9.27 Paper Presentation:
PCA & kernel methods

Yibin Xiong

September 27, 2021

Table of Contents

Big Picture

- Problem: "one sample per person" – given just 1 picture of each person as training data, predict the identity (class) of a test image where there are *different poses, lightings*, etc. [2]

Big Picture

- Problem: "one sample per person" – given just 1 picture of each person as training data, predict the identity (class) of a test image where there are *different poses, lightings*, etc. [2] robustness
- Challenge:
  ◦ High-dimensional input, "curse of dimensionality"
  ◦ Too few training examples per class *prevents* some more powerful algorithms, such as LDA-based, probabilistic-based, and SVM-based methods, to be used or *reduces* them to eigenface.
- Solutions: Reduce dimension & Enlarge training set

Big Picture

- Problem: "one sample per person" – given just 1 picture of each person as training data, predict the identity (class) of a test image where there are *different poses, lightings*, etc. [2] robustness
- Challenge:
  ◦ High-dimensional input, "curse of dimensionality"
  ◦ Too few training examples per class *prevents* some more powerful algorithms, such as LDA-based, probabilistic-based, and SVM-based methods, to be used or *reduces* them to eigenface.
- Solutions: Reduce dimension & Enlarge training set
  Holistic(global) models, **Local** models, and **Hybrid** models

Basic Model: Eigenface

▷ Idea: project the image into a lower-dimensional space in which important information are retained.

Basic Model: Eigenface

▷ Idea: project the image into a lower-dimensional space in which important information are retained.

▷ What information? Some patterns/factors that commonly appears in data and relates to the class label.

Basic Model: Eigenface

▷ Idea: project the image into a lower-dimensional space in which important information are retained.

▷ What information? Some patterns/factors that commonly appears in data and relates to the class label.

▷ PCA: Find the eigendecomposition (most significant factors) of the covariance matrix $C = E[XX^T]$;
Project the data into the directions that have most information (variation)

Let $X \in \mathbb{R}^{d\times N}$ be the data matrix where X_i is a long vector containing the pixel values of image i

$$C = \frac{1}{N}\sum_{i=1}^{N}(X_i - \mu)(X_i - \mu)^T$$

Base Model: Eigenface

$$C = \frac{1}{2N} \sum_{\ell(X_i)=\ell(X_j)} (X_i - X_j)(X_i - X_j)^T + \frac{1}{2N} \sum_{\ell(X_i)\neq\ell(X_j)} (X_i - X_j)(X_i - X_j)^T$$
$$\triangleq C_I + C_E$$

Here ℓ returns the class label (i.e. identity) of an image
C_I is the *intra*-person scatter matrix and C_E *inter*-person scatter matrix.
For one-shot case, we only have 1 example for each class, so C is just C_E.

Base Model: Eigenface

$$C = \frac{1}{2N} \sum_{\ell(X_i)=\ell(X_j)} (X_i - X_j)(X_i - X_j)^T + \frac{1}{2N} \sum_{\ell(X_i)\neq\ell(X_j)} (X_i - X_j)(X_i - X_j)^T$$
$$\triangleq C_I + C_E$$

Here ℓ returns the class label (i.e. identity) of an image
C_I is the *intra*-person scatter matrix and C_E *inter*-person scatter matrix.
For one-shot case, we only have 1 example for each class, so C is just C_E.

$$C = U\Lambda U^T$$

Then we can store the training data as lower-dimensional vectors
$x_i = \sum_{i=1}^{d} \alpha_i u_i \approx \sum_{i=1}^{m} \alpha_i u_i$ for some $m << d$.

Base Model: Eigenface

$$C = \frac{1}{2N} \sum_{\ell(X_i)=\ell(X_j)} (X_i - X_j)(X_i - X_j)^T + \frac{1}{2N} \sum_{\ell(X_i)\neq\ell(X_j)} (X_i - X_j)(X_i - X_j)^T$$
$$\triangleq C_I + C_E$$

Here ℓ returns the class label (i.e. identity) of an image
C_I is the *intra*-person scatter matrix and C_E *inter*-person scatter matrix.
For one-shot case, we only have 1 example for each class, so C is just C_E.

$$C = U\Lambda U^T$$

Then we can store the training data as lower-dimensional vectors $x_i = \sum_{i=1}^{d} \alpha_i u_i \approx \sum_{i=1}^{m} \alpha_i u_i$ for some $m << d$.

Given a new image x_{new}, we just first map it to the lower-dimensional feature space by $y_{new} = U^T x_{new}$ and find x_i whose coordinates in terms of u are closest to y_{new}

More Advanced Methods

Holistic Methods

- Utilizes the whole image as input (i.e. input is still high-dimensional)
- Key challenges:
  - How to address extremely small sample size
  - Intra-personal (within class) variation is not available
- Advantages:
  - Preserves *detailed* texture and shape information
  - Capture more global aspects compared with local methods

Local Methods

- Utilizes local facial features
- Key challenge is how to incorporate global configurational information into the model
- Advantage: lower-dimensional input

Hybrid Methods: use both

Holistic Methods: $(PC)^2A$

Let $I(x, y)$ be the intensity value of an $m \times n$ image at pixel (x, y). We compute the horizontal and vertical projections of the image

$$HI(x) = \sum_{y=1}^{n} I(x, y); \;\; VI(y) = \sum_{x=1}^{m} I(x, y)$$

Then the obtained projections are used to synthesize a new image, called *first-order* projection map

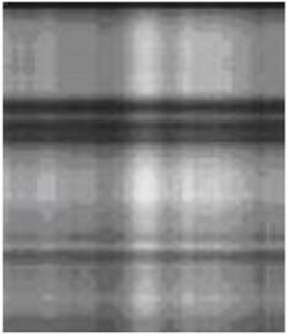
$$M_p(x, y) = \frac{HI(x)VI(y)}{\overline{J}}$$

This new image is *combined* with the original image as input to the model.

Holistic Methods: $(PC)^2A$

Intuition: important features are more salient and unimportant ones fade out

(a)



(b)

Enrich the information of eigenspace by perturbing the original image

Holistic Methods: Enriching the Training Set

We want to achieve high *robustness*, so more training examples with varying poses, lightings, etc. are needed.

$$\text{Generate } \begin{cases} \text{new representations} \\ \text{new training examples (using some prior knowledge)} \end{cases}$$

Holistic Methods: Enriching the Training Set

We want to achieve high *robustness*, so more training examples with varying poses, lightings, etc. are needed.

$$\text{Generate} \begin{cases} \text{new representations} \\ \text{new training examples (using some prior knowledge)} \end{cases}$$

◦ Representation: representational oriented component analysis (ROCA)

- Preprocess each image to **mitigate** the effects of light directions
- Apply some linear and non-linear models to generate 150 different representations
- For each representation, build an OCA classifier
- The final prediction is a linear combination of all OCA classifiers

◦ Image: Add a noise image to the original image

Holistic Methods: Enriching the Training Set

◦ Image: linear class

- Idea: exploit prior knowledge from prototypical examples in the domain
- Learn *class-specific* transformations

  Let the difference between the original image and reference image be ΔX, and the differences between the prototypical images and the reference image be ΔX_i.

  We assume $\Delta X \approx \sum_{i=1}^{q} \alpha_i \Delta_i$

  so we find $\alpha^* = \underset{\alpha}{argmin} \, \|\Delta X - \sum_{i=1}^{q} \alpha_i \Delta_i\|$

- The α_i's are transformation coefficients. Once we learn this, we can apply the transformation to generate new images

Prior knowledge $\Leftrightarrow$ regularization

Local Methods

1. local *feature-based* methods
   - Propose geometric features (e.g. the width of the head, the distances between eyes) to extract from images
   - Do similarity match on feature vectors to determine the similarity between a candidate and a training image

Difficulty: i) Sometimes difficult to extract;
ii) Not enough! lose global information

Local Methods

1. local *feature-based* methods
   - Propose geometric features (e.g. the width of the head, the distances between eyes) to extract from images
   - Do similarity match on feature vectors to determine the similarity between a candidate and a training image

Difficulty: i) Sometimes difficult to extract;
ii) Not enough! lose global information

Improved method: local features (Gabor) + global features (topological graph)

Local Methods

2. local *appearance* methods
   - local region partition (e.g. rectangles, strips)
   - local feature extraction (e.g. gray-value features, Gabor features)
   - (optional) feature selection: PCA or LDA
   - classification: the result of each feature's classifier is combined linearly

Hybrid Methods

Key challenges:

◦ Which features are chosen to combine and How

Table 2
Comparison of the local features and global features' sensitiveness to variations

Variation factors	Local features	Holistic features
Small variations	*Not sensitive*	*Sensitive*
Large variations	*Sensitive*	*Very sensitive*
Illuminations [103]	*Very sensitive*	*Sensitive*
Expressions [19,23]	*Not sensitive*	*Sensitive*
Pose [94]	*Sensitive*	*Very sensitive*
Noise [104]	*Very sensitive*	*Sensitive*
Occlusion [19,23]	*Not sensitive*	*Very sensitive*

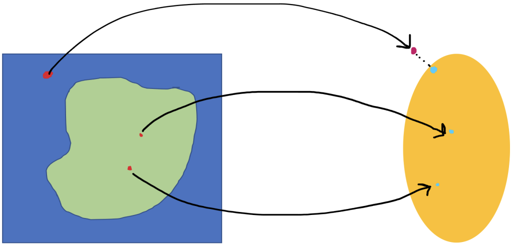
Table of Contents

Work [1]

- Problem: Given temporal sequences of images, predict the age of a new given image
- Challenges: Aging patterns are *personalized* and have *temporal* relations; data is highly incomplete (we only have images at some ages)
- Idea: First find the most appropriate aging pattern, then find the most appropriate position in this aging pattern to determine the age

Work [1]

- Problem: Given temporal sequences of images, predict the age of a new given image
- Challenges: Aging patterns are *personalized* and have *temporal* relations; data is highly incomplete (we only have images at some ages)
- Idea: First find the most appropriate aging pattern, then find the most appropriate position in this aging pattern to determine the age



AGES Algorithm: Aging Pattern Subspace

Idea: Construct a subspace using PCA
The projection is defined as

$$y = W^T(x - \mu)$$

but the aging pattern vector x is highly incomplete $\Longrightarrow$
use an EM-like algorithm to learn the mapping to the latent subspace

2 Interpretation of PCA:
i) Find directions that retain most variation of the data
ii) Find directions that minimizes the reconstruction error

AGES Algorithm

Let the k-th aging pattern be $x_k = \{x_k^a, x_k^m\}$, where x_k^a are the available features and x_k^m are the missing features.
We reconstruct x_k by

$$\hat{x}_k = \mu + \mathbf{W} y_k$$

Then we minimize the reconstruction error

$$\bar{\epsilon}^a = \frac{1}{N} \sum_{k=1}^{N} (x_k^a - \hat{x}_k^a)^T (x_k^a - \hat{x}_k^a)$$

AGES Algorithm

- Initialize x_k^m with the mean vector μ_k^m, which is calculated from other samples whose corresponding values are available
- With complete data, perform PCA to find $\mathbf{W}_0$ and μ_0
- (E-step) In each iteration $i+1$, first find the projection y_k using ONLY the available information. We solve the least-square solution of

$$[W_{i\binom{a}{k}}]y_k = x_k^a - [\mu_{i\binom{a}{k}}]$$

- (M-step) Calculate $\hat{x}_k$ by $\hat{x}_k = \mu + \mathbf{W}y_k$. Perform standard PCA to get $\mathbf{W}_{i+1}$ and μ_{i+1}

Repeat until the reconstruction error is smaller than some threshold

Interpretation

- Alternating between modeling *global* aging patterns and *personalized* aging patterns
- **W** captures *commonalities* of aging patterns
(think about when the covariance between x_i and x_j is large)
- In each iteration, the missing values are first estimated by the current global aging pattern model (i.e. using μ_i), then we refine the global model (subspace) using personalized aging patterns (available data).

Testing

- Given an unseen image I, first construct its feature vector b.
- We want to find z^* in the subspace that minimizes the reconstruction error.

  BUT, without knowing the **position** of I in an ordered sequence, we cannot evaluate the reconstruction error from an aging pattern to a single image
- Try to put I in each possible position in the aging pattern. Then we get p latent vectors z_j for $j = 1, \dots, p$ by placing b at position j of z^*.
- Note that now b is the only available information in z_j, so we find y_j by finding the least square solution of $W_{(j)} y_j = b - \mu_{(j)}$
- Then we can find y_j and evaluate the reconstruction error for each j

$$\epsilon^a(j) = (b - \mu_{(j)} - W_{(j)} y_j)^T (b - \mu_{(j)} - W_{(j)} y_j)$$

- Finally, we find $j^* = \underset{j}{argmin}\ \epsilon^a(j)$

Table of Contents

Incomplete supervision [3]

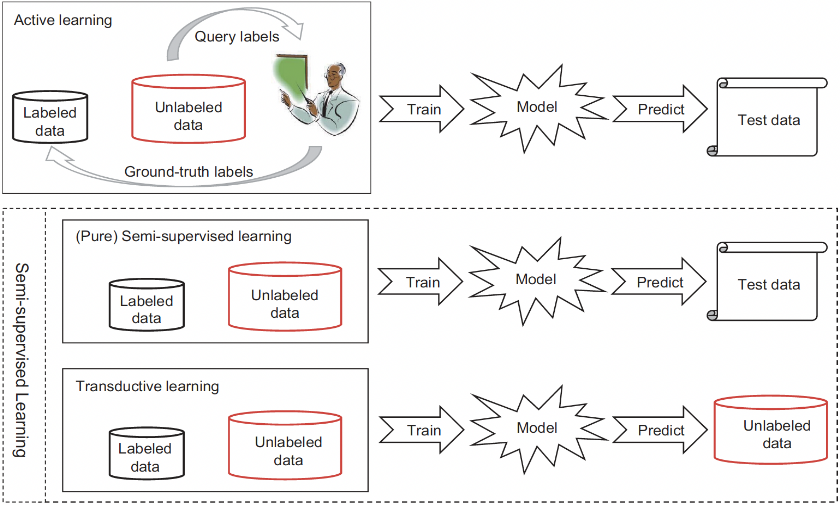


Figure 2. Active learning, (pure) semi-supervised learning and transductive learning.

Active Learning

Idea: given a small set of labeled data and abundant unlabeled data, select the most valuable unlabeled instances to query.

- Informativeness: how well an unlabeled instance helps reduce the uncertainty of a statistical model (or which one the model is NOT sure about)
  ◦ Uncertainty sampling: train 1 learner, then query the unlabeled instance on which the learner has least confidence
  ◦ Query-by-committee: train many learners, then query the one on which the learners disagree the most
  ▷ Unstable performance, highly dependent on the labeled data
- Representativeness: how well an unlabeled instance helps represent the structure of the input patterns
  ◦ Clustering methods
- Hybrid: leverage between the 2 criteria

Semi-supervised Learning

Idea: Model the structure/distribution of unlabeled data $p(x)$. Use MAP to model $p(y|x)$ indirectly.

Assumptions:

- Cluster assumption: Data have inherent cluster structure. Datapoints within a cluster have the same class label.
- Manifold assumption: Data lie on a manifold and nearby instances have similar predictions.

Algorithms:

- Generative methods: assume that both labeled and unlabeled data are generated from the same inherent model (e.g. EM)
- Graph-based methods construct a graph where nodes represent training instances and edges represent distance/similarity between them

  m instances $\Rightarrow \mathcal{O}(m^3)$, so not scalable

Semi-supervised Learning

- Low-density separation methods enforce the classification boundary to go across the less dense regions in the input space
  E.g. S3VM

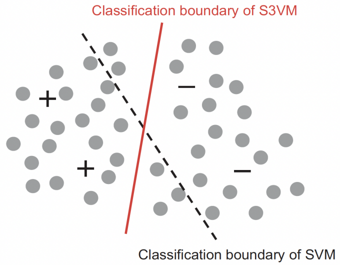


Figure 4. Illustration of different classification boundaries of SVM which considers only labeled data ("+/−" points), and S3VM which considers labeled and unlabeled data (gray points).

- Disagreement-based methods generate multiple learners for unlabeled data and focus on disagreement in the next iteration

Inexact Supervision: Multiinstance Learning

Task formulation:
Given a dataset $D = \{(X_1, y_1), \ldots, (X_m, y_m)\}$, where $X_i = \{x_{i1}, \ldots, x_{in}\}$ is a *bag*, predict the label of an unseen bag.
Note that instances within a bag are NOT i.i.d!

Methods:
∘ Most algorithms have their counterparts in multi-instance case, where the goal shifts from discrimination on instances to *bags*.
∘ Identify the *key instance* (assume it exists) in a bag that is strongly indicative of the label

Inaccurate Supervision

Assume that labels have random noise. We try to identify those that are potentially labeled incorrectly and correct them.

Crowdsourcing:

- Unlabeled data are outsourced (sent) to a large group of workers to label. There are some unreliable labels.
- If workers' quality and task difficulty can be modeled, then we can have a better estimate by a weighted sum of labels produced by different workers for different tasks

- Xin Geng, Zhi-Hua Zhou, and Kate Smith-Miles. "Automatic age estimation based on facial aging patterns". In: *IEEE Transactions on pattern analysis and machine intelligence* 29.12 (2007), pp. 2234–2240.
- Xiaoyang Tan et al. "Face recognition from a single image per person: A survey". In: *Pattern recognition* 39.9 (2006), pp. 1725–1745.
- Zhi-Hua Zhou. "A brief introduction to weakly supervised learning". In: *National science review* 5.1 (2018), pp. 44–53.