Unsupervised learning of Categories from Sets of Partially Matching Features

Kristen Grauman and Trevor Darrell
CVPR 2006

Presented By
Shilpa Gulati and Muhammad Zubair Malik
4-12-2007

[Most of the slides borrowed from Grauman presentation]

Motivation

- Automatically learn object categories from unlabeled images.
- Allow optional supervision.
- Train classifiers for new images.

Background

The Pyramid Match Kernel

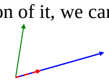
The Need of a good Kernel

- Discriminative classifiers require a good similarity measure
- Given a good similarity measure we can train a whole range of classifiers (perceptrons,pca, hyper-plane or any complicated shape classifiers)
- Of particular interest are hyper-plane, maximum margin classifiers: Support Vector Machines
- Looking at support vectors is far efficient than calculating all pair-wise comparisons (NN)

The Notion of similarity

- Dot Product works well for linearly separable data that is not noisy
- If the data is not linearly separable and if we can find a *symmetric* *semi-positive definite* function of it, we can use its dot product

$$\langle \mathbf{x}_1, \mathbf{x}_2 \rangle \leftarrow K(\mathbf{x}_1, \mathbf{x}_2) = \langle \varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2) \rangle$$



- The Kernel matrix encodes this information
- The eigen functions of Mercer kernel act as features
- Issues
  - Computational complexity increases (high dimension)
  - Risk Over fitting the data

Image representation

- An image is a **set** local features.

$$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\} \qquad \mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$$

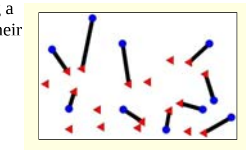
- The sets **X** and **Y** can have different sizes.

What is a good kernel in this case?

- Kernel matrix is the information bottleneck, and must be selected discreetly
  - A diagonal kernel matrix means we cannot discriminate, all data points are orthogonal to each other and no clusters exist
- In this case we have to consider:
  - Each instance is unordered set of vectors
  - Varying number of vectors per instance

Partial matching for sets of features

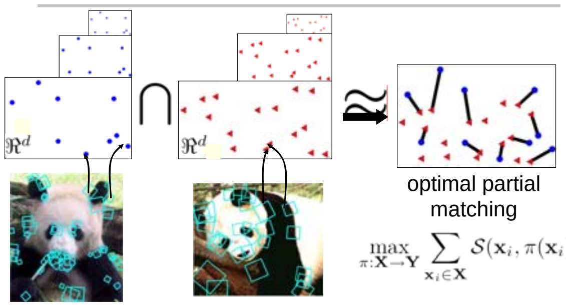
Compare sets by computing a *partial matching* between their features.



Robust to clutter, segmentation errors, occlusion…

Copied from Grauman Slides

Pyramid match



$$\max_{\pi: \mathbf{X} \to \mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} \mathcal{S}(\mathbf{x}_i, \pi(\mathbf{x}_i))$$

$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\vec{\mathbf{x}}_i \in \Re^d$

$\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$ $\vec{\mathbf{y}}_i \in \Re^d$

Copied from Grauman Slides

Pyramid match overview

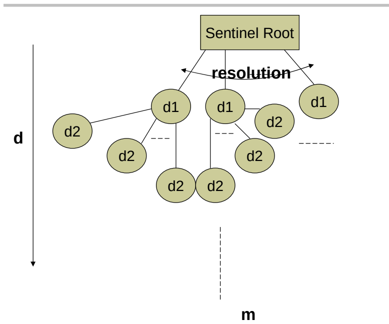
Pyramid match kernel measures similarity of a partial matching between two sets:

- Place multi-dimensional, multi-resolution grid over point sets
- Consider points matched at finest resolution where they fall into same grid cell
- Approximate similarity between matched points with worst case similarity at given level

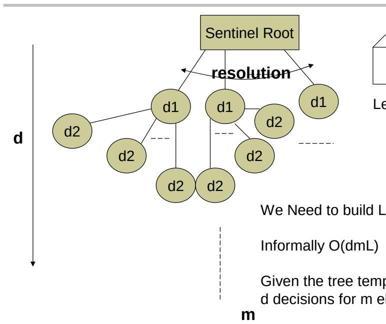
No explicit search for matches!

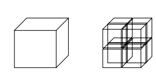
Copied from Grauman Slides

How to build a multi-dimensional Histogram?



How to build a multi-dimensional, multi-resolution Histogram?





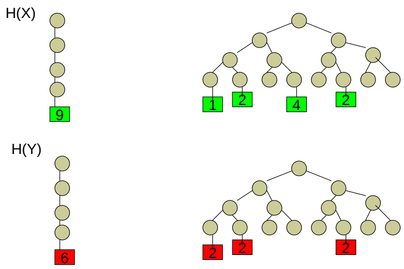
Level L to L-1 where d=3

We Need to build L such histograms

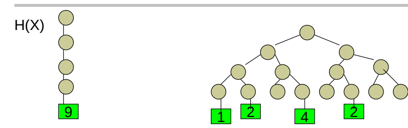
Informally O(dmL)

Given the tree template, we make d decisions for m elements in L such trees

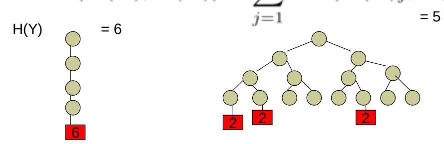
How to apply the Kernel



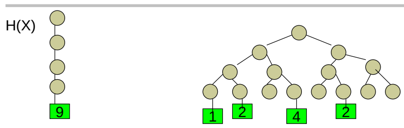
How to apply the Kernel



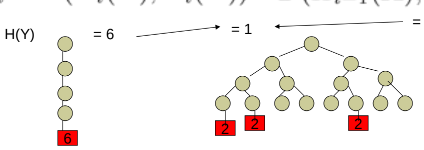
$$\mathcal{I}\left(H(\mathbf{X}), H(\mathbf{Y})\right) = \sum_{j=1}^{r} \min\left(H(\mathbf{X})_j, H(\mathbf{Y})_j\right)$$



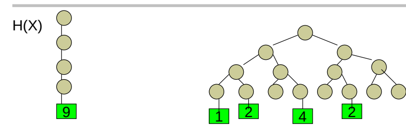
How to apply the Kernel



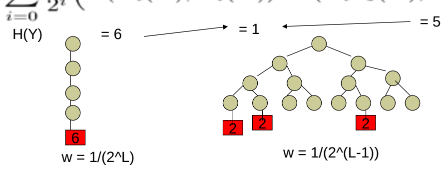
$$N_i = \mathcal{I}\left(H_i(\mathbf{X}), H_i(\mathbf{Y})\right) - \mathcal{I}\left(H_{i-1}(\mathbf{X}), H_{i-1}(\mathbf{Y})\right)$$



How to apply the Kernel



$$\sum_{i=0}^{L} \frac{1}{2^i}\left(\mathcal{I}\left(H_i(\mathbf{X}), H_i(\mathbf{Y})\right) - \mathcal{I}\left(H_{i-1}(\mathbf{X}), H_{i-1}(\mathbf{Y})\right)\right)$$

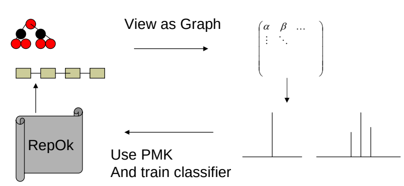


PMK is a Mercer Kernel

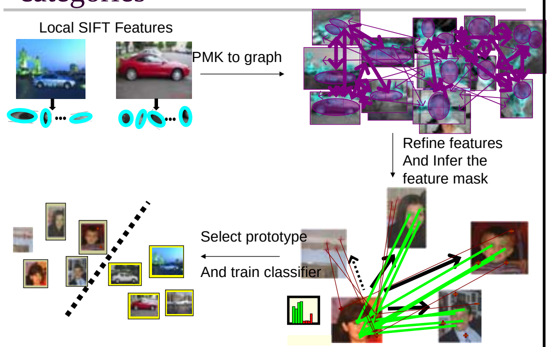
- Clearly histogram intersection is a positive definite function (min of two positive numbers cannot be less than zero)
- Mercer Kernels have good modularity properties. Given two kernels K1 and K2, and constants a,b
  - K1+K2 is also a mercer kernel
  - aK1 is also a kernel
  - aK1+bK2 is also a kernel

Application in Software Engineering

Given Example Structures find the Structural Invariants and output heuristics to check if a test structure satisfies them



Back to unsupervised learning of categories



Similarity between two images

- Similarity between two sets of features:
  - Use pyramid match kernel.
  - A scalar value for each pair of images.
- Partial Matching
  - The points (features) of smaller set are mapped to subset of points of larger set.
  - Not all features are matched.

Similarity Matrix

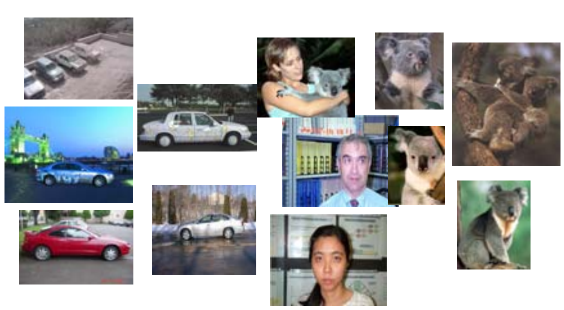
	X_1	X_2		X_i		X_N
X_1	K_{11}	K_{12}	..	..	..	..
X_2	K_{21}	K_{22}	..	..	..	..
	..	..	..	..	..	..
X_j	K_{j1}	K_{j2}		K_{ij}	..	..
	..	..	..	..	..	..
X_N	K_{N1}	K_{N2}	..	..	..	K_{NN}

K_{ij} is a scalar representing the similarity between sets X_i and X_j.

Clustering

- Partition the data into two sets
  - Shi and Malik's efficient approximation of Normalized Cuts algorithm.
- Optimal partitioning
  - Maximizes intra-cluster similarity.
  - Minimizes inter-cluster similarity.
- For k > 2 clusters
  - Recursively partition.

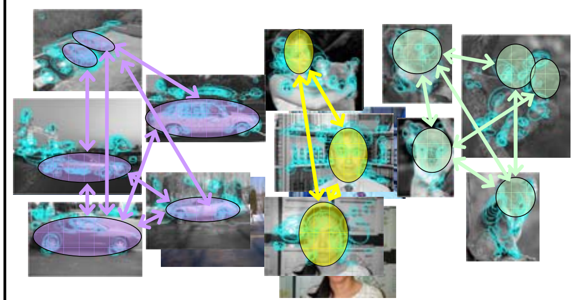
Clustering with a partial matching



Clustering with a partial matching

Clustering with a partial matching

Clustering with a partial matching



Partial Matching: Again

- Partial Matching: a key feature.
  - Not all features of a set are matched.

- This implies
  - Robustness to clutter and occlusion.
    - These features most likely will not be matched to anything.
  - But images with similar backgrounds may match.
    - Since some of the features match!

Limitation of partial match clustering

Background feature matches

Multiple object matches

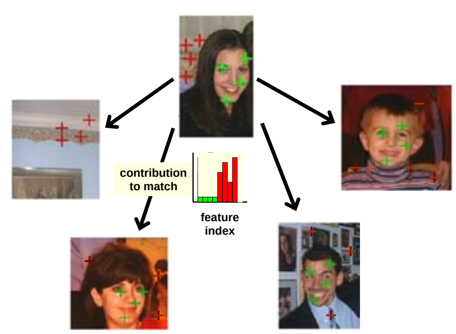
Overcoming the limitations

- Within each cluster, for each cluster member:

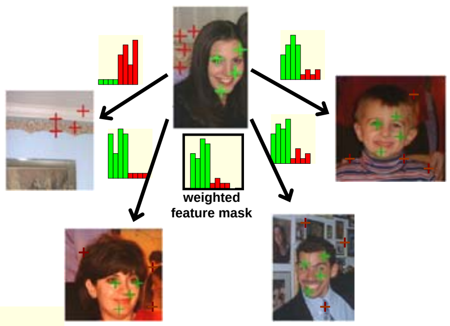
  - Identify features that form matching with other cluster members.

  - Greater number of matchings imply greater weight for that feature.

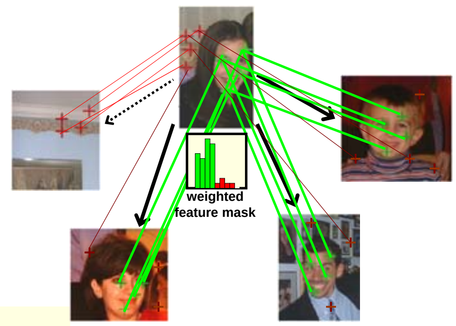
Inferring feature masks



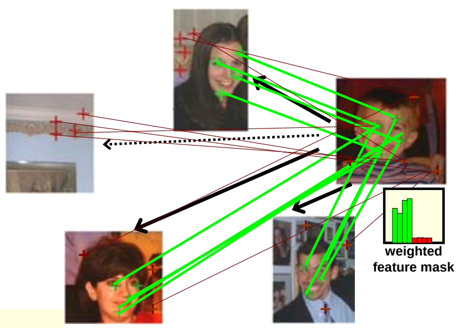
Inferring feature masks



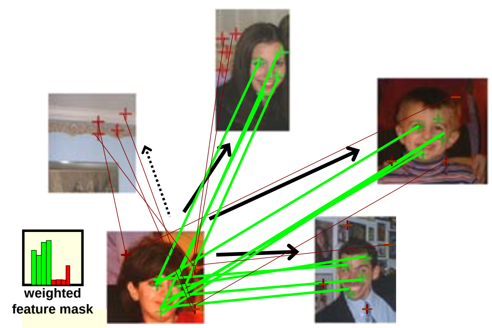
Refining intra-cluster matches



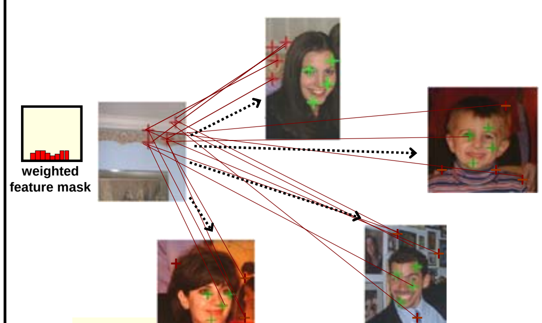
Refining intra-cluster matches



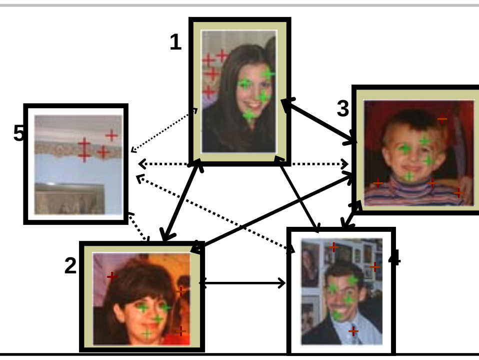
Refining intra-cluster matches



Refining intra-cluster matches

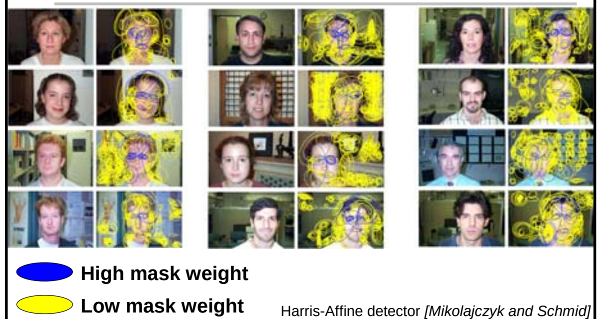


Selecting category prototypes



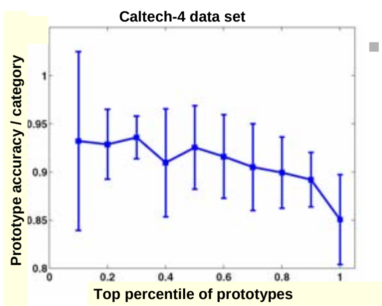
Selecting category prototypes

Inferred feature masks



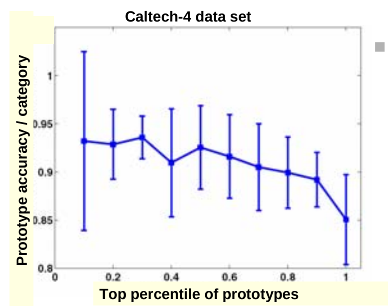
Harris-Affine detector *[Mikolajczyk and Schmid]*
SIFT descriptors *[Lowe]*

Unsupervised recovery of category prototypes



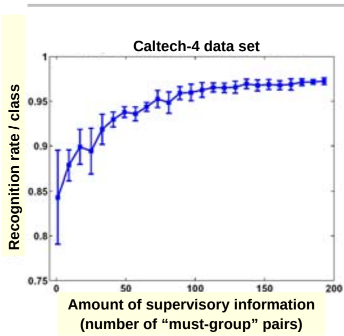
- 40 runs with 400 randomly selected images

Unsupervised recovery of category prototypes



- 40 runs with 400 randomly selected images

Semi-supervised category labeling



- Recover categories and SVM classifiers from 400 unlabeled images
- Classify 2788 unseen examples
- 40 runs with random cluster/test set/supervision selections

Conclusions

- Main contribution
  - Efficient unsupervised (semi-supervised) category learning.
  - Co-occurrences among features are naturally preserved.
  - Feature masks inferred automatically.
    - Roughly identify location of objects.