Dimensionality Reduction

Neil D. Lawrence

Department of Computer Science
Sheffield University

17th November 2015

Review

- Last time: Looked at classification.
- Introduced Naive Bayes and Logistic Regression.
- This time: Dimensionality reduction.

Outline

Clustering

Clustering

- Divide data into discrete groups according to characteristics.
  - For example different animal species.
  - Different political parties.
- Determine the allocation to the groups and (harder) number of different groups.

K-means Clustering
An Algorithm

- *Require*: Set of *K* cluster centers & assignment of each point to a cluster.
  - Initialize cluster centers as data points.
  - Assign each data point to nearest cluster center.
  - Update each cluster center by setting it to the mean of assigned data points.
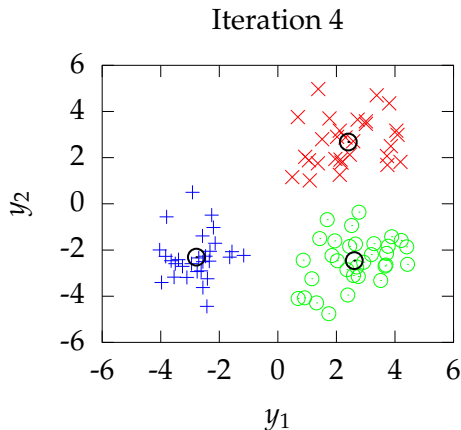
Objective Function

- This minimizes the objective:

$$\sum_{j=1}^{K} \sum_{i \text{ allocated to } j} \left(\mathbf{y}_{i,:} - \boldsymbol{\mu}_{j,:}\right)^{\top} \left(\mathbf{y}_{i,:} - \boldsymbol{\mu}_{j,:}\right)$$

- i.e. it minimizes the sum of Euclidean squared distances between points and their associated centers.
- The minimum is not guaranteed to be *global* or *unique*.
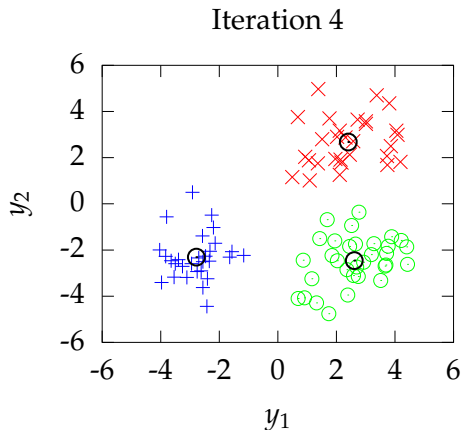 - This objective is a non-convex optimization problem.

K-means Clustering

- K-means clustering.
  - Update each center by setting to the mean of the allocated points.
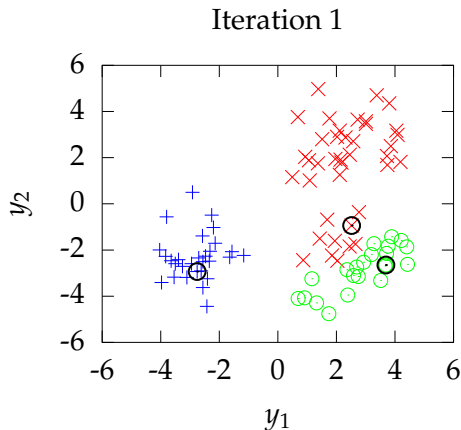
Iteration 4

K-means Clustering

- ▸ *K*-means clustering.
  - ▸ Allocate each data point to the nearest cluster center.

K-means Clustering

Iteration 1

- ▶ *K*-means clustering.
  - ▶ Update each center by setting to the mean of the allocated points.

K-means Clustering

- *K*-means clustering.
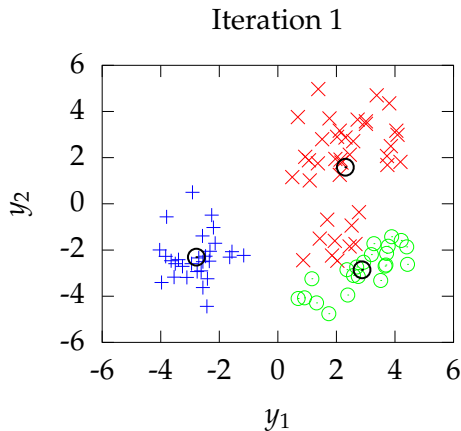 - Allocate each data point to the nearest cluster center.

K-means Clustering

- K-means clustering.
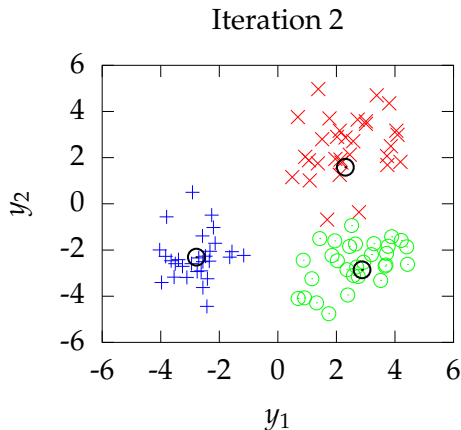 - Update each center by setting to the mean of the allocated points.

K-means Clustering

- *K*-means clustering.
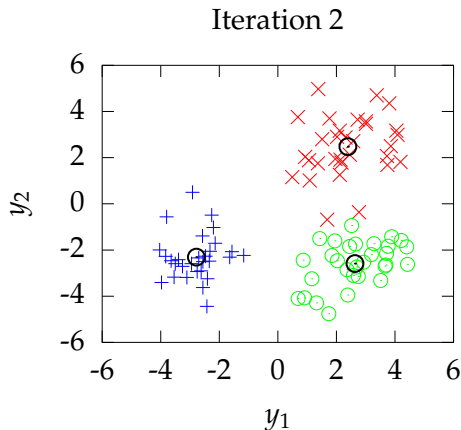 - Allocate each data point to the nearest cluster center.

Iteration 2

K-means Clustering

- K-means clustering.
  - Update each center by setting to the mean of the allocated points.

Iteration 3

K-means Clustering

- ▸ K-means clustering.
  - ▸ Allocate each data point to the nearest cluster center.

Iteration 3

K-means Clustering

Iteration 4

- ▶ *K*-means clustering.
  - ▶ Update each center by setting to the mean of the allocated points.
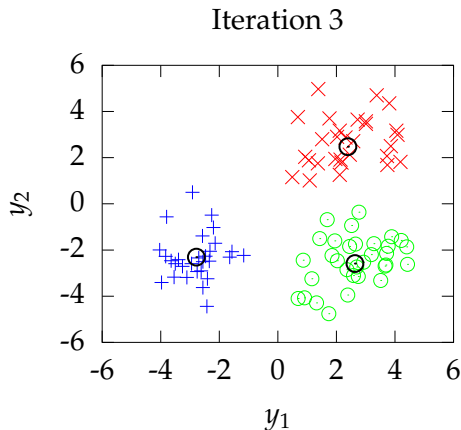
K-means Clustering

- *K*-means clustering.
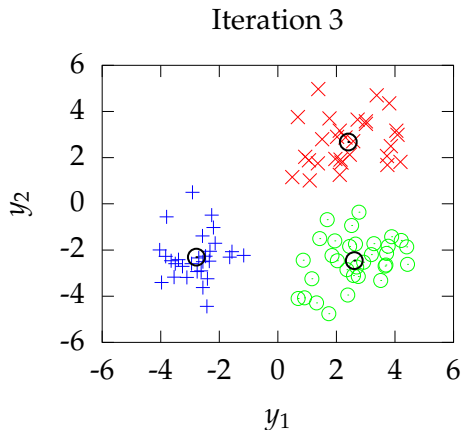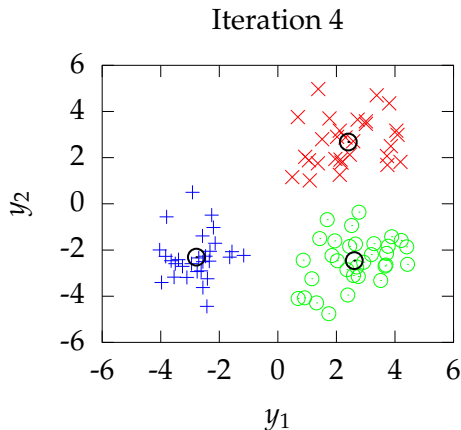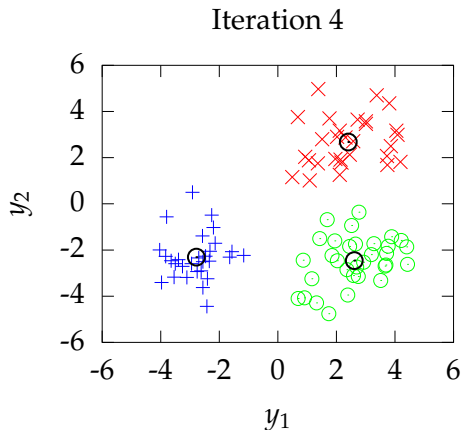 - Allocate each data point to the nearest cluster center.

K-means Clustering

- *K*-means clustering.
  - Allocation doesn't change so stop.

Iteration 4
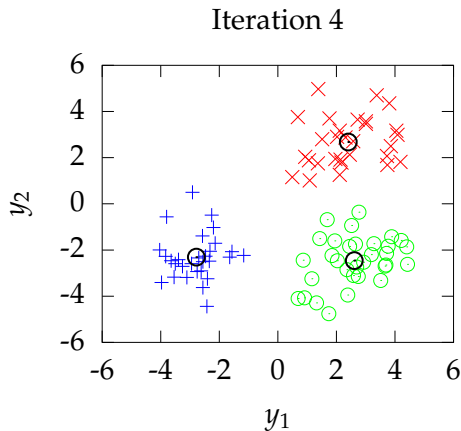
Other Clustering Approaches

- Spectral clustering (Shi and Malik, 2000; Ng et al., 2002).
  - Allows clusters which aren't convex hulls.
- Dirichlet processes
  - A probabilistic formulation for a clustering algorithm that is non-parameteric.

Outline

Clustering

Classification

High Dimensional Data

USPS Data Set Handwritten Digit

- ▶ 3648 Dimensions
- ▶ 64 rows by 57 columns
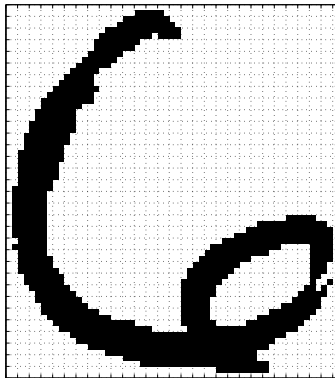
High Dimensional Data

USPS Data Set Handwritten Digit

- 3648 Dimensions
- 64 rows by 57 columns
- Space contains more than just this digit.

High Dimensional Data

USPS Data Set Handwritten Digit

- 3648 Dimensions
- 64 rows by 57 columns
- Space contains more than just this digit.
- Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!
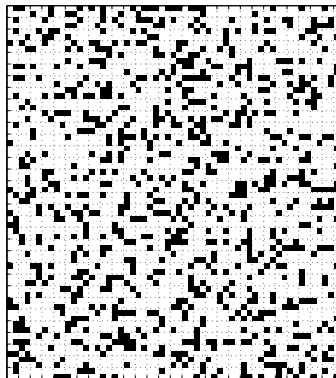
High Dimensional Data

USPS Data Set Handwritten Digit

- 3648 Dimensions
- 64 rows by 57 columns
- Space contains more than just this digit.
- Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!
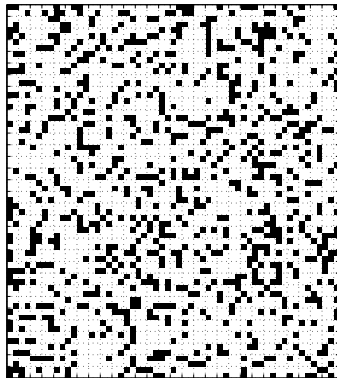
Simple Model of Digit
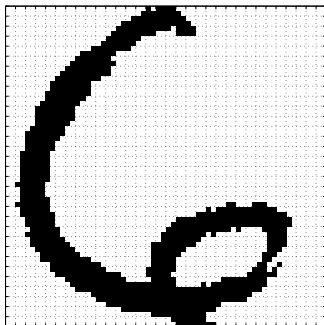
- Rotate a 'Prototype'

Simple Model of Digit

- Rotate a 'Prototype'

Simple Model of Digit
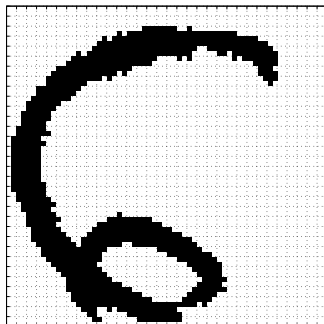
- Rotate a 'Prototype'

Simple Model of Digit
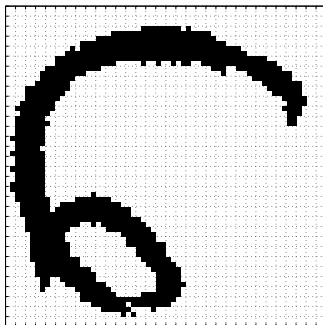
- Rotate a 'Prototype'

Simple Model of Digit
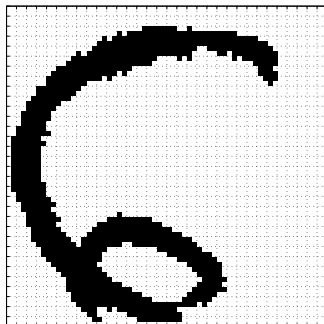
- Rotate a 'Prototype'

Simple Model of Digit
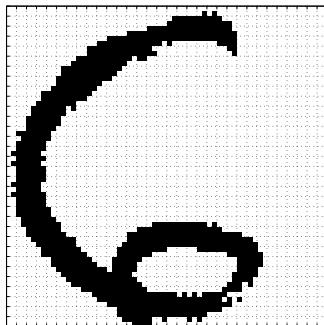
- Rotate a 'Prototype'

Simple Model of Digit
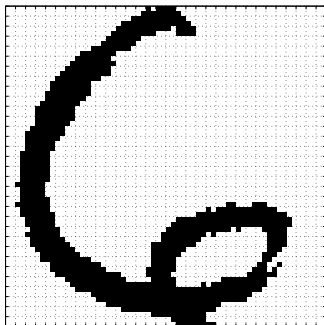
- Rotate a 'Prototype'

Simple Model of Digit
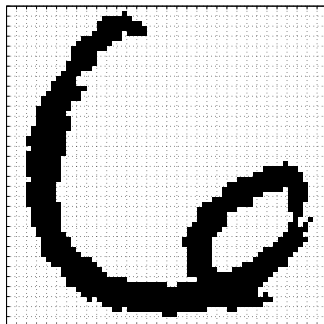
- Rotate a 'Prototype'

Simple Model of Digit

- Rotate a 'Prototype'

MATLAB Demo

```
demDigitsManifold([1 2], 'all')
```

MATLAB Demo

```
demDigitsManifold([1 2], 'all')
```
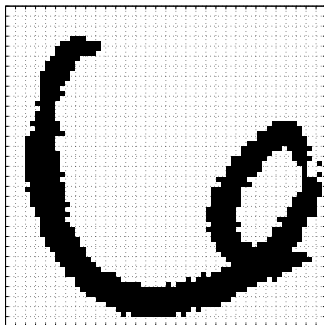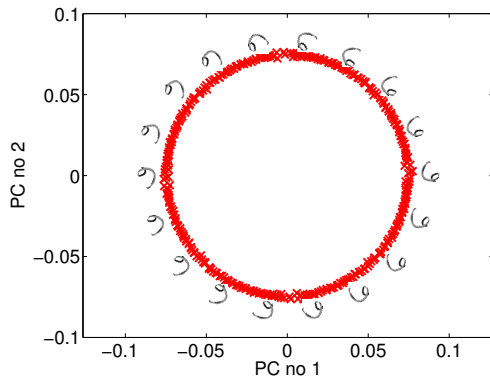
MATLAB Demo

```
demDigitsManifold([1 2], 'sixnine')
```
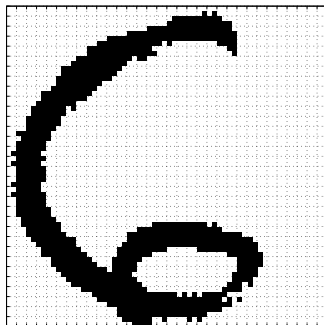
Low Dimensional Manifolds
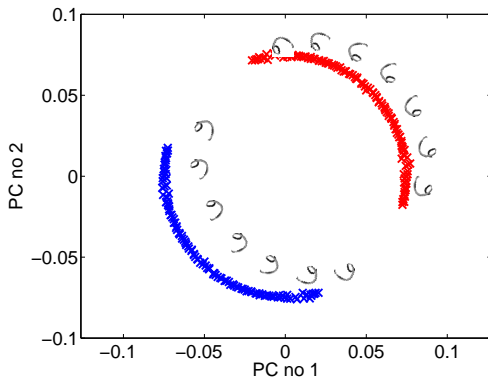
Pure Rotation is too Simple

- ▸ In practice the data may undergo several distortions.
    - ▸ *e.g.* digits undergo 'thinning', translation and rotation.
- ▸ For data with 'structure':
- ▸ we expect fewer distortions than dimensions;
- ▸ we therefore expect the data to live on a lower dimensional manifold.
- ▸ Conclusion: deal with high dimensional data by looking for lower dimensional non-linear embedding.

Principal Component Analysis

- How do we find these directions?
- Rotate to find directions in data with maximal variance.
    - This is known as PCA (Hotelling, 1933).
- Rotate data to extract directions of maximum variance.
- Do this by diagonalizing the sample covariance matrix

$$\mathbf{S} = n^{-1} \sum_{i=1}^{n} \left(\mathbf{y}_i - \mu \right) \left(\mathbf{y}_i - \mu \right)^{\top}$$

Principal Component Analysis

- Find a direction in the data, $\mathbf{x} = \mathbf{Ry}$, for which variance is maximized.

Lagrangian

- Solution is found via constrained optimisation (which uses *Lagrange* multipliers):

$$L(\mathbf{r}_1, \lambda_1) = \mathbf{r}_1^\top \mathbf{S} \mathbf{r}_1 + \lambda_1 \left(1 - \mathbf{r}_1^\top \mathbf{r}_1\right)$$

- Gradient with respect to $\mathbf{r}_1$

$$\frac{\mathrm{d}L(\mathbf{r}_1, \lambda_1)}{\mathrm{d}\mathbf{r}_1} = 2\mathbf{S}\mathbf{r}_1 - 2\lambda_1 \mathbf{r}_1$$

rearrange to form

$$\mathbf{S}\mathbf{r}_1 = \lambda_1 \mathbf{r}_1.$$

Which is known as an *eigenvalue* problem.

- Further directions can also be shown to be eigenvectors of the covariance.

Linear Dimensionality Reduction

Linear Latent Variable Model

- Represent data, $\mathbf{Y}$, with a lower dimensional set of latent variables $\mathbf{X}$.

- Assume a linear relationship of the form

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:},$$

where

$$\boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}\right).$$

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
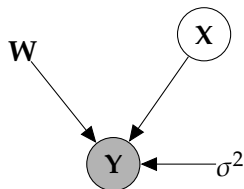
$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2 \mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Standard** Latent variable approach:
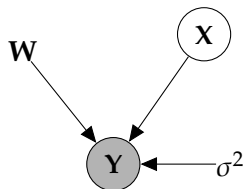
$$p\left(\mathbf{Y}|\mathbf{X}, \mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2 \mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.

- **Standard** Latent variable approach:

    - Define Gaussian prior over *latent space*, **X**.
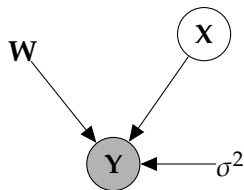
$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2 \mathbf{I}\right)$$

$$p(\mathbf{X}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{x}_{i,:}|\mathbf{0}, \mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

▸ Define *linear-Gaussian relationship* between latent variables and data.

▸ **Standard** Latent variable approach:

    ▸ Define Gaussian prior over *latent space*, $\mathbf{X}$.

    ▸ Integrate out *latent variables*.
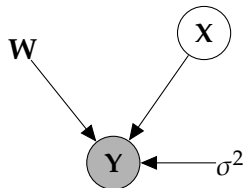
$$p\left(\mathbf{Y}|\mathbf{X}, \mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2 \mathbf{I}\right)$$

$$p\left(\mathbf{X}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{x}_{i,:}|\mathbf{0}, \mathbf{I}\right)$$

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{W}\mathbf{W}^{\top} + \sigma^2 \mathbf{I}\right)$$

Computation of the Marginal Likelihood

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:}, \quad \mathbf{x}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right), \quad \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$$

Computation of the Marginal Likelihood

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:}, \quad \mathbf{x}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right), \quad \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$$

$$\mathbf{W}\mathbf{x}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{W}\mathbf{W}^\top\right),$$

Computation of the Marginal Likelihood

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:}, \quad \mathbf{x}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right), \quad \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$$

$$\mathbf{W}\mathbf{x}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{W}\mathbf{W}^\top\right),$$

$$\mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)
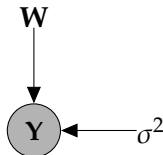
$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\top} + \sigma^2 \mathbf{I}$$

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0},\mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\top} + \sigma^{2}\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{W}\right) = -\frac{n}{2}\log|\mathbf{C}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\top}\mathbf{Y}\right) + \text{const.}$$

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\top} + \sigma^2 \mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{W}\right) = -\frac{n}{2}\log|\mathbf{C}| - \frac{1}{2}\text{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\top}\mathbf{Y}\right) + \text{const.}$$

If $\mathbf{U}_q$ are first q principal eigenvectors of $n^{-1}\mathbf{Y}^{\top}\mathbf{Y}$ and the corresponding eigenvalues are $\mathbf{\Lambda}_q$,

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\top} + \sigma^2 \mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{W}\right) = -\frac{n}{2}\log|\mathbf{C}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\top}\mathbf{Y}\right) + \mathrm{const.}$$

If $\mathbf{U}_q$ are first q principal eigenvectors of $n^{-1}\mathbf{Y}^{\top}\mathbf{Y}$ and the corresponding eigenvalues are $\mathbf{\Lambda}_q$,

$$\mathbf{W} = \mathbf{U}_q \mathbf{L} \mathbf{R}^{\top}, \quad \mathbf{L} = \left(\mathbf{\Lambda}_q - \sigma^2 \mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{R}$ is an arbitrary rotation matrix.

Reading

- Chapter 7 of Rogers and Girolami up to pg 249.

References I

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6): 417–441, 1933.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14, Cambridge, MA, 2002. MIT Press.

S. Rogers and M. Girolami. *A First Course in Machine Learning*. CRC Press, 2011. [Google Books] .

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8): 888–905, 2000.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, B*, 6(3):611–622, 1999. [PDF]. [DOI].