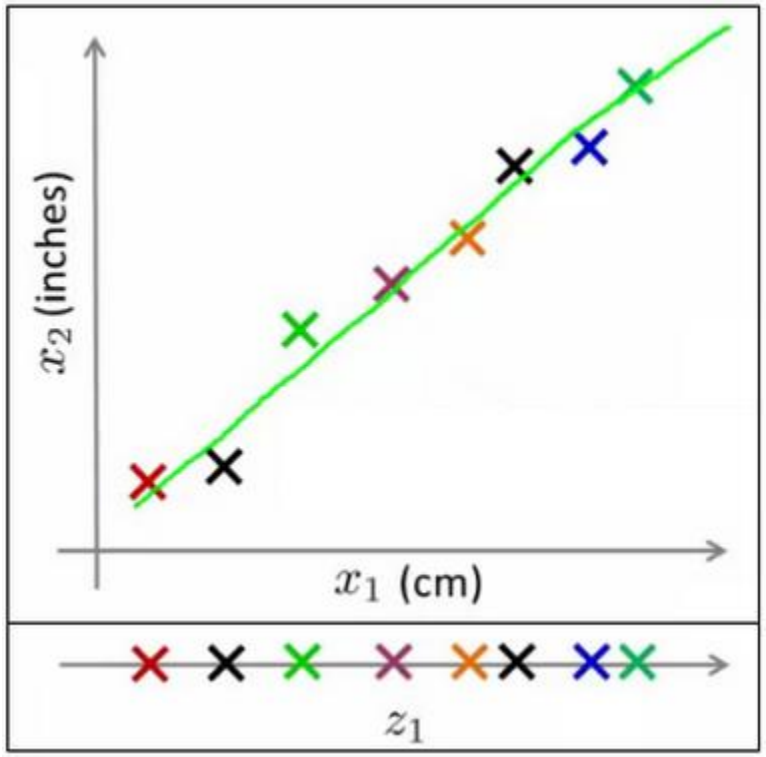
Dimensionality Reduction

Wenkai Mo

Background



Situations:

(1) Image Process: Large Image Windows.
(2) Text Process: Large Vocabulary.

Dimensionality Reduction:

A statistically optimal way of dimensionality reduction **is to project the data onto a lower-dimensional subspace that captures as much of variation of data of possible.**

Advantage:

1. It reduces the time and storage space required.
2. Removal of multi-collinearity improves the performance of the machine learning model.
3. Avoid overfitting.
4. It becomes easier to visualize the data when reduced to very low dimensions such as 2D or 3D.

Two Categories

- Feature Selection: Find a subset of the original variables.

$$Y = P^T X, P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- Feature Extraction: Transform the data in the high-dimensional space to a space of fewer dimensions. (Latent Factors)

$$Y = P^T X, P = \begin{bmatrix} 0.2 & 0.1 & 0.5 \\ 0.8 & 0.7 & 0.2 \\ 0 & 0.2 & 0.3 \end{bmatrix}$$

Two Categories

- Feature Selection: Find a subset of the original variables.
  - *Filter*: e.g. information gain;
  - *Wrapper:* e.g. search guided by the accuracy;
  - *Embedded:* Features are selected to add or be removed while building the model based on the prediction errors. (Bagging, Boosting, L1-Norm Least Square);

- Feature Extraction: Transform the data in the high-dimensional space to a space of fewer dimensions.

Filter

Easy but intuitive.

Feature Selection (Filter)

- **Missing Values Ratio**
  - Remove those dimensions whose missing values ratios are above a threshold.
- **Low Variance Filter**
  - If data in one dimension has a low variance, that dimension contains less information.
  - Remove those dimensions whose variance values are lower than a threshold.
- **High Correlation Filter**
  - If two dimensions are in high correlation, then remain one of them.
  - Numeric: Correlation Coefficient
  - Discrete: Pearson chi-square values
- **Other Criteria:** Information Gain, Mutual Information, etc.

Wrapper

Time-consuming, but effective in low dimension.

Feature Selection (Wrapper)

- **Backward Feature Elimination**
  - Remove a feature randomly, and compare the performance of models to decide to whether this feature should be removed.

- **Forward Feature Construction**
  - Select a feature randomly, and compare the performance of models to decide to whether this feature should be selected.

Embedded

Reflecting by Model Directly.

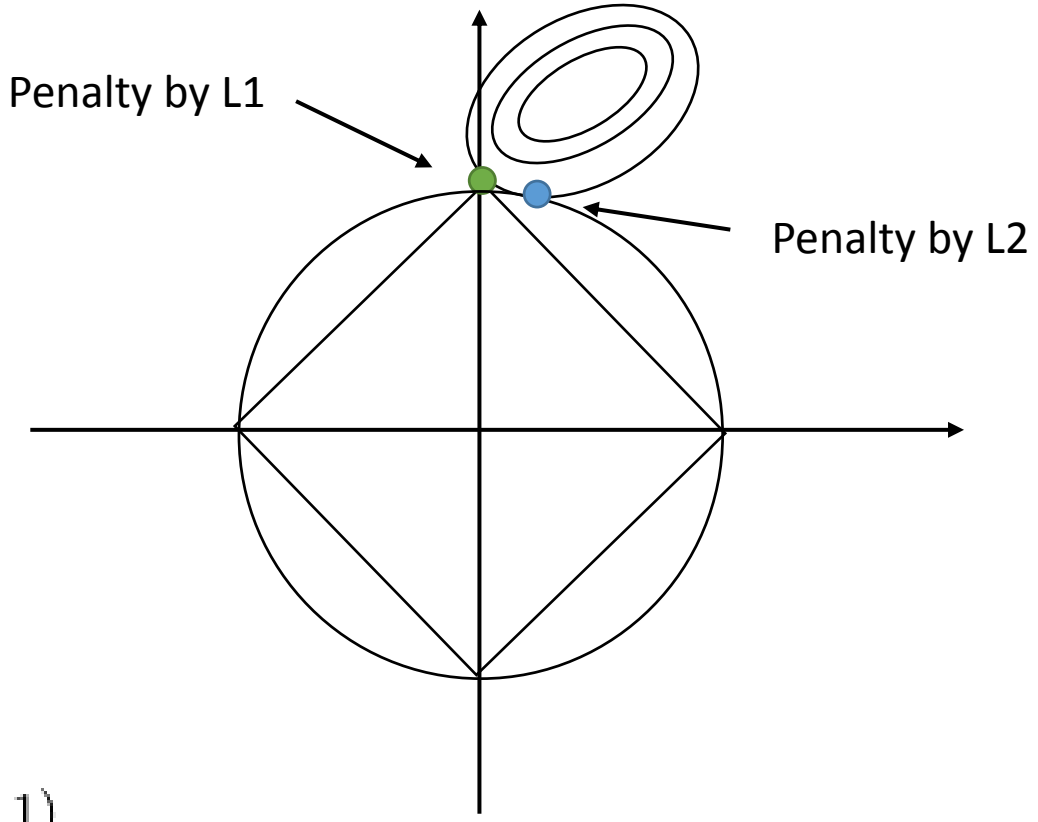
Feature Selection (Embedded)

- Bagging-Tree (Random Forest)
  - Learning:
    - 1. Sample, with replacement, n training examples from X, Y; call these X_b, Y_b.
    - 2. Train a decision or regression tree f_b on X_b, Y_b.
  - Predicting
    - Voting by all the trees.
  - Features which appear in most trees and in shallow levels are relatively important.

Feature Selection (Wrapper)

- Regression with L1-Norm (Lasso)
  - **Selecting non-zero coefficients**

$$\min_{w} \frac{1}{2n_{samples}} ||Xw - y||_2^2 + \alpha ||w||_1$$



- Logistic Regression
  - The relative values of coefficients (***w***).

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1)$$

Two Categories

- Feature Selection: Find a subset of the original variables.

- Feature Extraction: Transform the data in the high-dimensional space to a space of fewer dimensions.
  - Supervised
    - Fisher Discriminant Analysis (FDA) or Linear Discriminant Analysis (LDA).
  - Unsupervised
    - Principle Component Analysis (PCA)
    - Random Projection (RP, usually used in texts and images analysis)
    - Non-negative matrix factorization (NMF or NNMF)
    - Topic Model in Text Analysis (LSA, LDA, Another Tutorial)
  - Semi-Supervised
    - Deep Learning (Another Tutorial)

LDA

Linear Discriminant Analysis

Feature Extraction (Supervised)

- **Linear Discriminant Analysis**
  - Goal: Project a feature space onto a smaller subspace *k* (where *k* ≤ *n-1*), while maintaining the class-discriminatory information.

  - Fisher Principle:
  - $J(w) = \frac{(\widetilde{\mu_1} - \widetilde{\mu_2})}{(\widetilde{s_1}^2 + \widetilde{s_2}^2)}$, where $\widetilde{\mu_i}$ and $\widetilde{s_i}^2$ are the mean and variance data in class *i* in new space.
  - $\widetilde{\mu_i} = w^T \mu_i$ and $\widetilde{s_1}^2 = (w^T x - w^T \mu_i)^2$, where w^T is the projection.
  - Then it can be converted to a optimization problem:

$$\max_x J(w) = \frac{(\widetilde{\mu_1} - \widetilde{\mu_2})}{(\widetilde{s_1}^2 + \widetilde{s_2}^2)}$$

Feature Extraction (Supervised)

- Linear Discriminant Analysis
  - This method is suitable to the case where the number of classes *k* is very large, because LDA can only extraction at most *k-1*'s features. Such cases can be **image classification**, **hand-writing classification**, etc.
  - This method is based on Eigen Vector decomposition; therefore the performance may not be very well when data is too large.

Two Categories

- Feature Selection: Find a subset of the original variables.

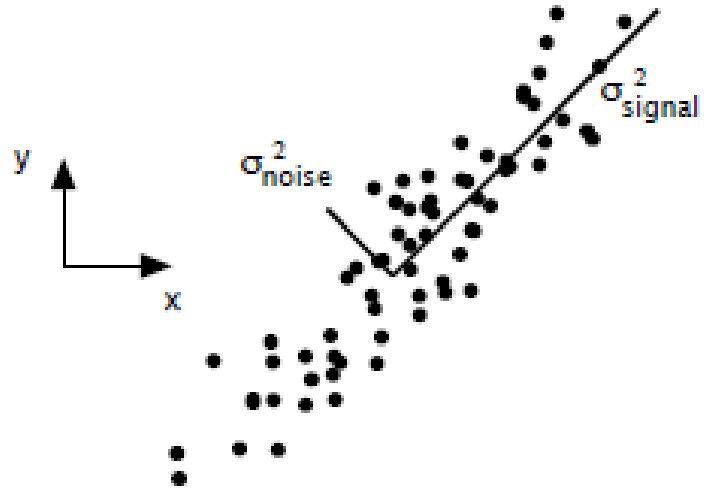
- Feature Extraction: Transform the data in the high-dimensional space to a space of fewer dimensions.
  - Supervised
    - **Fisher Discriminant Analysis (FDA) or Linear Discriminant Analysis (LDA).**
  - Unsupervised
    - **Principle Component Analysis (PCA)**
    - **Non-negative matrix factorization (NMF or NNMF)**
    - **Random Projection (RP, usually used in texts and images analysis)**
    - Topic Model in Text Analysis (LSA, LDA, Another Tutorial)
  - Semi-Supervised
    - Deep Learning (Another Tutorial)

PCA

Principle Component Analysis

Feature Extraction (Unsupervised)

- Principle Component Analysis (PCA)



Q: How to identify the most meaningful basis to re-express a data set?

A: Directions with largest variances in the measurement space.

Feature Extraction (Unsupervised)

- Consider ***X*** is a data matrix: $X = \begin{pmatrix} a_1 & a_2 & \cdots & a_m \\ b_1 & b_2 & \cdots & b_m \end{pmatrix}$

- Covariance matrix C_X: $\frac{1}{m} X X^\mathsf{T} = \begin{pmatrix} \frac{1}{m}\sum_{i=1}^{m} a_i^2 & \frac{1}{m}\sum_{i=1}^{m} a_i b_i \\ \frac{1}{m}\sum_{i=1}^{m} a_i b_i & \frac{1}{m}\sum_{i=1}^{m} b_i^2 \end{pmatrix}$

- Another constraint is that the covariance should be zero, because we do not want to select two redundant components.

Feature Extraction (Unsupervised)

- Our goal is to find a projection matrix ***P*** to project ***X*** to another space with largest variances. Suppose ***Y*** = ***PX*** is the target space, where the covariance matrix of ***Y*** is diagonal, and its variances are in ascending order.

- Now, let's see the relation between ***Y*** and ***X***

- Recall the eigenvalue decomposition of symmetric matrix. ***D*** is the eigenvalues of ***C*** and P is the orthogonal projection matrix.

$$
\begin{aligned}
D &= \frac{1}{m} Y Y^{\mathsf{T}} \\
&= \frac{1}{m} (PX)(PX)^{\mathsf{T}} \\
&= \frac{1}{m} P X X^{\mathsf{T}} P^{\mathsf{T}} \\
&= P(\frac{1}{m} X X^{\mathsf{T}}) P^{\mathsf{T}} \\
&= P C P^{\mathsf{T}}
\end{aligned}
$$

NMF

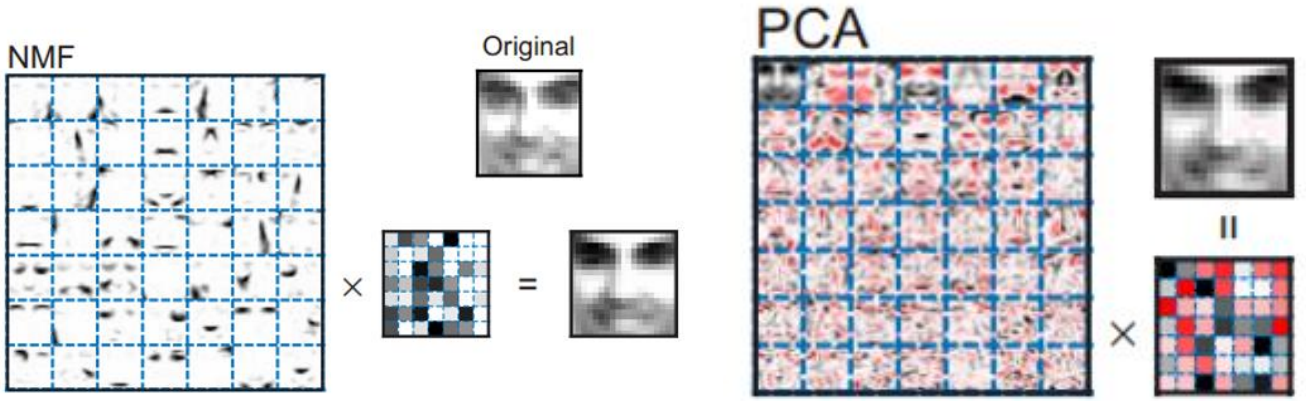
Non-negative matrix factorization

Feature Extraction (Unsupervised)

- **Non-negative matrix factorization (NMF or NNMF)**
  - Assumes that the data and the components are **non-negative.**
  - It finds a decomposition of samples ***X*** into two matrices ***V*** and ***H*** of non-negative elements, by optimizing for the squared Frobenius norm:

$$\arg\min_{W,H} ||X - WH||^2 = \sum_{i,j} X_{ij} - WH_{ij}$$

  - High interpretability and performance compared with PCA.



Feature Extraction (Unsupervised)

- **Non-negative matrix factorization (NMF or NNMF)**

	D1	D2	D3	D4
U1	5	3	-	1
U2	4	-	-	1
U3	1	1	-	5
U4	1	-	-	4
U5	-	1	5	4

	D1	D2	D3	D4
U1	4.97	2.98	2.18	0.98
U2	3.97	2.40	1.97	0.99
U3	1.02	0.93	5.32	4.93
U4	1.00	0.85	4.59	3.93
U5	1.36	1.07	4.89	4.12

Coping with missing data in Recommendation Systems.

RP

The Random Projections

Feature Extraction (Unsupervised)

- **Johnson-Lindenstrauss Theorem**

For any $0 < \varepsilon < 1$ and any positive integer n, let k be a positive integer such that

$$k \geq 4(\epsilon^2/2 - \epsilon^3/3)^{-1} \ln n$$

Then for any set V of n points in $\mathbf{R}^d$, there is a map $f : \mathbf{R}^d \to \mathbf{R}^k$ such that for all $u, v \in V$,

$$(1 - \epsilon)\|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \epsilon)\|u - v\|^2.$$

Furthermore, this map can be found in expected polynomial time.

Feature Extraction (Unsupervised)

- **The Random Projections**

Let A be a random $k \times d$ matrix that projects $\mathbf{R}^d$ onto a *uniform random* k-dimensional subspace.

Multiply A by a fixed scalar $\sqrt{\frac{d}{k}}$. For every $v \in \mathbf{R}^d$, v is mapped to $\sqrt{\frac{d}{k}}Av$.

- **Selection of the Random Matrix**
  - Sparse random projection

$$\begin{cases} -\sqrt{\frac{s}{n_{\text{components}}}} & 1/2s \\ 0 & \text{with probability} \quad 1 - 1/s \\ +\sqrt{\frac{s}{n_{\text{components}}}} & 1/2s \end{cases}$$

s is usually set to $\sqrt{\#\ of\ features}$

  - Gaussian random projection

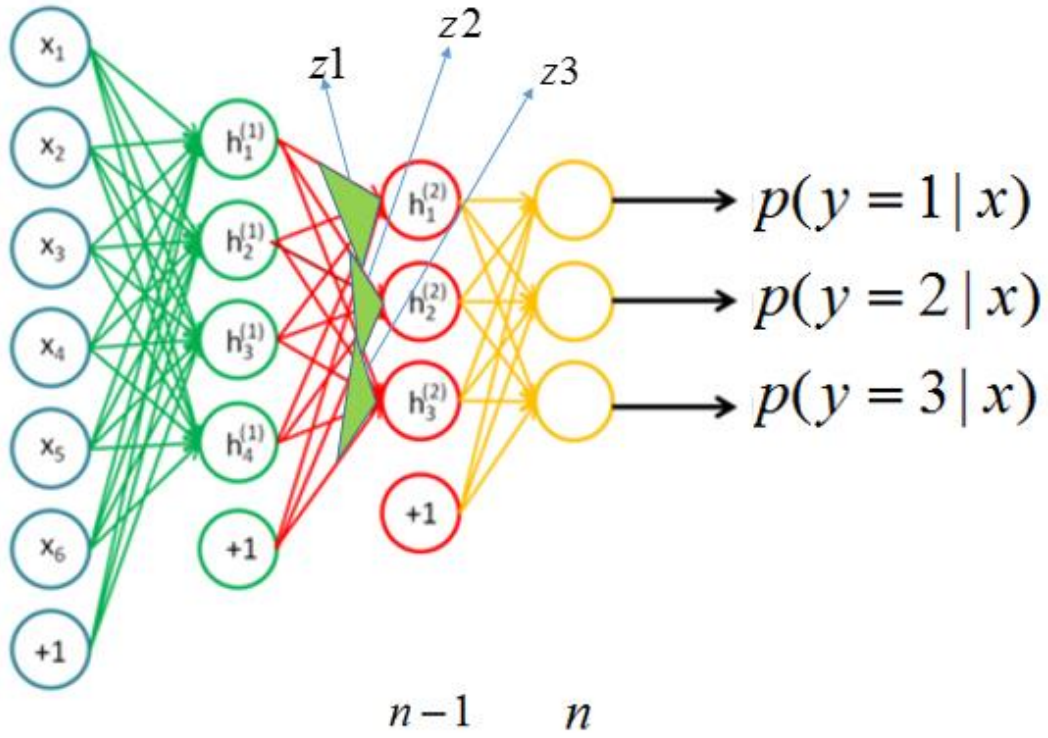
Advanced

- Some other methods:
  - Independent Component Analysis (ICA)
  - Factor Analysis (FA)

- Advanced:
  - Non-linear projection (kernel method).
  - Approximate algorithm for parallel computing (for big data).

Advanced

- Deep Learning (Feature Learning)



- Graph-based Topic Modeling