Sparsity-based Face Recognition using Discriminative Graphical Models

Umamahesh Srinivas[†] Vishal Monga[†] Yi Chen[‡] Trac Tran[‡]

[†]Pennsylvania State University
University Park, PA, USA

[‡]The Johns Hopkins University
Baltimore, MD, USA

Forty-fifth Asilomar Conference on Signals, Systems and Computers

November 08, 2011

Outline

1. Sparsity-based face recognition
2. Locally adaptive sparse representations
3. Probabilistic graphical models: Review
4. Contribution: Robust face recognition via discriminative graphical models
5. Results

Face recognition: Overview

Problem formulation:

- K unique faces (persons)
- Training: $\{\boldsymbol{v}_{1,1}, \ldots, \boldsymbol{v}_{1,N_1}\}, \ldots, \{\boldsymbol{v}_{K,1}, \ldots, \boldsymbol{v}_{K,N_k}\}$
- Goal: Given new face $\boldsymbol{y}$, assign one of the labels $\{1, \ldots, K\}$

Applications: Security, biometrics, online image search, etc.

Feature extraction for dimensionality reduction:

- Eigenfaces[1]
- Fisherfaces[2]

Classifier (decision engine):

- Nearest neighbor, nearest subspace[3]
- Support vector machines[4]

---

1 Turk et al., J Cogn. Neurosci., 1991
2 Belhumeur et al., IEEE Trans. PAMI, 1997
3 Kriegman et al., CVPR 2003
4 Vapnik, 1995

Face recognition: Overview

Problem formulation:

- K unique faces (persons)
- Training: $\{\boldsymbol{v}_{1,1}, \ldots, \boldsymbol{v}_{1,N_1}\}, \ldots, \{\boldsymbol{v}_{K,1}, \ldots, \boldsymbol{v}_{K,N_k}\}$
- Goal: Given new face $\boldsymbol{y}$, assign one of the labels $\{1, \ldots, K\}$

Applications: Security, biometrics, online image search, etc.

Feature extraction for dimensionality reduction:

- Eigenfaces[1]
- Fisherfaces[2]

Classifier (decision engine):

- Nearest neighbor, nearest subspace[3]
- Support vector machines[4]

---

[1] Turk et al., J Cogn. Neurosci., 1991
[2] Belhumeur et al., IEEE Trans. PAMI, 1997
[3] Kriegman et al., CVPR 2003
[4] Vapnik, 1995

Face recognition: Overview

Problem formulation:

- K unique faces (persons)
- Training: $\{\boldsymbol{v}_{1,1}, \ldots, \boldsymbol{v}_{1,N_1}\}, \ldots, \{\boldsymbol{v}_{K,1}, \ldots, \boldsymbol{v}_{K,N_k}\}$
- Goal: Given new face $\boldsymbol{y}$, assign one of the labels $\{1, \ldots, K\}$

Applications: Security, biometrics, online image search, etc.

Feature extraction for dimensionality reduction:

- Eigenfaces[1]
- Fisherfaces[2]

Classifier (decision engine):

- Nearest neighbor, nearest subspace[3]
- Support vector machines[4]

---

[1] Turk et al., J Cogn. Neurosci., 1991

[2] Belhumeur et al., IEEE Trans. PAMI, 1997

[3] Kriegman et al., CVPR 2003

[4] Vapnik, 1995

Sparse representation for face recognition[5]

- Assumption: New face of person i lies in linear span of training samples associated with class i

$$\boldsymbol{y} = \alpha_{i,1}\boldsymbol{v}_{i,1} + \alpha_{i,2}\boldsymbol{v}_{i,2} + \ldots + \alpha_{i,N_i}\boldsymbol{v}_{i,N_i} = \boldsymbol{A}_i\boldsymbol{\alpha}_i$$

$\left(\boldsymbol{y} \in \mathbb{R}^n, \boldsymbol{A}_i \in \mathbb{R}^{n \times N_i}, \boldsymbol{\alpha}_i \in \mathbb{R}^{N_i}\right)$

- $\boldsymbol{y} \rightarrow$ sparse linear combination of all training samples:

$$\boldsymbol{y} = \left[\begin{array}{cccc} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \ldots & \boldsymbol{A}_K \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \\ \vdots \\ \boldsymbol{\alpha}_K \end{array}\right] = \boldsymbol{A}\boldsymbol{\alpha}$$

$\left(\boldsymbol{A} \in \mathbb{R}^{n \times T}, T = \sum_{i=1}^{K} N_i, \boldsymbol{\alpha} \in \mathbb{R}^T\right)$

- Membership of $\boldsymbol{y}$ encoded by sparse representation

$$\boldsymbol{\alpha} = \left[\boldsymbol{0}^t \quad \ldots \quad \boldsymbol{0}^t \quad \boldsymbol{\alpha}_i^t \quad \boldsymbol{0}^t \quad \ldots \quad \boldsymbol{0}^t\right]^t.$$

[5] Wright et al., IEEE Trans. PAMI, 2009

Sparse representation for face recognition[5]

- Assumption: New face of person i lies in linear span of training samples associated with class i

$$\boldsymbol{y} = \alpha_{i,1}\boldsymbol{v}_{i,1} + \alpha_{i,2}\boldsymbol{v}_{i,2} + \ldots + \alpha_{i,N_i}\boldsymbol{v}_{i,N_i} = \boldsymbol{A}_i\boldsymbol{\alpha}_i$$

$\left(\boldsymbol{y} \in \mathbb{R}^n, \boldsymbol{A}_i \in \mathbb{R}^{n \times N_i}, \boldsymbol{\alpha}_i \in \mathbb{R}^{N_i}\right)$

- $\boldsymbol{y} \rightarrow$ sparse linear combination of all training samples:

$$\boldsymbol{y} = \left[\begin{array}{cccc} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \ldots & \boldsymbol{A}_K \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \\ \vdots \\ \boldsymbol{\alpha}_K \end{array}\right] = \boldsymbol{A}\boldsymbol{\alpha}$$

$\left(\boldsymbol{A} \in \mathbb{R}^{n \times T}, T = \sum_{i=1}^{K} N_i, \boldsymbol{\alpha} \in \mathbb{R}^T\right)$

- Membership of $\boldsymbol{y}$ encoded by sparse representation

$$\alpha = \left[\mathbf{0}^t \quad \ldots \quad \mathbf{0}^t \quad \alpha_i^t \quad \mathbf{0}^t \quad \ldots \quad \mathbf{0}^t\right]^t.$$

[5] Wright et al., IEEE Trans. PAMI, 2009

Sparse representation for face recognition[5]

- Assumption: New face of person i lies in linear span of training samples associated with class i

$$\boldsymbol{y} = \alpha_{i,1}\boldsymbol{v}_{i,1} + \alpha_{i,2}\boldsymbol{v}_{i,2} + \ldots + \alpha_{i,N_i}\boldsymbol{v}_{i,N_i} = \boldsymbol{A}_i\boldsymbol{\alpha}_i$$

$\left(\boldsymbol{y} \in \mathbb{R}^n, \boldsymbol{A}_i \in \mathbb{R}^{n\times N_i}, \boldsymbol{\alpha}_i \in \mathbb{R}^{N_i}\right)$

- $\boldsymbol{y} \rightarrow$ sparse linear combination of all training samples:

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \ldots & \boldsymbol{A}_K \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \\ \vdots \\ \boldsymbol{\alpha}_K \end{bmatrix} = \boldsymbol{A}\boldsymbol{\alpha}$$

$\left(\boldsymbol{A} \in \mathbb{R}^{n\times T}, T = \sum_{i=1}^{K} N_i, \boldsymbol{\alpha} \in \mathbb{R}^T\right)$

- Membership of $\boldsymbol{y}$ encoded by sparse representation

$$\boldsymbol{\alpha} = \begin{bmatrix}\mathbf{0}^t & \ldots & \mathbf{0}^t & \boldsymbol{\alpha}_i^t & \mathbf{0}^t & \ldots & \mathbf{0}^t\end{bmatrix}^t.$$

[5] Wright et al., IEEE Trans. PAMI, 2009

Sparse representation for face recognition (contd.)

- Solve the sparse recovery problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_0 \quad \text{subject to} \quad \|\boldsymbol{A}\boldsymbol{\alpha} - \boldsymbol{y}\|_2 \leq \epsilon$$

  Convex relaxation (if solution is sparse enough):

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_1 \quad \text{subject to} \quad \|\boldsymbol{A}\boldsymbol{\alpha} - \boldsymbol{y}\|_2 \leq \epsilon$$

- Class decision based on reconstruction residuals:

$$\text{identity}(\boldsymbol{y}) = \arg\min_i \|\boldsymbol{y} - \boldsymbol{A}\delta_i(\hat{\boldsymbol{\alpha}})\|_2$$

  $\delta_i(\hat{\boldsymbol{\alpha}}) \rightarrow$ only non-zero entries are those associated with class i

- Robustness to variety of distortions.

Sparse representation for face recognition (contd.)

- Solve the sparse recovery problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_0 \quad \text{subject to} \quad \|\boldsymbol{A}\boldsymbol{\alpha} - \boldsymbol{y}\|_2 \leq \epsilon$$

Convex relaxation (if solution is sparse enough):

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_1 \quad \text{subject to} \quad \|\boldsymbol{A}\boldsymbol{\alpha} - \boldsymbol{y}\|_2 \leq \epsilon$$

- Class decision based on reconstruction residuals:

$$\text{identity}(\boldsymbol{y}) = \arg\min_i \|\boldsymbol{y} - \boldsymbol{A}\delta_i(\hat{\boldsymbol{\alpha}})\|_2$$

$\delta_i(\hat{\boldsymbol{\alpha}}) \rightarrow$ only non-zero entries are those associated with class i

- Robustness to variety of distortions.

Sparse representation for face recognition (contd.)

- Solve the sparse recovery problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_0 \quad \text{subject to} \quad \|\boldsymbol{A}\boldsymbol{\alpha} - \boldsymbol{y}\|_2 \leq \epsilon$$

  Convex relaxation (if solution is sparse enough):

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_1 \quad \text{subject to} \quad \|\boldsymbol{A}\boldsymbol{\alpha} - \boldsymbol{y}\|_2 \leq \epsilon$$

- Class decision based on reconstruction residuals:

$$\text{identity}(\boldsymbol{y}) = \arg\min_i \|\boldsymbol{y} - \boldsymbol{A}\delta_i(\hat{\boldsymbol{\alpha}})\|_2$$

  $\delta_i(\hat{\boldsymbol{\alpha}}) \rightarrow$ only non-zero entries are those associated with class i

- Robustness to variety of distortions.

Sample result

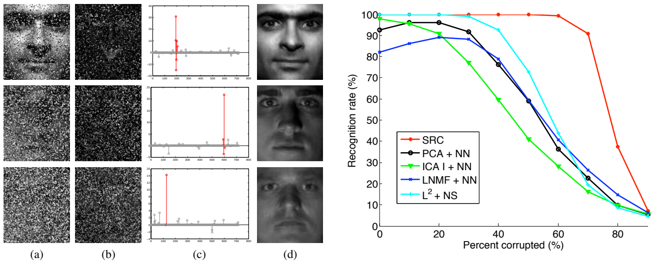
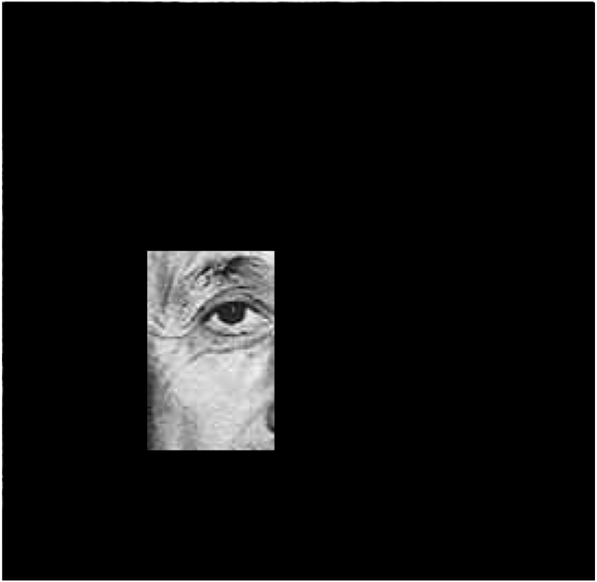


Figure: Left: Varying amounts of random pixel corruption. Right: Recognition rate variation with corruption.

Drawbacks and challenges

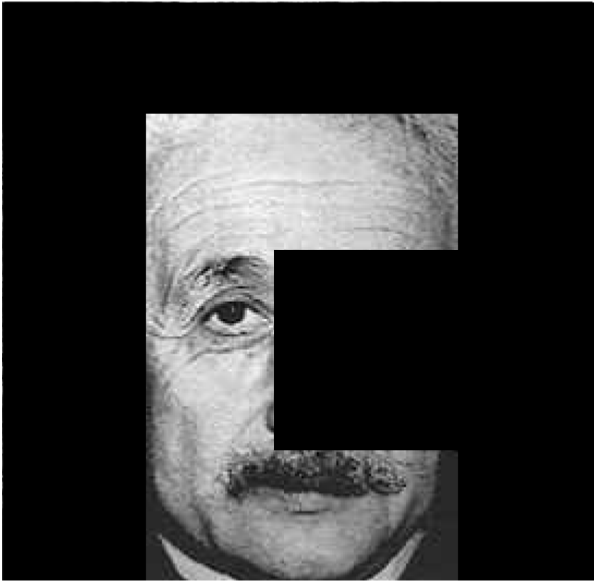
1. Accurate registration of training and test images necessary
   - Misalignment: translation, rotation, scale; pose and illumination variation; occlusion
   - Computational cost and feasibility in practical recognition systems
2. Class decision using reconstruction residuals
   - Does not capture inter-class variation
   - Sparse representations inherently discriminative.

Local features for recognition: Motivation

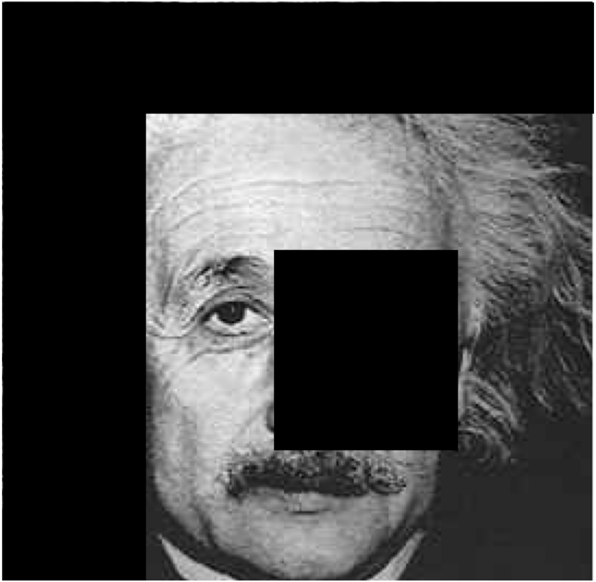


Local features for recognition: Motivation

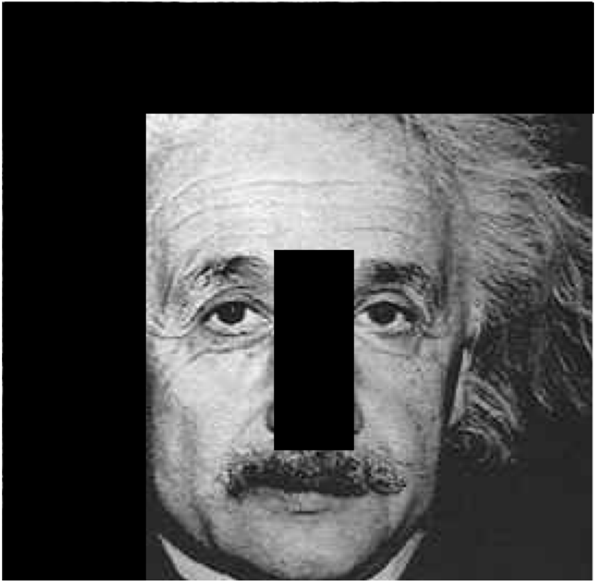
Local features for recognition: Motivation



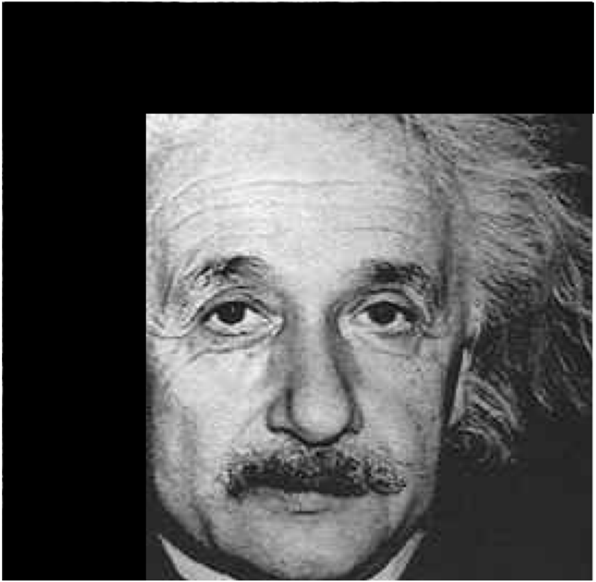
Local features for recognition: Motivation



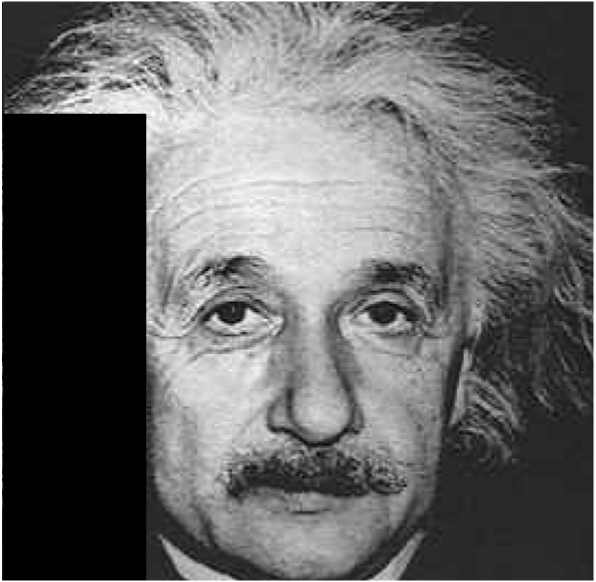
Local features for recognition: Motivation



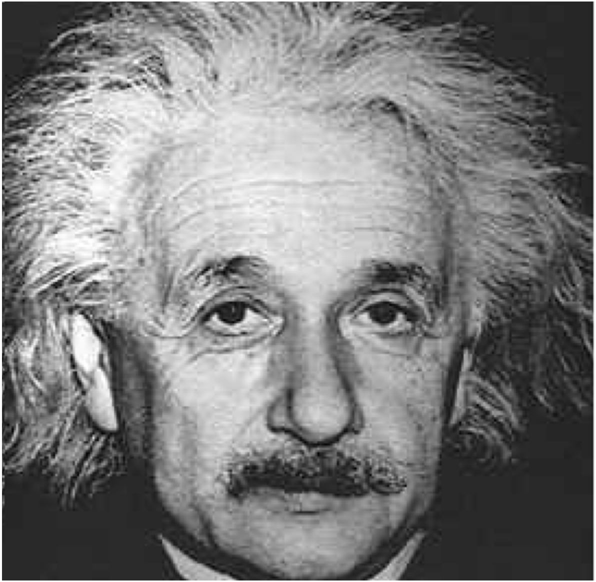
Local features for recognition: Motivation



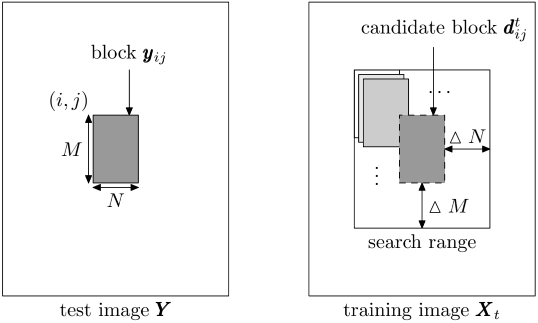
Local features for recognition: Motivation



Local features for recognition: Motivation



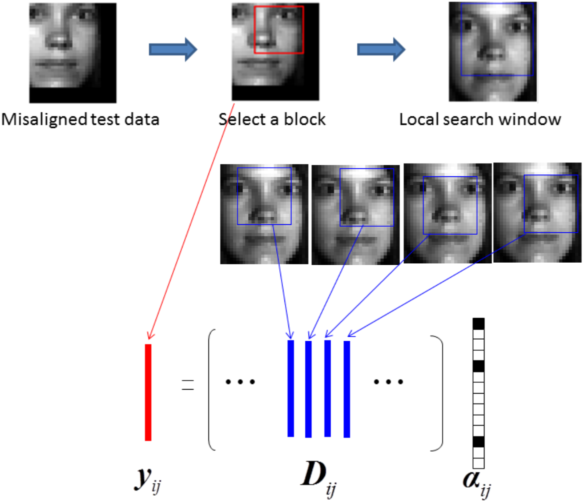
Local sparsity model for robust face recognition[6]



- Inspired by block-based motion estimation
- Block-sparsity model using locally adaptive dictionary $\boldsymbol{D}_{ij}$
- No explicit estimation of registration parameters.

[6] Chen et al., IEEE ICIP 2010

How to build the dictionary?



Block sparsity for face recognition

- For block $\boldsymbol{y}_{ij}$ in misaligned test image $\boldsymbol{Y}$,

$$\hat{\boldsymbol{\alpha}}_{ij} = \arg\min \|\boldsymbol{\alpha}_{ij}\|_0 \quad \text{subject to} \quad \|\boldsymbol{D}_{ij}\boldsymbol{\alpha}_{ij} - \boldsymbol{y}_{ij}\|_2 \leq \epsilon$$

- Identity of block $\boldsymbol{y}_{ij}$: determined by the residuals

$$\text{identity}\,(\boldsymbol{y}_{ij}) = \arg\min_{k=1,\ldots,K} r_{ij}^k,$$

$$r_{ij}^k = \left\| \boldsymbol{y}_{ij} - \boldsymbol{D}_{ij}^k \hat{\boldsymbol{\alpha}}_{ij}^k \right\|_2$$

- Select multiple local blocks from image → obtain individual classification decisions
- How to combine local decisions into global class decision?
  - Voting and ensemble classifiers
  - Challenge: Principled strategy to combine correlated sparse representations.

Block sparsity for face recognition

- For block $\boldsymbol{y}_{ij}$ in misaligned test image $\boldsymbol{Y}$,

$$\hat{\boldsymbol{\alpha}}_{ij} = \arg\min \|\boldsymbol{\alpha}_{ij}\|_0 \quad \text{subject to} \quad \|\boldsymbol{D}_{ij}\boldsymbol{\alpha}_{ij} - \boldsymbol{y}_{ij}\|_2 \leq \epsilon$$

- Identity of block $\boldsymbol{y}_{ij}$: determined by the residuals

$$\text{identity}\,(\boldsymbol{y}_{ij}) = \arg\min_{k=1,\dots,K} r_{ij}^k,$$

$$r_{ij}^k = \left\|\boldsymbol{y}_{ij} - \boldsymbol{D}_{ij}^k \hat{\boldsymbol{\alpha}}_{ij}^k\right\|_2$$

- Select multiple local blocks from image → obtain individual classification decisions
- How to combine local decisions into global class decision?
  - Voting and ensemble classifiers
  - Challenge: Principled strategy to combine correlated sparse representations.

Block sparsity for face recognition

- For block $\boldsymbol{y}_{ij}$ in misaligned test image $\boldsymbol{Y}$,

$$\hat{\boldsymbol{\alpha}}_{ij} = \arg\min \|\boldsymbol{\alpha}_{ij}\|_0 \quad \text{subject to} \quad \|\boldsymbol{D}_{ij}\boldsymbol{\alpha}_{ij} - \boldsymbol{y}_{ij}\|_2 \leq \epsilon$$

- Identity of block $\boldsymbol{y}_{ij}$: determined by the residuals

$$\text{identity}\,(\boldsymbol{y}_{ij}) = \arg\min_{k=1,\ldots,K} r_{ij}^k,$$

$$r_{ij}^k = \left\|\boldsymbol{y}_{ij} - \boldsymbol{D}_{ij}^k \hat{\boldsymbol{\alpha}}_{ij}^k\right\|_2$$

- Select multiple local blocks from image → obtain individual classification decisions
- How to combine local decisions into global class decision?
  - Voting and ensemble classifiers
  - Challenge: Principled strategy to combine correlated sparse representations.

Probabilistic graphical models: A brief review

- Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ defined by a set of nodes $\mathcal{V} = \{1, \ldots, n\}$, and a set of edges $\mathcal{E} \subset \binom{\mathcal{V}}{2}$.
- Graphical model: Random vector defined on a graph; nodes represent random variables, edges reveal conditional dependencies.
- Graph structure defines factorization of joint probability distribution

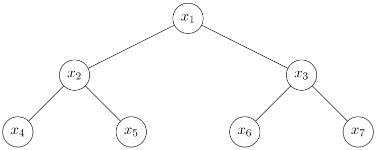


Figure: Tree - connected acyclic graph.

$$f(\boldsymbol{x}) = f(x_1)f(x_2|x_1)f(x_3|x_1)f(x_4|x_2)f(x_5|x_2)f(x_6|x_3)f(x_7|x_3).$$

Learning graphical models

- Generative learning: Single graph to minimize approximation error[7]

$$\text{Given } p, \text{ find } \hat{p} = \arg \min_{\hat{p} \text{ is a tree}} D(p||\hat{p}).$$

$$\left(D(p||\hat{p}) := \int p(\boldsymbol{x}) \log \left(\frac{p(\boldsymbol{x})}{\hat{p}(\boldsymbol{x})} \right) d\boldsymbol{x} \to \text{ KL-divergence.} \right)$$

- Discriminative learning: Simultaneously learn a pair of graphs to approximately minimize classification error[8]

[7] Chow et al., IEEE Trans. Inf. Theory, 1968

[8] Tan et al., IEEE Trans. Signal Process., 2010

Learning graphical models

- Generative learning: Single graph to minimize approximation error[7]

$$\text{Given } p, \text{ find } \hat{p} = \arg \min_{\hat{p} \text{ is a tree}} D(p||\hat{p}).$$

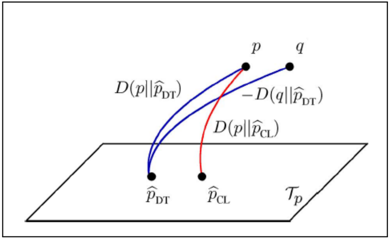
$$\left(D(p||\hat{p}) := \int p(\boldsymbol{x}) \log \left(\frac{p(\boldsymbol{x})}{\hat{p}(\boldsymbol{x})} \right) d\boldsymbol{x} \rightarrow \text{ KL-divergence.} \right)$$

- Discriminative learning: Simultaneously learn a pair of graphs to approximately minimize classification error[8]

Tree-approximate J-divergence:

$$\widehat{J}(\widehat{p}, \widehat{q}; p, q) := \int_{\Omega \subset \mathcal{X}^n} (p(\boldsymbol{x}) - q(\boldsymbol{x})) \log \left(\frac{\widehat{p}(\boldsymbol{x})}{\widehat{q}(\boldsymbol{x})} \right) d\boldsymbol{x}.$$

$$(\widehat{p}, \widehat{q}) = \arg \max_{\widehat{p} \in \mathcal{T}_p, \widehat{q} \in \mathcal{T}_q} \widehat{J}(\widehat{p}, \widehat{q}; p, q).$$



(Figure courtesy Tan et al.)

[7] Chow et al., IEEE Trans. Inf. Theory, 1968

[8] Tan et al., IEEE Trans. Signal Process., 2010

Discriminative graphical models for classification[9]

Two-stage framework:

1. Acquire multiple signal representations, which are conditionally correlated per class
2. Mine dependencies between different features via boosting on discriminative graphs.

[9] Srinivas et al., IEEE ICIP, Sep. 2011

Contribution: Robust face recognition using discriminative graphical models

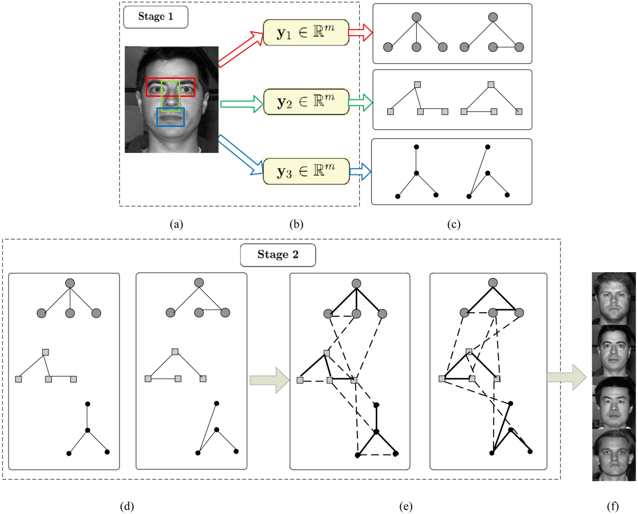
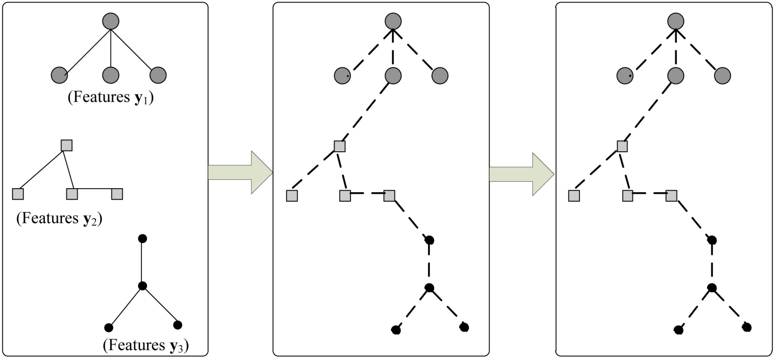


Figure: Learning graphs on sparse features.

Learning discriminative graphs: An illustration[10]

Iteration 1:



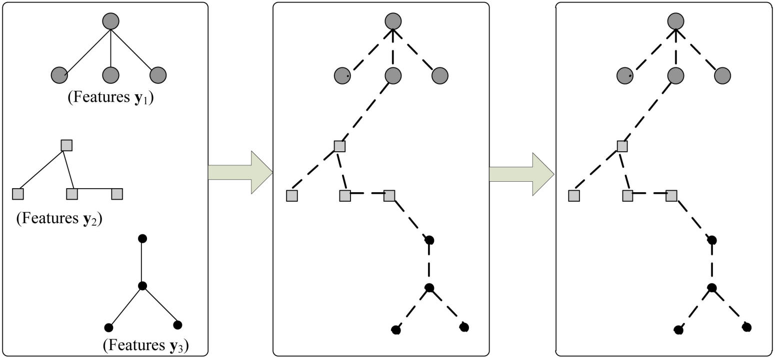
(a) Initial graph (b) Newly-learned tree (c) Augmented graph

Re-weighting of training samples (boosting) → learn another tree . . .

[10] Shown for distribution p; graph for q learnt analogously.

Learning discriminative graphs: An illustration[10]

Iteration 1:



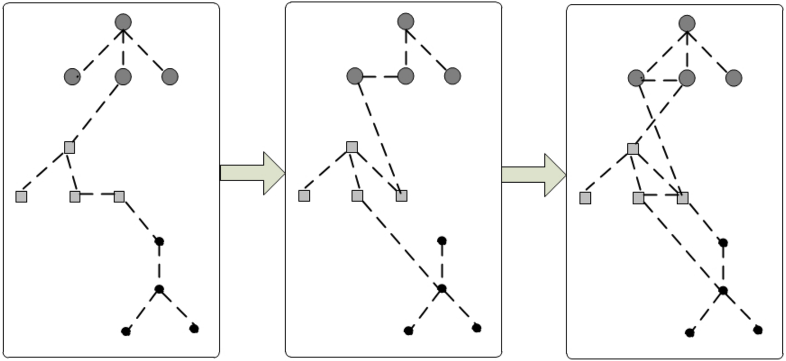
(a) Initial graph (b) Newly-learned tree (c) Augmented graph

Re-weighting of training samples (boosting) → learn another tree . . .

[10] Shown for distribution p; graph for q learnt analogously.

Learning discriminative graphs: An illustration

Iteration 2:

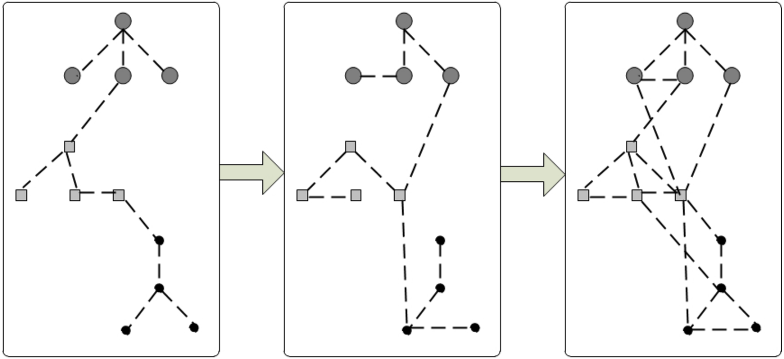


(a) Initial graph (b) Newly-learned tree (c) Augmented graph

Newly introduced edges crucial for capturing correlations amongst distinct signal representations.

Learning discriminative graphs: An illustration

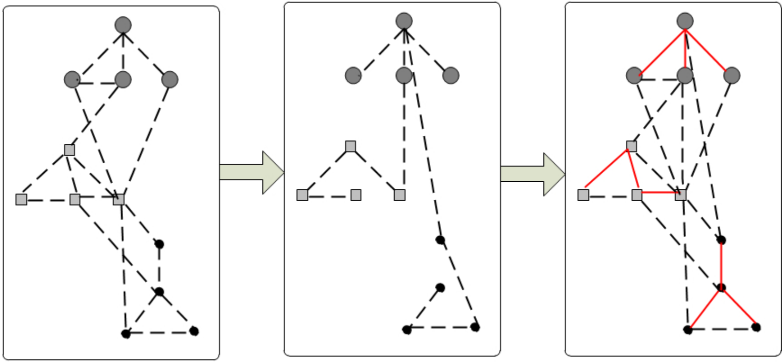
Iteration 3:



(a) Initial graph (b) Newly-learned tree (c) Augmented graph

Learning discriminative graphs: An illustration

Iteration 4:



(a) Initial graph (b) Newly-learned tree (c) Augmented graph

Stopping criterion

How many edges to learn?

1. Cross-validation

2. Using the J-divergence:

$$\widehat{J}(\widehat{p},\widehat{q};p,q) := \int_{\Omega\subset\mathcal{X}^n} (p(\boldsymbol{x}) - q(\boldsymbol{x})) \log\left(\frac{\widehat{p}(\boldsymbol{x})}{\widehat{q}(\boldsymbol{x})}\right) d\boldsymbol{x}.$$

Stopping criterion:
Stop after i boosting iterations if:

$$\frac{\widehat{J}^{(i+1)}(\widehat{p},\widehat{q};p,q) - \widehat{J}^{(i)}(\widehat{p},\widehat{q};p,q)}{\widehat{J}^{(i)}(\widehat{p},\widehat{q};p,q)} < \epsilon$$

Learning thicker graphical models

- Final boosted classifier:

$$\begin{aligned} H_T(\boldsymbol{x}) &= \operatorname{sgn}\left[\sum_{t=1}^{T} \alpha_t \log\left(\frac{\widehat{p}_t(\boldsymbol{x})}{\widehat{q}_t(\boldsymbol{x})}\right)\right] = \operatorname{sgn}\left[\log \prod_{t=1}^{T}\left(\frac{\widehat{p}_t(\boldsymbol{x})}{\widehat{q}_t(\boldsymbol{x})}\right)^{\alpha_t}\right] \\ &= \operatorname{sgn}\left[\log\left(\frac{\prod_{t=1}^{T}(\widehat{p}_t(\boldsymbol{x}))^{\alpha_t}}{\prod_{t=1}^{T}(\widehat{q}_t(\boldsymbol{x}))^{\alpha_t}}\right)\right] = \operatorname{sgn}\left[\log\left(\frac{\widehat{p}(\boldsymbol{x})}{\widehat{q}(\boldsymbol{x})}\right)\right] \end{aligned}$$

Define:

$$Z_p(\boldsymbol{\alpha}) = Z_p(\alpha_1, \ldots, \alpha_T) = \sum_{\boldsymbol{x}} \hat{p}(\boldsymbol{x}); Z_q(\boldsymbol{\alpha}) = \sum_{\boldsymbol{x}} \hat{q}(\boldsymbol{x})$$

- Normalized distributions for inference: $\frac{\hat{p}(\boldsymbol{x})}{Z_p(\boldsymbol{\alpha})}, \frac{\hat{q}(\boldsymbol{x})}{Z_q(\boldsymbol{\alpha})}$

  $\rightarrow$ Thicker graphical models learnt.

Robust face recognition using graphical models

$$i^* = \arg \max_{i \in \{1,\dots,K\}} \log \left(\frac{\hat{f}_p^i(\boldsymbol{\alpha})}{\hat{f}_q^i(\boldsymbol{\alpha})} \right). \quad (1)$$

Algorithm 1 Local-Sparse-Graphical-Model (LSGM) (Steps 1-4 offline)

1: **Feature extraction (training):** Obtain sparse representations $\boldsymbol{\alpha}_l, l = 1, \dots, P$ in $\mathbb{R}^m$ from facial features, using local block-sparsity model

2: **Initial disjoint graphs:**
For $l = 1, \dots, P$
Discriminatively learn pairs of m-node tree graphs $\mathcal{G}_l^p$ and $\mathcal{G}_l^q$ on $\{\boldsymbol{\alpha}_l\}$

3: Separately concatenate nodes corresponding to p and q respectively

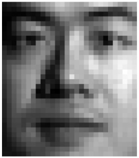
4: **Boosting on disjoint graphs**: Iteratively thicken initial disjoint graphs via boosting to obtain final graphs $\mathcal{G}^p$ and $\mathcal{G}^q$

{**Online process**}

5: **Feature extraction (test):** Obtain sparse representations $\boldsymbol{\alpha}_l, l = 1, \dots, P$ in $\mathbb{R}^m$ from test image

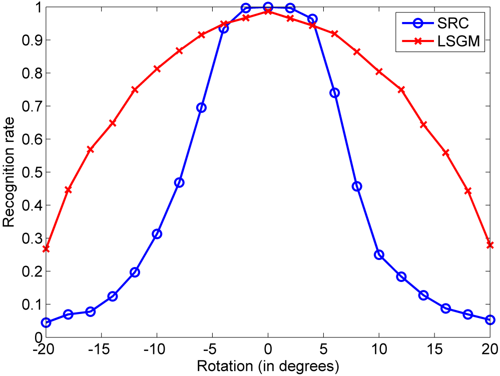
6: **Inference**: Classify based on output of the resulting classifier using (1).

Results: Rotation (Extended Yale B)



Original

Rotated



- SRC: sparse representation-based classification
- LSGM: local sparsity with graphical models.

Results: Scaling (Yale)

Original

Scaled

Table: Recognition rate using SRC and LSGM.

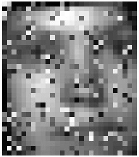
SF	1	1.071	1.143	1.214	1.286
1	100	100	98.0	88.2	76.5
	98.8	98.2	98.5	97.5	97.5
1.063	99.7	96.5	86.1	68.5	50.3
	97.5	96.7	96.0	96.0	93.5
1.125	83.8	70.2	49.8	33.6	26.2
	97.4	96.5	96.2	95.2	93.2
1.188	54.5	43.7	26.8	20.0	18.0
	94.9	92.9	91.6	89.4	87.1
1.25	36.1	27.2	20.9	16.6	12.3
	94.9	93.0	92.2	87.9	82.0
1.313	31.5	24.3	16.7	13.9	10.6
	90.7	90.4	84.1	81.0	75.5

Table: Overall recognition rates.

Method	Recog. rate (%)
LSGM	**89.4**
SRC	**60.8**
Eigen-NS	55.5
Eigen-SVM	56.7
Fisher-NS	54.1
Fisher-SVM	57.1

Results: Scaling and random pixel corruption (Yale)

Original



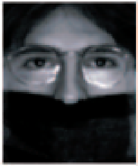
Distorted

Table: Test images scaled and subjected to random pixel corruption.

Method	Recognition rate (%)
LSGM	**96.3**
SRC	**93.2**
Eigen-NS	54.3
Eigen-SVM	58.5
Fisher-NS	56.2
Fisher-SVM	59.9

Results: Scaling and disguise (AR database)

Sunglasses

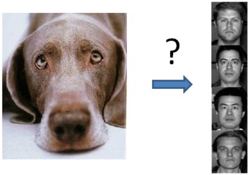


Scarf

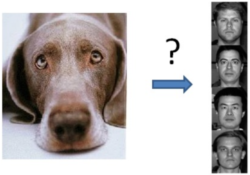
Table: Test images scaled and subjects wear disguise.

Method	Recog. rate (%) Sunglasses	Recog. rate (%) Scarves
LSGM	**96.0**	**92.9**
SRC	**93.5**	**90.1**
Eigen-NS	47.2	29.6
Eigen-SVM	53.5	34.5
Fisher-NS	57.9	41.7
Fisher-SVM	61.7	43.6

Results: Outlier rejection



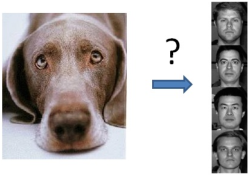
Results: Outlier rejection



- Subset of total number of classes for training
- Test images rotated by 5 degrees
- ROC for SRC:

$$\text{SCI}(\boldsymbol{\alpha}) = \frac{K \cdot \max_i \frac{\|\delta_i(\boldsymbol{\alpha})\|_1}{\|\boldsymbol{\alpha}\|_1} - 1}{K - 1} \in [0, 1].$$

Results: Outlier rejection



- Subset of total number of classes for training
- Test images rotated by 5 degrees
- ROC for SRC:

$$\mathrm{SCI}(\boldsymbol{\alpha}) = \frac{K \cdot \max_i \frac{\|\delta_i(\boldsymbol{\alpha})\|_1}{\|\boldsymbol{\alpha}\|_1} - 1}{K-1} \in [0,1].$$

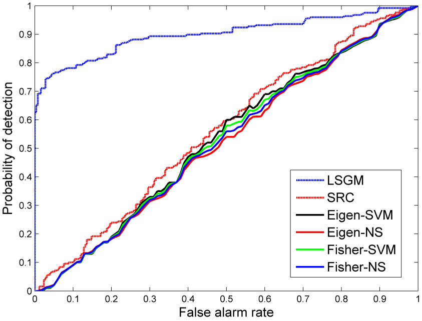


Figure: ROC curves for outlier rejection.

Conclusions

- Probabilistic graphical model framework for robust face recognition
  - Local block-based sparsity model $\rightarrow$ robustness to alignment errors and variety of distortions
  - Inspired by human perception $\rightarrow$ features built from informative regions of face
  - Explicitly captures conditional correlations between local sparse features.

Thank you

Questions?

Backup Slides

Edge weights:

$$\psi^p_{i,j} := \mathbb{E}_{\widetilde{p}_{i,j}}\left[\log \frac{\widetilde{p}_{i,j}}{\widetilde{p}_i \widetilde{p}_j}\right] - \mathbb{E}_{\widetilde{q}_{i,j}}\left[\log \frac{\widetilde{p}_{i,j}}{\widetilde{p}_i \widetilde{p}_j}\right]$$

$$\psi^q_{i,j} := \mathbb{E}_{\widetilde{q}_{i,j}}\left[\log \frac{\widetilde{q}_{i,j}}{\widetilde{q}_i \widetilde{q}_j}\right] - \mathbb{E}_{\widetilde{p}_{i,j}}\left[\log \frac{\widetilde{q}_{i,j}}{\widetilde{q}_i \widetilde{q}_j}\right].$$

Algorithm 2 Discriminative trees (DT)

Given: Training sets $\mathcal{T}_p$ and $\mathcal{T}_q$.

1: Estimate pairwise statistics $\widetilde{p}_{i,j}(x_i, x_j)$, $\widetilde{q}_{i,j}(x_i, x_j)$ for all edges (i, j).

2: Compute edge weights $\psi^p_{i,j}$ and $\psi^q_{i,j}$ for all edges (i, j).

3: Find $\mathcal{E}_{\widehat{p}} =$ MWST$(\psi^p_{i,j})$ and $\mathcal{E}_{\widehat{q}} =$ MWST$(\psi^q_{i,j})$.

4: Get $\widehat{p}$ by projection of $\widetilde{p}$ onto $\mathcal{E}_{\widehat{p}}$; likewise $\widehat{q}$.

5: LRT using $\widehat{p}$ and $\widehat{q}$.

Boosting

Algorithm 3 AdaBoost learning algorithm

1: Input data $(x_i, y_i),\ i = 1, 2, \ldots, N$, where $x_i \in S,\ y_i \in \{-1, +1\}$
2: Initialize $D_1(i) = \frac{1}{N}, i = 1, 2, \ldots, N$
3: For $t = 1, 2, \ldots, T$:

- Train weak learner using distribution D_t
- Determine weak hypothesis $h_t : S \mapsto \mathbb{R}$ with error ϵ_t
- Choose $\beta_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$
- $D_{t+1}(i) = \frac{1}{Z_t} \{D_t(i) \exp(-\beta_t y_i h_t(x_i))\}$, where Z_t is a normalization factor

4: Output soft decision $H(x) = \text{sign} \left[\sum_{t=1}^{T} \beta_t h_t(x) \right]$.

- Distribution of weights over the training set
- In each iteration, weak learner h_t minimizes weighted training error
- Weights on incorrectly classified samples increased $\rightarrow$ slow learners penalized for harder examples.