# Dimensionality Reduction
# 降维

雷森 [1] [2]

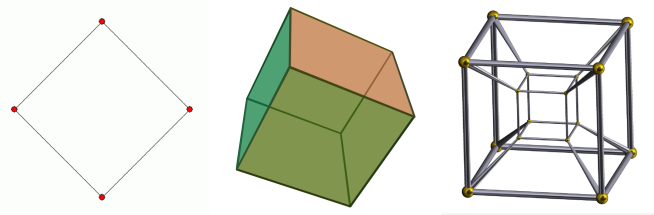
[1] UCSB
University of California, Santa Barbara
CA, U.S.

[2] Bayes Data Intelligence Technology Service Co., LTD
Xi'an, China

Sept, 2018

# Summary

1. Introduction
   - Dimension
   - The Curse of Dimensionality
   - Dimensionality Reduction
2. Feature Selection
   - Variance Thresholds
   - Correlation Thresholds
   - Genetic Algorithms (GA)
   - Stepwise Search
3. Feature Extraction
   - Principal Component Analysis (PCA)
   - Linear Discriminant Analysis (LDA)



(image source)

## Definition

Dimension (维度，aka 维数）：是数学中独立参数的数目。在物理学和哲学的领域内，指独立的时空坐标的数目。

在机器学习中，“维度”简单地指数据集的特征（也就是输入变量）的个数。

## Definition

Dimension (维度，aka 维数 )： 是数学中独立参数的数目。在物理学和哲学的领域内，指独立的时空坐标的数目。

在机器学习中，“ 维度 ” 简单地指数据集的特征（也就是输入变量）的个数。

# “ 维度诅咒 ”

curse of dimensionality (维度诅咒，aka 维数灾难 )： （最早由 Richard E. Bellman 在考虑优化问题时首次提出 ）用来描述当（数学）空间维度增加时，分析和组织高维空间（通常有成百上千维），因体积指数增加而遇到各种问题场景。

当数据集中的特征个数相比于观测值的个数非常多时，某些算法将很难有效地训练模型。这就是所谓的“ 维度诅咒 ”。

# “ 维度诅咒 ”

curse of dimensionality (维度诅咒，aka 维数灾难）：（最早由 Richard E. Bellman 在考虑优化问题时首次提出）用来描述当（数学）空间维度增加时，分析和组织高维空间（通常有成百上千维），因体积指数增加而遇到各种问题场景。

当数据集中的特征个数相比于观测值的个数非常多时，某些算法将很难有效地训练模型。这就是所谓的“ 维度诅咒 ”。

## 原因

当维数提高时，空间的体积提高太快，因而可用数据变得很稀疏。稀疏性对于任何要求有统计学意义的方法而言都是一个问题，为了获得在统计学上正确并且有可靠的结果，用来支撑这一结果所需要的数据量通常随着维数的提高而呈指数级增长。

# 类比 - 寻找硬币

在 $100$ 码的直线上丢一枚硬币，去寻找。 $\approx$ 两分钟

在 $100^2$ 码 $^2$ 的平面上丢一枚硬币，去寻找。
（相当于在两个并起来足球场那么大的地方找） $\approx$ 几天

在 $100^3$ 码 $^3$ 的空间内丢一枚硬币，去寻找。
（相当于在体育馆那么大的 30 层的楼里找） ☹

(From Quora)

# 类比 - 寻找硬币

在 100 码的直线上丢一枚硬币，去寻找。 ≈ 两分钟

在 $100^2$ 码 $^2$ 的平面上丢一枚硬币，去寻找。
(相当于在两个并起来足球场那么大的地方找) ≈ 几天

在 $100^3$ 码 $^3$ 的空间内丢一枚硬币，去寻找。
(相当于在体育馆那么大的 30 层的楼里找) ☹

(From Quora)

# 类比 - 寻找硬币

在 100 码的直线上丢一枚硬币，去寻找。 $\approx$ 两分钟

在 $100^2$ 码 $^2$ 的平面上丢一枚硬币，去寻找。
（相当于在两个并起来足球场那么大的地方找） $\approx$ 几天

在 $100^3$ 码 $^3$ 的空间内丢一枚硬币，去寻找。
（相当于在体育馆那么大的 30 层的楼里找） ☹

(From Quora)

# Dimensionality Reduction : Math

- $X \in \mathbb{R}^p$, where $p$ is the number of dimensions (possibly very large).

- Goal : Find a function $f : \mathbb{R}^p \rightarrow \mathbb{R}^d$, where $d \ll p$

- $f$ should preserve as much information about the original $X$ as possible.

# Dimensionality Reduction

降维：(机器学习和统计学领域）是指在某些限定条件下，降低随机变量个数，得到一组“不相关”主变量的过程。

两种方法:

- 特征选择
- 特征提取

**特征选择：** 假定数据中包含大量冗余或无关变量（或称特征、属性、指标等），旨在从原有变量中找出主要变量。其代表方法为 LASSO。

# Variance Thresholds

Variance thresholds remove features whose values don't change much from observation to observation (i.e. their variance falls below a threshold). These features provide little value.

For example, if you had a public health dataset where 96% of observations were for 35-year-old men, then the 'Age' and 'Gender' features can be eliminated without a major loss in information.

Because variance is dependent on scale, you should always normalize your features first.

# Strengths & Weaknesses

- Strengths : Applying variance thresholds is based on solid intuition : features that don't change much also don't add much information. This is an easy and relatively safe way to reduce dimensionality at the start of your modeling process.
- Weaknesses : If your problem does require dimensionality reduction, applying variance thresholds is rarely sufficient. Furthermore, you must manually set or tune a variance threshold, which could be tricky. We recommend starting with a conservative (i.e. lower) threshold.

```
import sklearn

sklearn.feature_selection.VarianceThreshold
```

Find more on : this site

## Correlation Thresholds

移除与其他特征高度相关特征（这些特征提供了冗余信息）

移除哪个特征？

calculate [pair-wise correlations] $\xrightarrow{\text{threshold}}$ compare [mean absolute correlation]

# Correlation Thresholds

移除与其他特征高度相关特征（这些特征提供了冗余信息）

移除哪个特征？

calculate [pair-wise correlations] $\xrightarrow{\text{threshold}}$ compare [mean absolute correlation]

# Correlation Thresholds

移除与其他特征高度相关特征（这些特征提供了冗余信息）

移除哪个特征？

calculate [pair-wise correlations] $\xrightarrow{\text{threshold}}$ compare [mean absolute correlation]

# Strengths & Weaknesses

- Strengths : Applying correlation thresholds is also based on solid intuition : similar features provide redundant information. Some algorithms are not robust to correlated features, so removing them can boost performance.
- Weaknesses : Again, you must manually set or tune a correlation threshold, which can be tricky to do. Plus, if you set your threshold too low, you risk dropping useful information. Whenever possible, we prefer algorithms with built-in feature selection over correlation thresholds. Even for algorithms without built-in feature selection, Principal Component Analysis (PCA) is often a better alternative.

# Genetic Algorithms (GA)

Genetic algorithms (遗传算法 )： 是计算数学中用于解决最优化的搜索算法，是进化算法的一种。进化算法最初是借鉴了进化生物学中的一些现象而发展起来的，这些现象包括遗传、突变、自然选择以及杂交等。

在机器学习中，GA 有两个主要用途：

- 优化 (eg：寻找神经网中的最优权重 )
- 有监督的特征选择
  features → “genes”；
  a candidate set of features → “organism”.

# Genetic Algorithms (GA)

Genetic algorithms (遗传算法 )： 是计算数学中用于解决最优化的搜索算法，是进化算法的一种。进化算法最初是借鉴了进化生物学中的一些现象而发展起来的，这些现象包括遗传、突变、自然选择以及杂交等。

在机器学习中，GA 有两个主要用途：

- 优化 (eg：寻找神经网中的最优权重 )
- 有监督的特征选择
  features → “genes”;
  a candidate set of features → “organism”.

## Flow Chart

1. 选择初始生命种群
2. 循环
    1. 评价种群中的个体适应度
    2. 以比例原则选择产生下一个种群
    3. 改变该种群（交叉和变异）
3. 直到停止循环的条件满足

# Flow Chart

1. 选择初始生命种群
2. 循环
    1. 评价种群中的个体适应度
    2. 以比例原则选择产生下一个种群
    3. 改变该种群（交叉和变异）
3. 直到停止循环的条件满足

# Flow Chart

1. 选择初始生命种群
2. 循环
    1. 评价种群中的个体适应度
    2. 以比例原则选择产生下一个种群
    3. 改变该种群（交叉和变异）
3. 直到停止循环的条件满足

## Flow Chart

1. 选择初始生命种群
2. 循环
    1. 评价种群中的个体适应度
    2. 以比例原则选择产生下一个种群
    3. 改变该种群（交叉和变异）
3. 直到停止循环的条件满足

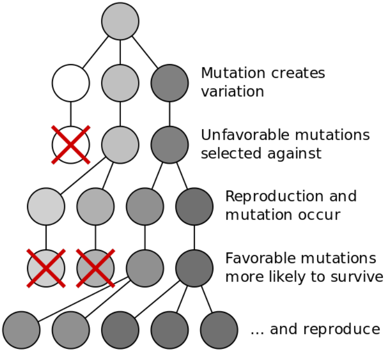
# Flow Chart

1. 选择初始生命种群
2. 循环
    1. 评价种群中的个体适应度
    2. 以比例原则选择产生下一个种群
    3. 改变该种群（交叉和变异）
3. 直到停止循环的条件满足

## Flow Chart

1. 选择初始生命种群
2. 循环
   1. 评价种群中的个体适应度
   2. 以比例原则选择产生下一个种群
   3. 改变该种群（交叉和变异）
3. 直到停止循环的条件满足

# The Loop


Mutation creates variation
Unfavorable mutations selected against
Reproduction and mutation occur
Favorable mutations more likely to survive
... and reproduce

## Strengths & Weaknesses

- Strengths : Genetic algorithms can efficiently select features from very high dimensional datasets, where exhaustive search is unfeasible. When you need to preprocess data for an algorithm that doesn't have built-in feature selection (e.g. nearest neighbors) and when you must preserve the original features (i.e. no PCA allowed), GA's are likely your best bet. These situations can arise in business/client settings that require a transparent and interpretable solution.
- Weaknesses : GA's add a higher level of complexity to your implementation, and they aren't worth the hassle in most cases. If possible, it's faster and simpler to use PCA or to directly use an algorithm with built-in feature selection.

## Stepwise Search

表现整体不如其他有监督的方法，如，正则化。

有很多记录在案的缺陷，不建议使用。

**特征提取：** 是将高维数据转化为低维数据的过程。在此过程中可能舍弃原有数据、创造新的变量，其代表方法为PCA。

# PCA

Given : Suppose we have $\mathbf{X} = (X_1, X_2, \cdots, X_p)^T$

Goal : Find a “lower dimensional” representation, say $d$-dimension, $d < p$, that captures main features of data.

# PCA : Math

- PCA defines transformed variables :

$$Z_z = \phi_{1z} X_1 + \phi_{2z} X_2 + \cdots + \phi_{pz} X_p$$

- Columns of $\Phi$ are unit vectors (length 1) and orthogonal :
  - $\|\boldsymbol{\phi}_j\| = \sqrt{\sum_i \phi_{ij}^2} = 1$
  - $\boldsymbol{\phi}_j^T \boldsymbol{\phi}_k = \sum_i (\phi_{ij} \phi_{ik}) = 0, \ \forall j \neq k$

- $Corr(Z_i, Z_j) = 0, \ \forall i \neq j$ (principal components are uncorrelated)
- $Var(Z_i) \geq Var(Z_j), \ \forall i < j$ (ordered by variables with highest variance)

# PCA : Math (cont.)

- The first principal component's rotation :

$$\phi_1 = \underset{\|\phi\|=1}{\arg\max} \left\{ \sum_i (\mathbf{x_i} \cdot \phi)^2 \right\}$$

$$= \underset{\|\phi\|=1}{\arg\max} \left\{ \|\mathbf{X}\phi\|_2 \right\} = \underset{\|\phi\|=1}{\arg\max} \left\{ \phi^T \mathbf{X}^\mathbf{T} \mathbf{X} \phi \right\} = \arg\max \left\{ \frac{\phi^T \mathbf{X}^\mathbf{T} \mathbf{X} \phi}{\phi^T \phi} \right\}$$

- The following ($k^{th}$) principal component's rotation :

$$\tilde{\mathbf{X}}_k = \mathbf{X} - \sum_{\mathbf{s=1}}^{\mathbf{k-1}} \mathbf{X} \phi_\mathbf{s} \phi_\mathbf{s}^\mathbf{T}$$

$$\phi_k = \underset{\|\phi\|=1}{\arg\max} \left\{ \|\mathbf{X_k}\phi\|_2 \right\}$$

The full principal components decomposition of $\mathbf{X}$ :

$$\mathbf{Z} = \mathbf{X}\,\phi$$

# PCA : Math (cont.)

- The first principal component's rotation :

$$\boldsymbol{\phi}_1 = \underset{\|\boldsymbol{\phi}\|=1}{\arg\max} \left\{ \sum_i (\mathbf{x_i} \cdot \boldsymbol{\phi})^2 \right\}$$

$$= \underset{\|\boldsymbol{\phi}\|=1}{\arg\max} \left\{ \|\mathbf{X}\boldsymbol{\phi}\|_2 \right\} = \underset{\|\boldsymbol{\phi}\|=1}{\arg\max} \left\{ \boldsymbol{\phi}^T \mathbf{X^T X} \boldsymbol{\phi} \right\} = \arg\max \left\{ \frac{\boldsymbol{\phi}^T \mathbf{X^T X} \boldsymbol{\phi}}{\boldsymbol{\phi}^T \boldsymbol{\phi}} \right\}$$

- The following ($k^{th}$) principal component's rotation :

$$\tilde{\mathbf{X}}_k = \mathbf{X} - \sum_{\mathbf{s=1}}^{\mathbf{k-1}} \mathbf{X} \boldsymbol{\phi}_\mathbf{s} \boldsymbol{\phi}_\mathbf{s}^\mathbf{T}$$

$$\boldsymbol{\phi}_k = \underset{\|\boldsymbol{\phi}\|=1}{\arg\max} \left\{ \|\mathbf{X_k}\boldsymbol{\phi}\|_2 \right\}$$

The full principal components decomposition of $\mathbf{X}$ :

$$\mathbf{Z} = \mathbf{X}\,\phi$$

# PCA : Math (cont.)

- The first principal component's rotation :

$$\boldsymbol{\phi}_1 = \underset{\|\boldsymbol{\phi}\|=1}{\arg\max} \left\{ \sum_i (\mathbf{x_i} \cdot \boldsymbol{\phi})^2 \right\}$$

$$= \underset{\|\boldsymbol{\phi}\|=1}{\arg\max} \left\{ \|\mathbf{X}\boldsymbol{\phi}\|_2 \right\} = \underset{\|\boldsymbol{\phi}\|=1}{\arg\max} \left\{ \boldsymbol{\phi}^T \mathbf{X^T} \mathbf{X} \boldsymbol{\phi} \right\} = \arg\max \left\{ \frac{\boldsymbol{\phi}^T \mathbf{X^T} \mathbf{X} \boldsymbol{\phi}}{\boldsymbol{\phi}^T \boldsymbol{\phi}} \right\}$$

- The following ($k^{th}$) principal component's rotation :

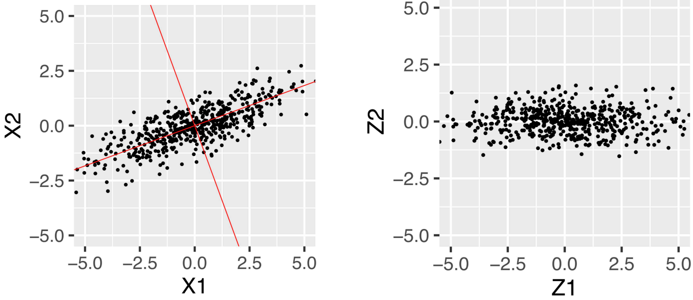
$$\tilde{\mathbf{X}}_k = \mathbf{X} - \sum_{\mathbf{s=1}}^{\mathbf{k-1}} \mathbf{X} \boldsymbol{\phi}_{\mathbf{s}} \boldsymbol{\phi}_{\mathbf{s}}^{\mathbf{T}}$$

$$\boldsymbol{\phi}_k = \underset{\|\boldsymbol{\phi}\|=1}{\arg\max} \left\{ \|\mathbf{X_k}\boldsymbol{\phi}\|_2 \right\}$$

The full principal components decomposition of $\mathbf{X}$ :

$$\mathbf{Z} = \mathbf{X}\,\boldsymbol{\phi}$$

# Simple example



Another classic case : Eigenfaces

# Strengths & Weaknesses

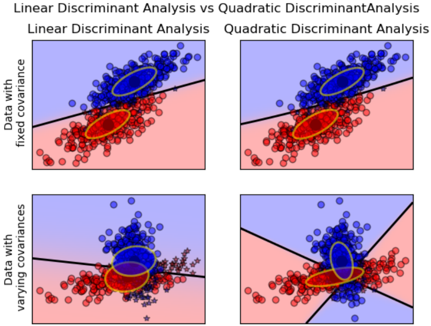
- Strengths : PCA is a versatile technique that works well in practice. It's fast and simple to implement, which means you can easily test algorithms with and without PCA to compare performance. In addition, PCA offers several variations and extensions (i.e. kernel PCA, sparse PCA, etc.) to tackle specific roadblocks.
- Weaknesses : The new principal components are not interpretable, which may be a deal-breaker in some settings. In addition, you must still manually set or tune a threshold for cumulative explained variance.

# LDA

Linear Discriminant Analysis (LDA), not ~~Latent Dirichlet Allocation~~

有监督的方法 - 只能用于带标签的数据

# LDA : example

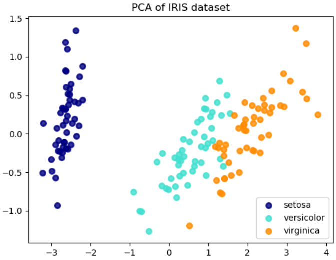


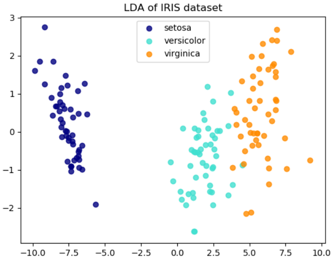
(image source)

The dimension of the output is necessarily less than the number of classes, so this is a in general a rather strong dimensionality reduction, and ONLY makes senses in a multi-class setting.

# Comparison of LDA and PCA 2D projection of Iris dataset

The Iris dataset represents 3 kind of Iris flowers (Setosa, Versicolour and Virginica) with 4 attributes : sepal length, sepal width, petal length and petal width.





explained variance ratio (first two components) : [0.92461621, 0.05301557]

## Strengths & Weaknesses

- Strengths : LDA is supervised, which can (but doesn't always) improve the predictive performance of the extracted features. Furthermore, LDA offers variations (i.e. quadratic LDA) to tackle specific roadblocks.
- Weaknesses : As with PCA, the new features are not easily interpretable, and you must still manually set or tune the number of components to keep. LDA also requires labeled data, which makes it more situational.

1. Introduction
   - Dimension
   - The Curse of Dimensionality
   - Dimensionality Reduction
2. Feature Selection
   - Variance Thresholds
   - Correlation Thresholds
   - Genetic Algorithms (GA)
   - Stepwise Search
3. Feature Extraction
   - Principal Component Analysis (PCA)
   - Linear Discriminant Analysis (LDA)

# 问题？