Learning distance functions

Xin Sui
CS395T Visual Recognition and Search
The University of Texas at Austin

Outline

- Introduction
- Learning one Mahalanobis distance metric
- Learning multiple distance functions
- Learning one classifier represented distance function
- Discussion Points

Outline

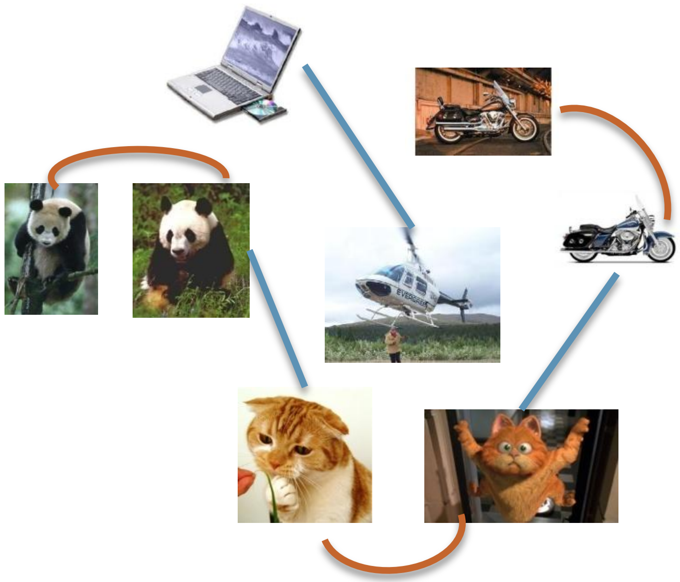
- **Introduction**
- Learning one Mahalanobis distance metric
- Learning multiple distance functions
- Learning one classifier represented distance function
- Discussion Points

Distance function vs. Distance Metric

- Distance Metric:
  - Satisfy non-negativity, symmetry and triangle inequation
- Distance Function:
  - May not satisfy one or more requirements for distance metric
  - More general than distance metric

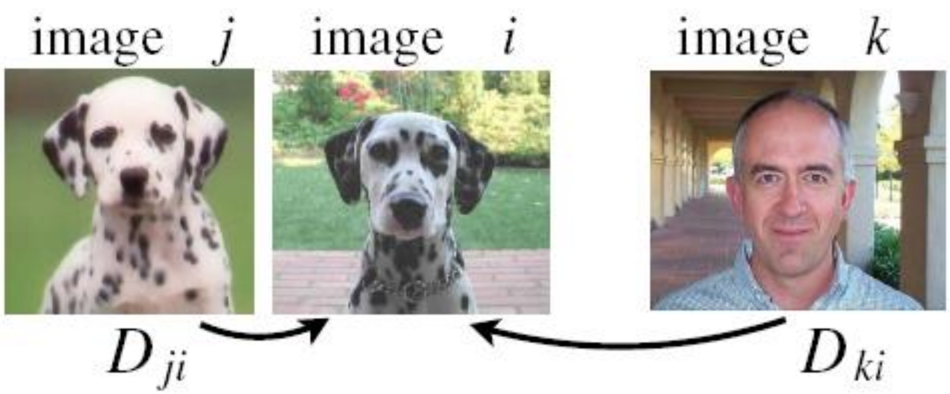
Constraints

- Pairwise constraints
  - Equivalence constraints
    - Image i and image j is similar
  - Inequivalence constraints
    - Image i and image j is not similar



Red line: equivalence constraints
Blue line: in-equivalence constraints

- Triplet constraints
  - Image j is more similar to image i than image k



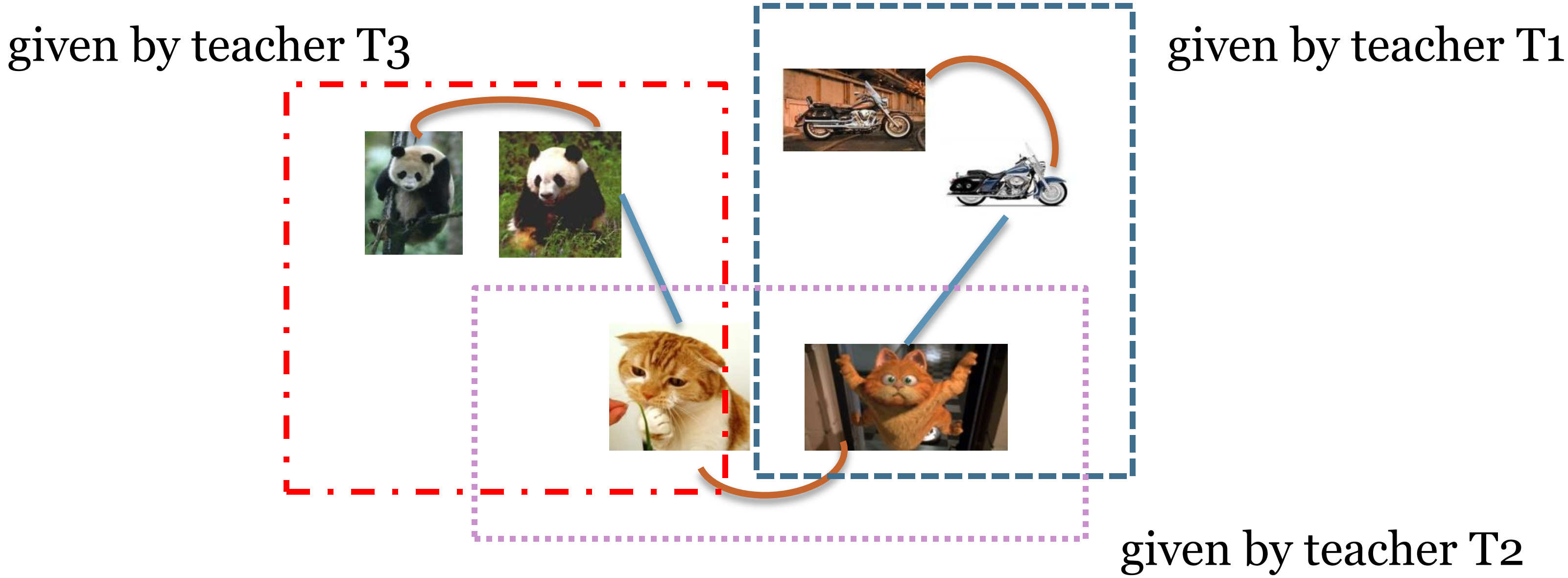
Constraints are the supervised knowledge for the distance learning methods

Why not labels?

- Sometimes constraints are easier to get than labels
  - faces extracted from successive frames in a video in roughly the same location can be assumed to come from the same person

Why not labels?

- Sometimes constraints are easier to get than labels
  - Distributed Teaching
    - Constraints are given by teachers who don't coordinate with each other



Why not labels?

- Sometimes constraints are easier to get than labels
  - Search engine logs

Problem

- Given a set of constraints
- Learn one or more distance functions for the input space of data from that preserves the distance relation among the training data pairs

Importance

- Many machine learning algorithms, heavily rely on the distance functions for the input data patterns. e.g. kNN
- The learned functions can significantly improve the performance in classification, clustering and retrieval tasks:

  e.g. KNN classifier, spectral clustering, content-based image retrieval (CBIR).

Outline

- Introduction
- **Learning one Mahalanobis distance metric**
  - **Global methods**
  - **Local methods**
- Learning one classifier represented distance function
- Discussion Points

Parameterized Mahalanobis Distance Metric

$$d(x, y) = d_A(x, y) = ||x - y||_A = \sqrt{(x-y)^T A (x-y)}.$$

x, y: the feature vectors of two objects,
for example, a words-of-bag representation of an image

Parameterized Mahalanobis Distance Metric

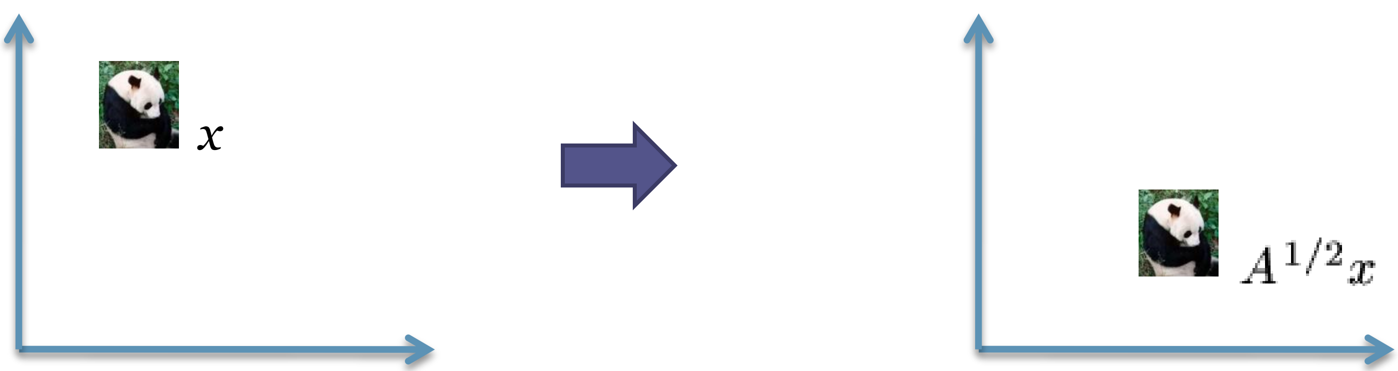
$$d(x,y) = d_A(x,y) = ||x-y||_A = \sqrt{(x-y)^T A (x-y)}.$$

To be a metric, A must be semi-definite

Parameterized Mahalanobis Distance Metric

$$d(x, y) = d_A(x, y) = ||x - y||_A = \sqrt{(x-y)^T A (x-y)}.$$

It is equivalent to finding a rescaling of a data that replaces each point x with $A^{1/2}x$ and applying standard Euclidean distance



Parameterized Mahalanobis Distance Metric

$$d(x,y) = d_A(x,y) = ||x-y||_A = \sqrt{(x-y)^T A (x-y)}.$$

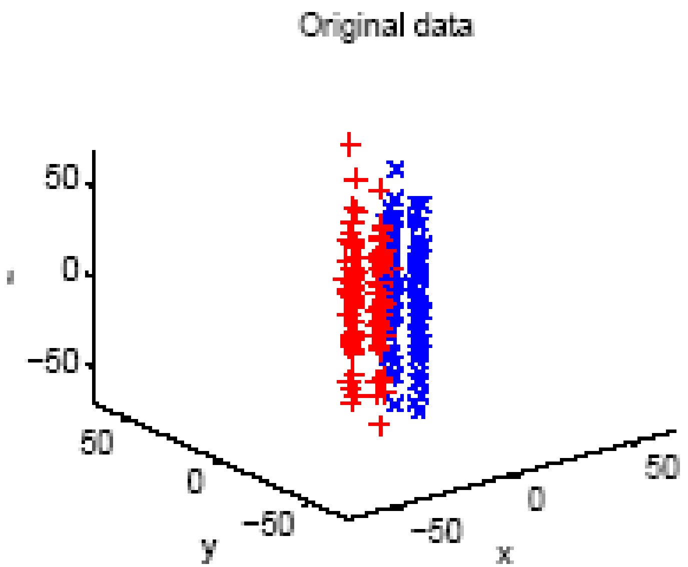
- If A=I, Euclidean distance
- If A is diagonal, this corresponds to learning a metric in which the different axes are given different "weights"

Global Methods

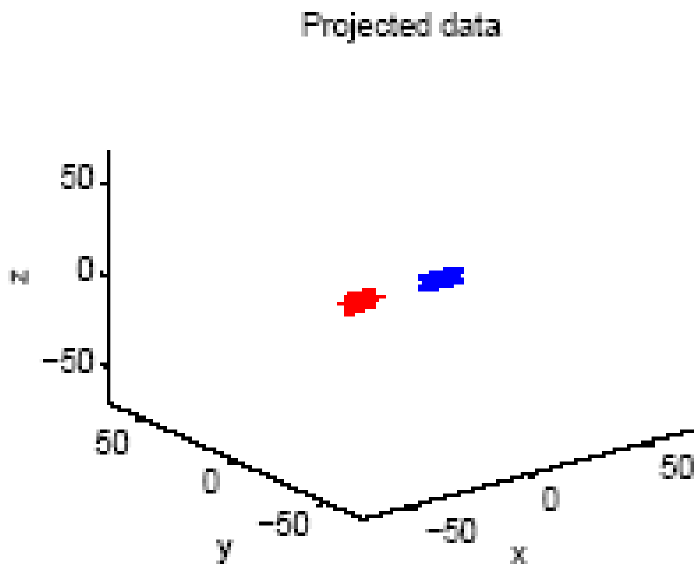
- Try to satisfy *all the constraints simultaneously*
  - keep ***all*** the data points within the same classes close, while separating ***all*** the data points from different classes

- Distance Metric Learning, with Application to Clustering with Side-information [Eric Xing . Et, 2003]

A Graphical View



(a) Data Dist. of the original dataset



(b) Data scaled by the global metric

- Keep *all* the data points within the same classes close
- Separate *all* the data points from different classes

(the figure from [Eric Xing . Et, 2003])

Pairwise Constraints

- A set of Equivalence constraints

$$S = \{(x_i, x_j) | x_i \text{ and } x_j \text{ are similar}\}$$

- A set of In-equivalence constraints

$$D = \{(x_i, x_j) | x_i \text{ and } x_j \text{ are dissimilar}\}$$

The Approach

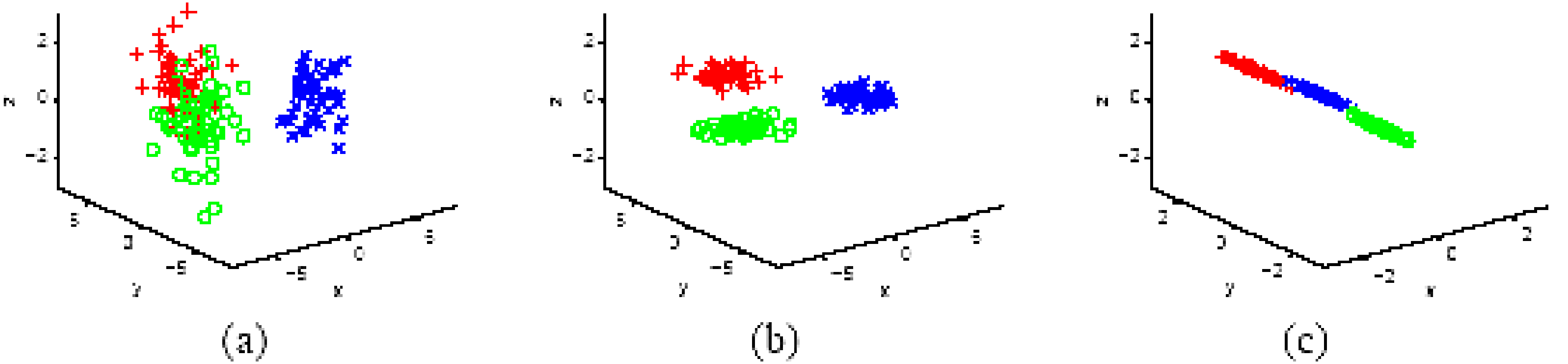
- Formulate as a constrained convex programming problem
  - Minimize the distance between the data pairs in S
  - Subject to data pairs in D are well separated

$$\begin{aligned} \min_{A} \quad & \sum_{(x_i, x_j) \in \mathcal{S}} ||x_i - x_j||_A^2 \\ \text{s.t.} \quad & \sum_{(x_i, x_j) \in \mathcal{D}} ||x_i - x_j||_A \geq 1, \\ & A \succeq 0. \end{aligned}$$

ensure that A does not collapse the dataset to a single point

- Solving an iterative gradient ascent algorithm

Another example



(a) Original data

(b) Rescaling by learned diagonal A

(c) rescaling by learned full A

(the figure from [Eric Xing . Et, 2003])

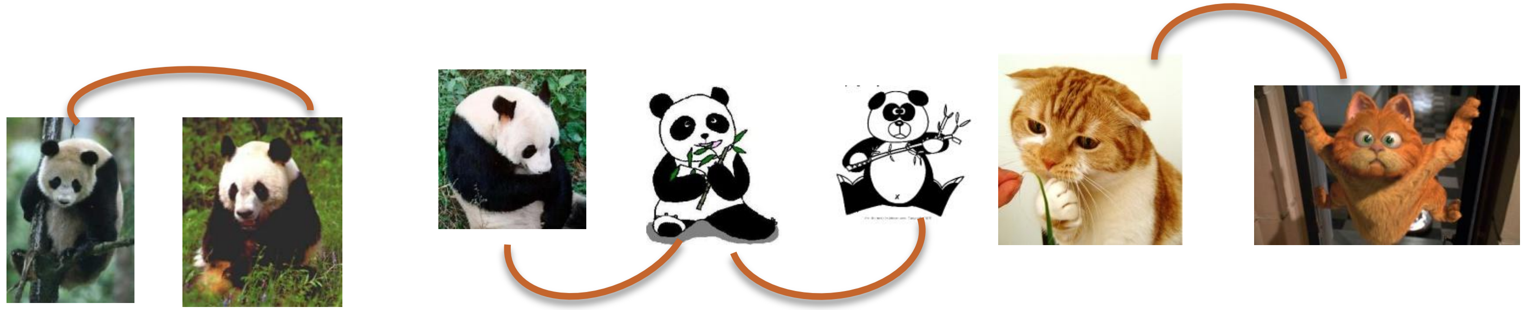
RCA

- Learning a Mahalanobis Metric from Equivalence Constraints [BAR HILLEL, et al. 2005]

RCA(Relevant Component Analysis)

- Basic Ideas
  - Changes the feature space by assigning large weights to “relevant dimensions” and low weights to “irrelevant dimensions”.
  - These “relevant dimensions” are estimated using equivalence constraints

Another view of equivalence constraints: chunklets

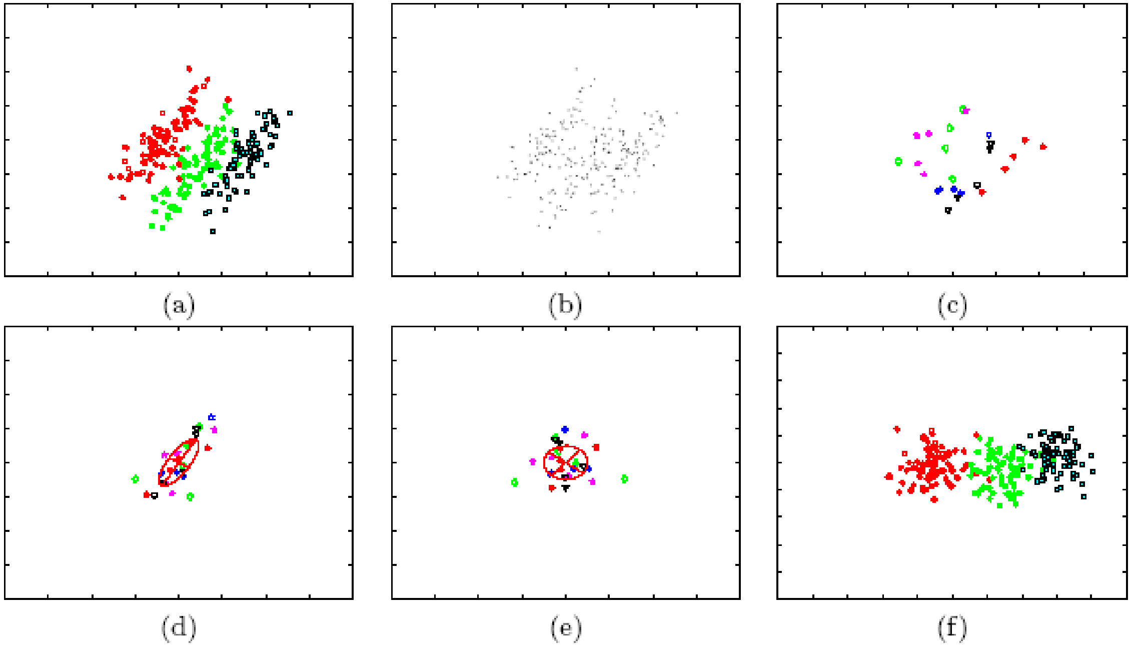


Equivalence constraints

Chunklets formed by applying transitive closure

Estimate the within class covariance
dimensions correspond to large with-in covariance are not relevant
dimensions correspond to small with-in covariance are relevant

Synthetic Gaussian data



(a) The fully labeled data set with 3 classes.
(b) Same data unlabeled; classes' structure is less evident.
(c) The set of chunklets that are provided to the RCA algorithm
(d) The centered chunklets, and their empirical covariance.
(e) The RCA transformation applied to the chunklets. (centered)
(f) The original data after applying the RCA transformation.

(BAR HILLEL, et al. 2005)

RCA Algorithm

- Sum of in-chunklet covariance matrices for p points in k chunklets

$$\hat{C} = \frac{1}{p}\sum_{j=1}^{k}\sum_{i=1}^{n_j}(x_{ji} - \hat{m}_j)(x_{ji} - \hat{m}_j)^T, \text{ chunklet } j : \{x_{ji}\}_{i=1}^{n_j}, \text{ with mean } \hat{m}_j$$

- Compute the whitening transformation associated with $\hat{C}$: $W = \hat{C}^{-\frac{1}{2}}$, and apply it to the data points, Xnew = WX
  - (The whitening transformation *W assigns* lower weights to directions of large variability)

Applying to faces

Top: facial images of two subjects under different lighting conditions.
Bottom: the same images from the top row after applying PCA and RCA and then reconstructing the images

RCA dramatically reduces the effect of different lighting conditions, and the reconstructed images of each person look very similar to each other. [Bar-Hillel, et al. , 2005]

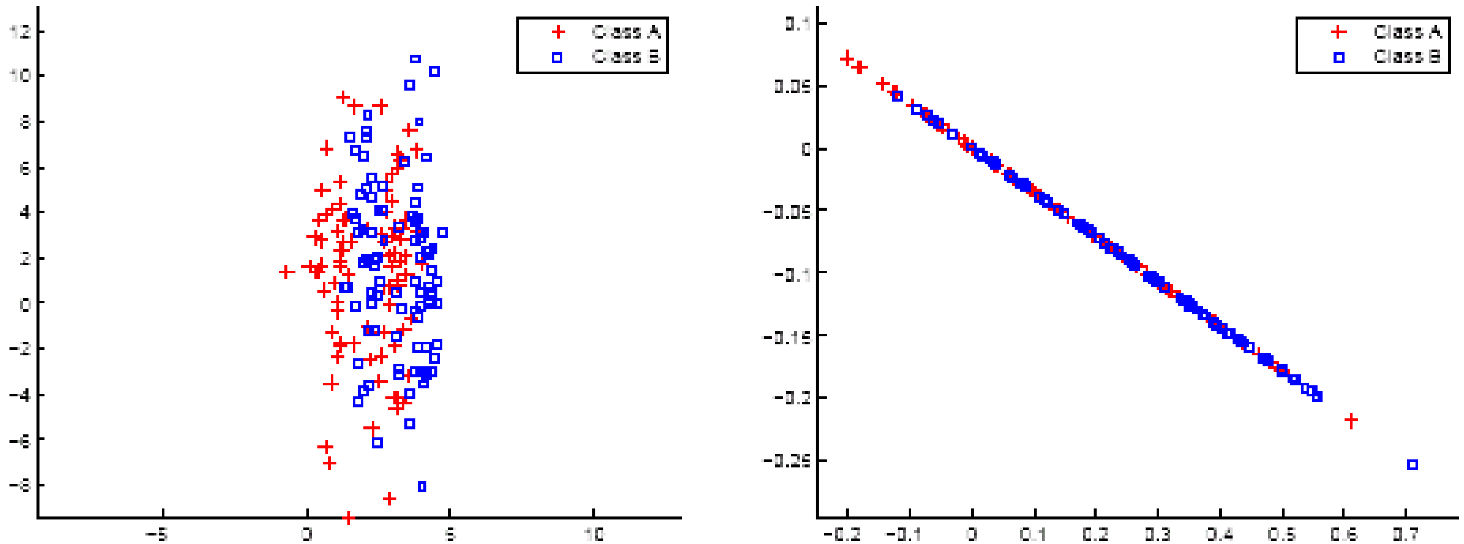
Comparing Xing's method and RCA

- Xing's method
  - Use both equivalence constraints and in-equivalence constraints
  - The iterative gradient ascent algorithm leading to high computational load and is sensitive to parameter tuning
  - Does not explicitly exploit the transitivity property of positive equivalence constraints
- RCA
  - Only use equivalence constraints
  - explicitly exploit the transitivity property of positive equivalence constraints
  - Low computational load
  - Empirically show that RCA is similar or better than Xing' method using UCI data

Problems with Global Method

- Satisfying some constraints may be conflict to satisfying other constraints

Multimodal data distributions



(a) Data Dist. of the original dataset

(b) Data scaled by the global metric

Multimodal data distributions prevent global distance metrics from simultaneously satisfying constraints on within-class compactness and between-class separability.
[[Yang, et al, AAAI, 2006]]

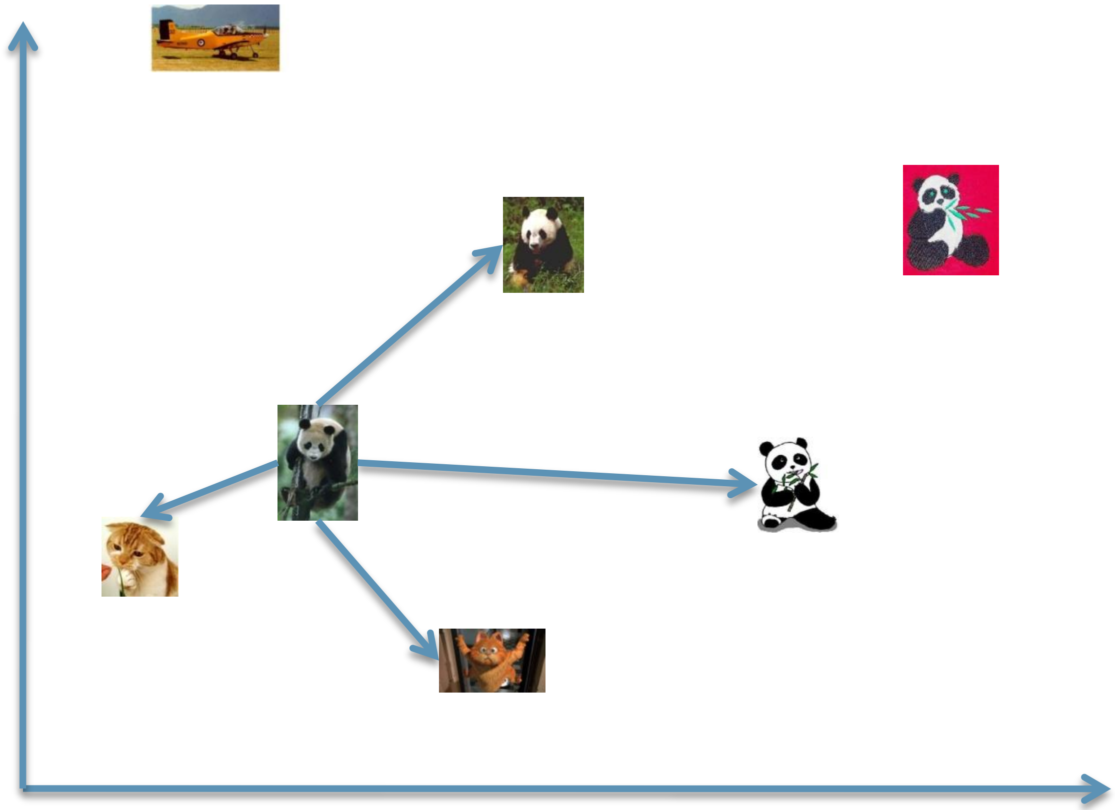
Local Methods

- Not try to satisfy all the constraints, but try to satisfy the *local constraints*

LMNN

- Large Margin Nearest Neighbor Based Distance Metric Learning [Weinberger et al., 2005]

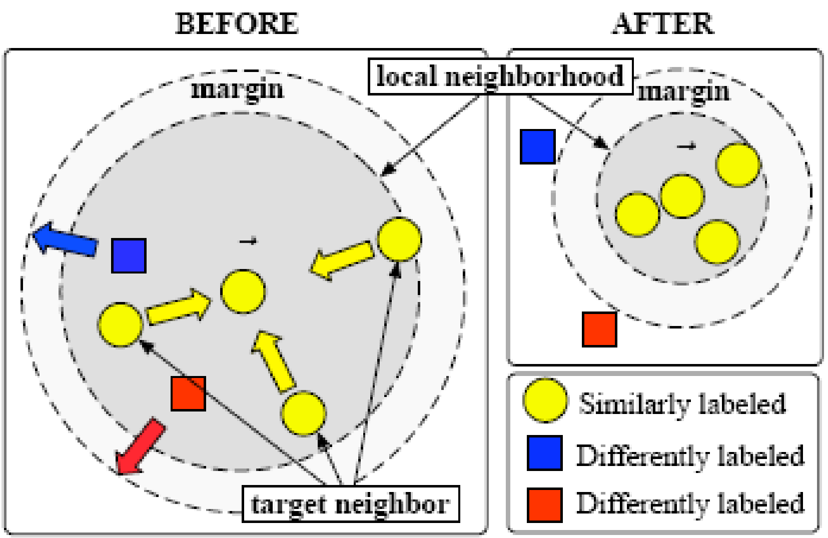
K-Nearest Neighbor Classification



We only care the nearest k neighbors

LMNN

- Learns a Mahanalobis distance metric, which
  - Enforces the k-nearest neighbors belong to the same class
  - Enforces examples from different classes are separated by a large margin



Approach

- Formulated as a optimization problem
- Solving using semi-definite programming method

Cost Function

$$\varepsilon(\mathbf{L}) = \sum_{ij} \eta_{ij} \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 + c \sum_{ijl} \eta_{ij}(1 - y_{il}) \left[1 + \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 - \|\mathbf{L}(\vec{x}_i - \vec{x}_l)\|^2\right]_+$$

Distance Function: $\mathcal{D}(\vec{x}_i, \vec{x}_j) = \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2$

Another form of Mahalanobis Distance: $\mathcal{D}(\vec{x}_i, \vec{x}_j) = (\vec{x}_i - \vec{x}_j)^\top \mathbf{M}(\vec{x}_i - \vec{x}_j)$

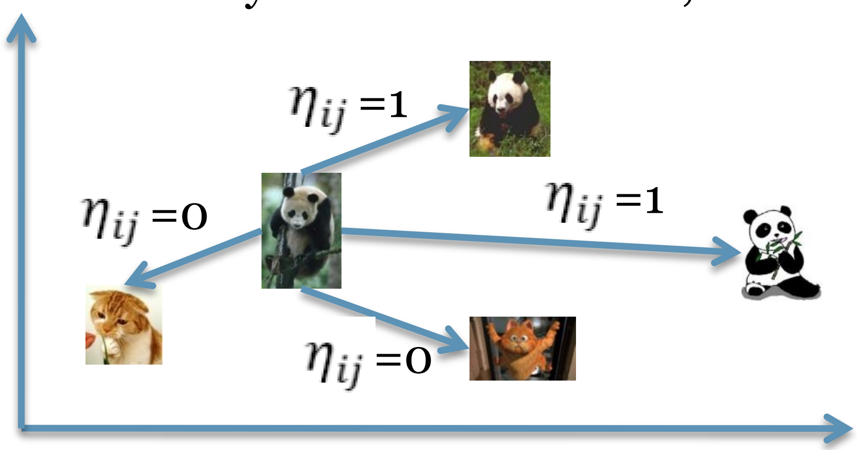
$$\mathbf{M} = \mathbf{L}^\top \mathbf{L}.$$

Cost Function

$$\varepsilon(\mathbf{L}) = \sum_{ij} \eta_{ij} \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 + c \sum_{ijl} \eta_{ij}(1 - y_{il}) \left[1 + \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 - \|\mathbf{L}(\vec{x}_i - \vec{x}_l)\|^2\right]_+$$

$\eta_{ij} \in \{0, 1\}$ indicate whether input $\vec{x}_j$ is a target neighbor of input $\vec{x}_i$

Target Neighbors: identified as the k-nearest neighbors, determined by Euclidean distance, that share the same label

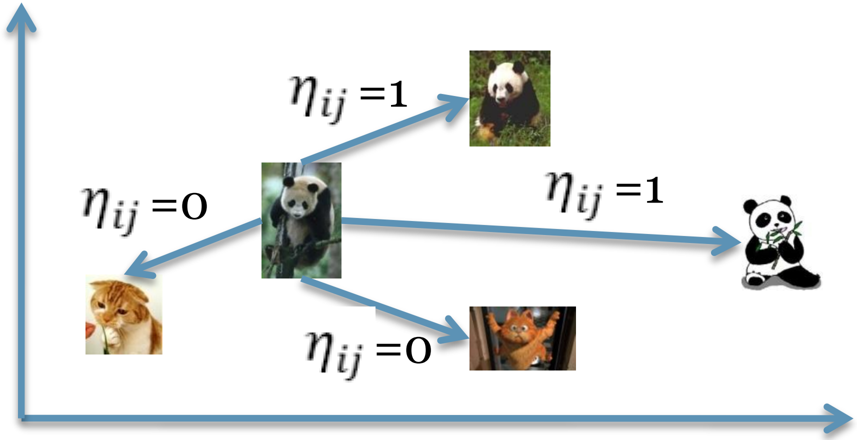


When K=2

Cost Function

$$\varepsilon(\mathbf{L}) = \sum_{ij} \eta_{ij} \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 + c \sum_{ijl} \eta_{ij}(1 - y_{il}) \left[1 + \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 - \|\mathbf{L}(\vec{x}_i - \vec{x}_l)\|^2\right]_+$$

Penalizes large distances between inputs and target neighbors. In other words, making similar neighbors close



Cost Function

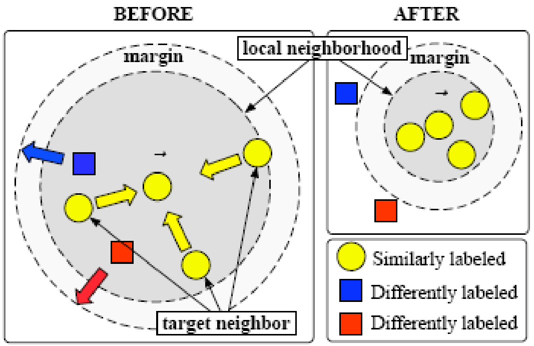
$$\varepsilon(\mathbf{L}) = \sum_{ij} \eta_{ij} ||\mathbf{L}(\vec{x}_i - \vec{x}_j)||^2 + c \sum_{ijl} \eta_{ij}(1 - y_{il}) \left[1 + ||\mathbf{L}(\vec{x}_i - \vec{x}_j)||^2 - ||\mathbf{L}(\vec{x}_i - \vec{x}_l)||^2\right]_+$$

$$[z]_+ = \max(z, 0)$$

Cost Function

$$\varepsilon(\mathbf{L}) = \sum_{ij} \eta_{ij} \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 + c \sum_{ijl} \eta_{ij}(1 - y_{il}) \left[1 + \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 - \|\mathbf{L}(\vec{x}_i - \vec{x}_l)\|^2\right]_+$$

For inputs and target neighbors
It is equal to 1

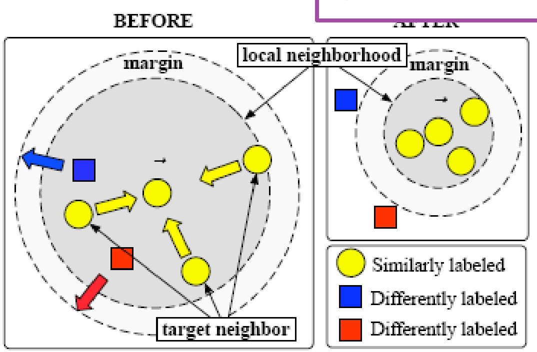


Approach-Cost Function

$$\varepsilon(\mathbf{L}) = \sum_{ij} \eta_{ij} \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 + c \sum_{ijl} \eta_{ij} (1 - y_{il}) \left[1 + \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 - \|\mathbf{L}(\vec{x}_i - \vec{x}_l)\|^2\right]_+$$

For inputs and target neighbors
It is equal to 1

$y_{il} \in \{0,1\}$ indicates if $\vec{x}_i$ and $\vec{x}_l$ has same label. So For input and neighbors having different labels, it is equal to 1

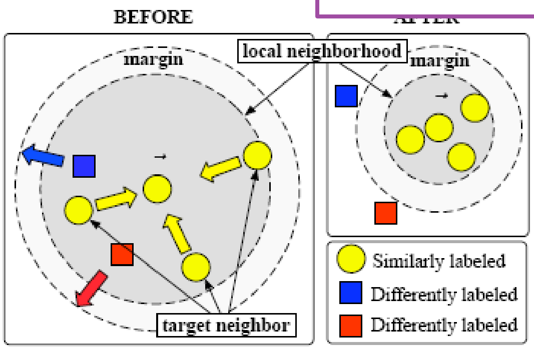


Approach-Cost Function

$$\varepsilon(\mathbf{L}) = \sum_{ij} \eta_{ij} \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 + c \sum_{ijl} \eta_{ij}(1 - y_{il}) \left[1 + \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 - \|\mathbf{L}(\vec{x}_i - \vec{x}_l)\|^2\right]_+$$

For inputs and target neighbors
It is equal to 1

$y_{il} \in \{0,1\}$ indicates if $\vec{x}_i$ and $\vec{x}_l$ has same label. So For input and neighbors having different labels, it is equal to 1



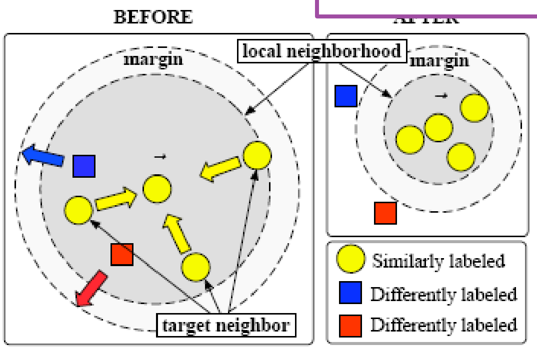
Approach-Cost Function

$$\varepsilon(\mathbf{L}) = \sum_{ij} \eta_{ij} \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 + c \sum_{ijl} \eta_{ij}(1 - y_{il}) \left[1 + \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 - \|\mathbf{L}(\vec{x}_i - \vec{x}_l)\|^2\right]_+$$

For inputs and target neighbors
It is equal to 1

$y_{il} \in \{0,1\}$ indicates if $\vec{x}_i$ and $\vec{x}_l$ has same label. So For input and neighbors having different labels, it is equal to 1

Distance between inputs and target neighbors



Approach-Cost Function

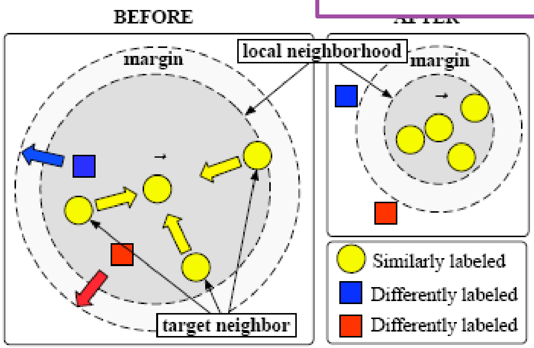
$$\varepsilon(\mathbf{L}) = \sum_{ij} \eta_{ij} \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 + c \sum_{ijl} \eta_{ij}(1 - y_{il}) \left[1 + \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 - \|\mathbf{L}(\vec{x}_i - \vec{x}_l)\|^2\right]_+$$

For inputs and target neighbors
It is equal to 1

$y_{il} \in \{0,1\}$ indicates if $\vec{x}_i$ and $\vec{x}_l$ has same label. So For input and neighbors having different labels, it is equal to 1

Distance between inputs and target neighbors

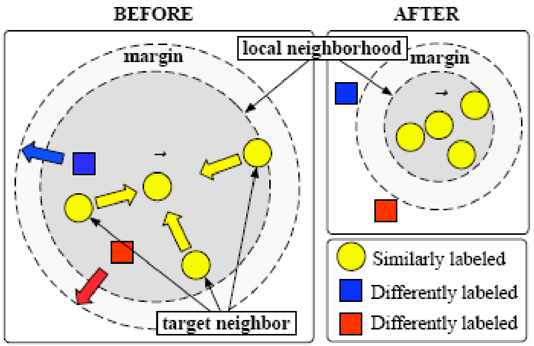
Distance between input and neighbors with different labels



Cost Function

$$\varepsilon(\mathbf{L}) = \sum_{ij} \eta_{ij} \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 + c \sum_{ijl} \eta_{ij}(1 - y_{il}) \left[1 + \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 - \|\mathbf{L}(\vec{x}_i - \vec{x}_l)\|^2\right]_+$$

Differently labeled neighbors lie outside the smaller radius with a margin of at least one unit distance



Test on Face Recognition

Images from the AT&T face recognition data base, kNN classification (k = 3)

•Top row: an image correctly recognized with Mahalanobis distances, but not with Euclidean distances

•Middle row: correct match among the k=3 nearest neighbors according to Mahalanobis distance, but not Euclidean distance.

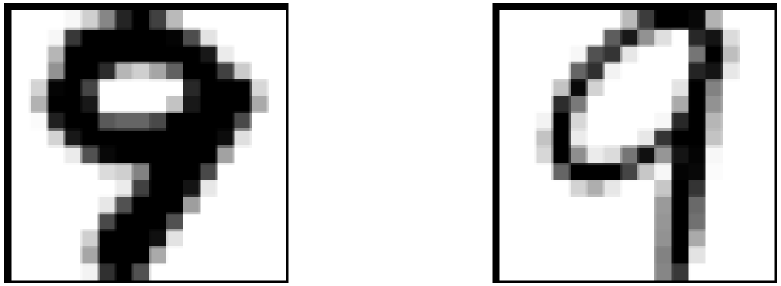
•Bottom row: incorrect match among the k=3 nearest neighbors according to Euclidean distance, but not Mahalanobis distance.

[K. Weinberger et al., 2005]

ILMNN

- An Invariant Large Margin Nearest Neighbor Classifier [Mudigonda, et al, 2007]

Transformation Invariance



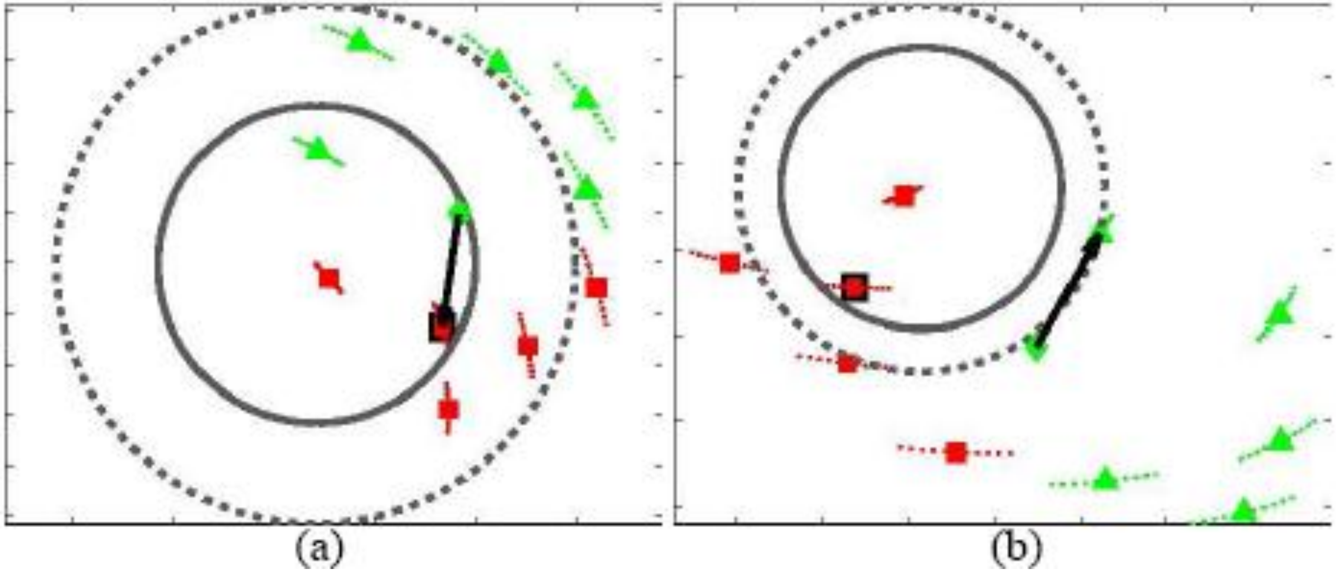
Same after rotation transformation and thickness transformation

When do classification, the classifier needs to regard the two images as the same image.

Figure from [Simard et al., 1998]

ILMNN

- An extension to LMNN[K.Weinberger et al., 2005]
  - Add regularization to LMNN to avoid overfitting
  - Incorporating invariance using Polynomial Transformations (Such as Euclidean, Similarity, Affine, usually used in computer vision)



Green Diamond is test point,
(a) Trajectories defined by rotating the points by an angle $-5^\circ < \theta < 5^\circ$
(b) Mapped trajectories After learning

[Mudigonda, et al, 2007]

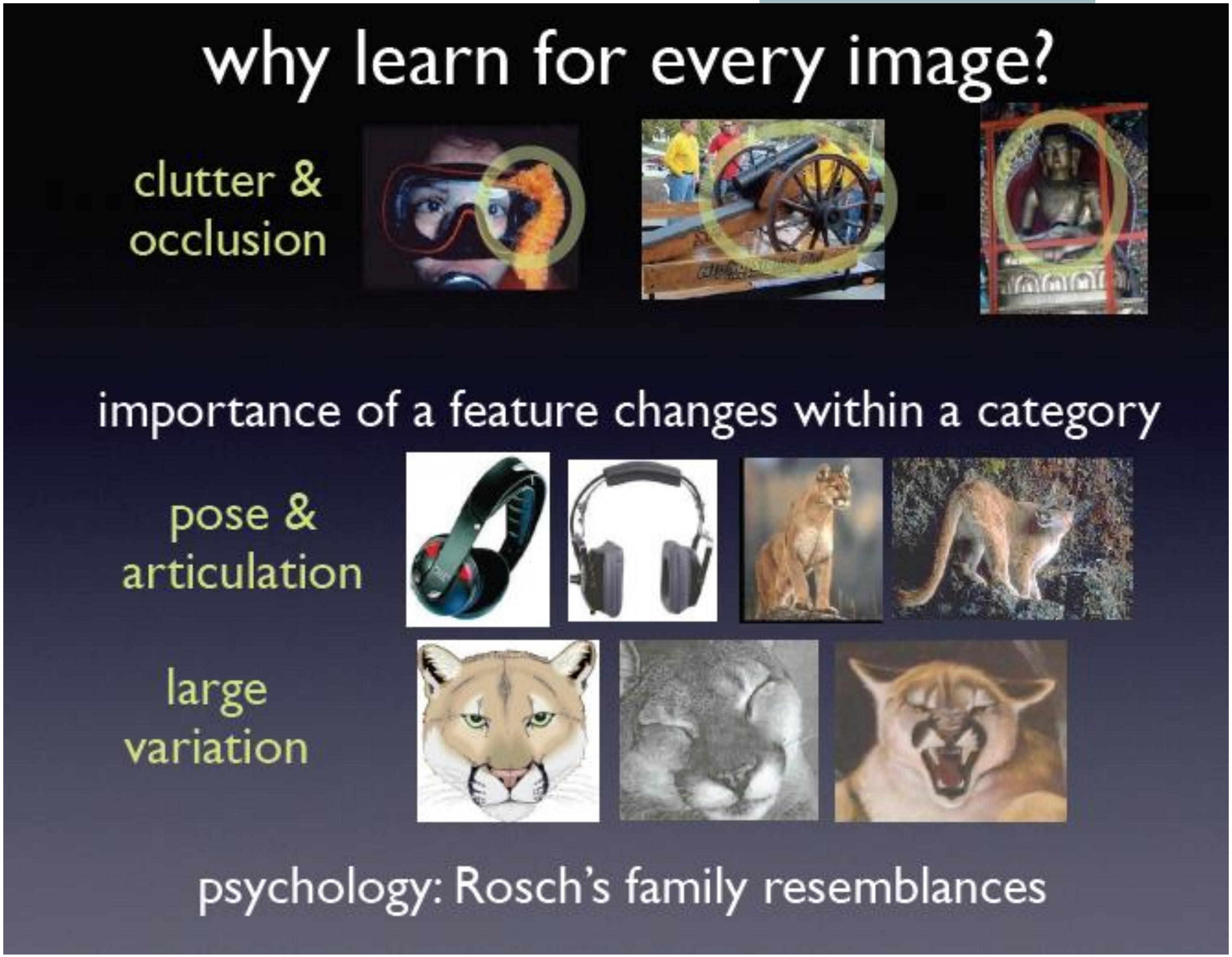
Outline

- Introduction
- Learning Mahalanobis distance metric
- **Learning multiple distance functions**
- Learning one classifier represented distance function
- Conclusion

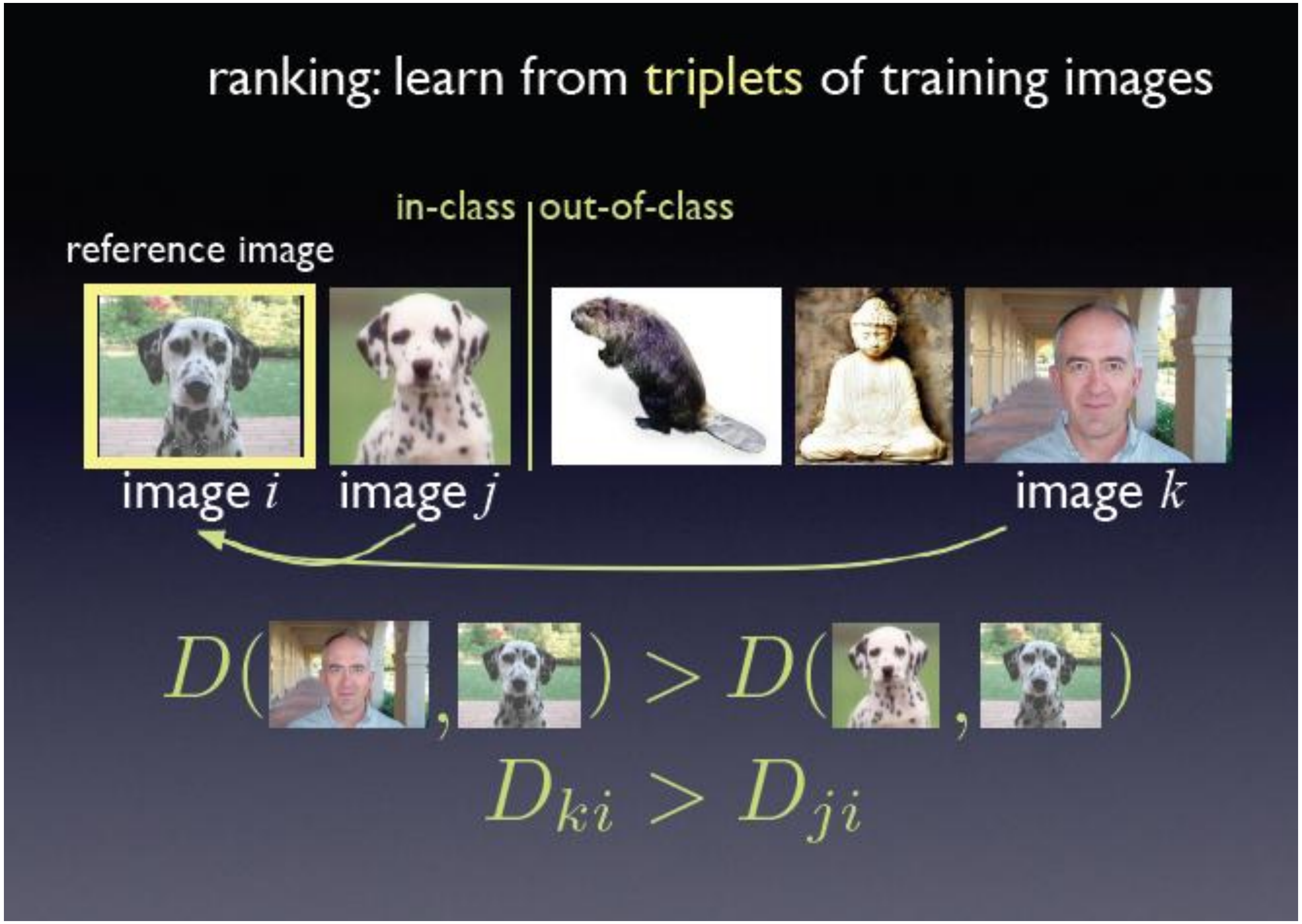
- Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification[Frome, et al., 2007]
  - The slides are adapted from Frome' talk on ICCV 2007 (http://www.cs.berkeley.edu/~afrome/papers/iccv2007_talk.pdf)

Globally-Consistent Local Distance Functions [Frome, et al., 2007]

- Previous methods only learn one distance function for all images, while this method learns one distance function for each image
  - From this perspective, it's a local distance function learning method while all the previous methods are global

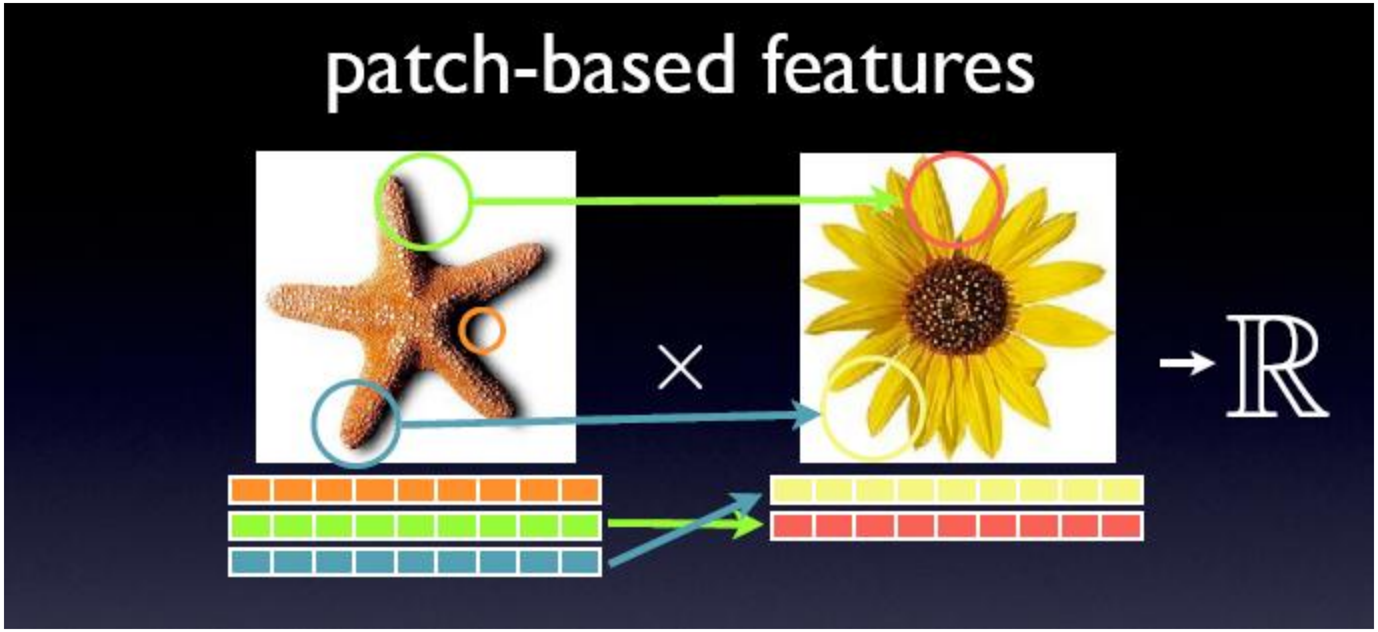

why learn for every image?
clutter & occlusion
importance of a feature changes within a category
pose & articulation
large variation
psychology: Rosch's family resemblances

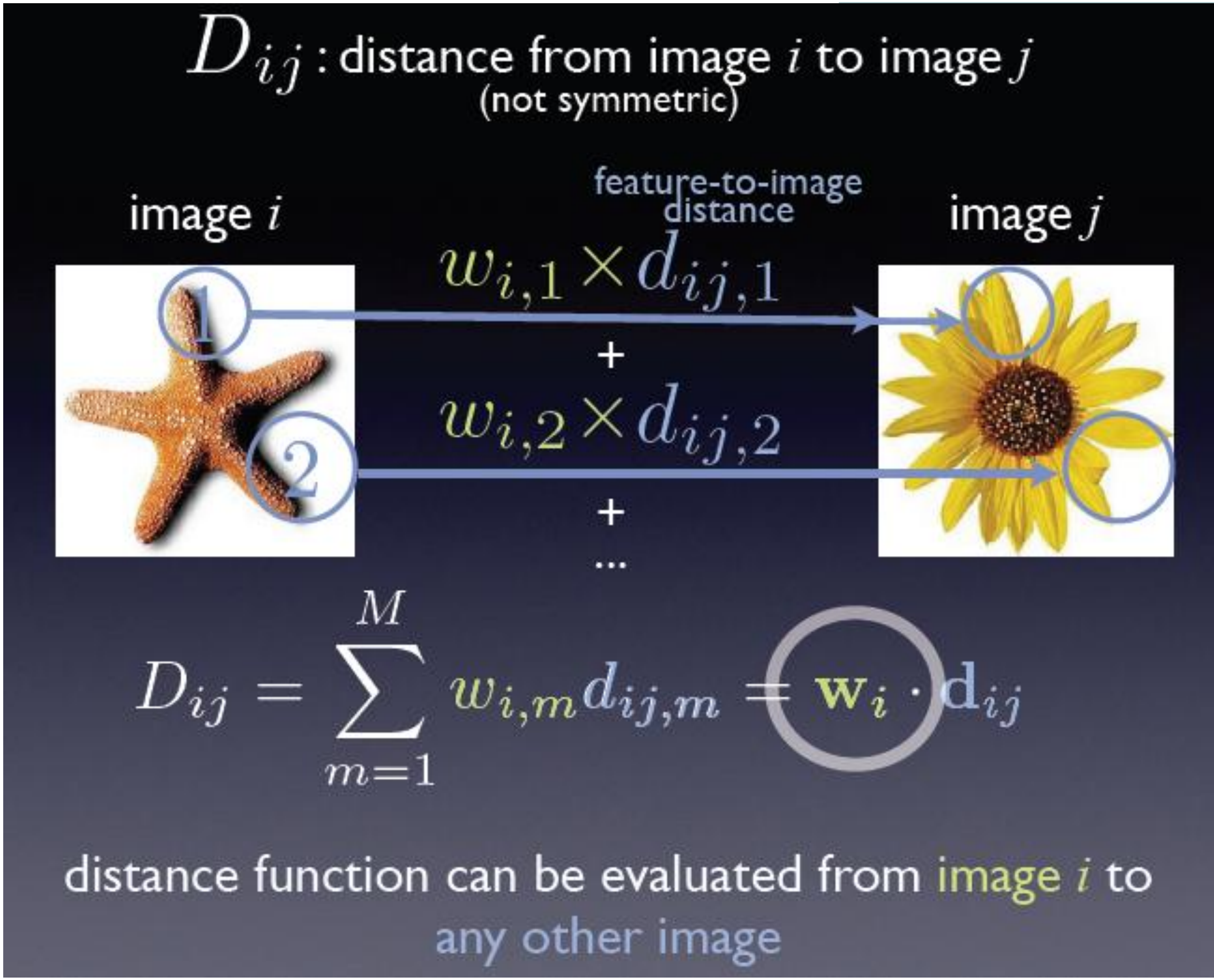
Using triplet constraints



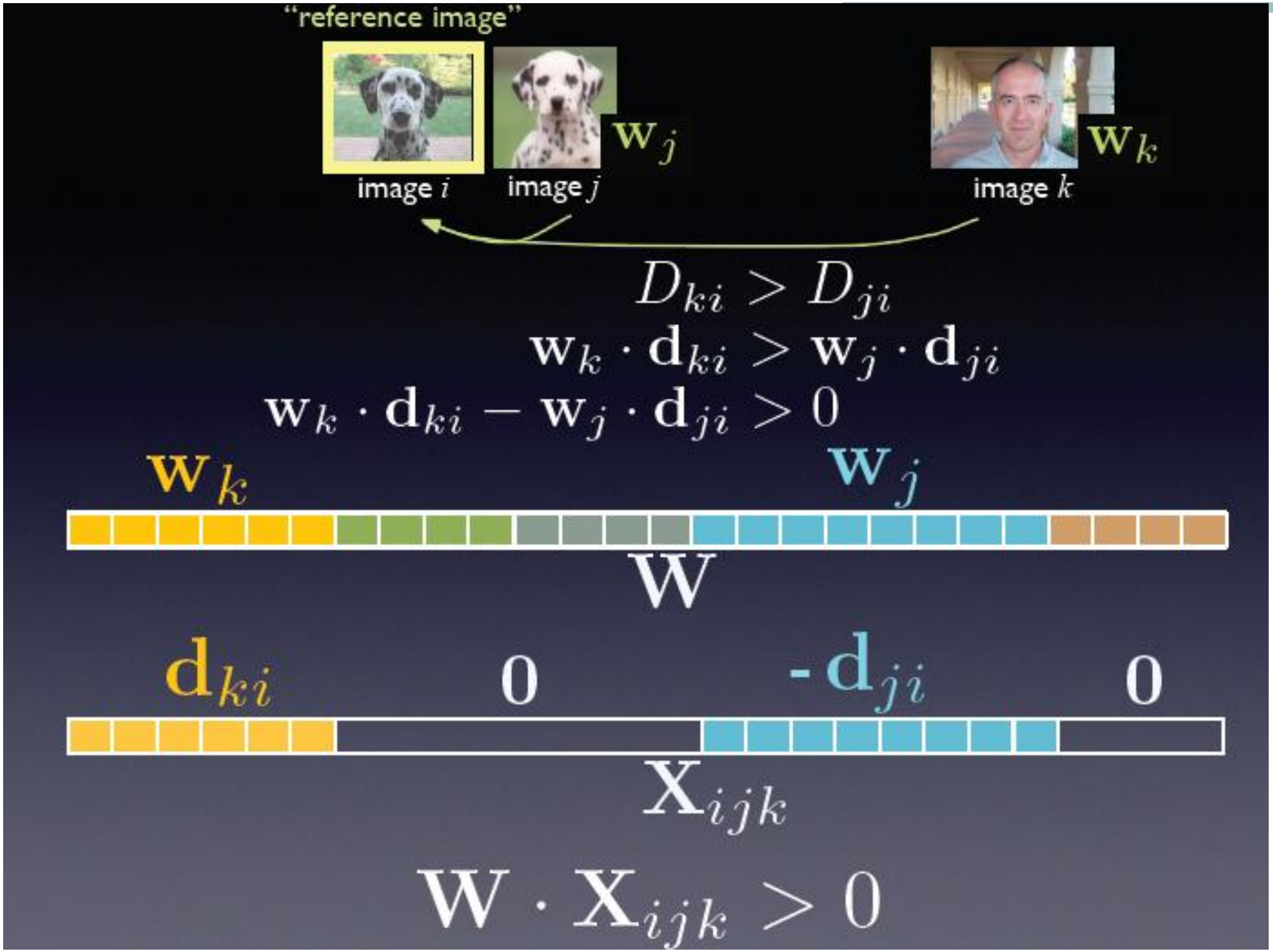
Patch-based features

- Different images may have different number of features.





D_{ij} : distance from image i to image j
(not symmetric)
feature-to-image distance
image i
image j
$w_{i,1} \times d_{ij,1}$
1
+
$w_{i,2} \times d_{ij,2}$
2
+
...
$D_{ij} = \sum_{m=1}^{M} w_{i,m} d_{ij,m} = \mathbf{w}_i \cdot \mathbf{d}_{ij}$
distance function can be evaluated from image i to any other image



[Frome, et al., 2007]

$$\mathbf{W} \cdot \mathbf{X}_{ijk} > 0$$

$$\mathbf{W} \cdot \mathbf{X}_{ijk} \geq 1$$

empirical loss: $$\sum_{i,j,k \in \text{triplets}} [1 - \mathbf{W} \cdot \mathbf{X}_{ijk}]_+$$

$$\min_{\mathbf{W},\xi} \quad \frac{1}{2}\|\mathbf{W}\|^2 + C \sum_{ijk} \xi_{ijk}$$

$$\text{s.t.} \quad \mathbf{W} \cdot \mathbf{X}_{ijk} \geq 1 - \xi_{ijk}$$

$$\xi_{ijk} \geq 0$$

$$\mathbf{W} \succeq 0$$

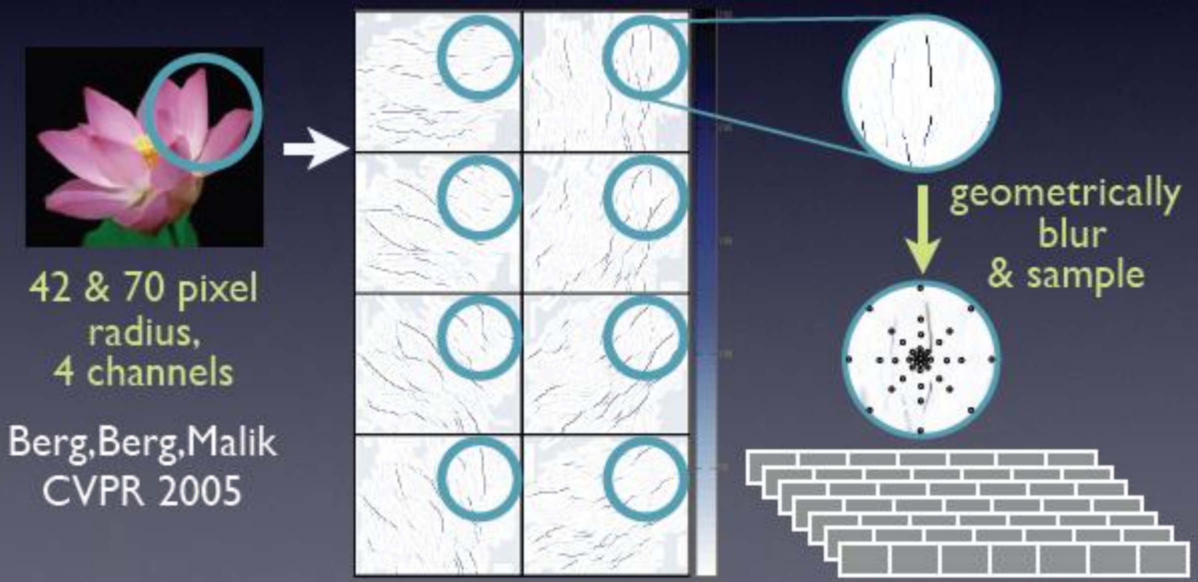
Schultz,Joachims NIPS 2003
Frome,Singer,Malik NIPS 2006

experiments

Caltech-101 (without using absolute position)

features: geometric blur (2 sizes) and color

L_2 feature-to-image distance



problem scale
(15 images/category, 101 categories)

~1,200 features/image: weight vector has 1.8M elements
using in- vs. out-of-class,
exhaustive set of triplets is 31.8 M triplets

speeding it up

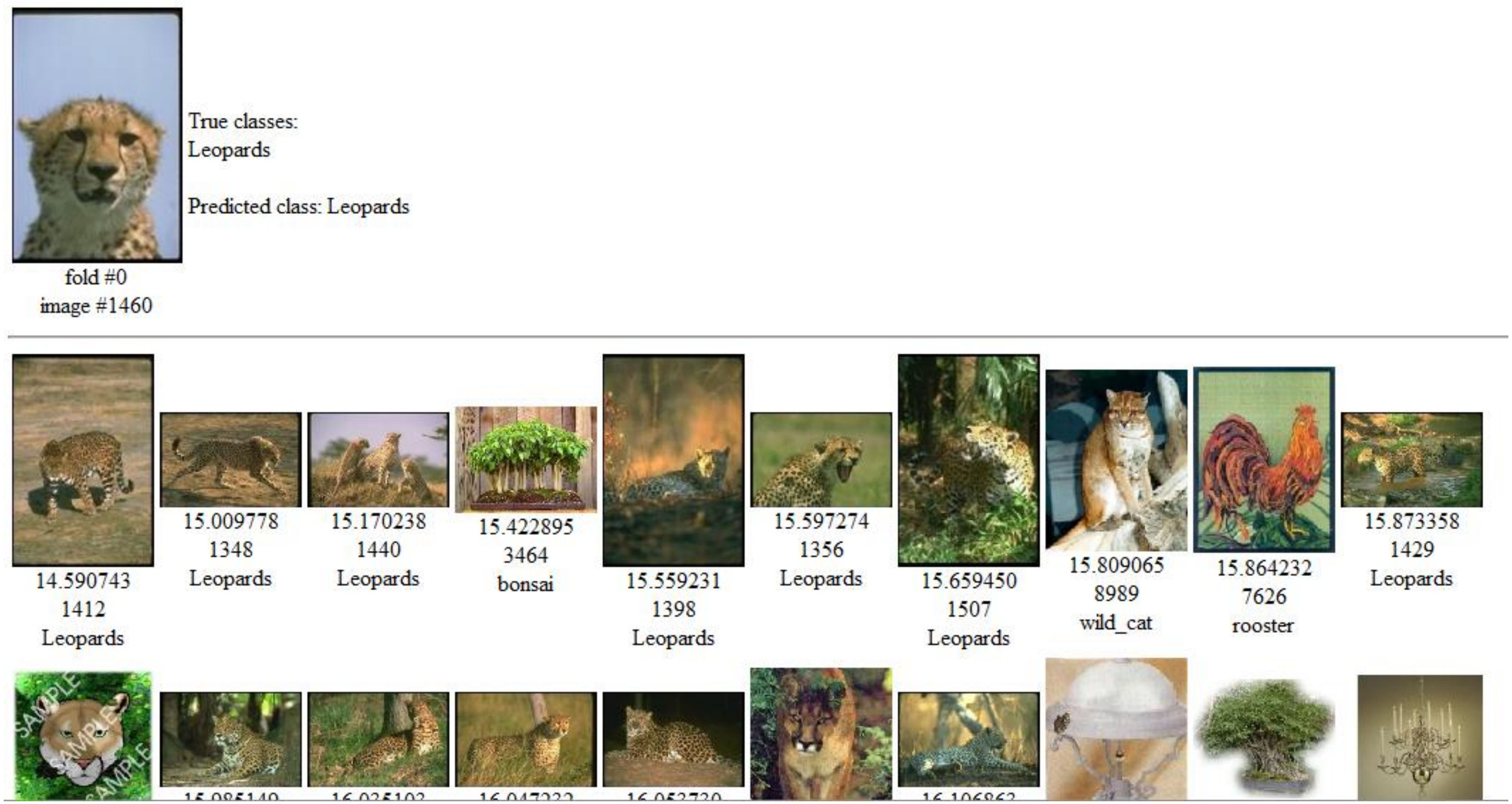
pare down to 15.7 M triplets

solve the dual problem
similar to on-line algorithms
early stopping: 10 hours to 1 hour
set trade-off parameter: one run through triplets

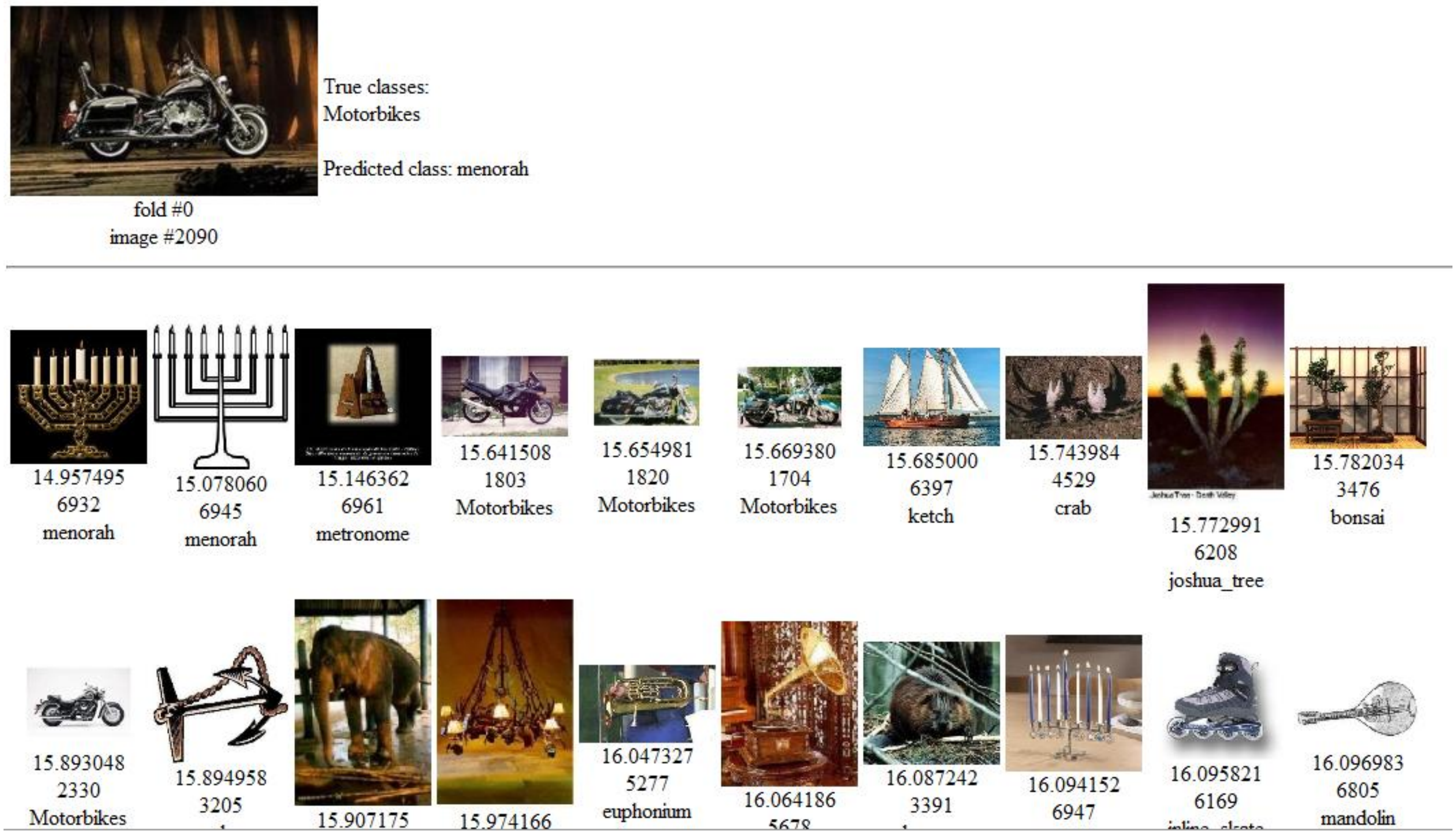
weight vectors are surprisingly sparse.
on average, 68% of weights are zero

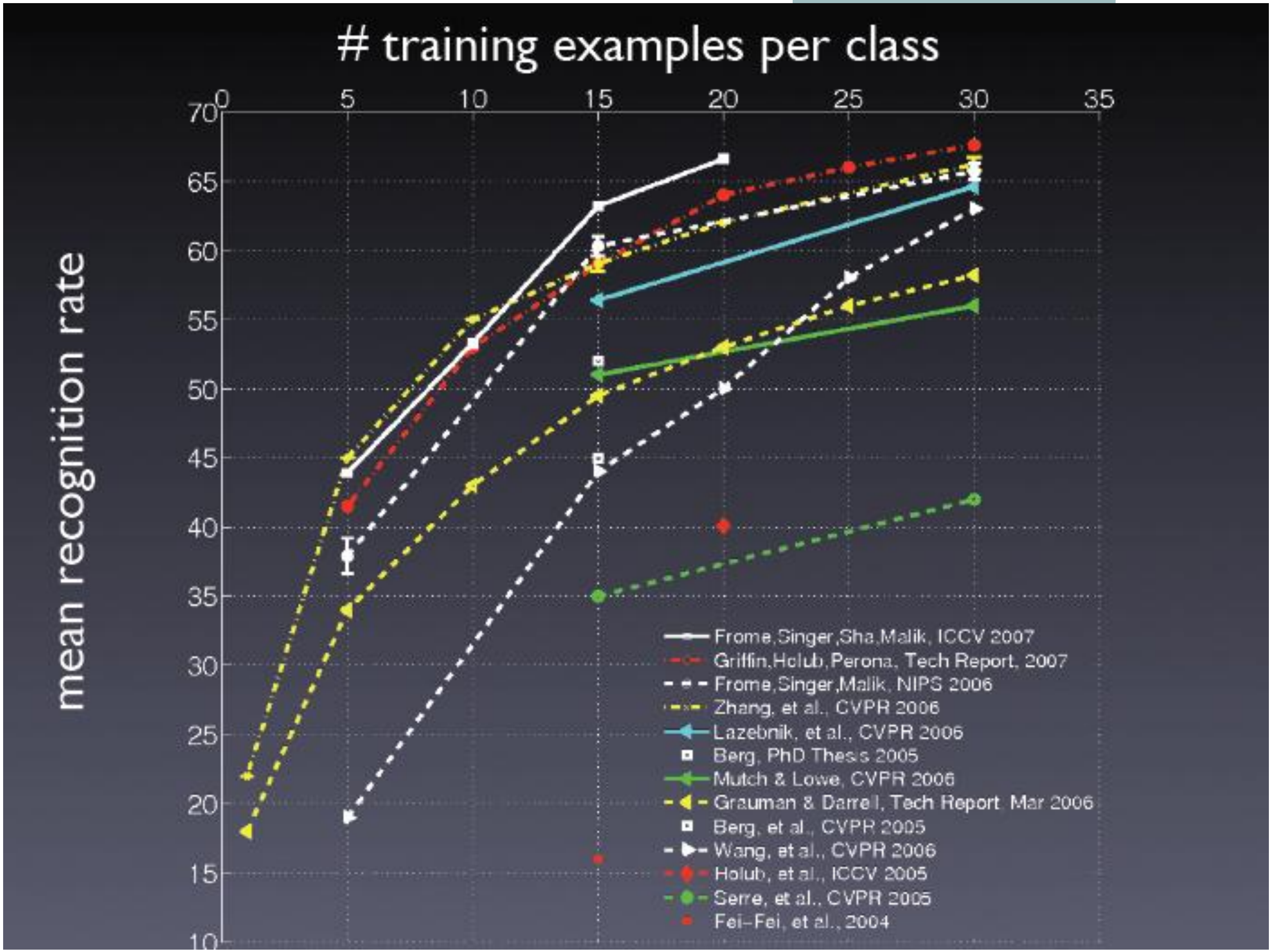
[Frome, et al., 2007]

Good Result


True classes:
Leopards
Predicted class: Leopards
fold #0
image #1460
14.590743
1412
Leopards
15.009778
1348
Leopards
15.170238
1440
Leopards
15.422895
3464
bonsai
15.559231
1398
Leopards
15.597274
1356
Leopards
15.659450
1507
Leopards
15.809065
8989
wild_cat
15.864232
7626
rooster
15.873358
1429
Leopards

Bad Results


True classes:
Motorbikes
Predicted class: menorah
fold #0
image #2090
14.957495
6932
menorah
15.078060
6945
menorah
15.146362
6961
metronome
15.641508
1803
Motorbikes
15.654981
1820
Motorbikes
15.669380
1704
Motorbikes
15.685000
6397
ketch
15.743984
4529
crab
15.772991
6208
joshua_tree
15.782034
3476
bonsai
15.893048
2330
Motorbikes
15.894958
3205
15.907175
15.974166
16.047327
5277
euphonium
16.064186
16.087242
3391
16.094152
6947
16.095821
6169
16.096983
6805
mandolin



training examples per class
mean recognition rate
0
5
10
15
20
25
30
35
70
65
60
55
50
45
40
35
30
25
20
15
10
Frome,Singer,Sha,Malik, ICCV 2007
Griffin,Holub,Perona, Tech Report, 2007
Frome,Singer,Malik, NIPS 2006
Zhang, et al., CVPR 2006
Lazebnik, et al., CVPR 2006
Berg, PhD Thesis 2005
Mutch & Lowe, CVPR 2006
Grauman & Darrell, Tech Report Mar 2006
Berg, et al., CVPR 2005
Wang, et al., CVPR 2006
Holub, et al., ICCV 2005
Serre, et al., CVPR 2005
Fei-Fei, et al., 2004

Summary

- Extremely local, having more ability to learn a good distance function for complex feature space
- Too many weights to learn
- Too many constraints

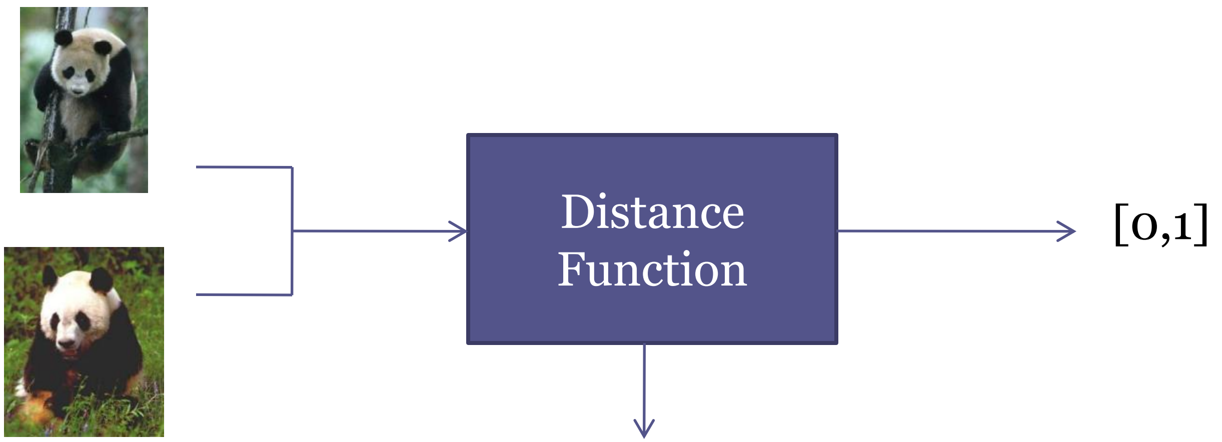
Outline

- Introduction
- Learning one Mahalanobis distance metric
- Learning multiple distance functions
- **Learning one classifier represented distance function**
- Discussion Points

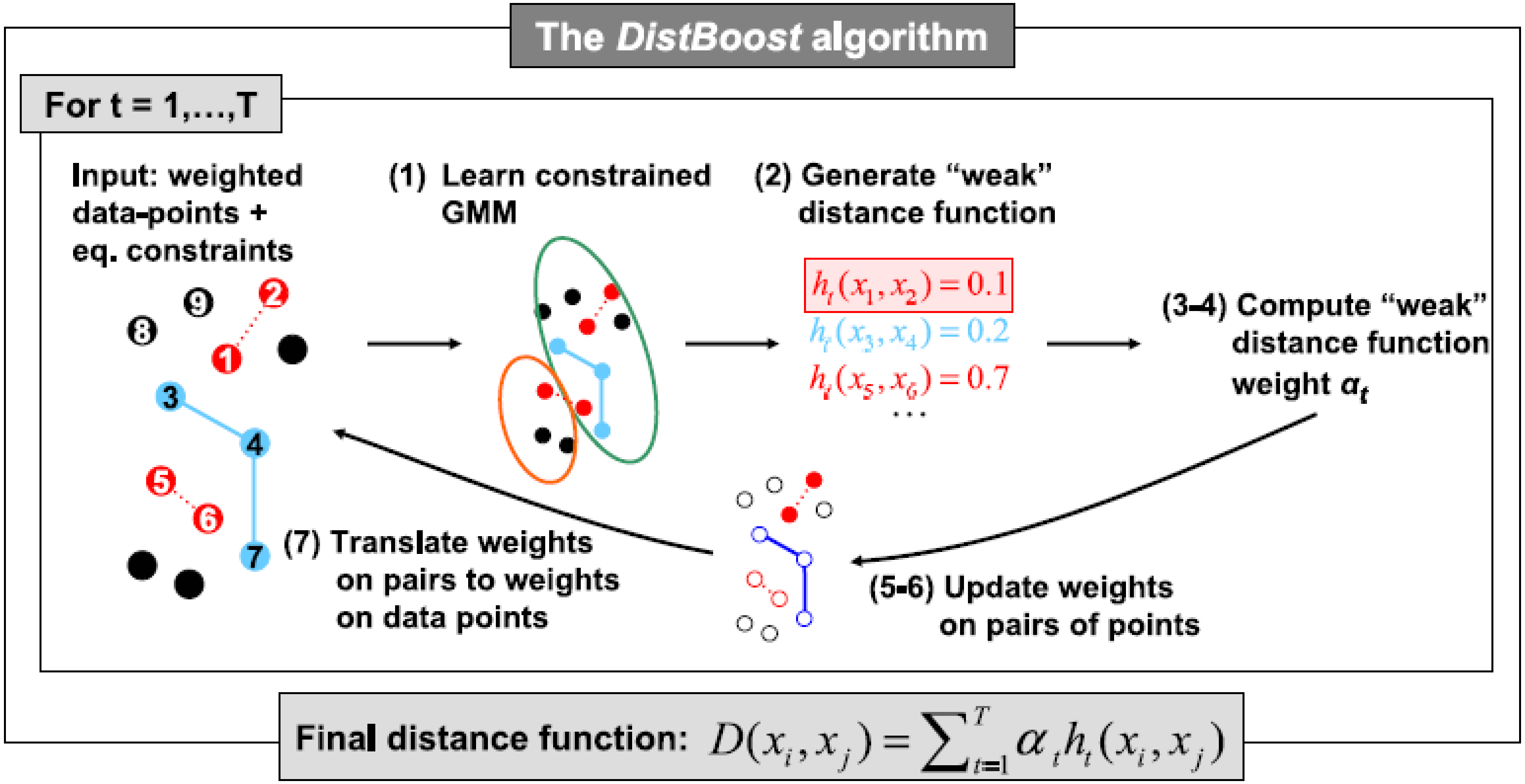
DistBoost

- T. Hertz, A. Bar-Hillel and D. Weinshall, Learning Distance Functions for Image Retrieval, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2004 [Hertz, et al, 2004]

DistBoost



Can be seen as a binary classifier (Adaboost)
The constraints are the labeled training examples for the classifier.



- Figure from [**Hertz, Ph.D Thesis, 2006**]

Algorithm 3 The *DistBoost* algorithm.

Input:

Data points: $(x_1, ..., x_n), \quad x_k \in \mathcal{X}$

A set of equivalence constraints: (x_{i_1}, x_{i_2}, y_i), where $y_i \in \{-1, 1\}$

Unlabeled pairs of points:$(x_{i_1}, x_{i_2}, y_i = *)$, implicitly defined by all unconstrained pairs of points

- Initialize $W^1_{i_1 i_2} = 1/(n^2) \;\; i_1, i_2 = 1, \ldots, n$ (weights over pairs of points)

  $w_k = 1/n \qquad k = 1, \ldots, n$ (weights over data points)

- For $t = 1, .., T$

1. Fit a constrained GMM (weak learner) on weighted data points in $\mathcal{X}$ using the equivalence constraints.
2. Generate a weak hypothesis $\tilde{h}_t : \mathcal{X} \times \mathcal{X} \rightarrow [-\infty, \infty]$ and define a weak distance function as $h_t(x_i, x_j) = \frac{1}{2}\left(1 - \tilde{h}_t(x_i, x_j)\right) \in [0, 1]$
3. Compute $r_t = \sum_{(x_{i_1}, x_{i_2}, y_i = \pm 1)} W^t_{i_1 i_2} y_i \tilde{h}_t(x_{i_1}, x_{i_2})$, only over **labeled** pairs. Accept the current hypothesis only if $r_t > 0$.
4. Choose the hypothesis weight $\alpha_t = \frac{1}{2}\ln(\frac{1+r_t}{1-r_t})$
5. Update the weights of **all** points in $\mathcal{X} \times \mathcal{X}$ as follows:

$$W^{t+1}_{i_1 i_2} = \begin{cases} W^t_{i_1 i_2} \exp(-\alpha_t y_i \tilde{h}_t(x_{i_1}, x_{i_2})) & y_i \in \{-1, 1\} \\ W^t_{i_1 i_2} \exp(-\alpha_t) & y_i = * \end{cases}$$

6. Normalize: $W^{t+1}_{i_1 i_2} = \frac{W^{t+1}_{i_1 i_2}}{\sum\limits_{i_1, i_2 = 1}^{n} W^{t+1}_{i_1 i_2}}$
7. Translate the weights from $\mathcal{X} \times \mathcal{X}$ to $\mathcal{X}$: $w^{t+1}_k = \sum_j W^{t+1}_{kj}$

Output: A final distance function $\mathcal{D}(x_i, x_j) = \sum_{t=1}^{T} \alpha_t h_t(x_i, x_j)$

Results

- Each row presents a query image and its first 5 nearest neighbors comparing DistBoost and normalized L1 CCV distance

Results

- Each row presents a query image and its first 5 nearest neighbors comparing DistBoost and normalized L1 CCV distance

Results

- Each row presents a query image and its first 5 nearest neighbors comparing DistBoost and normalized L1 CCV distance

Summary

- Another view of distance function learning
- A global method, since it try to satisfy all the constraints
- Can learn non-linear distance functions

Discussion Points

- Currently most of the work focus on learning linear distance function, how can we learn non-linear distance function?
- Learning one distance function for every image is really good? Will lead to overfitting? Should we learn higher level distance function?
- The triplet constraints are huge for [Frome, 2007], how to improve the triplet selection method?

References

- [Hertz, et al, 2004]T. Hertz, A. Bar-Hillel and D. Weinshall, Learning Distance Functions for Image Retrieval, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2004
- [Hertz, PhD Thesis, 2006] Learning Distance Functions: Algorithms and Applications, Hebrew University, 2006
- [Bar-Hillel, et al, 2005]A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, Learning a Mahalanobis Metric from Equivalence Constraints, in Journal of Machine Learning Research (JMLR), 2005
- [Frome, et al, 2007]A. Frome, Y. Singer, F. Sha, J. Malik , Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification, in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2007
- [Mudigonda, et al, 2007]P. Mudigonda, P. Torr, and A. Zisserman , Invariant Large Margin Nearest Neighbor Classifier, in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2007
- [Yang, et al, 2006]L. Yang, Distance Metric Learning: A Comprehensive Survey, Michigan State University, 2006
- [Yang, et al, AAAI, 2006] L. Yang, R. Jin, R. Sukthankar, Y. Liu. An Efficient Algorithm for Local Distance Metric Learning. (Oral Prensentation) Proceedings of AAAI, 2006
- [Weinberger et al., 2005] K. Q.Weinberger, J. Blitzer, and L. K. Saul. Distance metriclearning for large margin nearest neighbor classification. In *NIPS, 2005*
- [Xing et al., 2002] E. Xing, A. Ng, and M. Jordan. Distancemetric learning with application to clustering with side-information. In *NIPS,* 2002.
- [Simard et al., 1998]P. Simard, Y. LeCun, J. Denker, and B. Victorri. Transformation invariance in pattern recognition, tangent distance and tangent propagation. In G. Orr and M. K., editors, Neural Networks: Tricks of the trade. Springer, 1998.


thank
you