Dimensionality Reduction

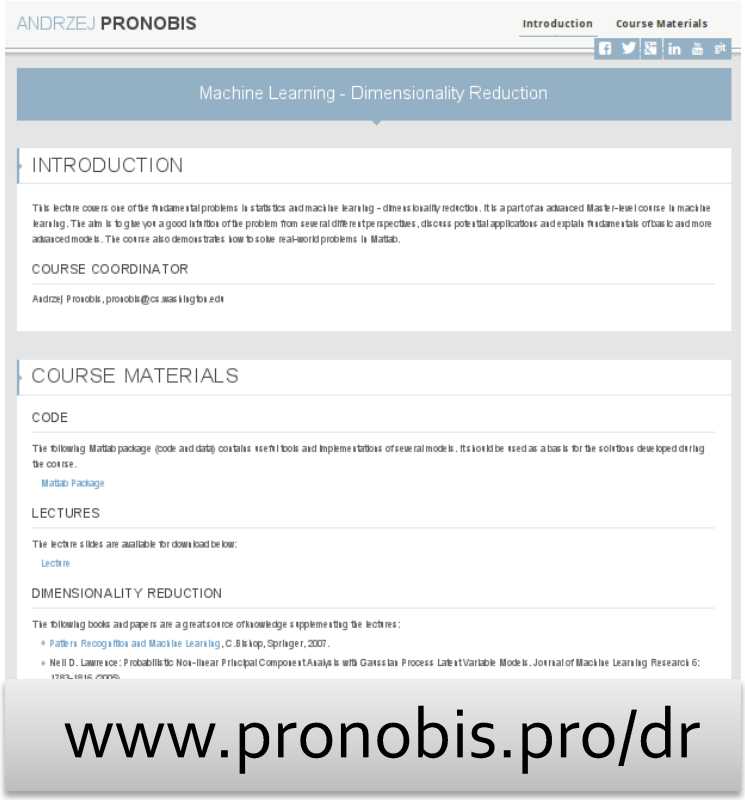
KTH 2013
Andrzej Pronobis

Goals of Lecture

- Advanced Master-level course in ML
- Give intuition of ***Dimensionality Reduction*** from several perspectives
  - Low-dimensional embedded space estimation
  - Unsupervised learning of continuous latent variable models
- Explain fundamentals of
  - Basic models (PCA)
  - Probabilistic models (PPCA)
  - Non-linear models (GP-LVM)
- Demonstration of real-world problems

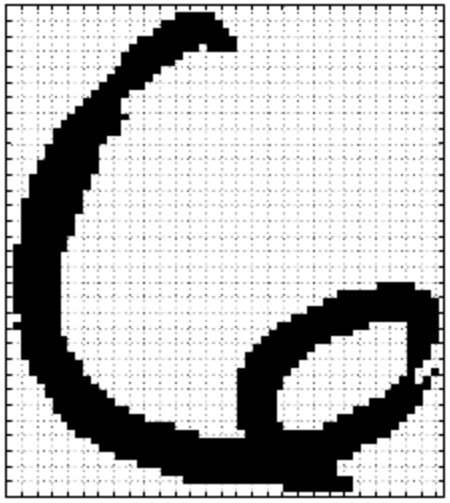
Requirements and Materials

- Materials available online
  - Code and demos
  - Lecture notes
  - Books and papers
- Requirements
  - Mathematical analysis
  - Linear algebra
  - Statistical modeling
  - Basic concepts in machine learning
  - Scientific Programming (EL2310)

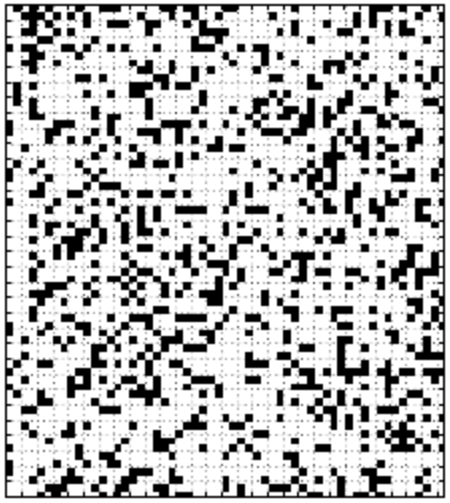


www.pronobis.pro/dr

Why dimensionality reduction?



Grid 64x57
= 3648 dimensions

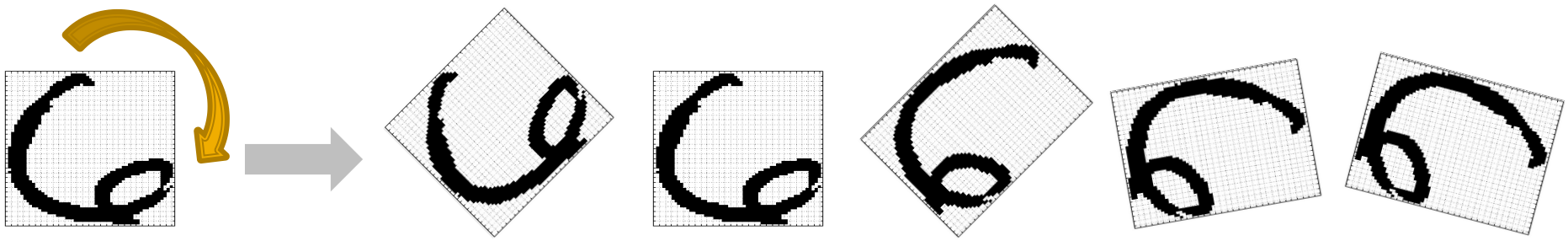


2^{3648} possibilities
Very large space to model!

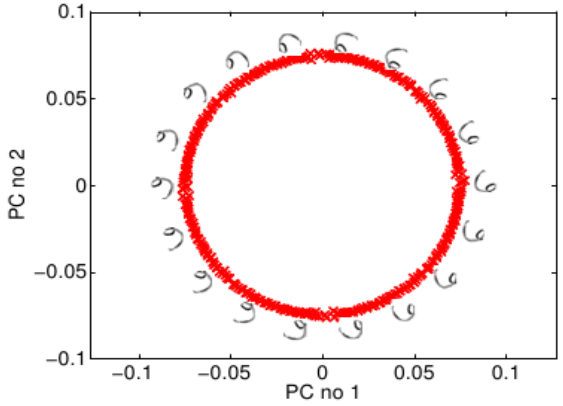
Examples from N. Lawrence '06

Why dimensionality reduction?

- Digits undergo a set of transformations e.g. rotation



- Transformations occupy only a small portion of space
- Structured Data typically lives on low dimensional manifolds



- Goal: find that manifold and model data there

Examples from N. Lawrence '06

Outline

- **Linear** Mapping to Embedded Space
  - Principal Component Analysis
  - Applications
  - Probabilistic PCA
- **Non-linear** Dimensionality Reduction
  - Gaussian Process Latent Variable Models
- Summary
  - New Promising Solutions

Outline

- **Linear Mapping to Embedded Space**
  - Principal Component Analysis
  - Applications
  - Probabilistic PCA
- **Non-linear** Dimensionality Reduction
  - Gaussian Process Latent Variable Models
- Summary
  - New Promising Solutions

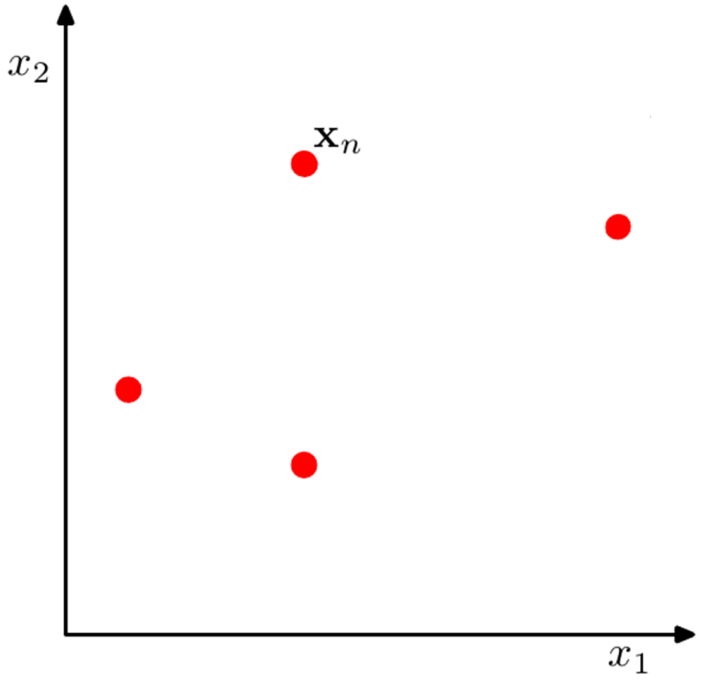
Input Space

- N input data points in D-dimensional input space

$$\mathbf{x}_n \in \Re^{D \times 1}$$ Single point

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix} \in \Re^{N \times D}$$

All data points



- Covariance Matrix

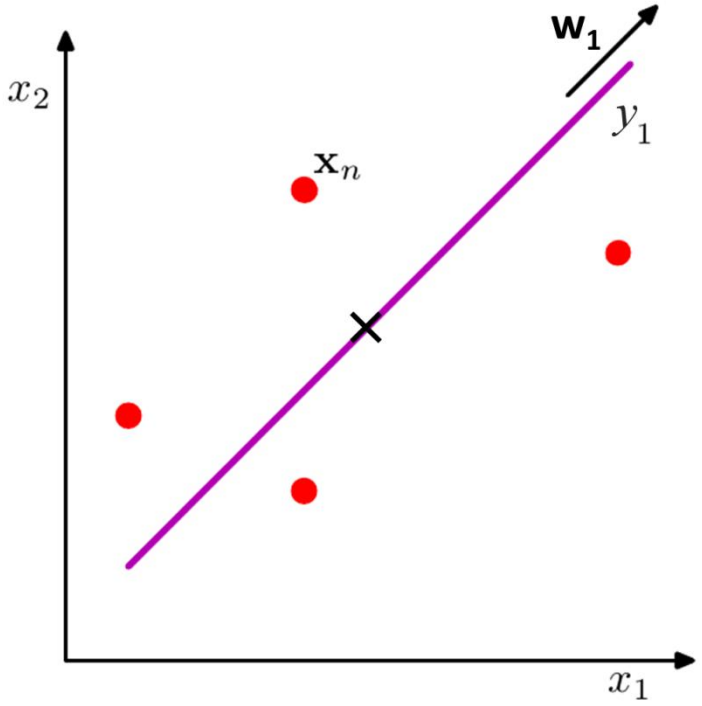
$$\mathbf{S} = \frac{1}{N}\mathbf{X}^T\mathbf{X} \in \Re^{D \times D}$$

Embedded Space

- Q-dimensional embedded (latent) space

$$\mathbf{y}_n \in \Re^{Q \times 1}$$ Single point

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_n^T \\ \vdots \\ \mathbf{y}_N^T \end{bmatrix} \in \Re^{N \times Q}$$ All data points



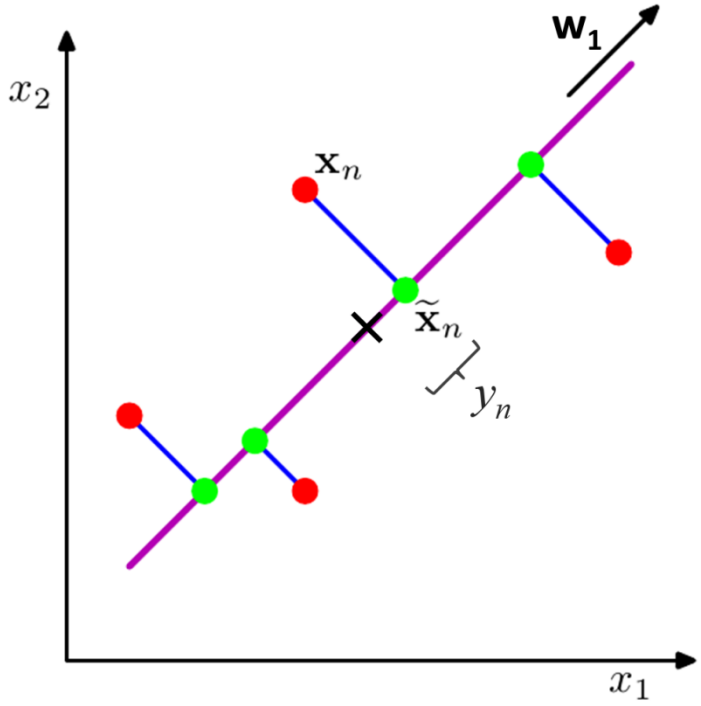
- Typically $Q < D$

Linear Mapping

- Matrix defines linear mapping between spaces

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_Q \end{bmatrix} \in \Re^{D \times Q}$$

$$\mathbf{w}_q \in \Re^{D \times 1}$$



- Project to embedded space

$$y_{n,q} = \mathbf{x}_n^T \mathbf{w}_q \qquad \mathbf{Y} = \mathbf{X}\mathbf{W}$$

- Re-create from embedded space representation

$$\widetilde{\mathbf{x}}_n = \sum_{q=1}^{Q} y_{n,q}\mathbf{w}_q = \mathbf{W}\mathbf{y}_n \qquad \widetilde{\mathbf{X}} = \mathbf{Y}\mathbf{W}^T$$

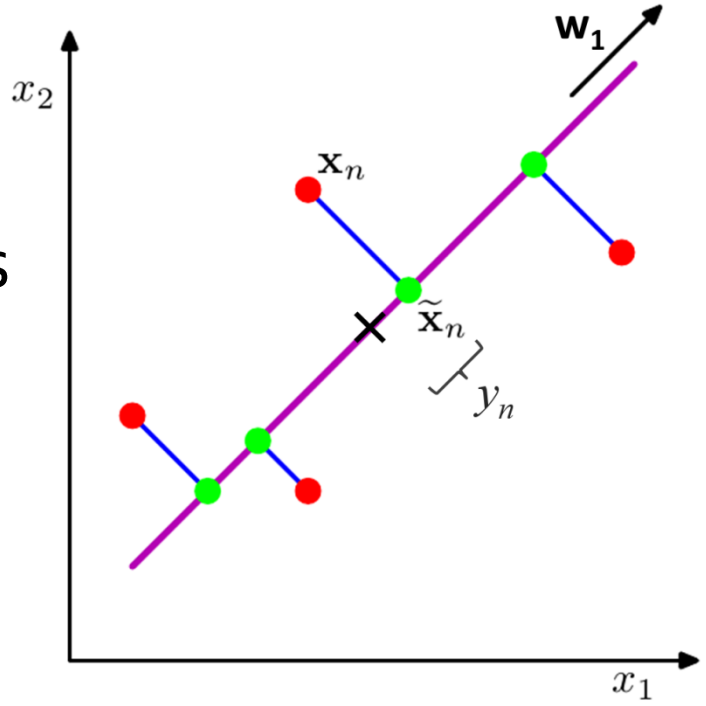
- $\boldsymbol{w}_q$ are basis vectors of embedded space (orthonormal)

Outline

- **Linear** Mapping to Embedded Space
  - **Principal Component Analysis**
  - Applications
  - Probabilistic PCA
- **Non-linear** Dimensionality Reduction
  - Gaussian Process Latent Variable Models
- Summary
  - New Promising Solutions

Principal Component Analysis

- Find a lower-dimensional sub-space
- Project data to that sub-space
- Two equivalent PCA formulations
  - Retain as much data variance as possible
  - Minimize projection error
- One common solution
  - Derivation in the lecture notes



Principal Component Analysis

- PCA solution

$$S\mathbf{w}_i = \lambda_i \mathbf{w}_i$$

- This makes $\mathbf{w}_i$ an eigenvector of $\mathbf{S}$ and λ_i the corresponding eigenvalue
- Projection error

$$\sum_{i=Q+1}^{D} \lambda_i$$

Eigenvalues for rejected eigenvectors

- Minimized by choosing $\mathbf{Q}$ eigenvectors with largest eigenvalues

Outline

- **Linear** Mapping to Embedded Space
  - **Principal Component Analysis**
  - **Applications**
  - Probabilistic PCA
- **Non-linear** Dimensionality Reduction
  - Gaussian Process Latent Variable Models
- Summary
  - New Promising Solutions

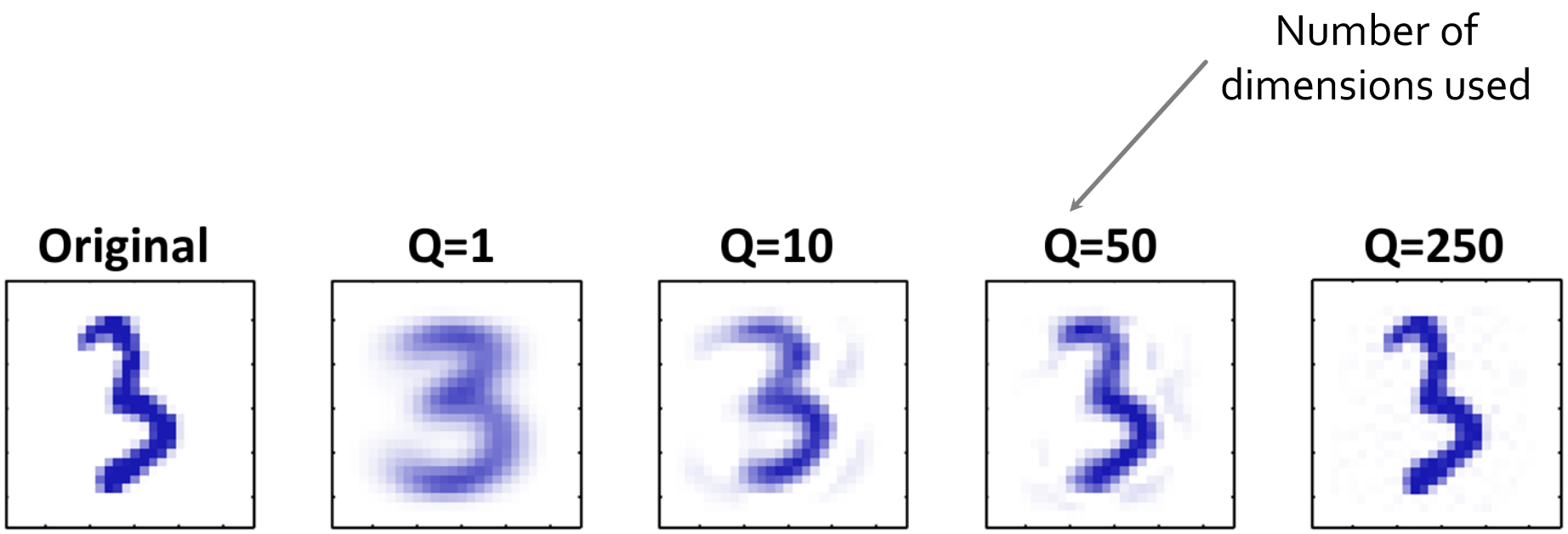
Feature Extraction

- Face Recognition using "Eigenfaces"

Retained eigenvectors

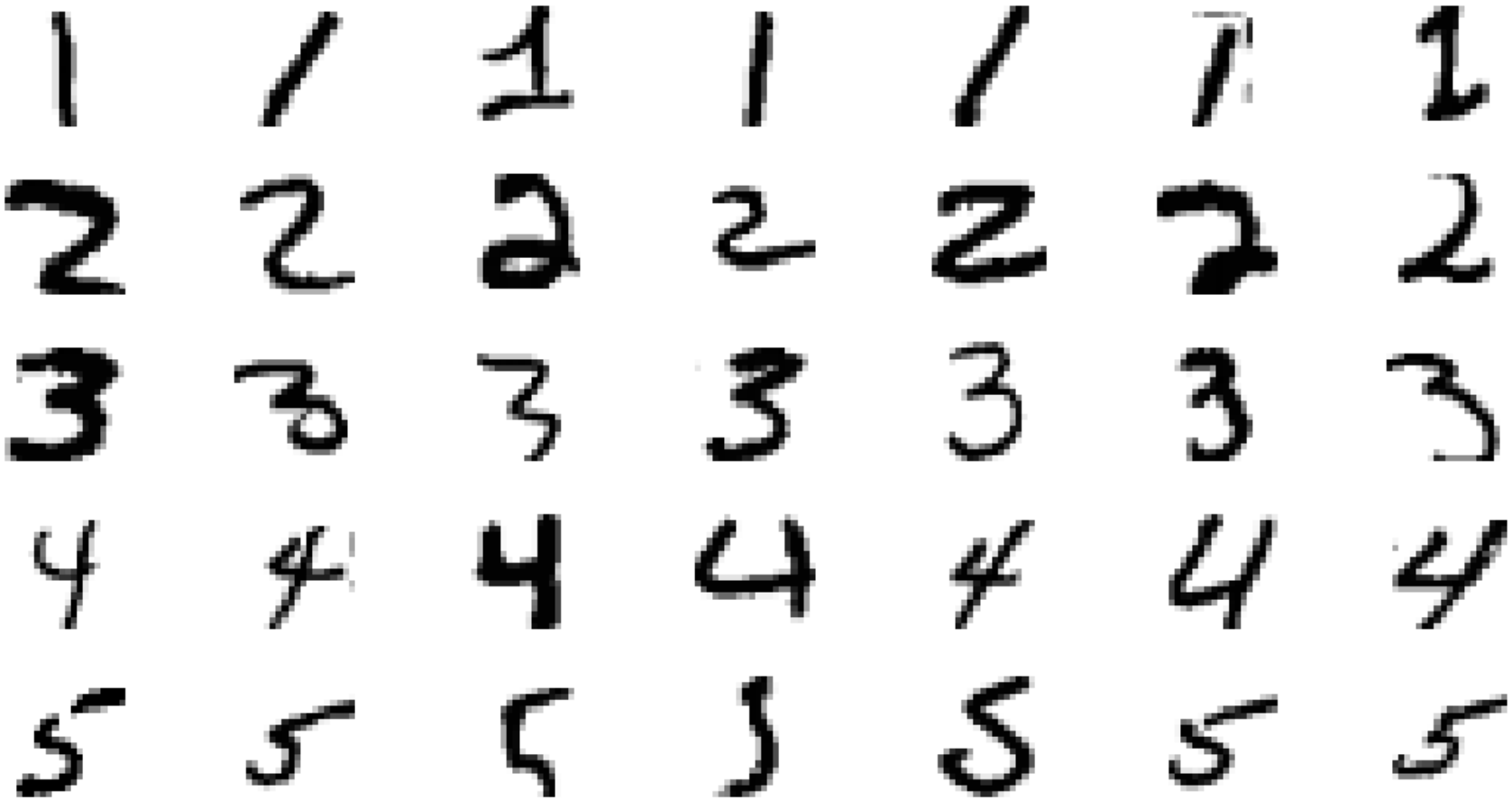
- Linear combination of eigenvectors

Compression



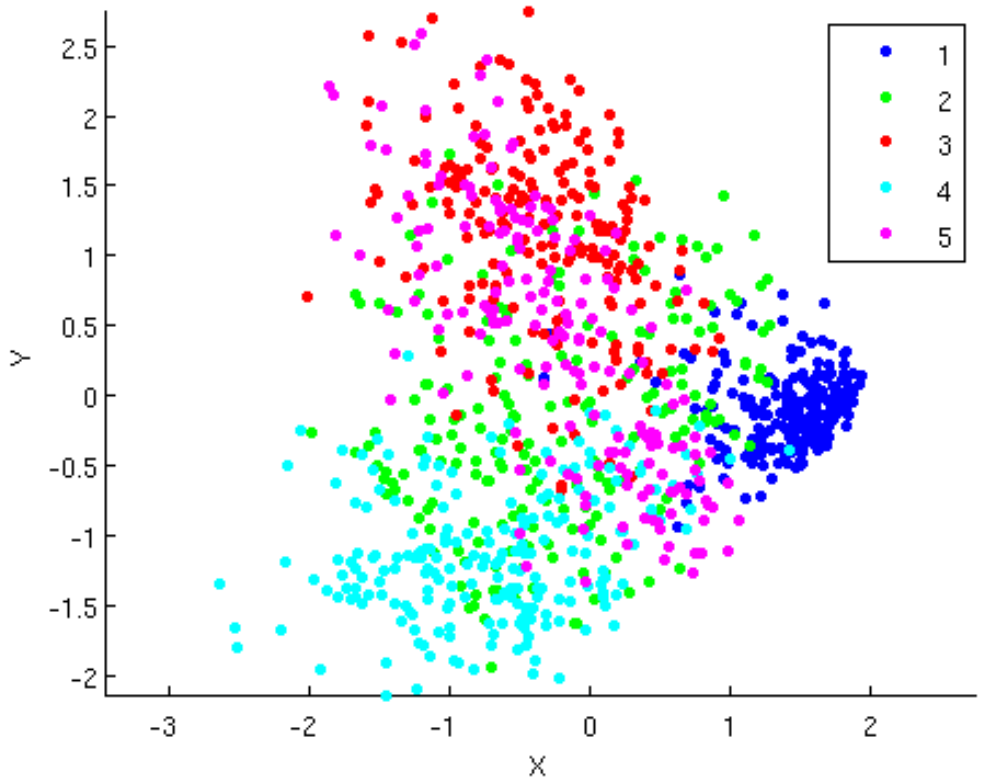
High-dimensional Data Visualization

- U.S. Postal Service dataset of handwritten digits



High-dimensional Data Visualization

- Handwritten digits - two first principle components



DEMO
in Matlab

- Can we find a non-linear manifold?

Outline

- **Linear** Mapping to Embedded Space
  - **Principal Component Analysis**
  - **Applications**
  - **Probabilistic PCA**
- **Non-linear** Dimensionality Reduction
  - Gaussian Process Latent Variable Models
- Summary
  - New Promising Solutions

Continuous Latent Variable Models

- Dimensionality reduction = **unsupervised learning** of **continuous latent variable models**
  - Embedded space becomes latent space
- We use *Maximum Likelihood (ML)* to learn model

Continuous Latent Variable Models

- Dimensionality reduction = **unsupervised learning** of **continuous latent variable models**
  - Embedded space becomes latent space
- We use *Maximum Likelihood (ML)* to learn model
- Recap *ML*:

Marginal likelihood over latent variables

$$p(\mathbf{x}_n|\mathbf{\Theta}) = \int p(\mathbf{x}_n|\mathbf{y}_n, \mathbf{\Theta})p(\mathbf{y}_n)d\mathbf{y}_n$$

Continuous Latent Variable Models

- Dimensionality reduction = **unsupervised learning** of **continuous latent variable models**
  - Embedded space becomes latent space
- We use *Maximum Likelihood (ML)* to learn model
- Recap *ML*:

Latent variable — Model Parameters

Marginal likelihood over latent variables

$$p(\mathbf{x}_n|\mathbf{\Theta}) = \int p(\mathbf{x}_n|\mathbf{y}_n, \mathbf{\Theta})p(\mathbf{y}_n)d\mathbf{y}_n$$

Continuous Latent Variable Models

- Dimensionality reduction = **unsupervised learning** of **continuous latent variable models**
  - Embedded space becomes latent space
- We use *Maximum Likelihood (ML)* to learn model
- Recap *ML*:

Marginal likelihood over latent variables

Latent variable

Model Parameters

$$p(\mathbf{x}_n|\mathbf{\Theta}) = \int \underbrace{p(\mathbf{x}_n|\mathbf{y}_n, \mathbf{\Theta})}_{\text{Likelihood}} p(\mathbf{y}_n) d\mathbf{y}_n$$

Continuous Latent Variable Models

- Dimensionality reduction = **unsupervised learning** of **continuous latent variable models**
  - Embedded space becomes latent space
- We use *Maximum Likelihood (ML)* to learn model
- Recap *ML*:

Marginal likelihood over latent variables

Latent variable

Model Parameters

$$p(\mathbf{x}_n|\mathbf{\Theta}) = \int \underbrace{p(\mathbf{x}_n|\mathbf{y}_n, \mathbf{\Theta})}_{\text{Likelihood}} \underbrace{p(\mathbf{y}_n)}_{\text{Prior over latent variables}} d\mathbf{y}_n$$

Continuous Latent Variable Models

- Dimensionality reduction = **unsupervised learning** of **continuous latent variable models**
  - Embedded space becomes latent space
- We use *Maximum Likelihood (ML)* to learn model
- Recap *ML*:

Marginal likelihood over latent variables

$$p(\mathbf{x}_n|\boldsymbol{\Theta}) = \int \underbrace{p(\mathbf{x}_n|\mathbf{y}_n, \boldsymbol{\Theta})}_{\text{Likelihood}} \underbrace{p(\mathbf{y}_n)}_{\text{Prior over latent variables}} d\mathbf{y}_n$$

Latent variable

Model Parameters

Maximum likelihood estimate of parameters

$$\underset{\boldsymbol{\Theta}}{\operatorname{argmax}} \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\boldsymbol{\Theta})$$

Maximum Likelihood for PCA

- Previously, we defined the reconstruction as

$$\widetilde{\mathbf{x}}_n = \mathbf{W}\mathbf{y}_n \qquad \mathbf{x}_n = \widetilde{\mathbf{x}}_n + \epsilon_n$$

- We can express this linear relationship using noise (to account for reconstruction error)

$$\mathbf{x}_n = \mathbf{W}\mathbf{y}_n + \boldsymbol{\eta}_n$$

where the noise $\boldsymbol{\eta}_n$ is $p(\boldsymbol{\eta}_n) = \mathcal{N}(\boldsymbol{\eta}_n|\mathbf{0}, \sigma^2\mathbf{I})$

- Therefore, the <u>likelihood</u> of input data becomes

$$p(\mathbf{X}|\mathbf{Y}, \mathbf{W}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\mathbf{W}\mathbf{y}_n, \sigma^2\mathbf{I})$$

We use independence of data points

Maximum Likelihood for PCA

- For ML estimation, we need marginal likelihood

$$p(\mathbf{X}|\mathbf{W}) = \int p(\mathbf{X}|\mathbf{Y}, \mathbf{W}) p(\mathbf{Y}) d\mathbf{Y}$$

- We need prior over latent variables
- For PCA, we assume

$$p(\mathbf{Y}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{y}_n|\mathbf{0}, \mathbf{I})$$

- Marginal likelihood can be found analytically

$$p(\mathbf{X}|\mathbf{W}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\mathbf{0}, \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I})$$

Probabilistic PCA

$$p(\mathbf{X}|\mathbf{W}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\mathbf{0}, \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I})$$

- We maximize it to find the parameters $\mathbf{W}$
- The result is found analytically

$$\mathbf{W} = \mathbf{U}_Q\mathbf{L}\mathbf{V}^T \qquad \mathbf{L} = (\mathbf{\Lambda}_Q - \sigma^2\mathbf{I})^{\frac{1}{2}}$$

Q eigenvectors with largest eigenvalues

Arbitrary rotation matrix

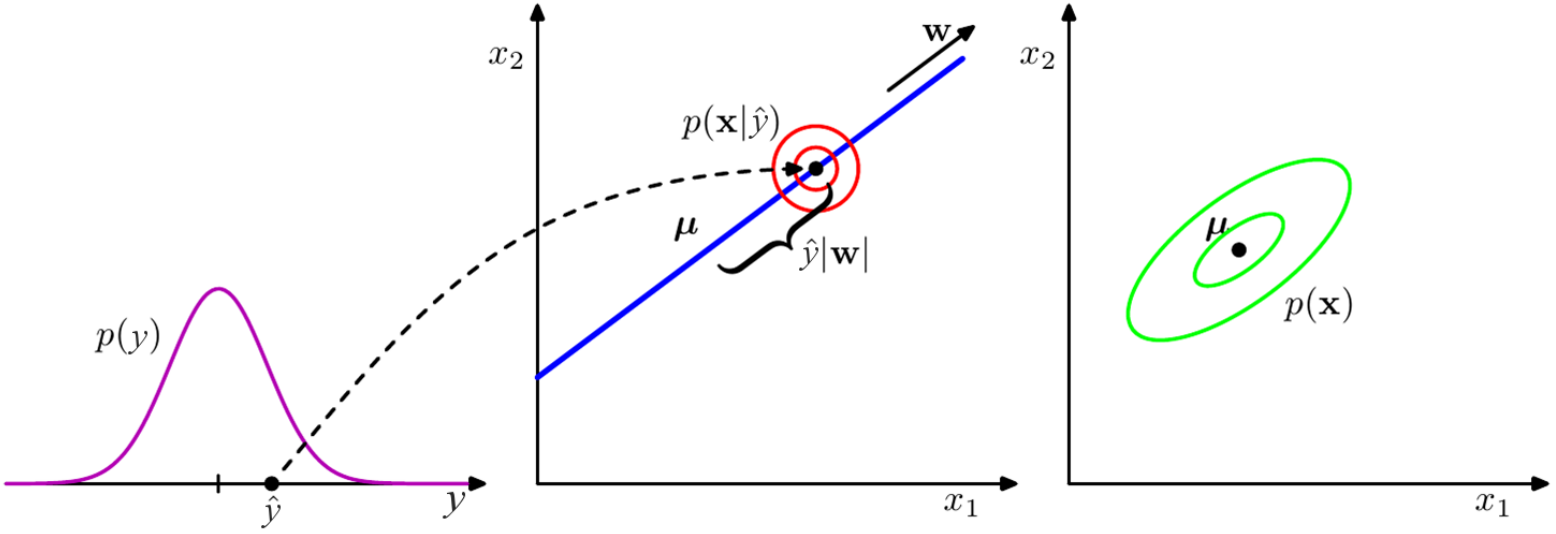
Corresponding eigenvalues

[Tipping and Bishop '99]

- This corresponds to PCA

Probabilistic PCA

- Benefits
  - Expectation-maximization algorithms for large datasets
  - Basis for Bayesian treatment (discover Q)
  - PPCA can be run generatively



Probabilistic PCA dual formulation

- PPCA
  - Marginalize latent variables Y
  - Optimize W
- Dual PPCA
  - Marginalize parameters W
  - Optimize Y

Final marginal likelihood

$$p(\mathbf{X}|\mathbf{W}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\mathbf{0}, \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I})$$

$$p(\mathbf{X}|\mathbf{Y}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{x}_n|\mathbf{0}, \mathbf{Y}\mathbf{Y}^T + \sigma^2\mathbf{I})$$

Leads to equivalent solutions
but with different formulations

Outline

- **Linear** Mapping to Embedded Space
  - Principal Component Analysis
  - Applications
  - Probabilistic PCA
- **Non-linear Dimensionality Reduction**
  - **Gaussian Process Latent Variable Models**
- Summary
  - New Promising Solutions

Gaussian Processes

- GP $p(f)$ defines a distribution over functions

$$f : \Re^Q \rightarrow \Re$$

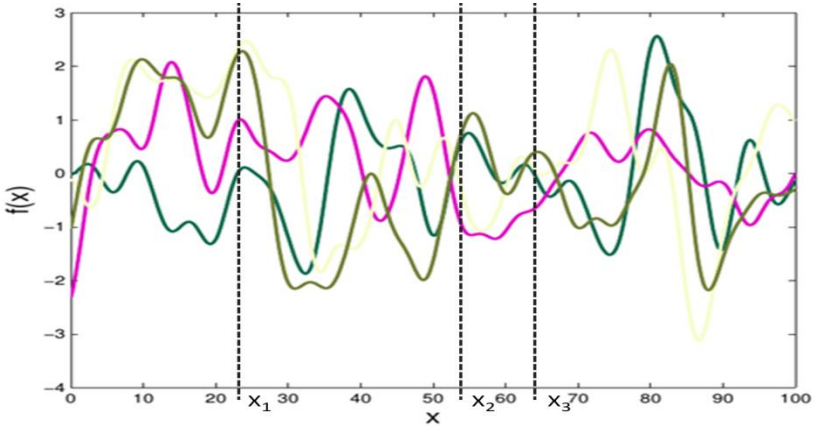
- We can sample that function for a subset

$$\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\} \subset \Re^Q$$

- The marginal distribution for any subset

$$p(f(\mathbf{y}_1), f(\mathbf{y}_2), \ldots, f(\mathbf{y}_N))$$

is a multivariate Gaussian distribution



Gaussian Processes

- GPs are parameterized by mean **function** and covariance **function** $k(\mathbf{y}_i, \mathbf{y}_j)$
- Often mean is assumed to be 0, only covariance function need

$$p(f(\mathbf{y}_1), f(\mathbf{y}_2), \ldots, f(\mathbf{y}_N)) = \mathcal{N}(\mathbf{0}, \mathbf{K})$$

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{y}_1, \mathbf{y}_1) & \cdots & k(\mathbf{y}_1, \mathbf{y}_N)) \\ & \ddots & \\ \ldots & & \ldots \\ k(\mathbf{y}_N, \mathbf{y}_1) & \cdots & k(\mathbf{y}_N, \mathbf{y}_N)) \end{bmatrix}$$

GP Latent Variable Models

- Consider a GP modeling functions that are
  - linear
  - corrupted by Gaussian noise of variance $\sigma^2\mathbf{I}$

  then, covariance function

$$k(\mathbf{y}_i, \mathbf{y}_j) = \mathbf{y}_i^T \mathbf{y}_j + \sigma^2 \delta_{ij}$$

Kronecker Delta

- If we calculate *k* for every point in the embedded (low dimensional) space

$$\mathbf{K} = \mathbf{Y}\mathbf{Y}^T + \sigma^2\mathbf{I}$$

➡ This is covariance for marginal likelihood of Dual PPCA

$$p(\mathbf{X}|\mathbf{Y}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{x}_n|\mathbf{0}, \mathbf{Y}\mathbf{Y}^T + \sigma^2\mathbf{I})$$

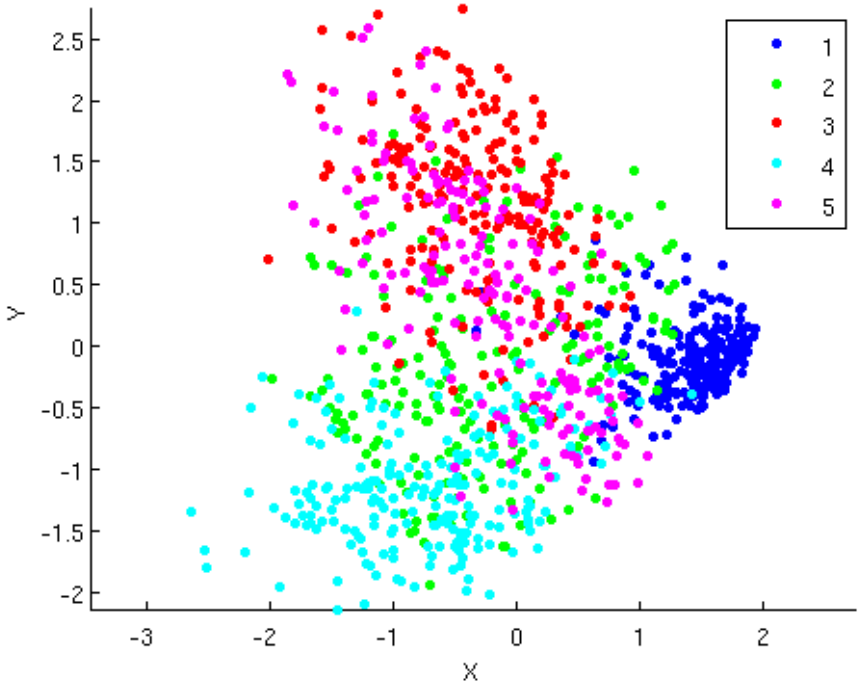
GP Latent Variable Models

- The marginal likelihood of dual PPCA can be seen as a product of Gaussian Processes
  - For a linear GP covariance function, we get PCA
  - But, we can now **use non-linear covariance functions** to obtain **non-linear low-dimensional manifolds**

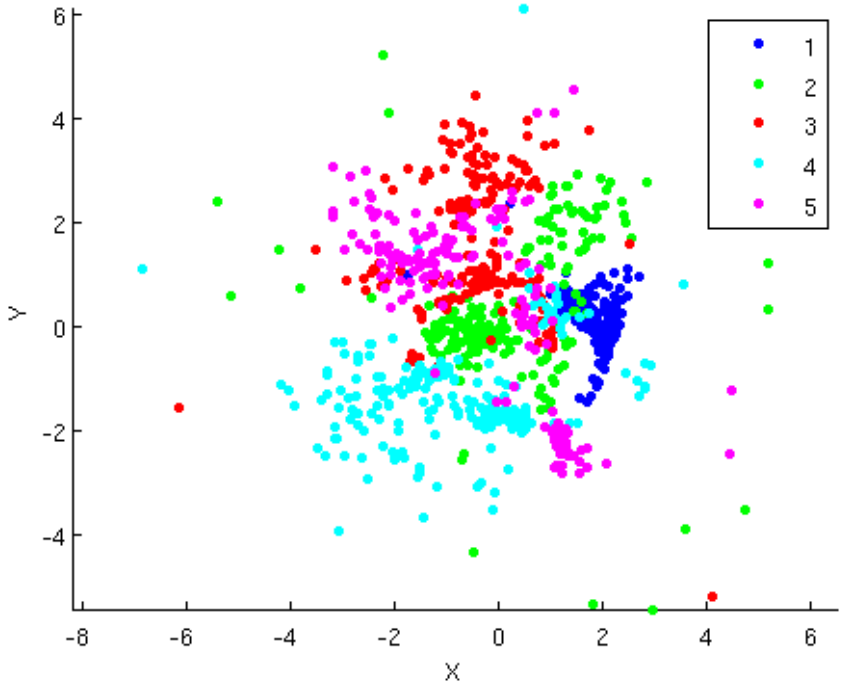
$$k(\mathbf{x}_i, \mathbf{x}_j) = \theta_{\text{rbf}} \exp\left(-\frac{\gamma}{2}(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j)\right) + \theta_{\text{bias}} + \theta_{\text{white}}\delta_{ij}$$

Comparison of PCA and GPLVM

PCA



GPLVM



DEMO in Matlab

Outline

- **Linear** Mapping to Embedded Space
  - Principal Component Analysis
  - Applications
  - Probabilistic PCA
- **Non-linear** Dimensionality Reduction
  - Gaussian Process Latent Variable Models
- **Summary**
  - **New Promising Solutions**

Summary

- Dimensionality reduction as:
  - Estimation of embedded low-dimensional space
  - Unsupervised learning of continuous latent-variable models
- Different approaches
  - Basic models (PCA)
  - Probabilistic models (PPCA)
  - Non-linear models (GP-LVM)
- Try this code at home!

New Promising Solutions

- Deep Learning
  - Convolutional Neural Networks
    - Instead of traditional 2-step process
    - Integrate
      - Regularization (dimensionality reduction)
      - Lease squares objective (supervised classification)
  - Hierarchical Sparse Coding
    - Dimensionality explosion
    - Simple linear classification problem
    - Works great for large Internet databases
  - Many state-of-the-art results

Questions?