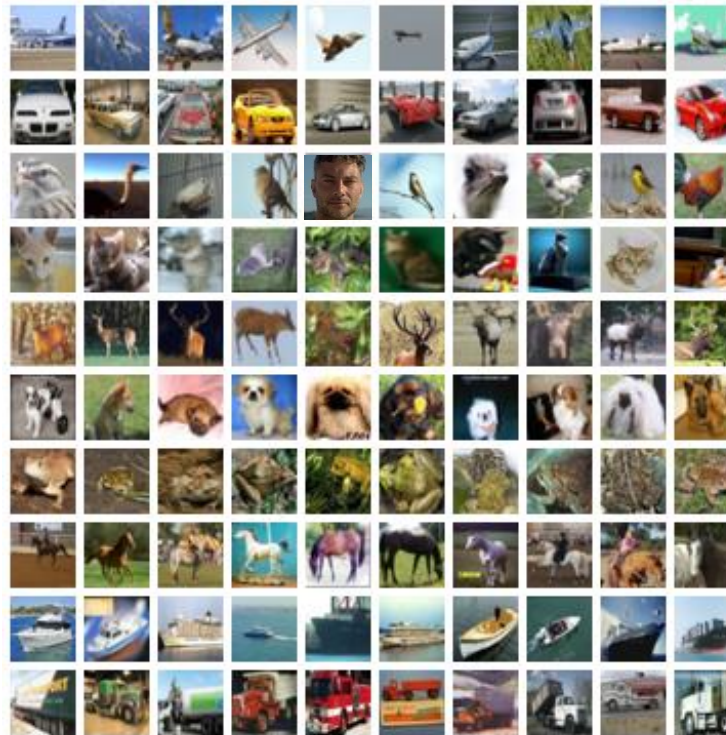


Sistemi avanzati per il Riconoscimento



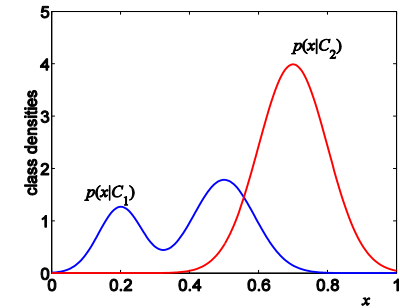
Riconoscimento di oggetti e scene

Prof. Marco Cristani

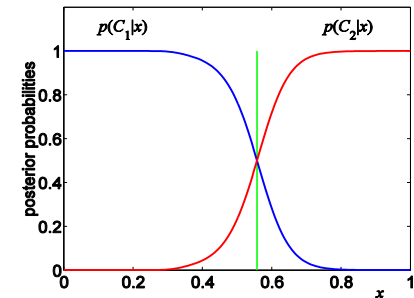


Learning and Recognition

1. Generative method:
- graphical models



2. Discriminative method:
- SVM



**category models
(and/or) classifiers**



2 generative models

1. Naïve Bayes classifier

- Csurka Bray, Dance & Fan, 2004

2. Hierarchical Bayesian text models (pLSA and LDA)

- Background: Hoffman 2001, Blei, Ng & Jordan, 2004
- Object categorization: Sivic et al. 2005, Sudderth et al. 2005
- Natural scene categorization: Fei-Fei et al. 2005

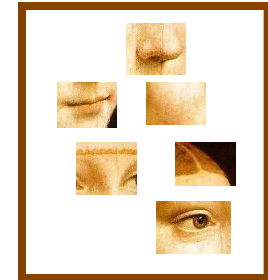
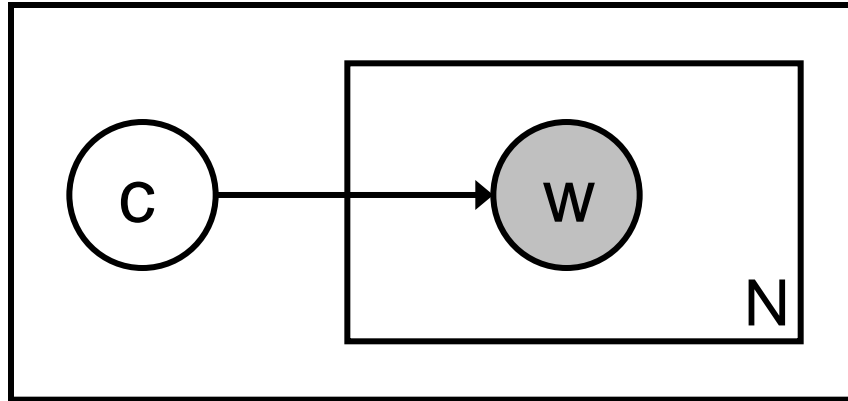


First, some notations

- w_n : each patch in an image
 - $w_n = [0, 0, \dots, 1, \dots, 0, 0]^T$
- N size of the vocabulary
- D number of images
- d_j : the j^{th} image in an image collection
- c : category of the image
- z : theme or topic of the patch



Case #1: the Naïve Bayes model



$$c^* = \arg \max_c p(c | w) \propto p(c) p(w | c) = p(c) \prod_{n=1}^N p(w_n | c)$$

Object class
decision

Prior prob. of
the object classes

Image likelihood
given the class



Our in-house database contains 1776 images in seven classes¹: faces, buildings, trees, cars, phones, bikes and books. Fig. 2 shows some examples from this dataset.

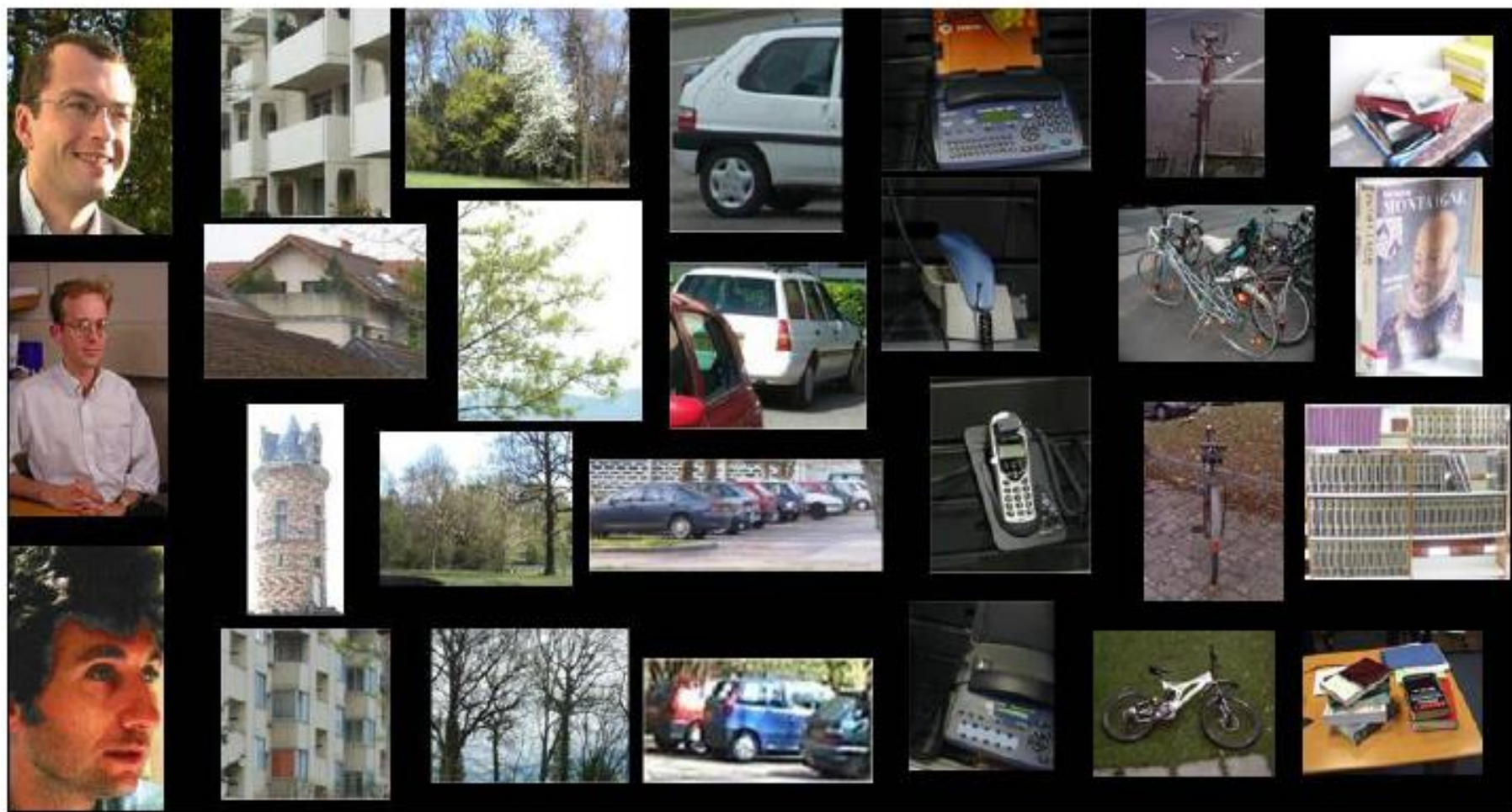


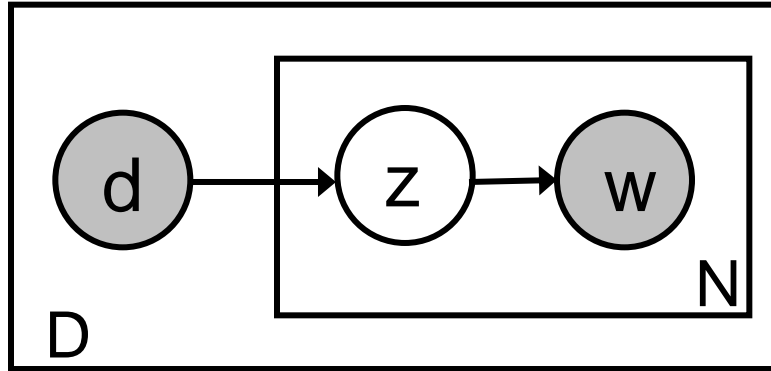
Table 1. Confusion matrix and the mean rank for the best vocabulary ($k=1000$).

True classes →	<i>faces</i>	<i>buildings</i>	<i>trees</i>	<i>cars</i>	<i>phones</i>	<i>bikes</i>	<i>books</i>
<i>faces</i>	76	4	2	3	4	4	13
<i>buildings</i>	2	44	5	0	5	1	3
<i>trees</i>	3	2	80	0	0	5	0
<i>cars</i>	4	1	0	75	3	1	4
<i>phones</i>	9	15	1	16	70	14	11
<i>bikes</i>	2	15	12	0	8	73	0
<i>books</i>	4	19	0	6	7	2	69
<i>Mean ranks</i>	1.49	1.88	1.33	1.33	1.63	1.57	1.57



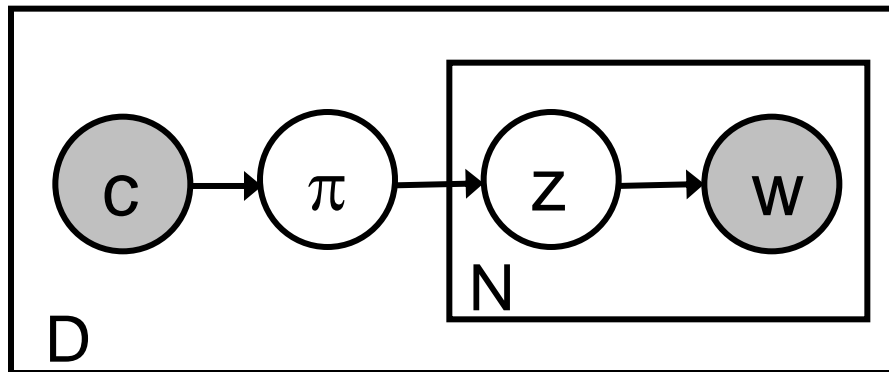
Case #2: Hierarchical Bayesian text models

Probabilistic Latent Semantic Analysis (pLSA)



Hoffman, 2001

Latent Dirichlet Allocation (LDA)

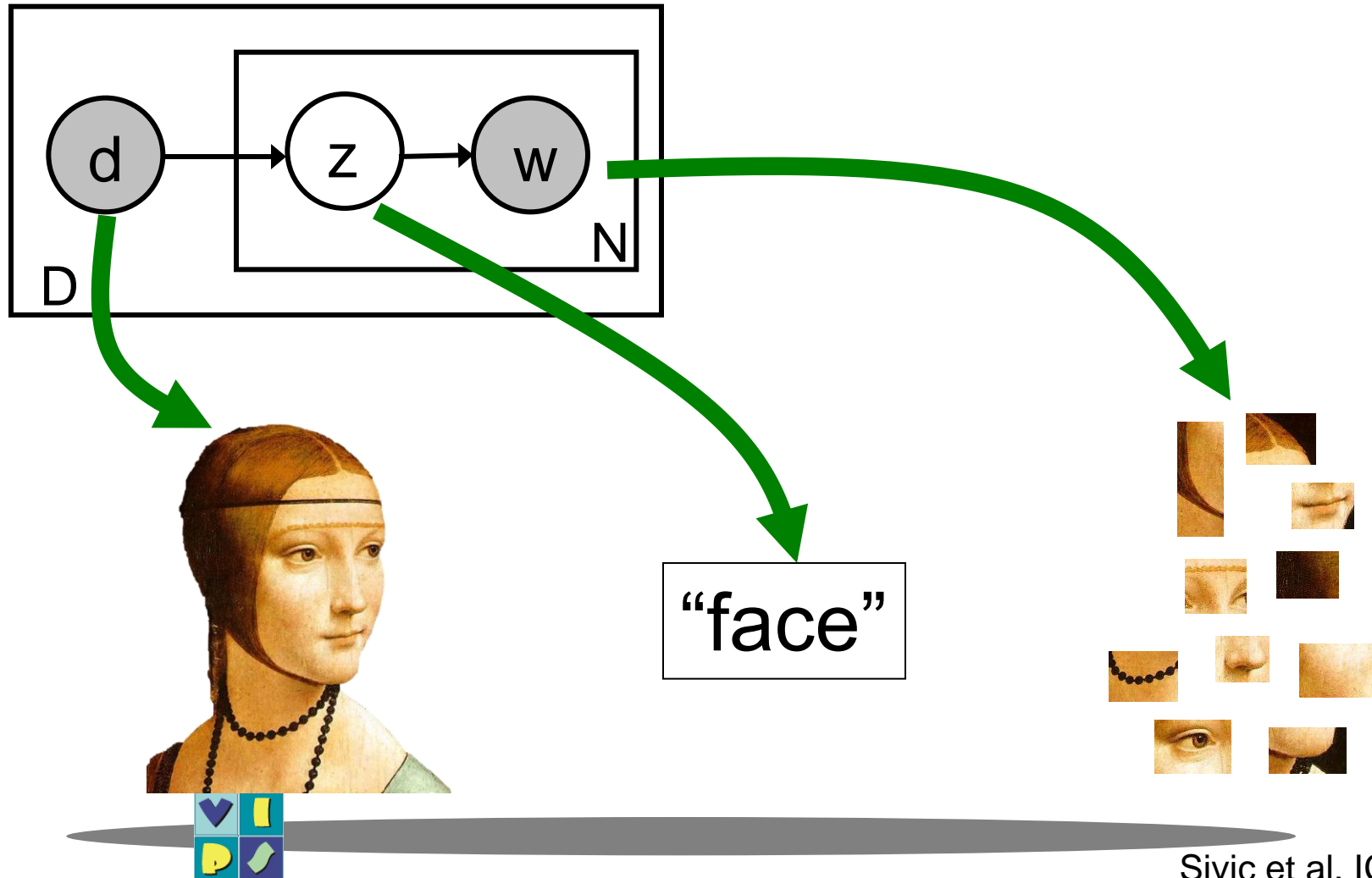


Blei et al., 2001

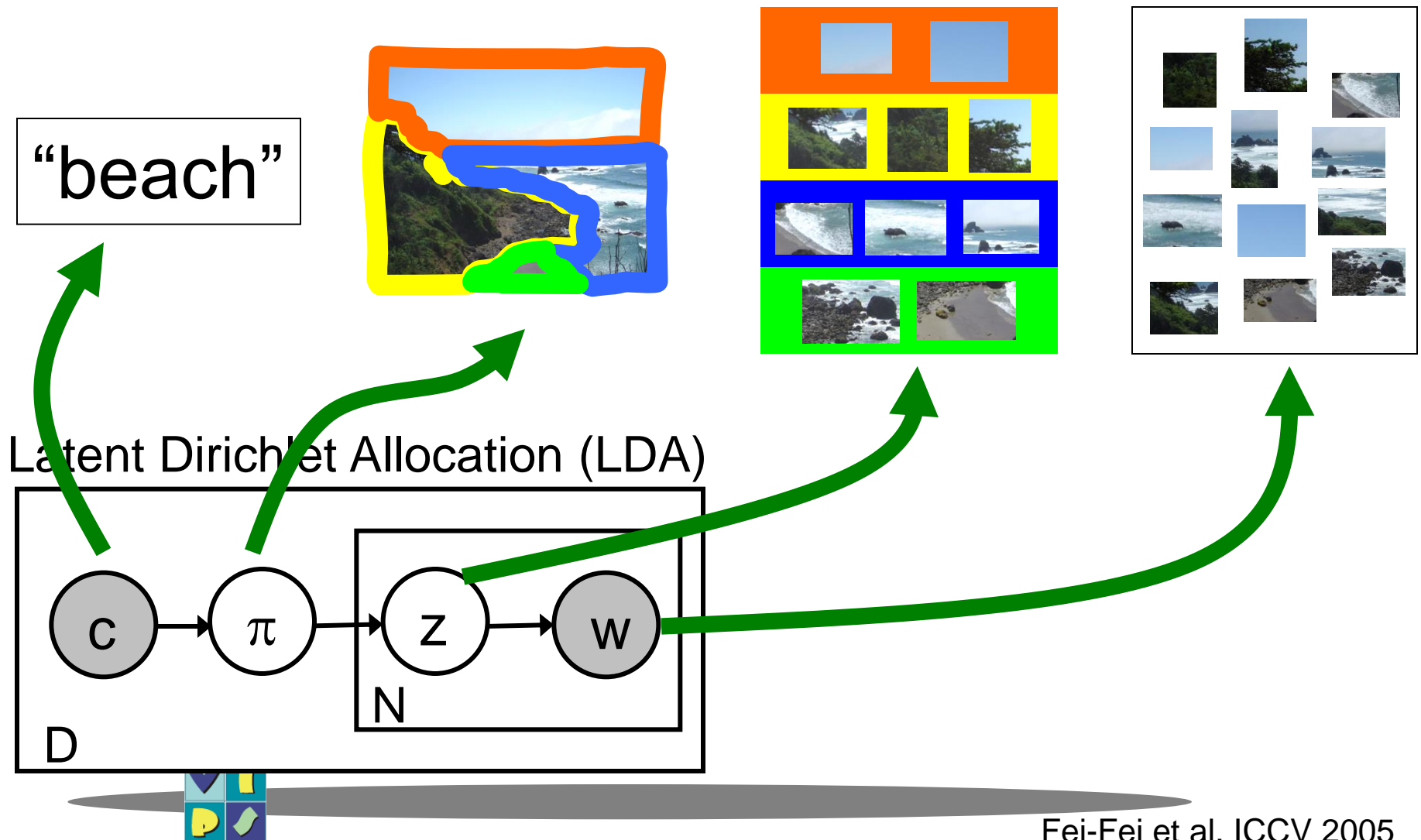


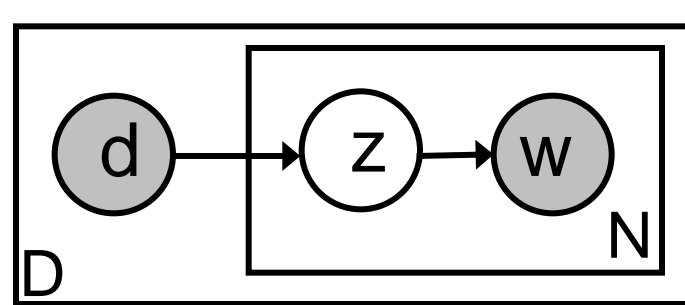
Case #2: Hierarchical Bayesian text models

Probabilistic Latent Semantic Analysis (pLSA)

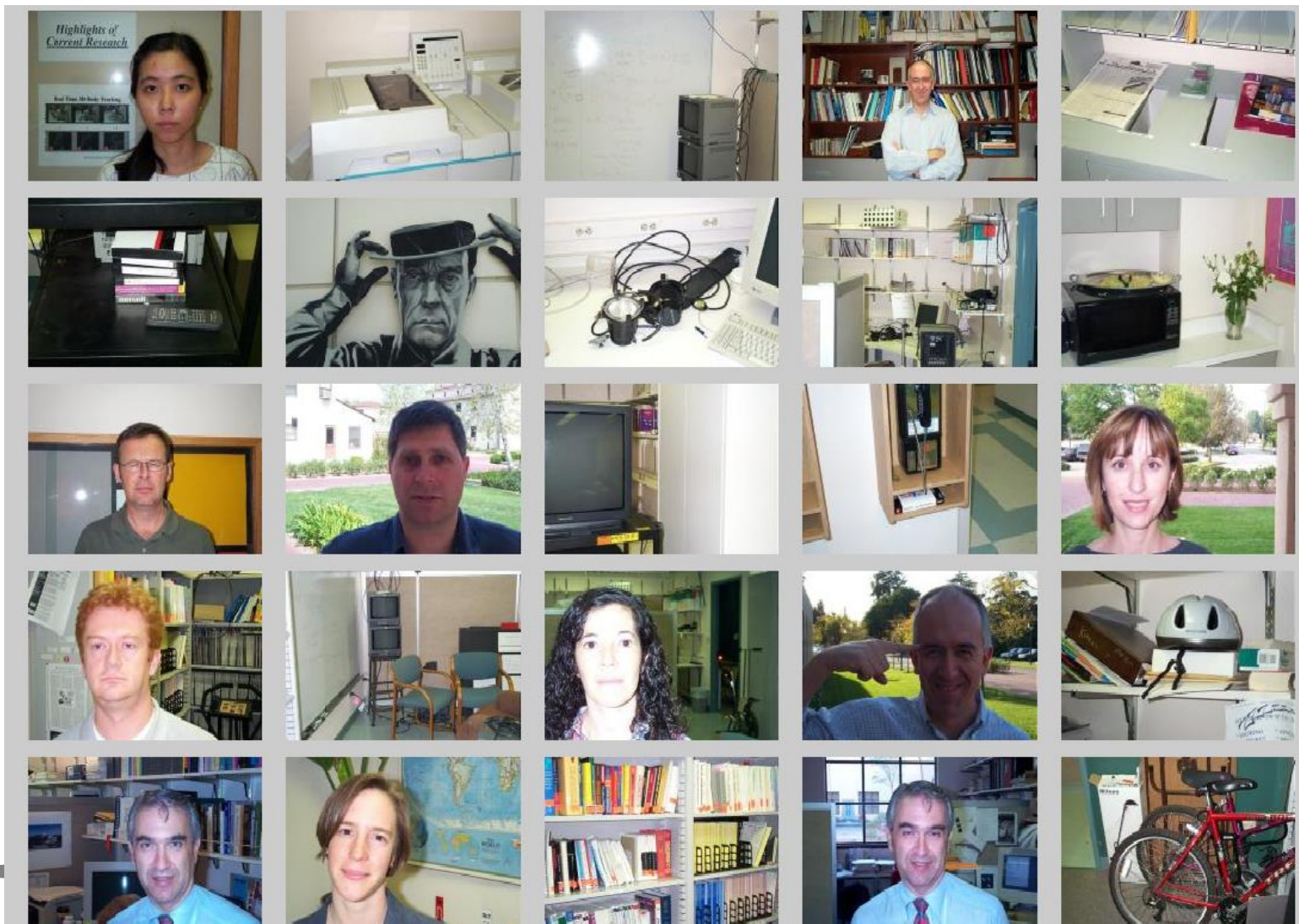


Case #2: Hierarchical Bayesian text models

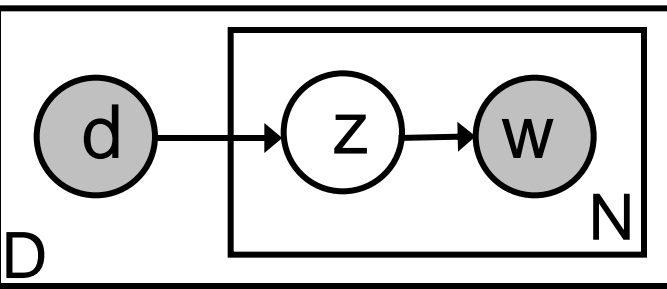




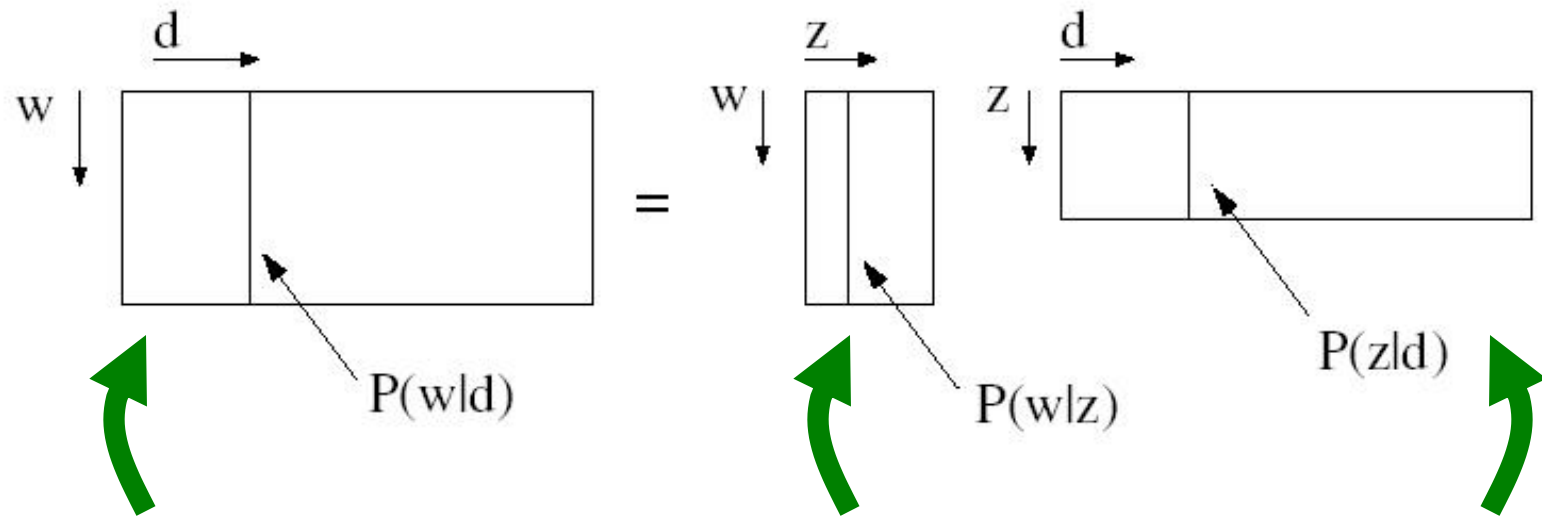
Case #2: the pLSA model



Case #2: the pLSA model



$$p(w_i | d_j) = \sum_{k=1}^K p(w_i | z_k) p(z_k | d_j)$$



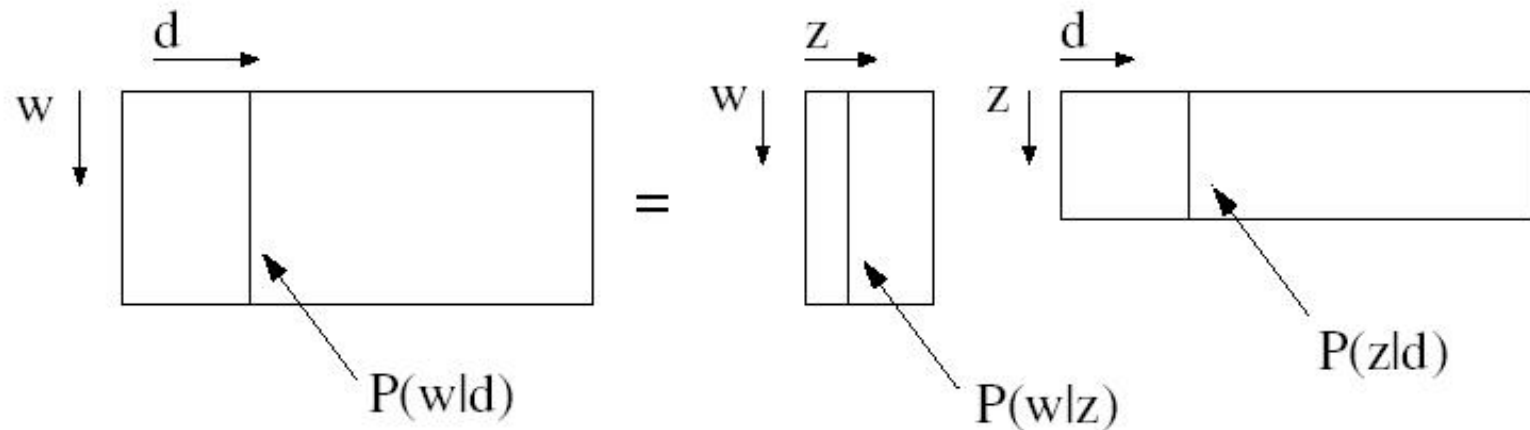
Observed codeword
distributions

Codeword distributions
per theme (topic)

Theme distributions
per image

Case #2: Recognition using pLSA

$$z^* = \arg \max_z p(z | d)$$



Case #2: Learning the pLSA parameters

Observed counts of word i in document j

$$L = \prod_{i=1}^M \prod_{j=1}^N P(w_i | d_j)^{n(w_i, d_j)}$$

$\sum_{k=1}^K P(z_k | d_j) P(w_i | z_k)$

Maximize likelihood of data using EM

M ... number of codewords

N ... number of images



Demo

- Course website


A demonstration of bag-of-words classifiers - Microsoft Internet Explorer provided by Insight Broadband

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Mail Print Fax New Tab

Address <http://people.csail.mit.edu/fergus/iccv2005/bagwords.html>

Google Search 100 blocked Check AutoLink AutoF



Two bag-of-words classifiers

ICCV 2005 short courses on
[Recognizing and Learning Object Categories](#)

A simple approach to classifying images is to treat them as a collection of regions, describing only their appearance and ignoring their location. This approach has been successfully used in the text community for analyzing documents and are known as "bag-of-words" models, since each document is represented by the distribution over fixed vocabulary(s). Using such a representation, methods such as probabilistic latent semantic analysis (pLSA) [1] (LDA) [2] are able to extract coherent topics within document collections in an unsupervised manner.

Recently, Fei-Fei et al. [3] and Sivic et al. [4] have applied such methods to the visual domain. The demo code implements pLSA, including a Naïve Bayes classifier. For comparison, a Naïve Bayes classifier is also provided which requires labelled training data, unlike pLSA.

The code consists of Matlab scripts (which should run under both Windows and Linux) and a couple of 32-bit Linux binaries for doing matrix operations. Hence the whole system will need to be run on Linux. The code is for teaching/research purposes only. If you find a bug, please email fergus@csail.mit.edu where csail point mit point edu.

Download

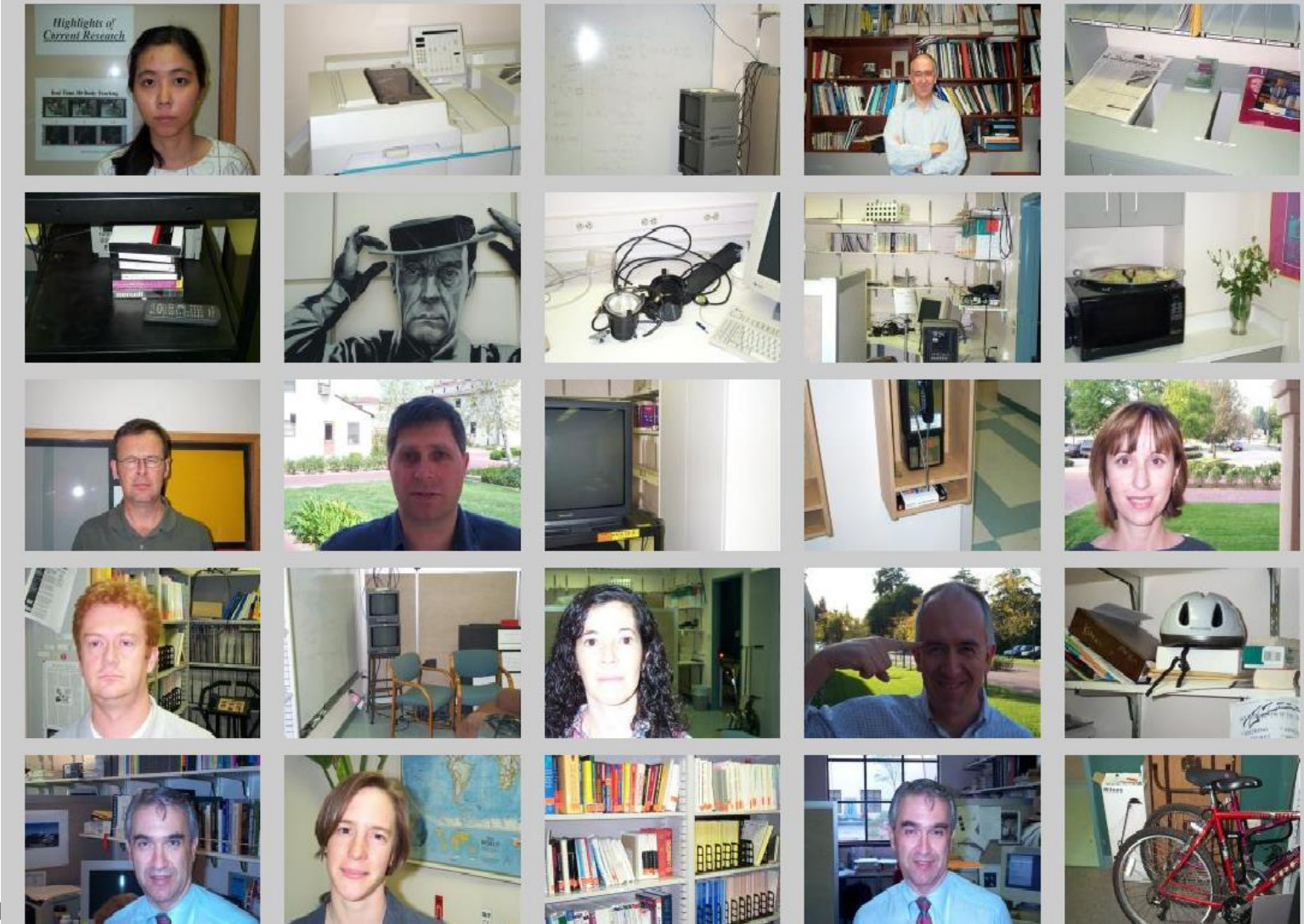
[Download](#) the code and datasets (32 Mbytes)

Operation of code

To run the demos:

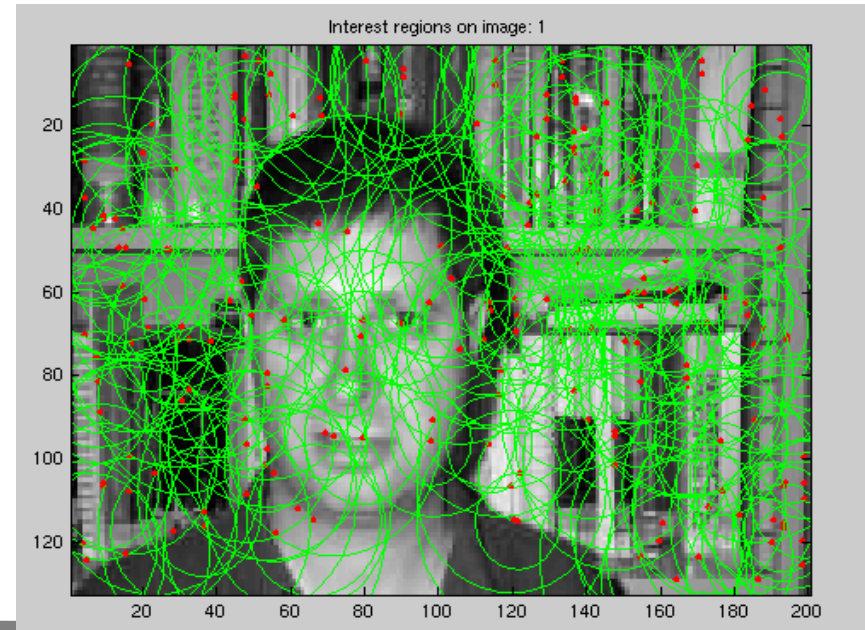
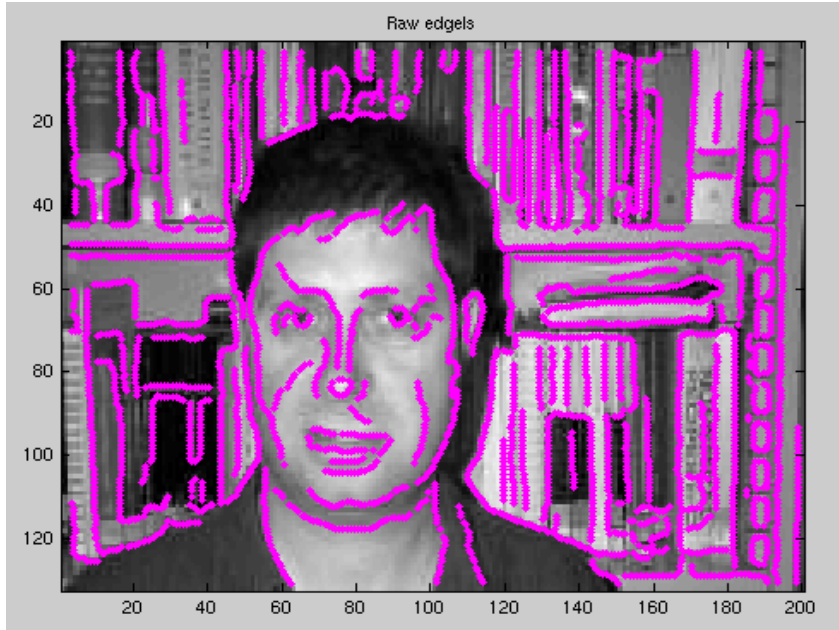
start Microsoft Outlook We... 未名空间(mitbbs.co... A demonstration of b... ICCV2005

task: face detection – no labeling



Demo: feature detection

- Output of crude feature detector
 - Find edges
 - Draw points randomly from edge set
 - Draw from uniform distribution to get scale

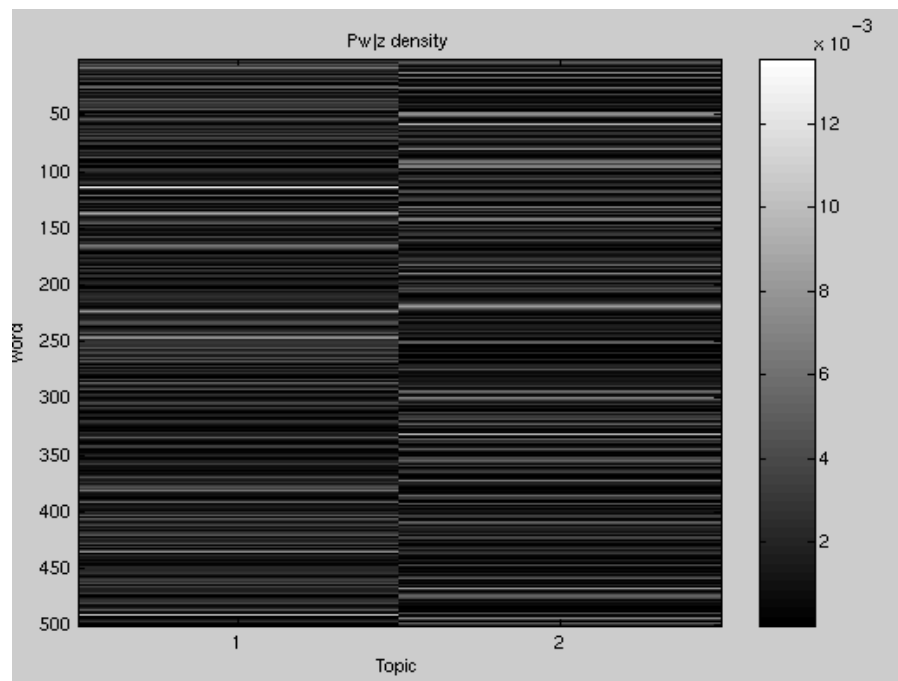


Demo: learnt parameters

- Learning the model: `do_plsa('config_file_1')`
- Evaluate and visualize the model: `do_plsa_evaluation('config_file_1')`

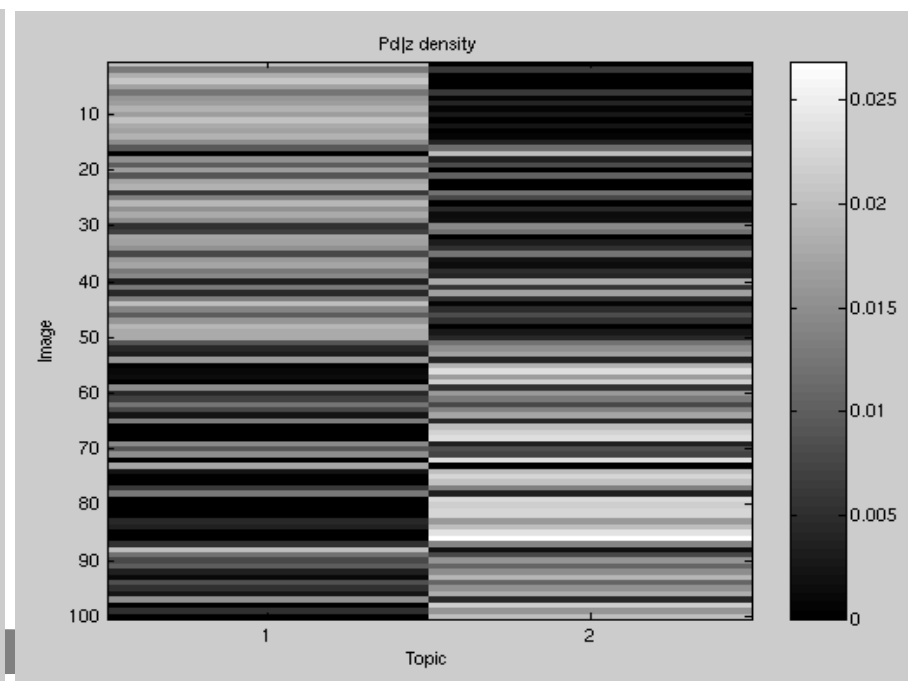
Codeword distributions
per theme (topic)

$$p(w | z)$$



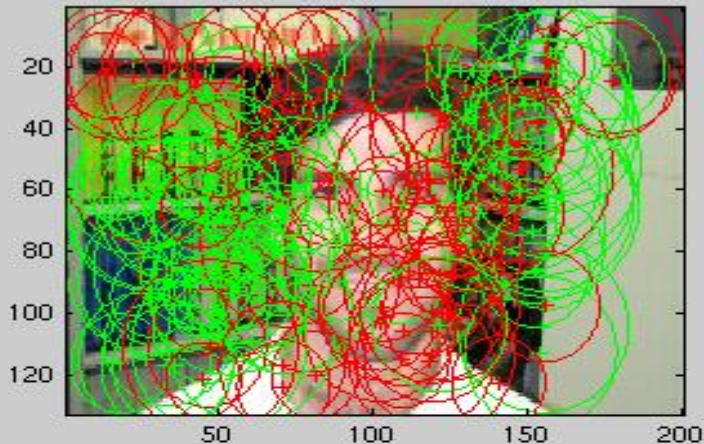
Theme distributions
per image

$$p(z | d)$$

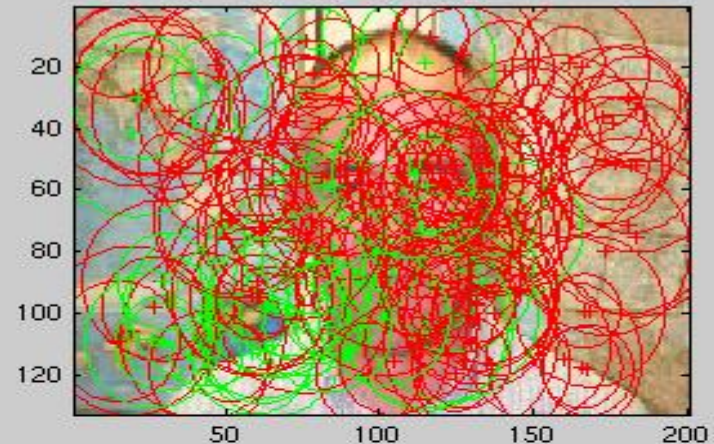


Demo: recognition examples

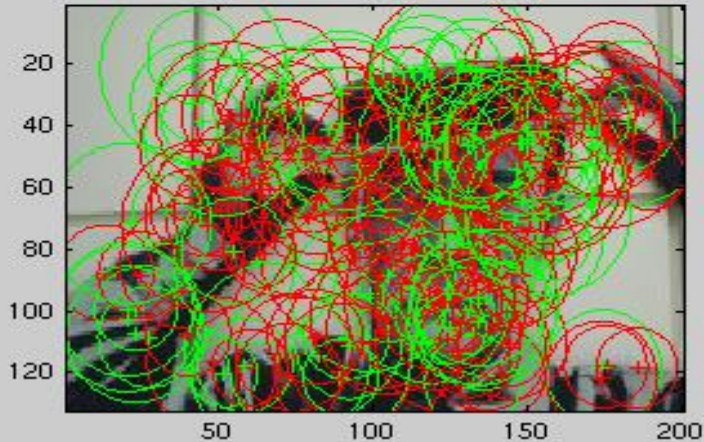
Correct - Image: 1 $P(z|d)=0.3662$ 0.6338



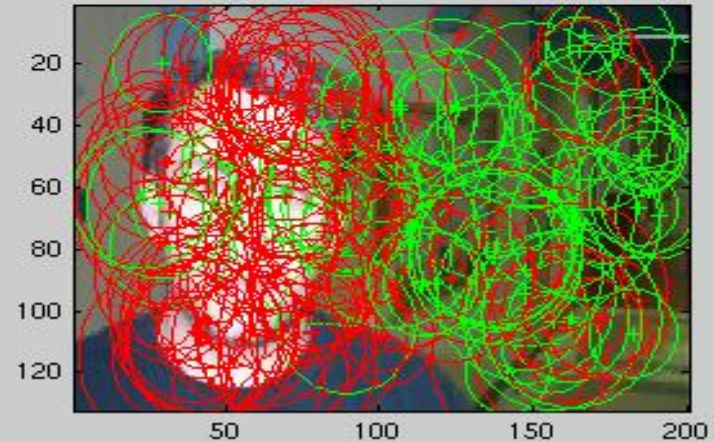
Correct - Image: 2 $P(z|d)=0.83087$ 0.16913



Correct - Image: 3 $P(z|d)=0.59906$ 0.40094



Correct - Image: 5 $P(z|d)=0.68534$ 0.31466



Demo: categorization results

- Performance of each theme

