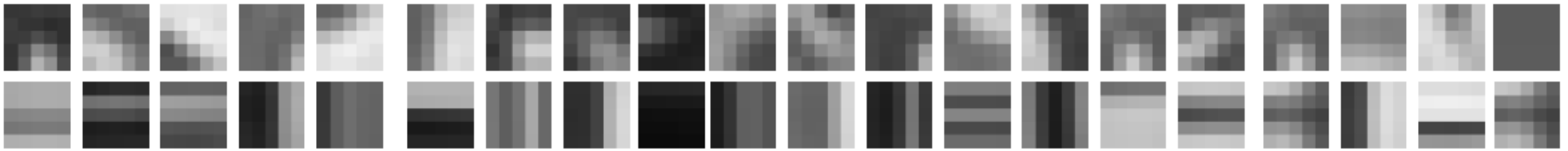
- Top row: SIFT *M*=10, *r* = 4,8,12,16
- Bottom row: grey patches *N*=5, *M*=3

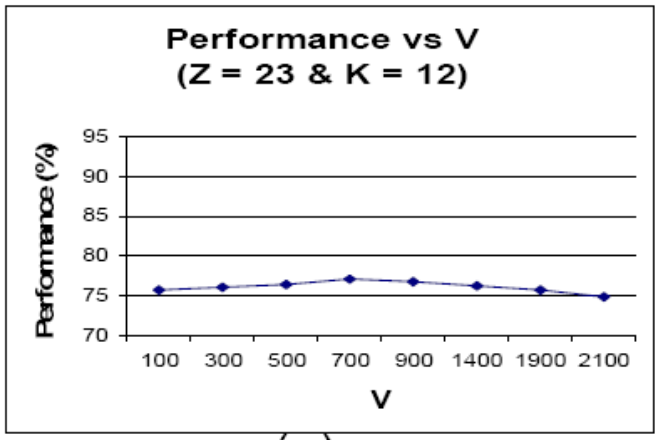

h
s
v


Performance vs V
(Z = 25 & K = 10)
Perfomance (%)
95
90
85
80
75
70
100 500 900 1200 1500 1700 1900 2100
V

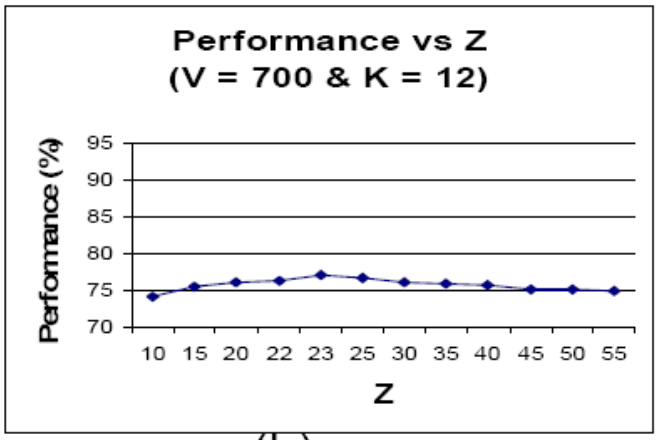

Performance vs Z
(V = 1500 & K = 10)
Perfomance (%)
95
90
85
80
75
70
10 15 20 25 30 35 40 45 50 55
Z


Performance vs K
(V = 1500 & Z = 25)
Perfomance (%)
95
90
85
80
75
70
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
K

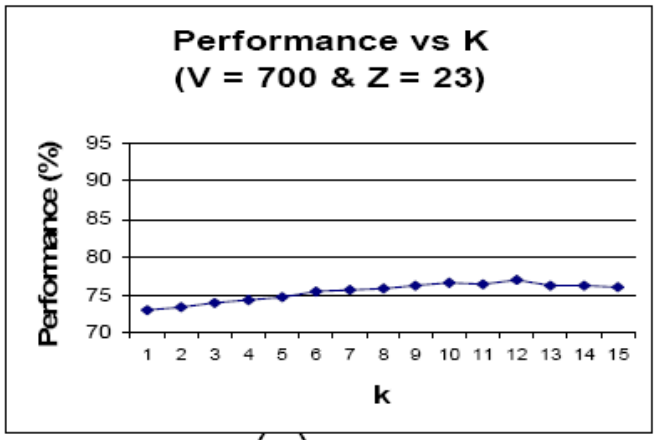



Performance vs V
(Z = 23 & K = 12)
Perfomance (%)
95
90
85
80
75
70
100 300 500 700 900 1400 1900 2100
V

(a)


Performance vs Z
(V = 700 & K = 12)
Perfomance (%)
95
90
85
80
75
70
10 15 20 22 23 25 30 35 40 45 50 55
Z

(b)


Performance vs K
(V = 700 & Z = 23)
Perfomance (%)
95
90
85
80
75
70
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
k

(c)

Examples of 'Aspects'

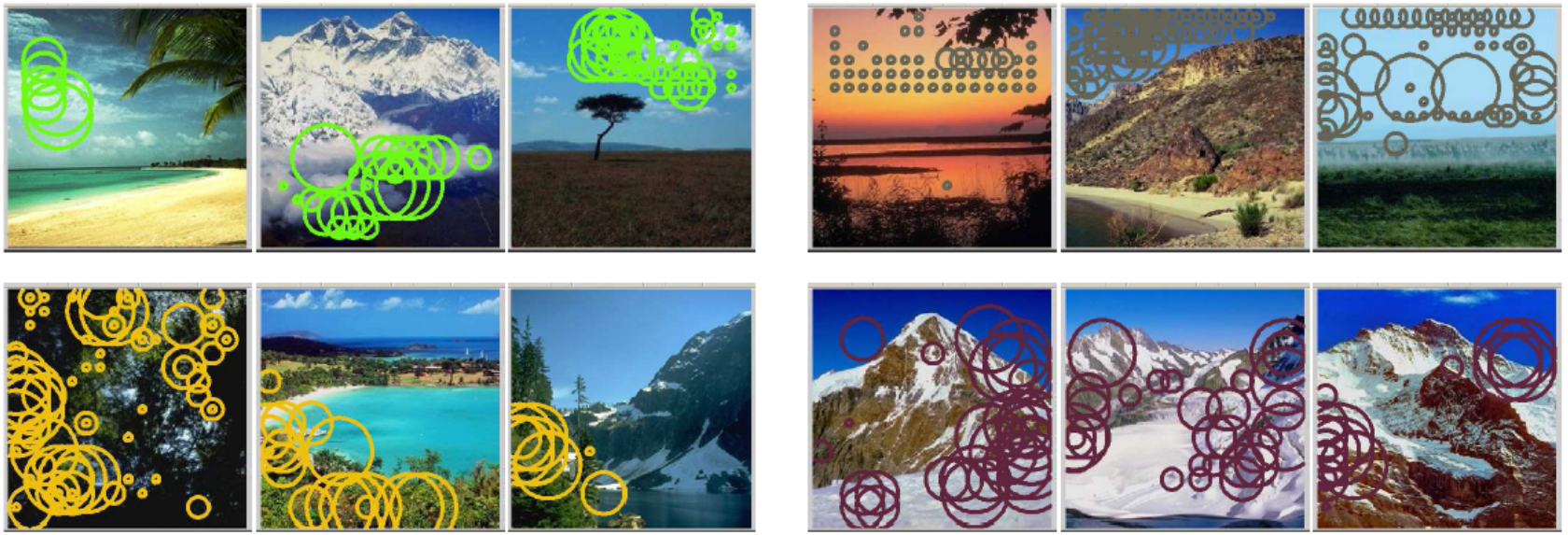


Fig. 5. Topics segmentation. Four topics (clouds – top left, sky – top right, vegetation – lower left, and snow/rocks in mountains – lower right) are shown. Only circular regions with a topic posterior $P(z|w,d)$ greater than 0.8 are shown.

More Classification Examples

Fig. 6. Example frames from the film Pretty Woman with their classification. The classifier is trained on the OT dataset.