Image Retrieval

D.A. Forsyth, UIUC

LOTS of BIG collections of images

Corel Image Data	40,000 images
Fine Arts Museum of San Francisco	83,000 images online
Cal-flora	20,000 images, species information
News photos with captions (yahoo.com)	1,500 images per day available from yahoo.com
Hulton Archive	40,000,000 images (only 230,000 online)
internet.archive.org	1,000 movies with no copyright
TV news archives (televisionarchive.org, informedia.cs.cmu.edu)	Several terabytes already available
Google Image Crawl	>330,000,000 images (with nearby text)
Satellite images (terrarserver.com, nasa.gov, usgs.gov)	(And associated demographic information)
Medial images	(And associated with clinical information)

1.5e9 or so

* and the BBC is releasing its video archive, too;
and we collected 500,000 captioned news images;
and it's easy to get scanned mediaeval manuscripts;
etc., etc.,

Imposing order

- Iconic matching
  - child abuse prosecution
  - managing copyright (BayTSP)

  Current, practical applications
- Clustering
  - Browsing for:
    - web presence for museums (Barnard et al, 01)
    - home picture, video collections
    - selling pictures

  Maybe applications
- Searching
  - scanned writing (Manmatha, 02)
  - collections of insects

  Maybe applications
- Building world knowledge
  - a face gazetteer (Miller et al, 04)

Searching

- Specify a need
  - picture based interface
  - word based interface
- Get it met
  - weighted match of search terms

Picture Queries

Jacobs et al, 1995

Simple things

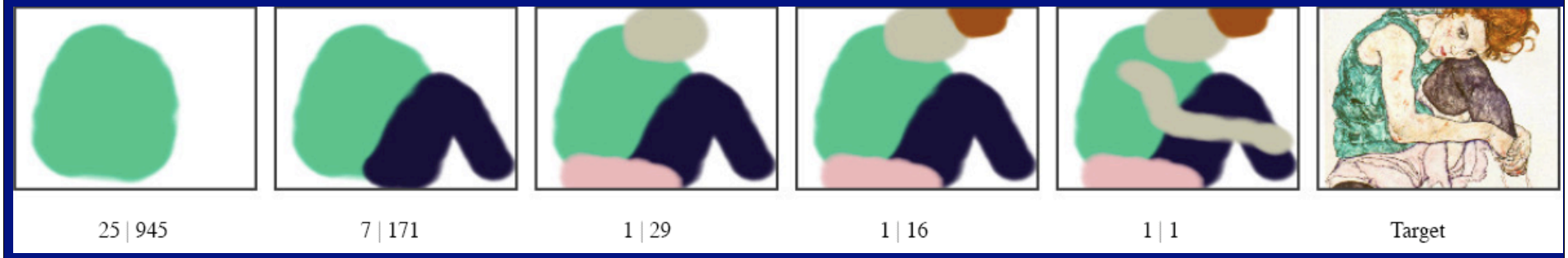
- Match color histograms
  - not great, insufficient spatial support
- Pyramid matcher
  - ok matches

Procedure

- For each image in collection
  - compute a signature (wavelet coefficients)
- For query
  - compute match score to each image

- Speedup
  - Quantize wavelet coefficients
  - “Bin” wavelet coefficients
  - Only look at those where query is not zero

Query



Jacobs et al, 1995

Technical problem

- Score for two vectors (query, collection)
  - that is small for the right answer, big for wrong
  - why does Euclidean distance make sense? (doesn't necessarily)
- General issue - Metric learning
- Solutions
  - Jacobs et al do logistic regression
  - large-margin metric learning

Logistic regression

- Given a set of pairs (x_i, y_i)
  - y_i is 1 or zero
  - model P(y_i=1|x_i)
- Model $\log P(y_i = 1|\mathbf{x}_i) - \log P(y_i = 0|\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i$
- Notice that this gives a linear decision boundary

$$P(1|\mathbf{x}_i) = \frac{\exp \mathbf{w}^T \mathbf{x}_i}{1 + \exp \mathbf{w}^T \mathbf{x}_i}$$

Logistic Regression - II

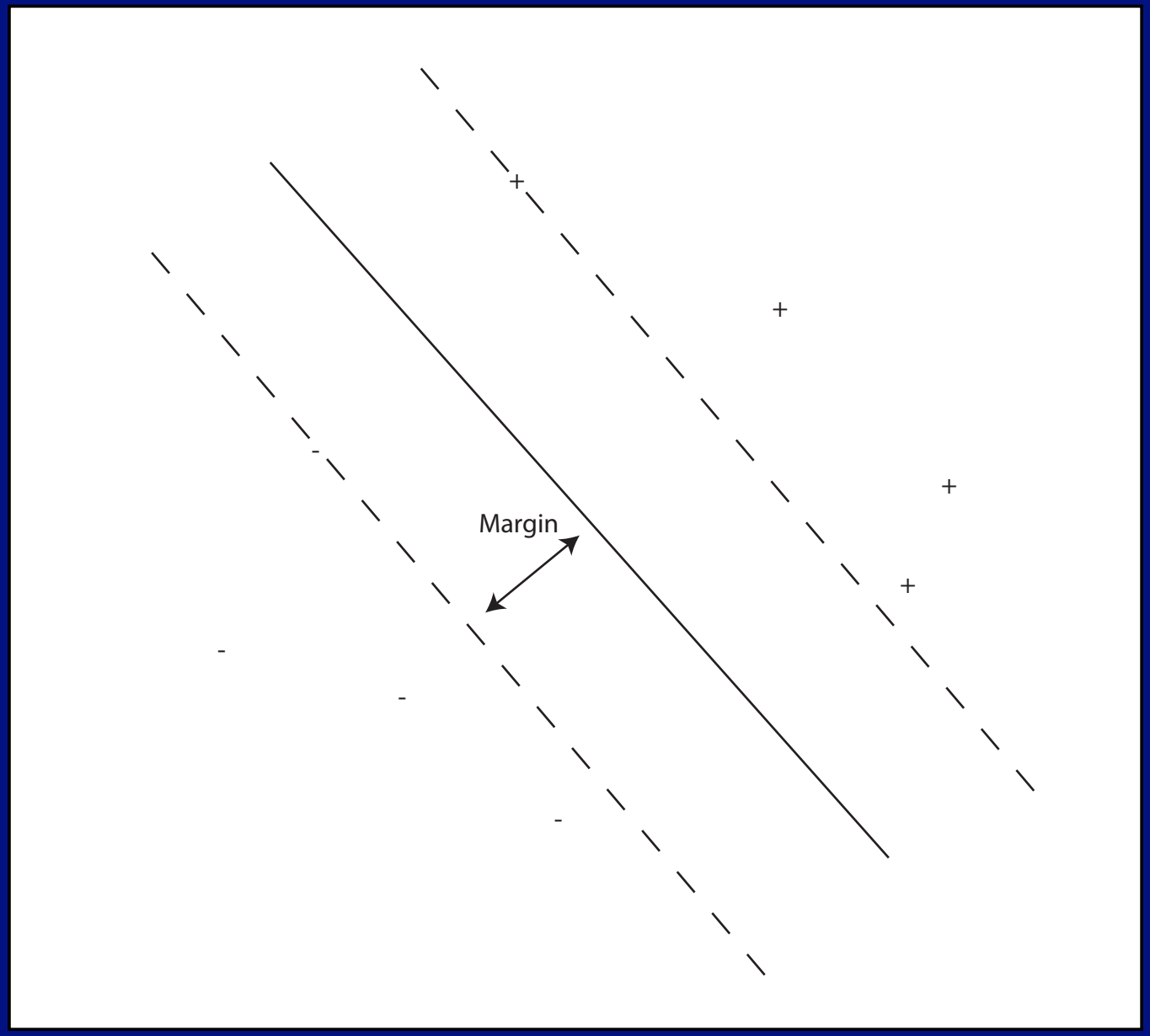
- Log likelihood is

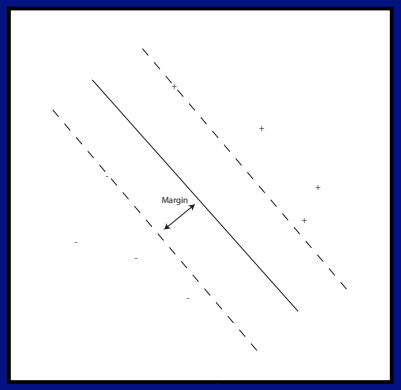
$$\mathcal{L}(\mathbf{w}) = \sum_i \left[y_i \mathbf{w}^T \mathbf{x}_i - \log\left(1 + \exp \mathbf{w}^T \mathbf{x}_i\right)\right]$$

- Convex
  - generally pretty well behaved
  - variety of methods to deal with excessively long x (later, perhaps)

Logistic regression for metric learning

- As used by Jacobs et al, 95
- Learning
  - Feature vector for pair
    - Compute vector of differences of binned wavelet coeffs
    - Quantize to 0-1
  - Apply logistic regression
- Matching
  - compute LR score
    - actually w^T x is enough
  - rank on this score


+
+
+
+
Margin
-
-
-



- Constraints: $y_i(\mathbf{w}^T\mathbf{x}_i + c) \geq 1$

- Non-separable problem:

$$\begin{array}{ll} \text{minimize:} & (1/2)\mathbf{w}^T\mathbf{w} + \sum_i \xi_i \\ \text{subject to:} & y_i(\mathbf{w}^T\mathbf{x}_i + c) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{array}$$

- Alternative view:

$$\xi_i = max((1 - y_i(\mathbf{w}^T\mathbf{x} + c)), 0)$$

The hinge loss

- Rewrite xi

$$\xi_i = max((1 - y_i(\mathbf{w}^T\mathbf{x} + c)), 0)$$

- as

$$L_h((1 - yy_{pred})) = max((1 - yy_{pred}), 0)$$

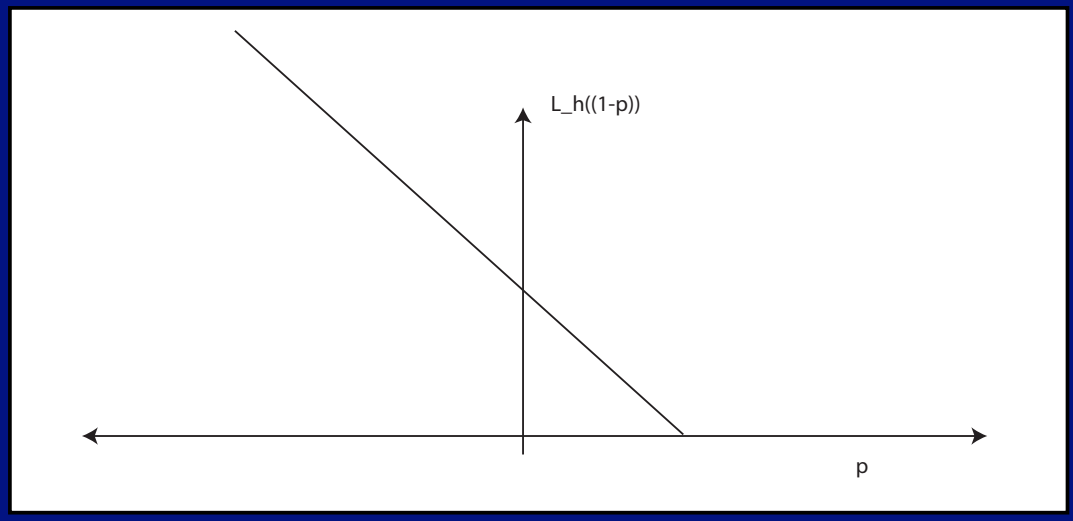
- Problem

$$\begin{array}{ll} \text{minimize:} & (1/2)\mathbf{w}^T\mathbf{w} + \sum_i \xi_i \\ \text{subject to:} & y_i(\mathbf{w}^T\mathbf{x}_i + c) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{array}$$

- becomes

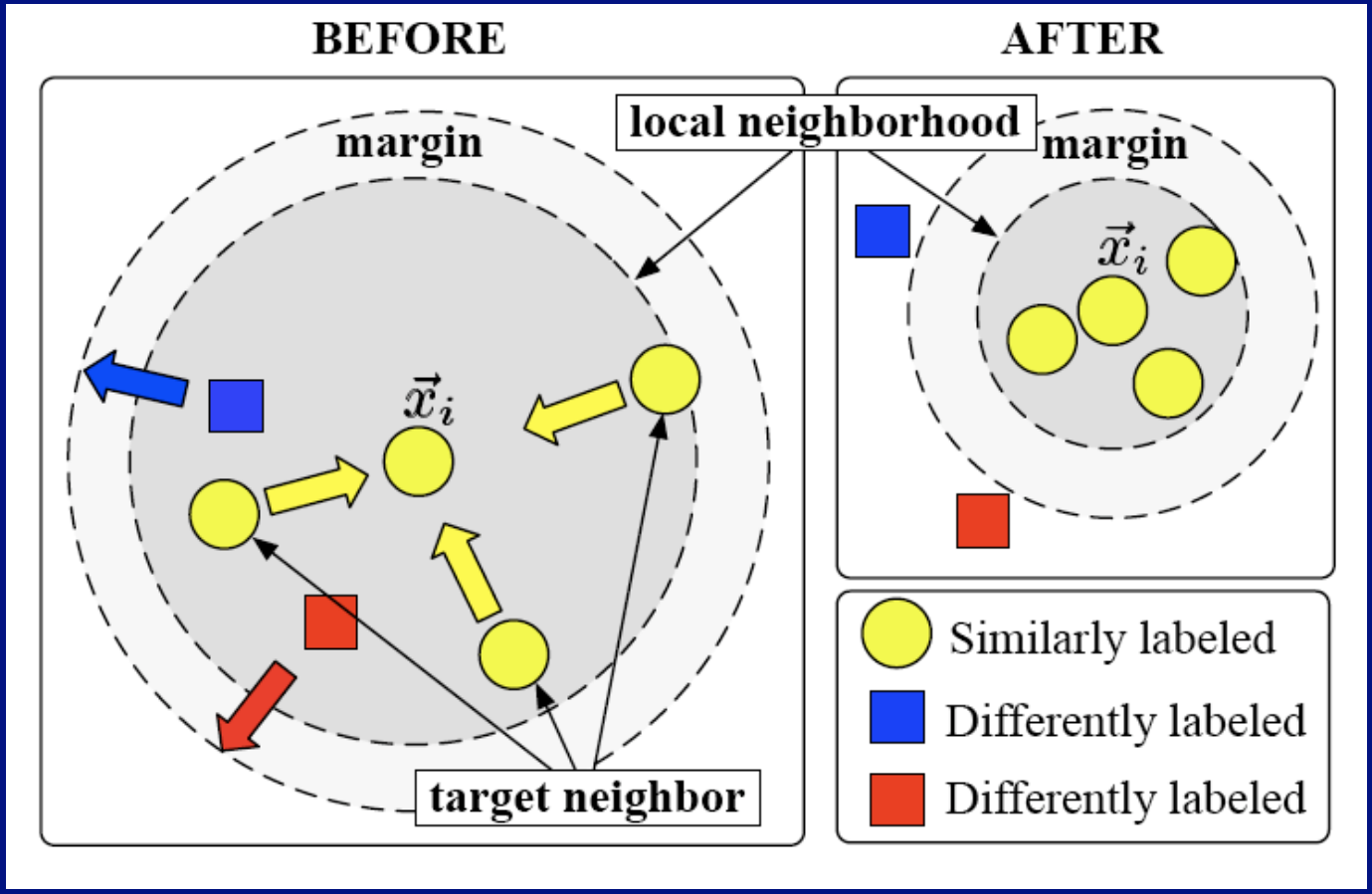
$$\text{minimize:} \quad (1/2)\mathbf{w}^T\mathbf{w} + \sum_i L_h((1 - y_i(\mathbf{w}^T\mathbf{x}_i + c)))$$

The hinge loss



Metric learning, revisited

Want: points with same label to be close
points with distant label to be far
margin



Metric learning, revisited- II

- Notation

examples $(\mathbf{x}_i, y_i)$

distance $(D(\mathbf{x}_i, \mathbf{x_j})) = (\mathbf{x}_i - \mathbf{x_j})^T \mathcal{M} (\mathbf{x}_i - \mathbf{x_j})$

similarity $y_{il} = \begin{cases} 1 & y_i,\ y_l \text{ same} \\ 0 & \text{otherwise} \end{cases}$

target neighbours $\eta_{ij} = \begin{cases} 1 & \mathbf{x}_i,\ \mathbf{x}_j \text{ should be close} \\ 0 & \text{otherwise} \end{cases}$

Metric learning, revisited - III

- Want
  - points with same label to be close
  - points with different label to be far
  - margin
- Cost function

$$\sum_{i,j} \eta_{ij} D(\mathbf{x}_i, \mathbf{x}_j) + c \sum_{i,j,l} \eta_{ij} (1 - y_{il}) L_h (1 - D(\mathbf{x}_i, \mathbf{x}_l) + D(\mathbf{x}_i, \mathbf{x_j}))$$

examples	$(\mathbf{x}_i, y_i)$
distance	$(D(\mathbf{x}_i, \mathbf{x_j})) = (\mathbf{x}_i - \mathbf{x_j})^T \mathcal{M} (\mathbf{x}_i - \mathbf{x_j})$
similarity	$y_{il} = \begin{cases} 1 & y_i, y_l \text{ same} \\ 0 & \text{otherwise} \end{cases}$
target neighbours	$\eta_{ij} = \begin{cases} 1 & \mathbf{x}_i, \mathbf{x}_j \text{ should be close} \\ 0 & \text{otherwise} \end{cases}$

Optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{i,j} \eta_{ij} (\mathbf{x}_i - \mathbf{x_j})^T \mathcal{M} (\mathbf{x}_i - \mathbf{x_j}) + c \sum_{i,j,l} \eta_{ij} (1 - y_{il}) \xi_{ijl} \\ \text{subject to} & (\mathbf{x}_i - \mathbf{x_l})^T \mathcal{M} (\mathbf{x}_i - \mathbf{x_l}) - (\mathbf{x}_i - \mathbf{x_j})^T \mathcal{M} (\mathbf{x}_i - \mathbf{x_j}))) \geq 1 - \xi_{ijl} \\ & \mathcal{M} \geq 0 \end{array}$$

i.e. M is positive semidefinite

- Manageable optimization problem in M
  - convex
  - semi-definite program
- Reasonable results in other applications
- Gets nasty when x is big

The curse of dimension

- In high dimensions, volume is on the “skin” of a body
  - e.g. high dimensional cube
- Example: uniform data in unit cube in dimension p
  - want fraction r of data to be in subcube
  - so subcube must have volume r
  - so edge length must be r^(1/p)

  - numbers: p=10, r=0.1 gives edge length of 0.794
  - hardly local!

Curse of dimension-II

- General phenomenon of high dimensions
  - volume is concentrated at the boundary
- Parameter estimation is hard for high dimensional distributions
  - even Gaussians
    - where probability is concentrated further and further from the mean
    - and covariance has too many parameters
      - dodge: assume covariance is diagonal
- Idea: reduce the dimension of the feature set
  - Principal components
  - Linear discriminants

Principal components

- Find linear features that explain most of the variance of the data

Assume we have a set of n feature vectors $\boldsymbol{x}_i$ $(i = 1, \ldots, n)$ in $\mathbb{R}^d$. Write

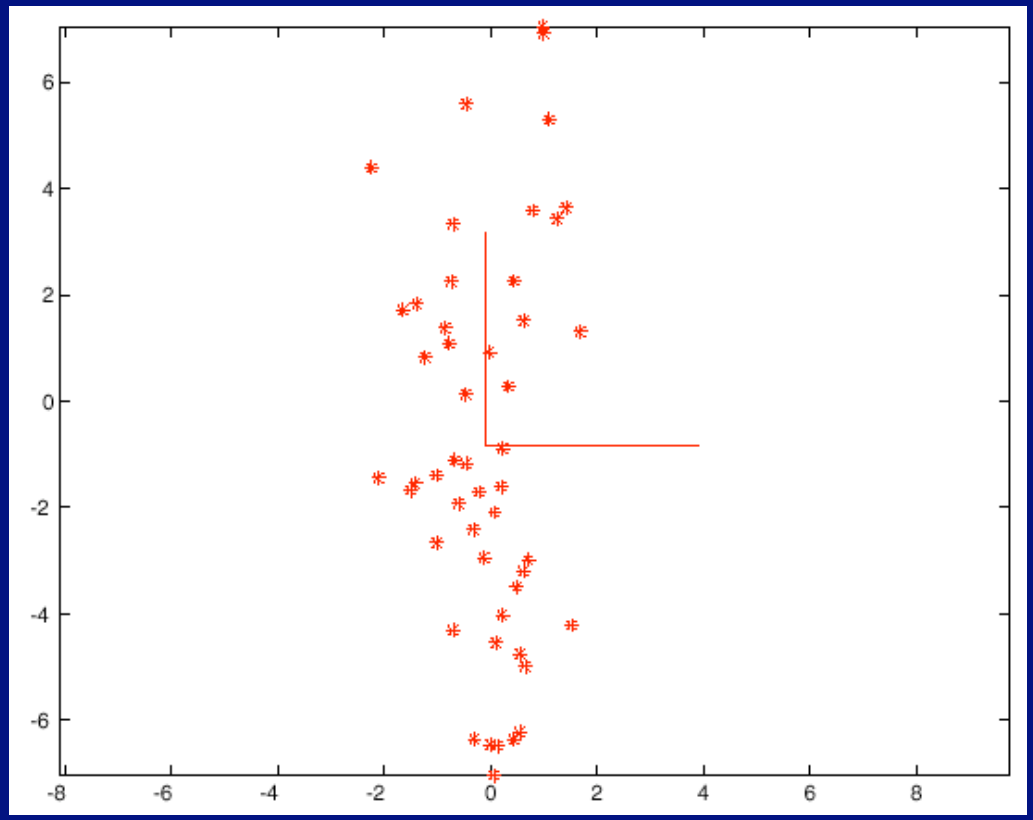
$$\boldsymbol{\mu} = \frac{1}{n} \sum_i \boldsymbol{x}_i$$

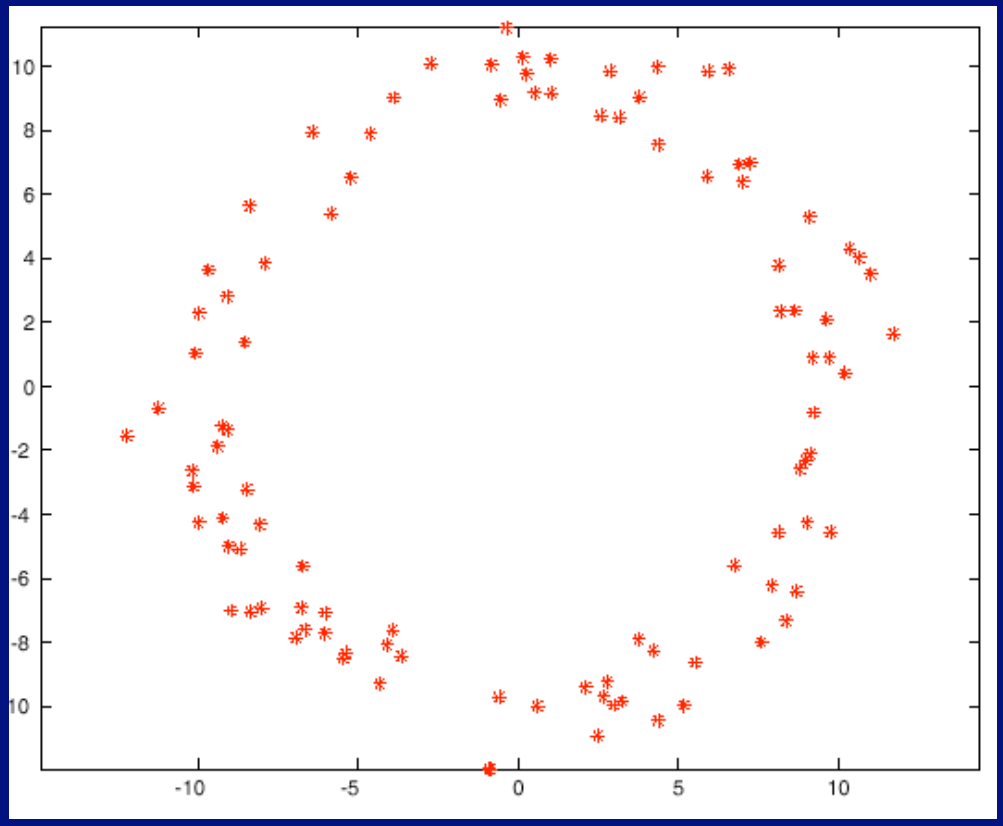
$$\Sigma = \frac{1}{n-1} \sum_i (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T$$

The unit eigenvectors of Σ — which we write as $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d$, where the order is given by the size of the eigenvalue and $\boldsymbol{v}_1$ has the largest eigenvalue — give a set of features with the following properties:

- They are independent.
- Projection onto the basis $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ gives the k-dimensional set of linear features that preserves the most variance.

Algorithm 22.5: *Principal components analysis identifies a collection of linear features that are independent, and capture as much variance as possible from a dataset.*

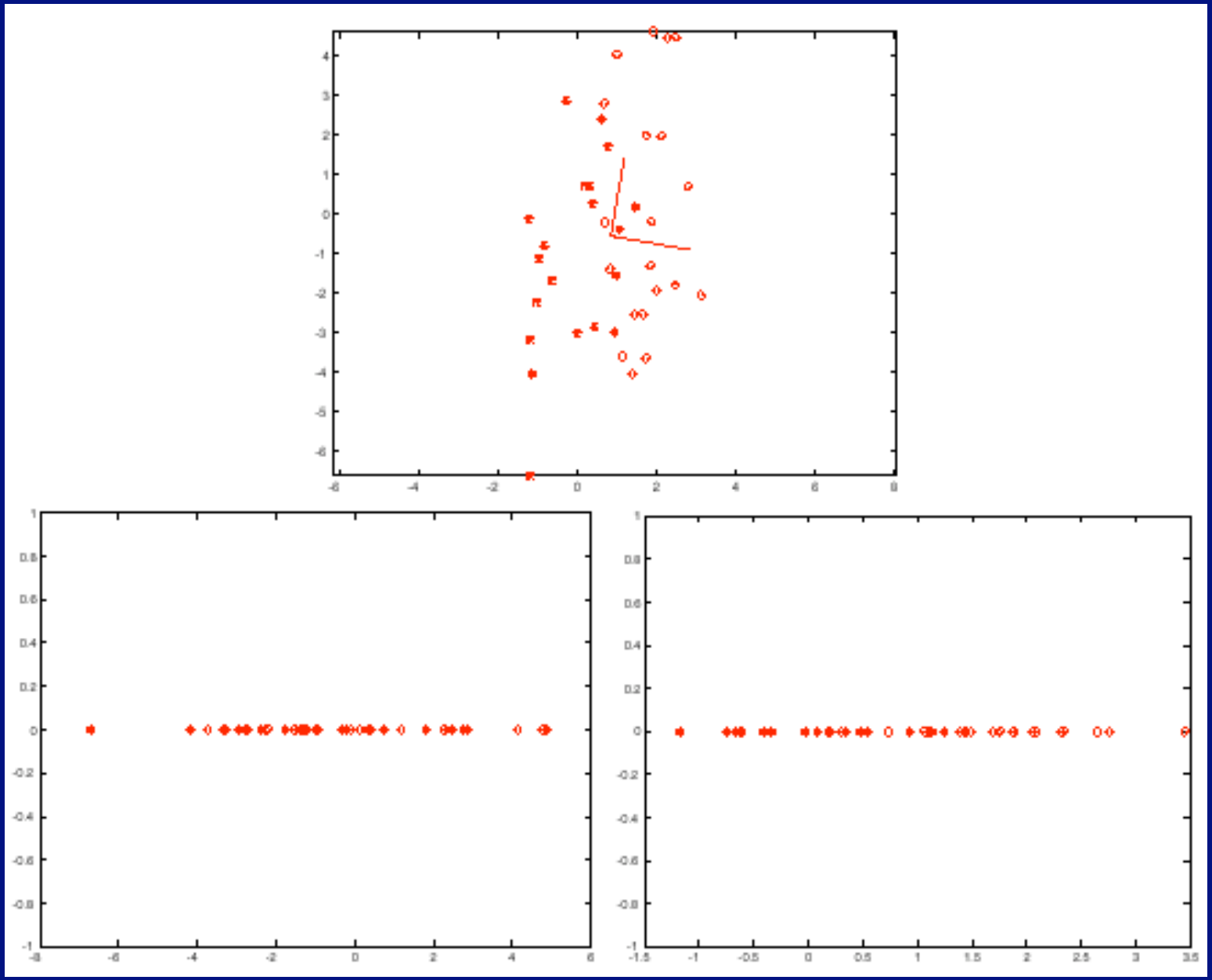

6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8

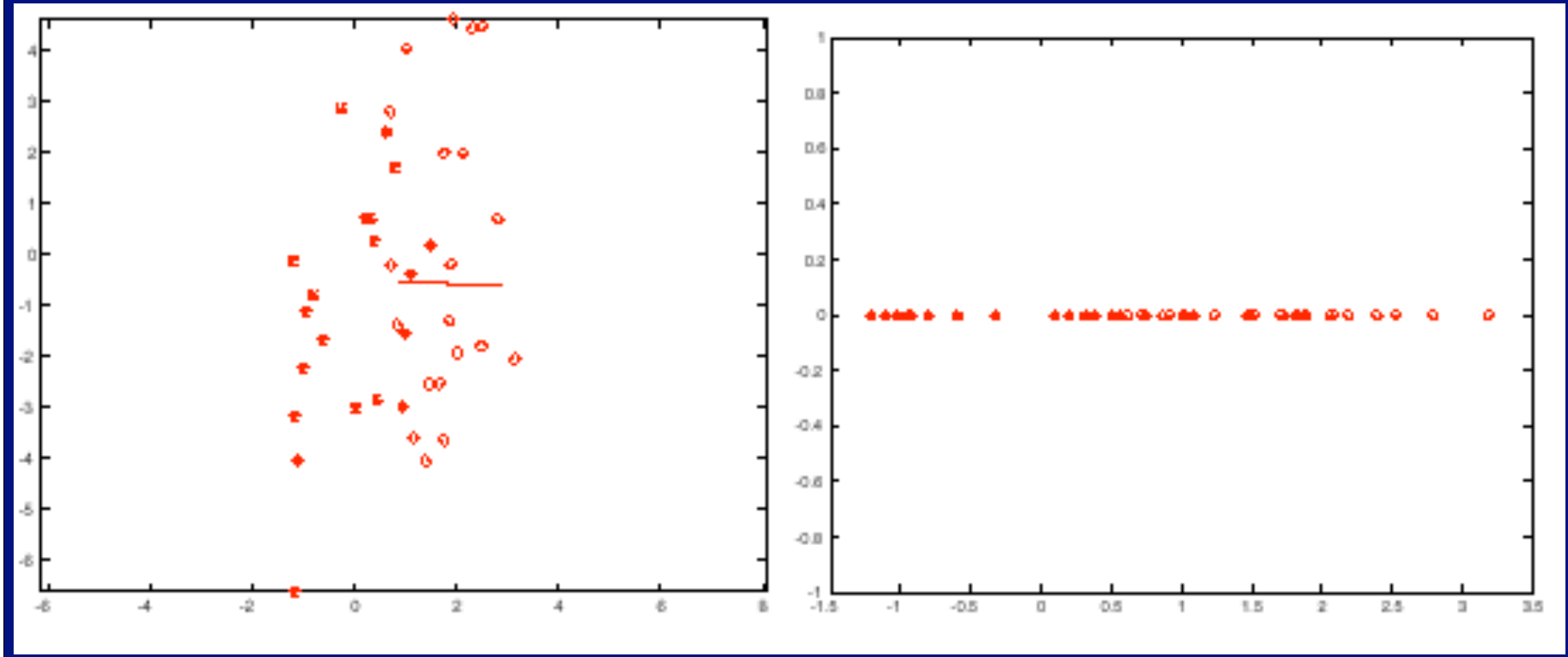

10
8
6
4
2
0
-2
-4
-6
-8
10
-10
-5
0
5
10

Principal components for face images, from
http://vismod.www.media.mit.edu/vismod/demos/facerec/basic.html

Linear discriminant analysis

- Principal components do not preserve discrimination
  - so we could have features that don't distinguish, see picture
- Assume (pretend) class conditional densities are normal, with the same covariance
  - Choose linear features so that
    - between class variation is big compared to within class variation
      - between class variation
        - covariance of class means
      - within class variation
        - class covariance




-6
-4
-2
0
2
4
6
8
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3
3.5

Assume that we have a set of data items of g different classes. There are n_k items in each class, and a data item from the k'th class is $\boldsymbol{x}_{k,i}$, for $i \in \{1, \ldots, n_k\}$. The j'th class has mean $\boldsymbol{\mu}_j$. We assume that there are p features (i.e. that the $\boldsymbol{x}_i$ are p-dimensional vectors).
Write $\overline{\boldsymbol{\mu}}$ for the mean of the class means, i.e.

$$\overline{\boldsymbol{\mu}} = \frac{1}{g} \sum_{j=1}^{g} \boldsymbol{\mu}_j$$

Write

$$\mathcal{B} = \frac{1}{g-1} \sum_{j=1}^{g} (\boldsymbol{\mu}_j - \overline{\boldsymbol{\mu}})(\boldsymbol{\mu}_j - \overline{\boldsymbol{\mu}})^T$$

Assume that each class has the same covariance Σ, which is either known or estimated as

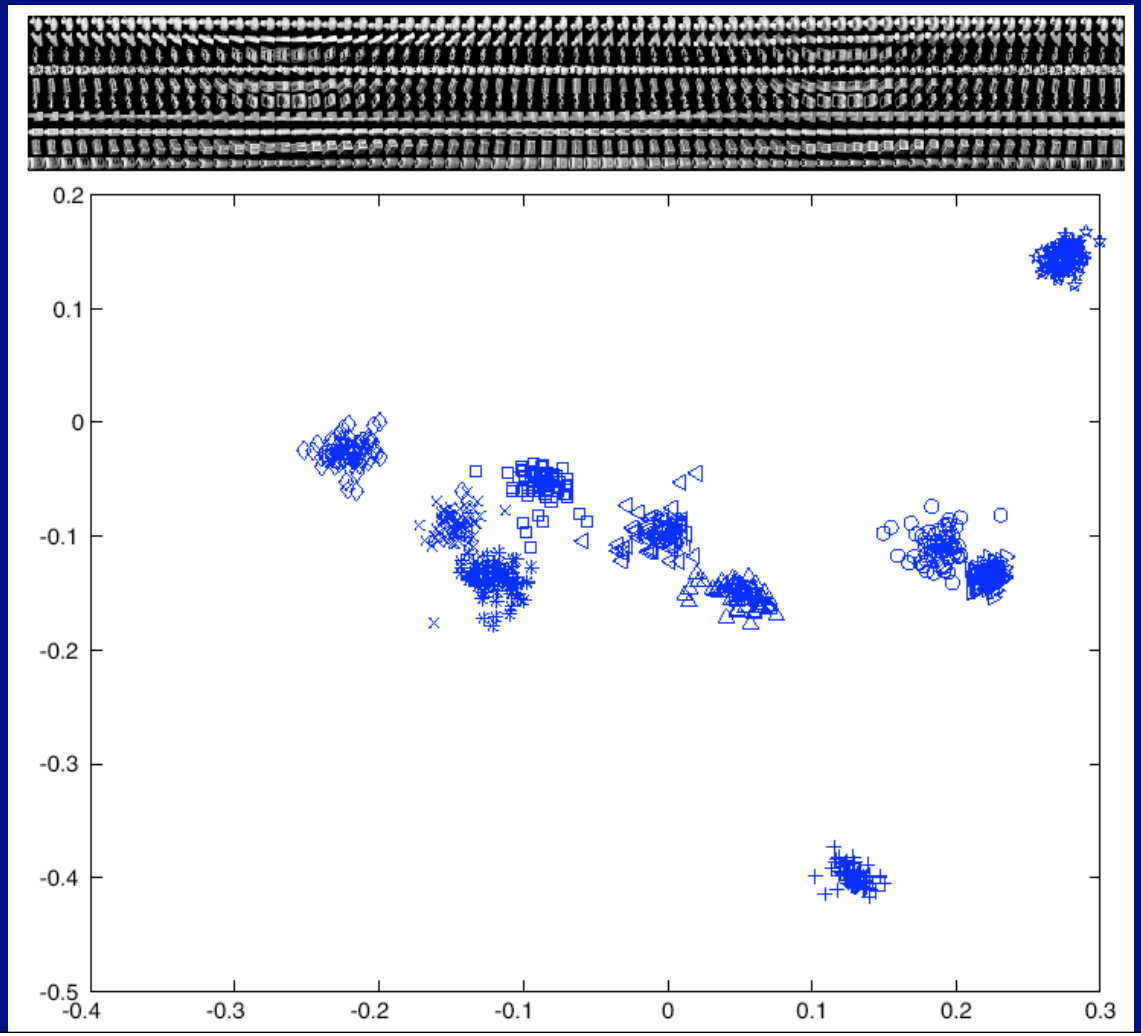
$$\Sigma = \frac{1}{N-1} \sum_{c=1}^{g} \left\{ \sum_{i=1}^{n_c} (\boldsymbol{x}_{c,i} - \boldsymbol{\mu}_c)(\boldsymbol{x}_{c,i} - \boldsymbol{\mu}_c)^T \right\}$$

The unit eigenvectors of $\Sigma^{-1}\mathcal{B}$ — which we write as $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d$, where the order is given by the size of the eigenvalue and $\boldsymbol{v}_1$ has the largest eigenvalue — give a set of features with the following property:

- Projection onto the basis $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ gives the k-dimensional set of linear features that best separates the class means.

Algorithm 22.6: *Canonical variates identifies a collection of linear features that separating the classes as well as possible.*

First two canonical variates for well known image collection



Back to image retrieval

- Whole image queries don’t work that well
- Alternatives:
  - segment image, query on segment matches (Blobworld)
  - search with words

What will users pay for?

- Work by Peter Enser and colleagues on the use of photø movie collections
  (Enser McGregor 92; Ornager 96; Armitage Enser 97; Markkula Sormunen 00; Frost et al 00; Enser 00)
- Typical queries:

What is this about?

"… smoking of kippers…"

"The depiction of vanity in painting, the depiction of the female figure looking in the mirror, etc."

"Cheetahs running on a greyhound course in Haringey in 1932"

Query on

Example from Berkeley
Blobworld system

Query on

“Rose”

Example from Berkeley
Blobworld system







Annotation results in complementary words and pictures

Query on

“Rose”

and

Example from Berkeley Blobworld system

Searching with words

- Most pictures don’t have words “attached”
- Attach in simple ways
  - in image name
  - in caption
  - in nearby text
- All really useful, but dangerous
  - 12739.jpg?
  - common to have pictures on web pages without easily identified captions
  - nearby text might, might not, be relevant
- Predict annotations from picture

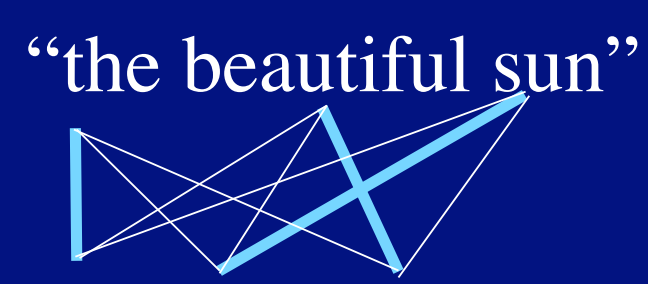
Annotation vs Recognition

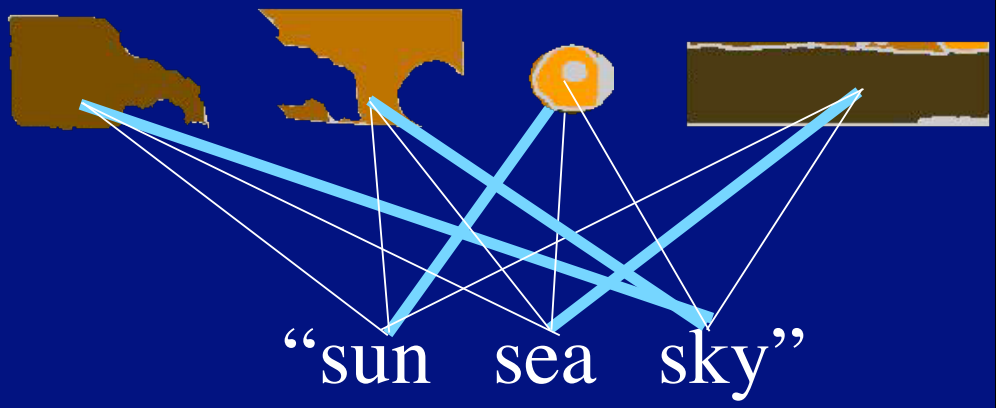
?

tiger cat grass

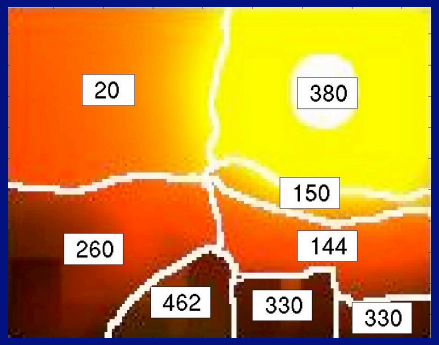
Lexicon building

- In its simplest form, missing variable problem
- Pile in with EM
  - given correspondences, conditional probability table is easy (count)
  - given cpt, expected correspondences could be easy
- Caveats
  - might take a lot of data; symmetries, biases in data create issues

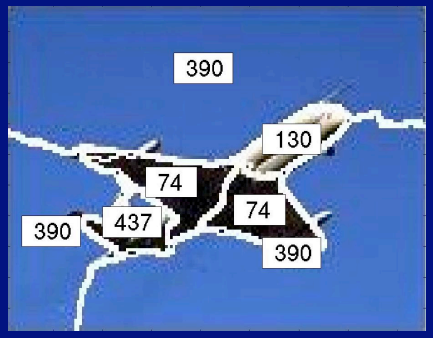




Brown, Della Pietra, Della Pietra & Mercer 93; Melamed 01



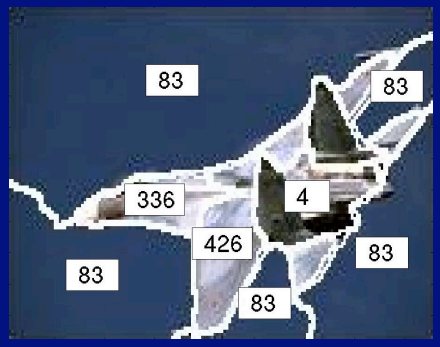
city mountain sky sun



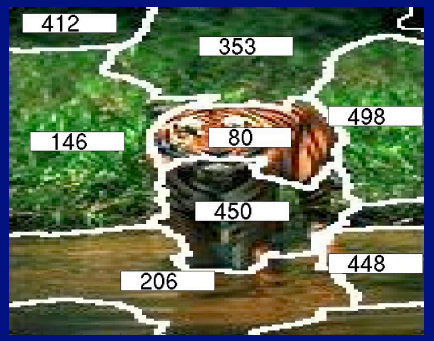
jet plane sky

cat forest grass tiger

beach people sun water



jet plane sky



cat grass tiger water

“Lexicon” of “meaning”

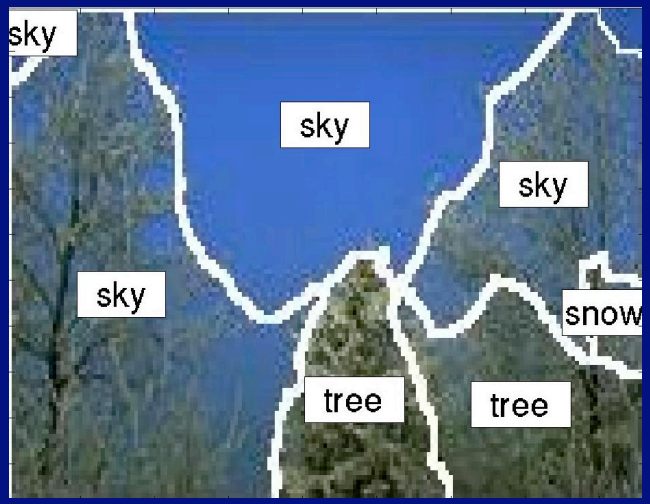
sun

sky

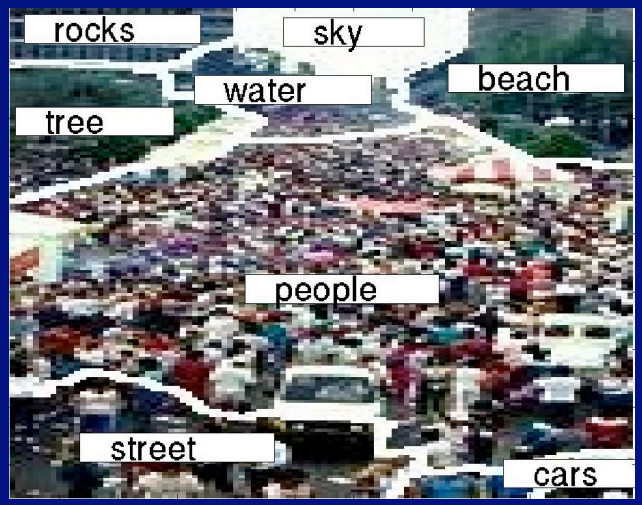
cat

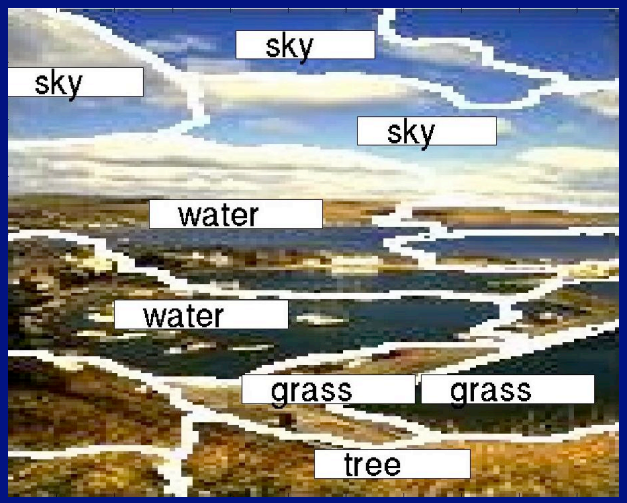
horse

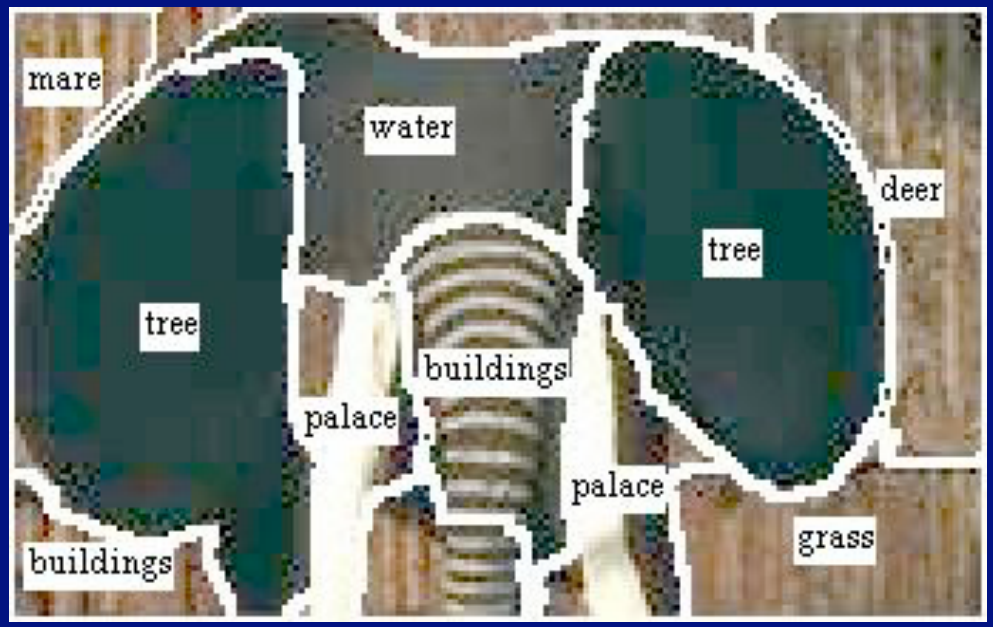
This could be either a conditional probability table or a joint probability table; each has significant attractions for different applications


sky
sky
sky
sky
snow
tree
tree


sky
sky
water
buildings
sky
tree
buildings
buildings


rocks
sky
water
beach
tree
people
street
cars


sky
sky
sky
water
water
grass
grass
tree

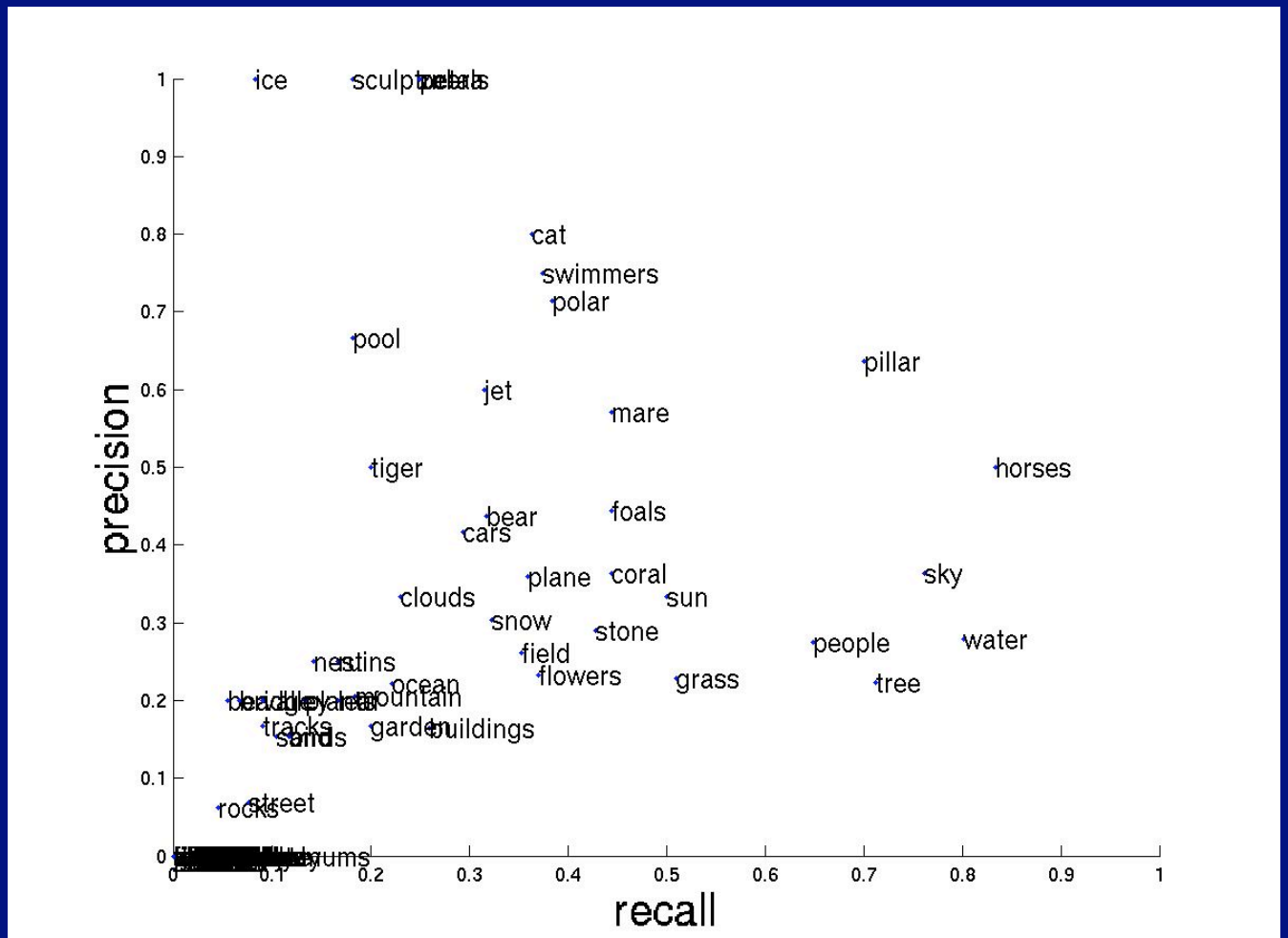

mare
water
deer
tree
tree
buildings
palace
palace
grass
buildings

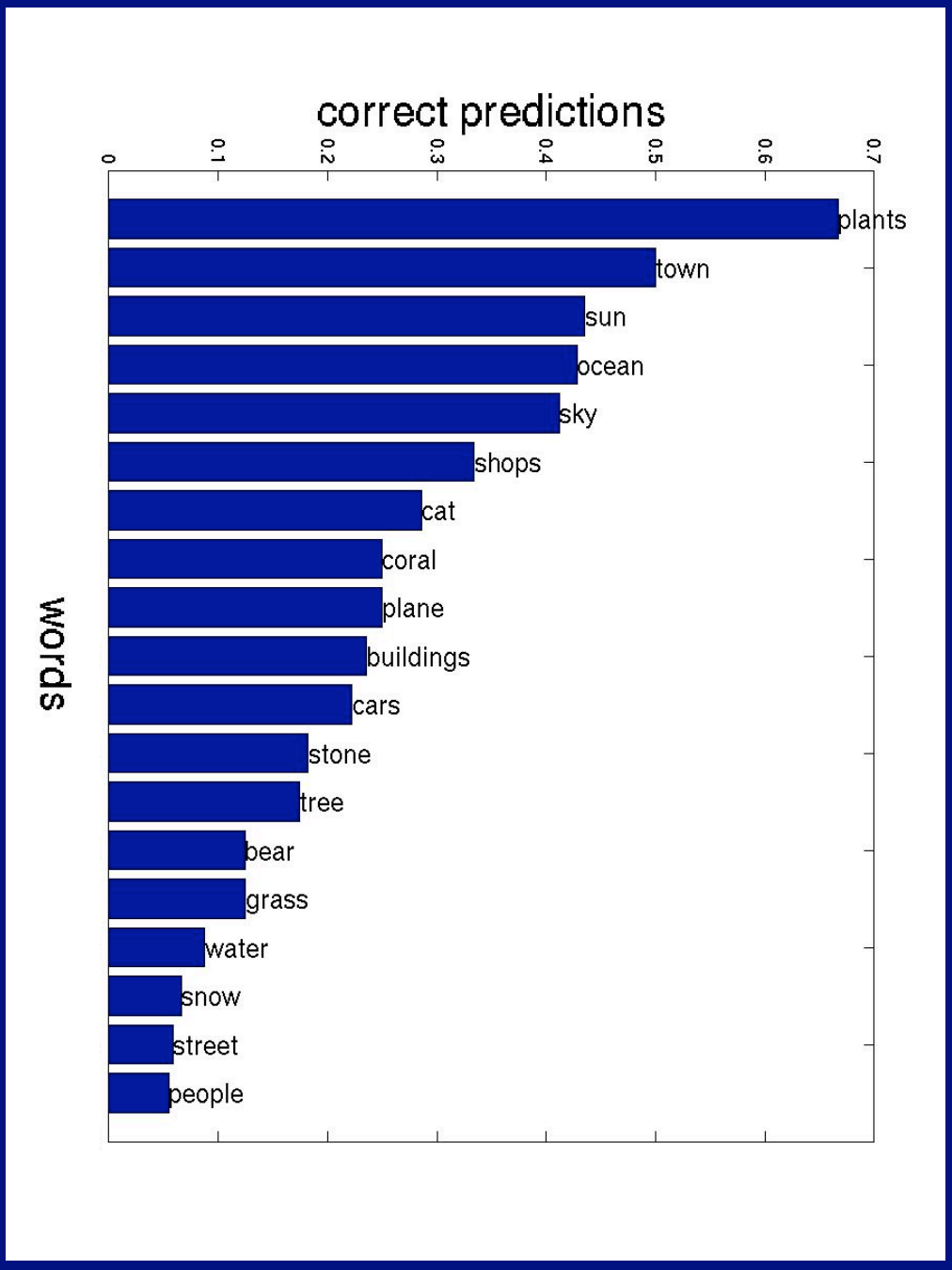
Performance measurement

By hand

By proxy

Grass Cat Buildings
Horses Tiger Mare


precision
recall
ice
cat
swimmers
polar
pool
pillar
jet
mare
tiger
horses
bear
foals
cars
plane
coral
sky
clouds
sun
snow
stone
people
water
field
flowers
grass
tree
ocean
mountain
tracks
garden
buildings
rocks
street
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


correct predictions
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
words
plants
town
sun
ocean
sky
shops
cat
coral
plane
buildings
cars
stone
tree
bear
grass
water
snow
street
people

Precision and recall

- Precision
  - Percentage of the retrieved items that are relevant
- Recall
  - Percentage of the relevant items that were retrieved
- Importance varies by application
  - high recall: patent search
  - high precision: celebrity search

Translation isn't that great

Method	P	R	F1	Ref
Co-occ	0.03	0.02	0.02	[53]
Trans	0.06	0.04	0.05	[27]
CMRM	0.10	0.09	0.10	[37]
TSIS	0.10	0.09	0.10	[19]
MaxEnt	0.09	0.12	0.10	[39]
CRM	0.16	0.19	0.17	[44]
CT-3×3	0.18	0.21	0.19	[82]
CRM-rect	0.22	0.23	0.23	[31]
InfNet	0.17	0.24	0.23	[50]
MBRM	0.24	0.25	0.25	[31]
MixHier	0.23	0.29	0.26	[17]
(section 2.2)	0.27	0.27	0.27	
(section 2.2, kernel)	0.29	0.29	0.29	
PicSOM	0.35*	0.35*	0.35*	[73]

Discriminative annotation

- Idea:
  - use a linear SVM to predict each word from image features
- Issues:
  - training data tends to be noisy
  - awful lot of SVM's
  - words tend to be correlated

Discriminative annotation

- Word matrix $\mathcal{D}$
  - d_ij is 1 if j'th image has i'th word
- Feature matrix $\mathcal{X}$
  - x_ij is i'th feature of j'th image

- Problem

$$\text{predict } \mathcal{D} \text{ with } \mathcal{C}\mathcal{X}$$

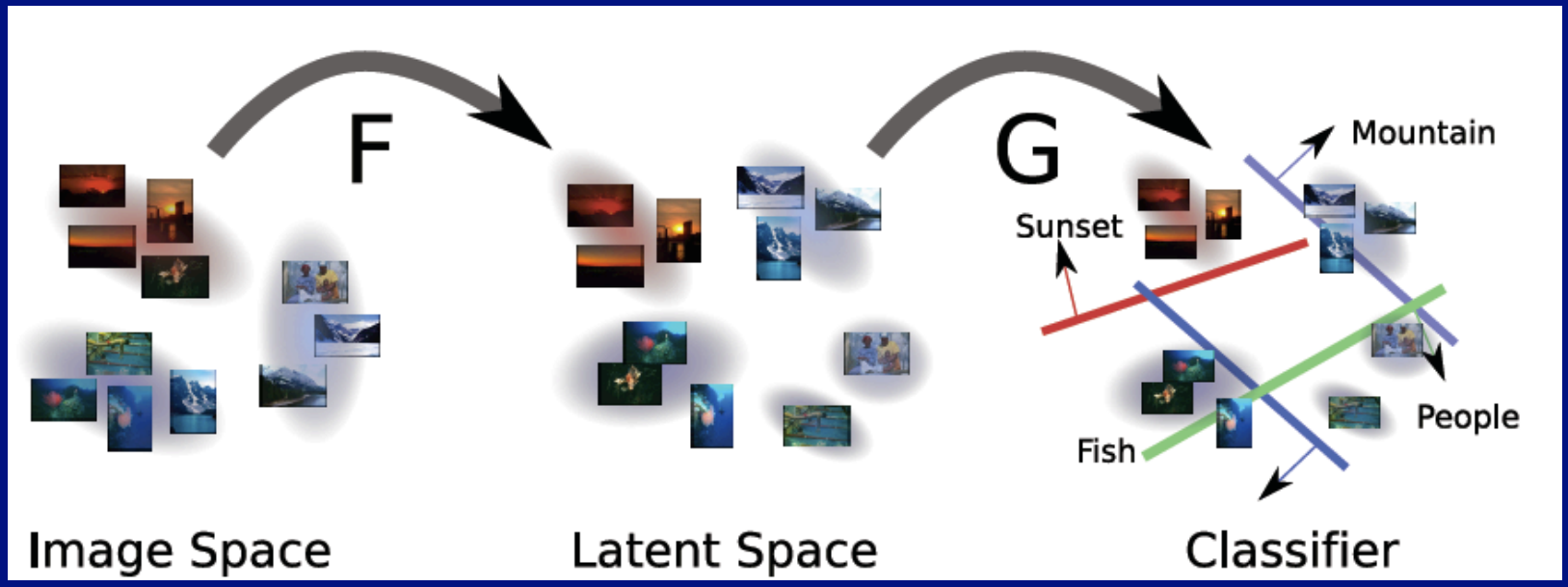
- loss

$$L_h((1 - \text{trace}(\mathcal{D}(\mathcal{C}\mathcal{X}))))$$

Discriminative annotation

- But what is the penalty?
- C should not be “too big”
  - so we have a margin
- C should not have “too high a rank”
  - because words are correlated
  - because good features are reused
  - or a kind of projection
- Trace norm
  - where the sigma are singular values

$$\|\mathcal{C}\|_{tr} = \sum_i |\sigma_i|$$



Train a system of svm classifiers, one per word but penalize that matrix for rank, after Rennie+Srebro 05

The latent space reveals scenes because it is good at word prediction and takes appearance into account

Loeff Farhadi Forsyth??

It was there and we didn't

It was there and we predicted it

It wasn't and we did

Loeff Farhadi Forsyth??

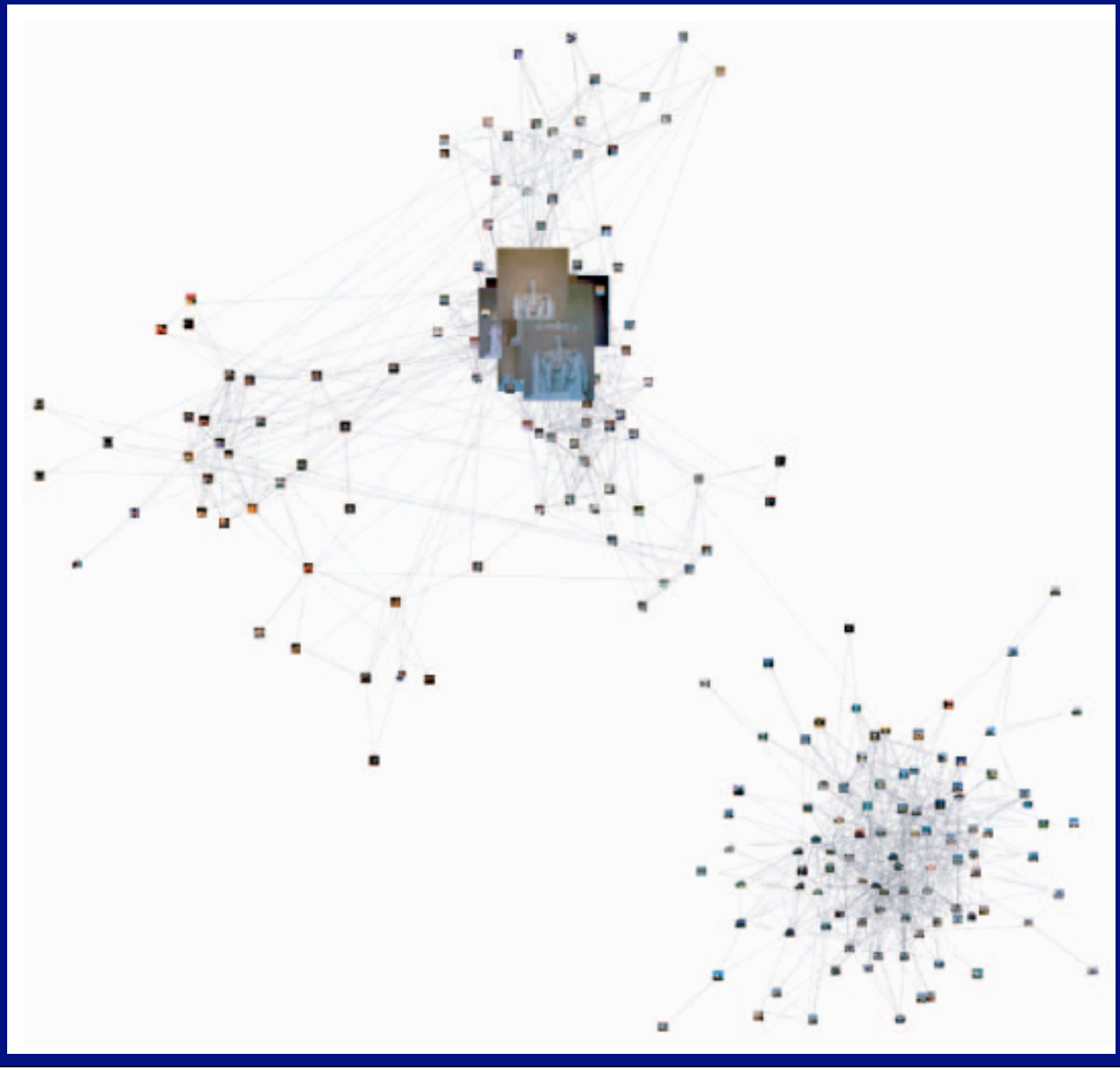
Correlated annotations are better

Method	P	R	F1	Ref
Co-occ	0.03	0.02	0.02	[53]
Trans	0.06	0.04	0.05	[27]
CMRM	0.10	0.09	0.10	[37]
TSIS	0.10	0.09	0.10	[19]
MaxEnt	0.09	0.12	0.10	[39]
CRM	0.16	0.19	0.17	[44]
CT-3×3	0.18	0.21	0.19	[82]
CRM-rect	0.22	0.23	0.23	[31]
InfNet	0.17	0.24	0.23	[50]
MBRM	0.24	0.25	0.25	[31]
MixHier	0.23	0.29	0.26	[17]
(section 2.2)	0.27	0.27	0.27	
(section 2.2, kernel)	0.29	0.29	0.29	
PicSOM	0.35*	0.35*	0.35*	[73]

Reranking

- Idea:
  - Once we have a set of search results, find the “important” ones
- Motivation:
  - in image search, precision matters, recall doesn’t
    - (usually?)
- What is an “important” image?
  - one that is similar in many features to many of the pictures returned
- Build a graph linking search results, find important ones
  - links based on local features
    - (e.g. SIFT features at big interest points are the same)

The largest number of neighbours is not a good “importance” test because it tends to find large clusters of very similar images and ignore large scale structure



Ying Baluja 08

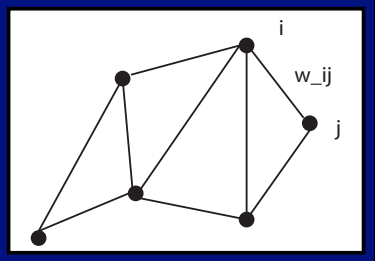
The largest number of neighbours is not a good “importance” test because it tends to find large clusters of very similar images and ignore large scale structure

Ying Baluja 08

Measuring “importance”

- graph is weighted
  - by the number of interest points that match
- Model:
  - random walk on weighted graph
    - a high probability of arriving at an image, it’s important
    - i.e. degree counts, but so do weights on links and degree of neighbours
-

Random walk on a graph



- state at k'th step = x^k
  - which is a node on the graph
- Represent connections in graph with transition matrix

$$M_{ij} = P(x^{(k+1)} = i | x^{(k)} = j) \propto w_{ij}$$

- Notice that

$$\sum_i M_{ij} = 1$$

$$\text{if } P(x^{(k)}) = p^{(k)} \text{ then } P(x^{(k+1)}) = Mp^{(k)}$$

Random walk on a graph - II

- So

$$\text{if } P(x^{(0)}) = \pi \text{ then } P(x^{(k)}) = M^k \pi$$

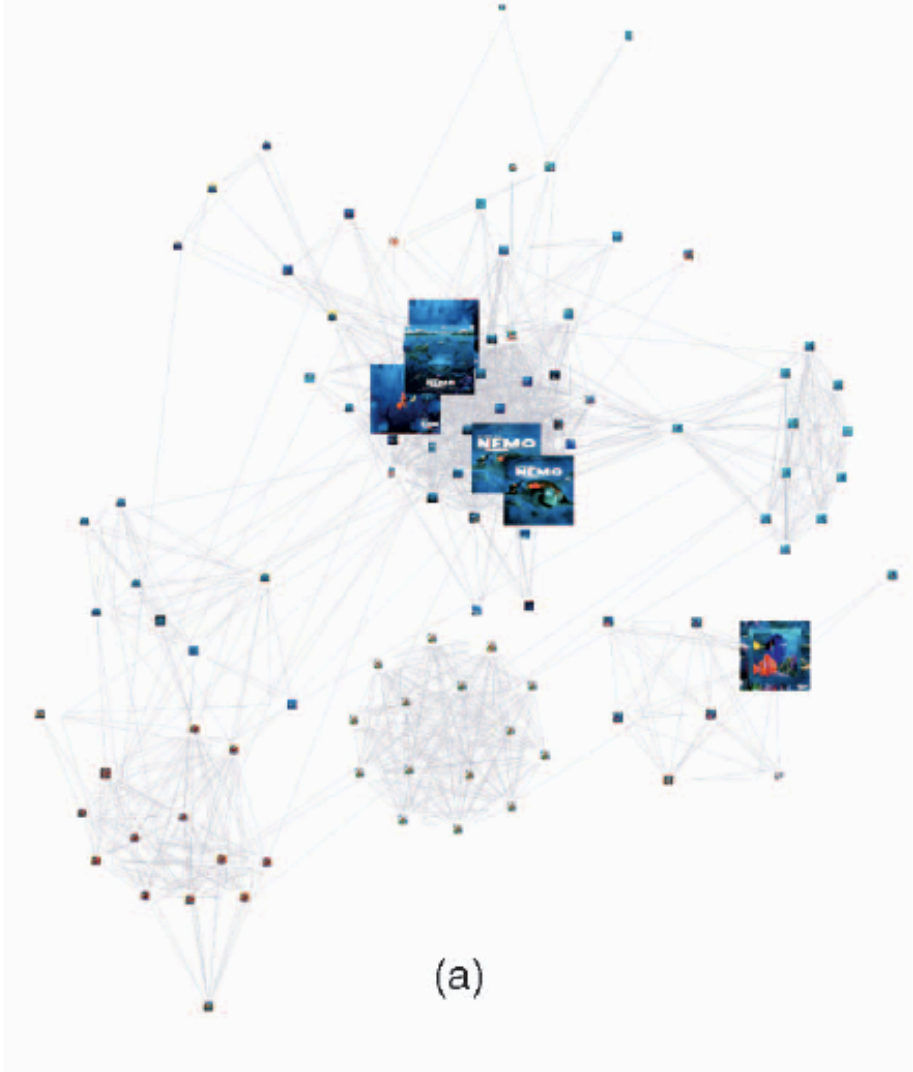
- Under simple conditions on M, we have

$$\lim_{k \to \infty} M^k \pi = s$$

- s is known as the stationary distribution
- s(node) is its importance

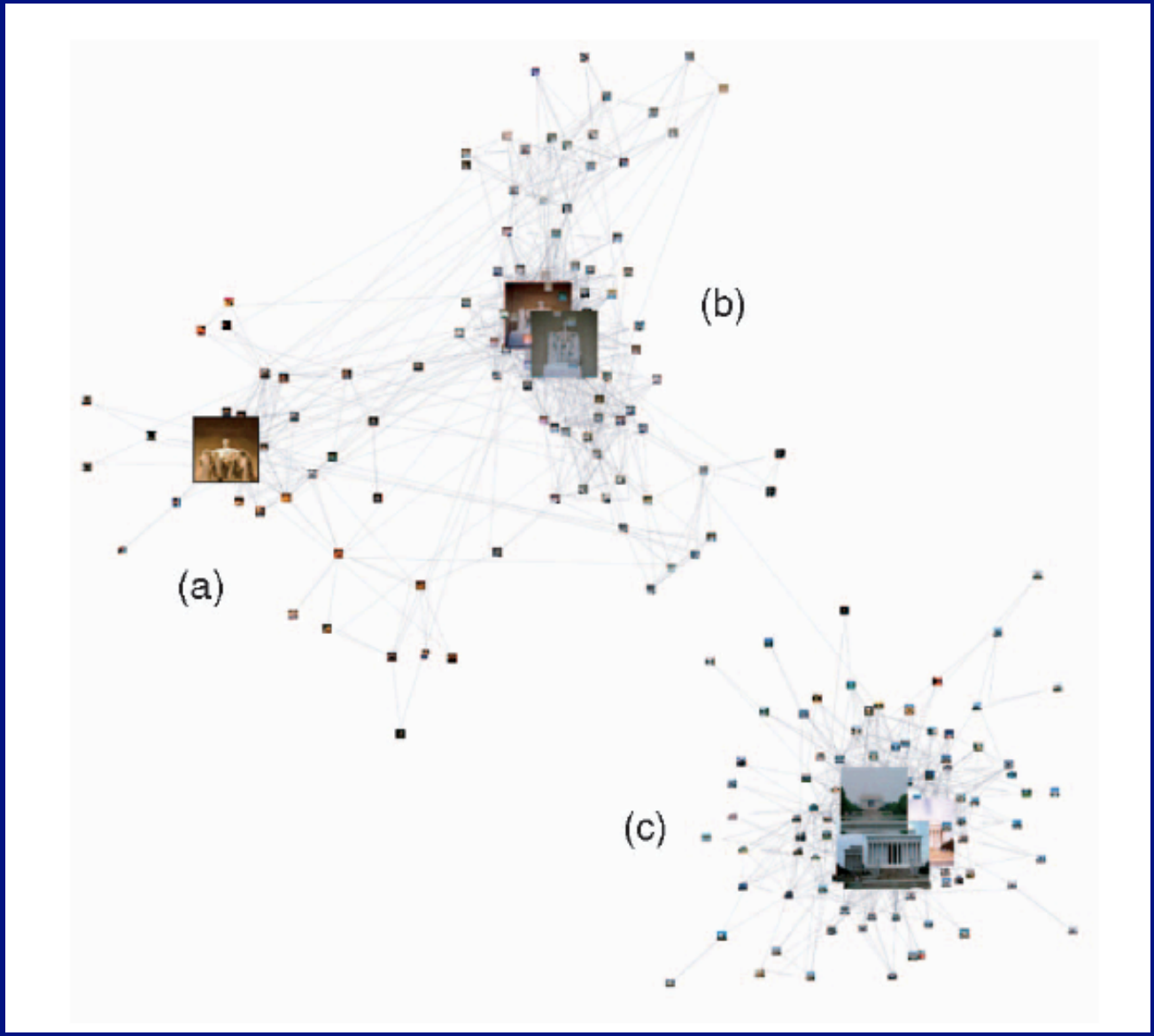
Random walk on a graph - III

- small graphs: s is eigenvector of M with eval = 1
- big graphs:
  - can't build M
  - method:
    - use a hash table to build the links
    - simulate the random walk
      - trick: at any node, with small probability transition to any other
        - otherwise, follow weights
      - trick: exploit the hash table
        - for transition, choose feature, then choose collisions in hash bucket
  - importance is frequency with which one visits points



Ying Baluja 08

Ying Baluja 08



Ying Baluja 08