Feature Engineering

- Practical Machine Learning

QiChao 2016.10

Question

- What is Feature Engineering?
- Feature Engineering Pipeline?
- How to Apply Feature Engineering?

Outline

- **Definition of Feature Engineering**
- Feature Engineering Pipeline
- Non-Linearity and Model Stacking
- Feature Learning
- Application of Feature Engineering

Definition

- What is Feature Engineering?

- Feature engineering is the process of the transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved on unseen data.(作用)

- Feature engineering is the process of formulating the most appropriate features given the data, the model, and the task.(目的)

- Feature engineering is a super-set of activities which include feature extraction, feature construction and feature selection, each of the three are important steps and none should be ignored.(feature selection > feature extraction > feature construction)(包含内容)

Outline

- Definition of Feature Engineering
- **Feature Engineering Pipeline**
- Non-Linearity and Model Stacking
- Feature Learning
- Application of Feature Engineering

Pipeline

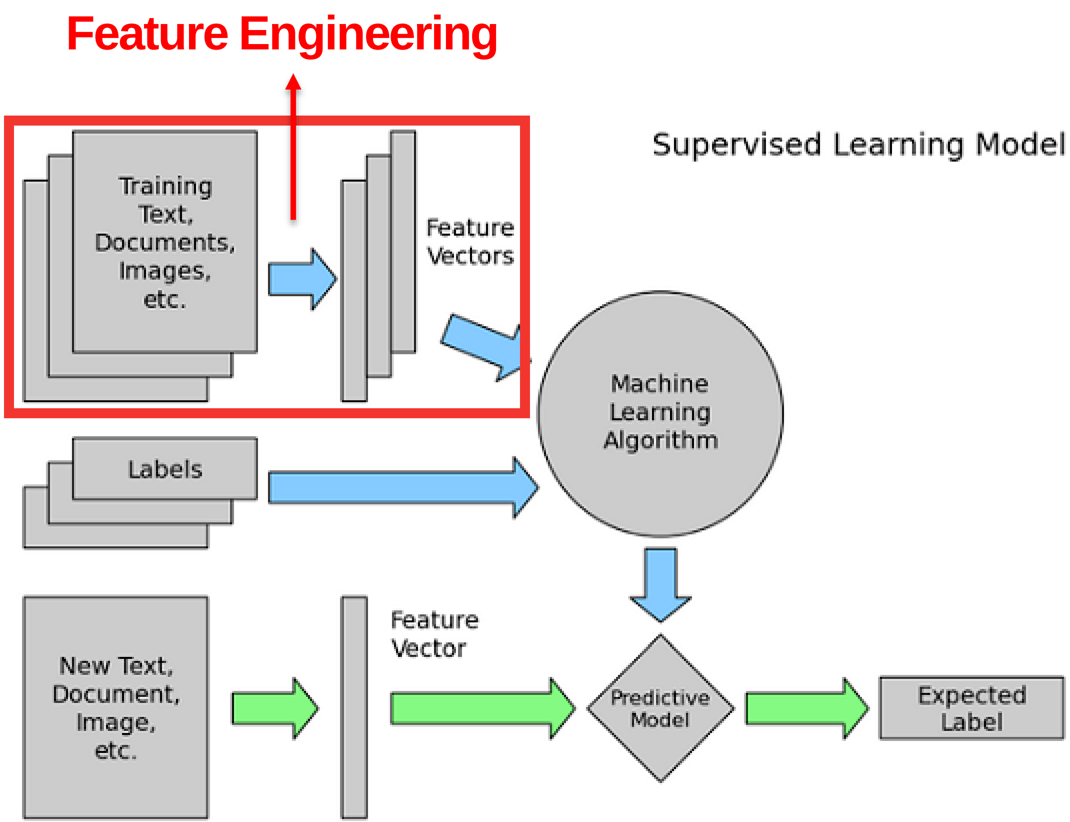
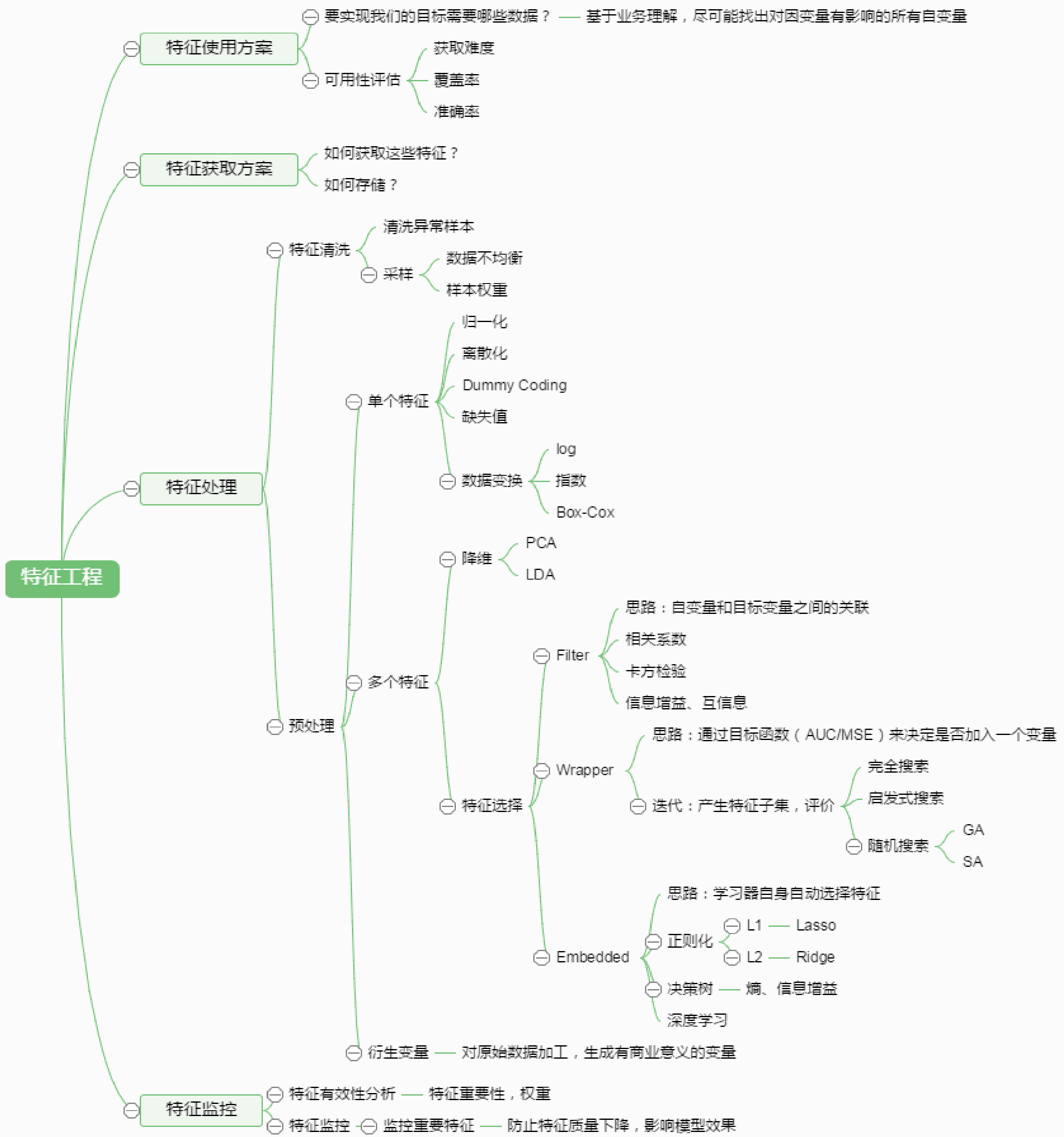
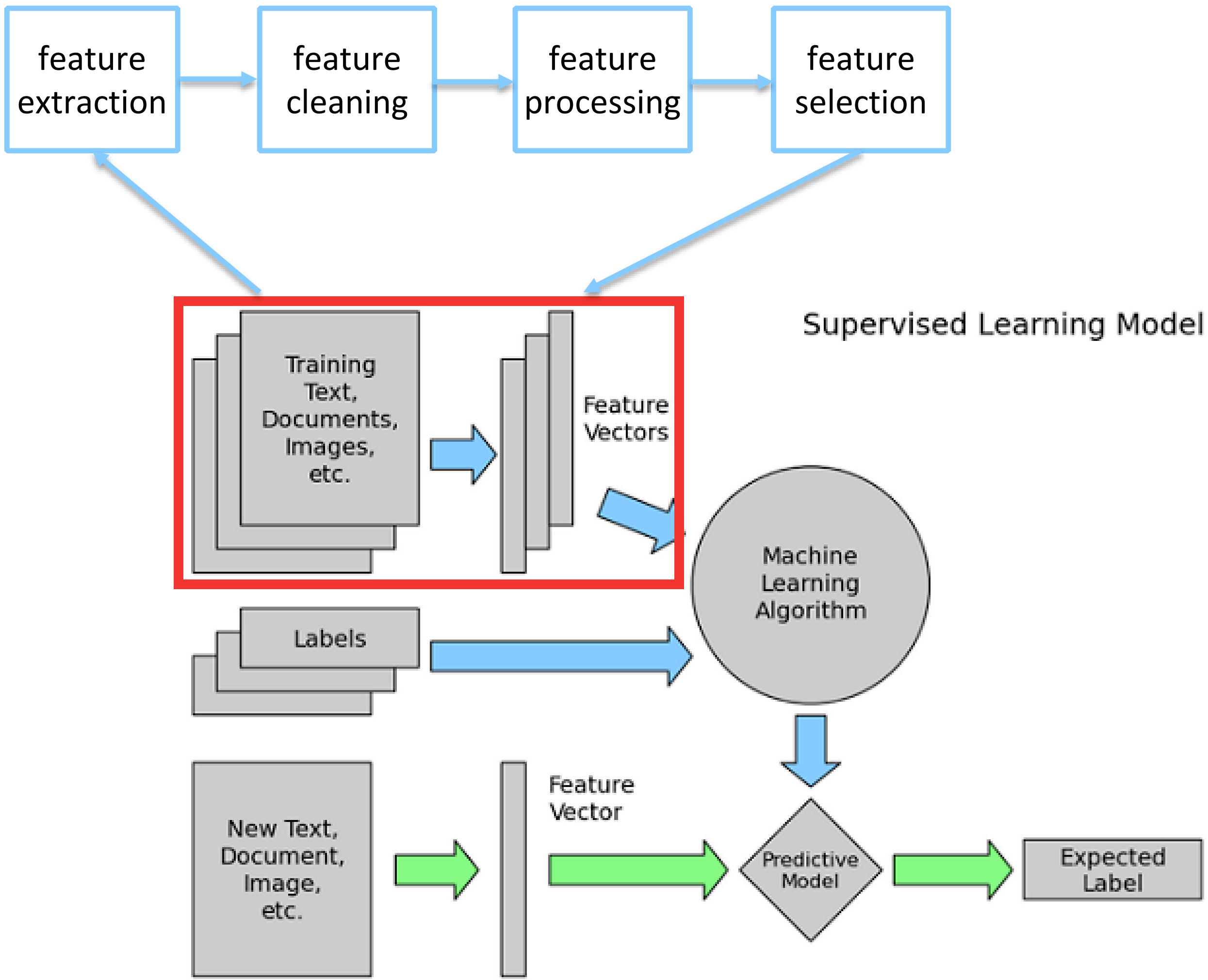


Figure. The place of feature engineering in the machine learning workflow


特征工程
特征使用方案
要实现我们的目标需要哪些数据？
基于业务理解，尽可能找出对因变量有影响的所有自变量
可用性评估
获取难度
覆盖率
准确率
特征获取方案
如何获取这些特征？
如何存储？
特征处理
特征清洗
清洗异常样本
采样
数据不均衡
样本权重
预处理
单个特征
归一化
离散化
Dummy Coding
缺失值
数据变换
log
指数
Box-Cox
多个特征
降维
PCA
LDA
特征选择
Filter
思路：自变量和目标变量之间的关联
相关系数
卡方检验
信息增益、互信息
Wrapper
思路：通过目标函数（AUC/MSE）来决定是否加入一个变量
迭代：产生特征子集，评价
完全搜索
启发式搜索
随机搜索
GA
SA
Embedded
思路：学习器自身自动选择特征
正则化
L1
Lasso
L2
Ridge
决策树
熵、信息增益
深度学习
衍生变量
对原始数据加工，生成有商业意义的变量
特征监控
特征有效性分析
特征重要性，权重
特征监控
监控重要特征
防止特征质量下降，影响模型效果

Pipeline



feature extraction
feature cleaning
feature processing
feature selection
Supervised Learning Model
Training Text, Documents, Images, etc.
Feature Vectors
Machine Learning Algorithm
Labels
New Text, Document, Image, etc.
Feature Vector
Predictive Model
Expected Label

Pipeline

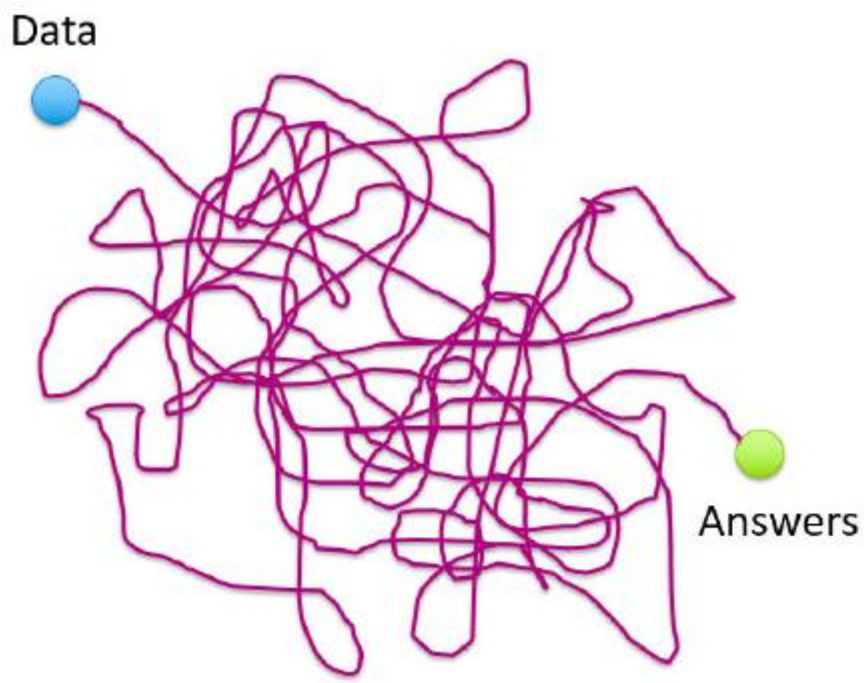


Figure. The messy path from data to answers

Pipeline

- Feature Categories
  - Raw Data:
    - text, image, Speech
  - Single Feature
    - 0/1([0, 1, 0, …])
    - continuous(0.12)
    - category(enumeration, [‘male’, ‘female’])
  - Stable, Dynamic
    - stable(hotel star, gender)
    - dynamic(geographic, age)
  - Low level, High level
    - low(gender, age)
    - high(log(x), sin(x))

Pipeline

- Feature Categories
- Feature Extraction & Construction
- Extraction
  - Image(sift, hog, ...)
  - Document(html, blog, email, …)
  - Text(BOW, TFIDF, word2vec, doc2vec)
  - Database(structured data)
- Construction
  - Discretization
  - Binarization
  - Transformation

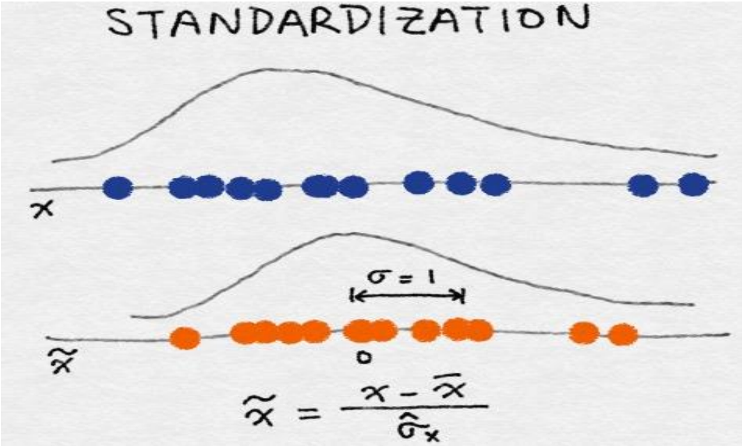
Pipeline

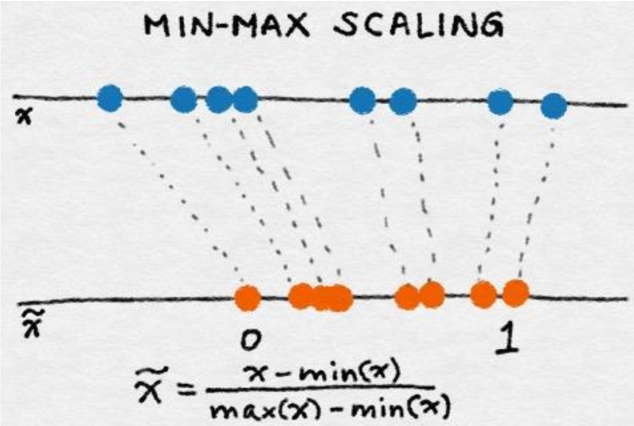
- Feature Categories
- Feature Extraction & Construction
- Feature Cleaning
  - Missing Value
    - (knn, mean, random and median)
  - Character Encoding
  - Outliers

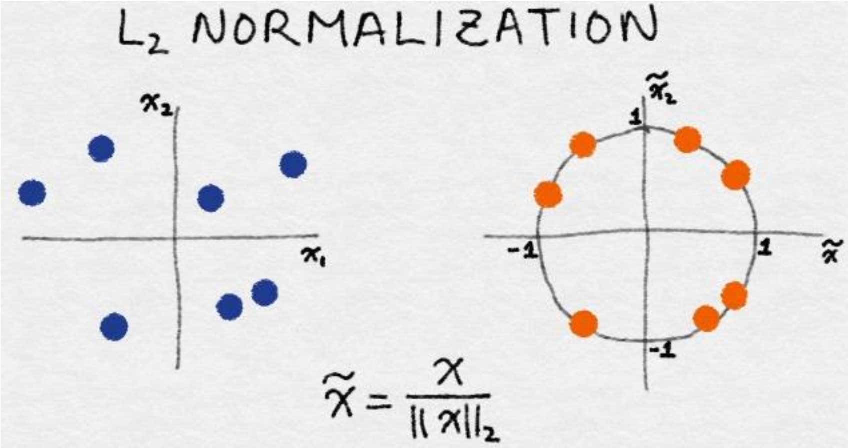
Pipeline

- Feature Categories
- Feature Extraction & Construction
- Feature Cleaning
- Feature Processing
  - Normalization(z-score, min-max, normalizer)

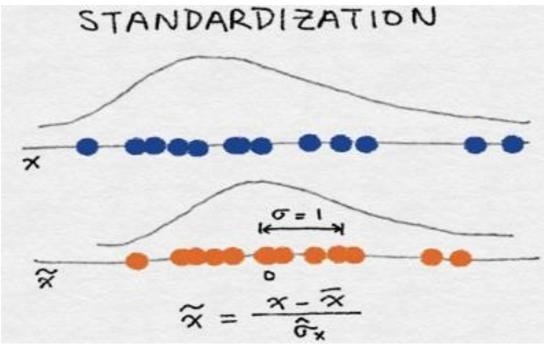
Pipeline

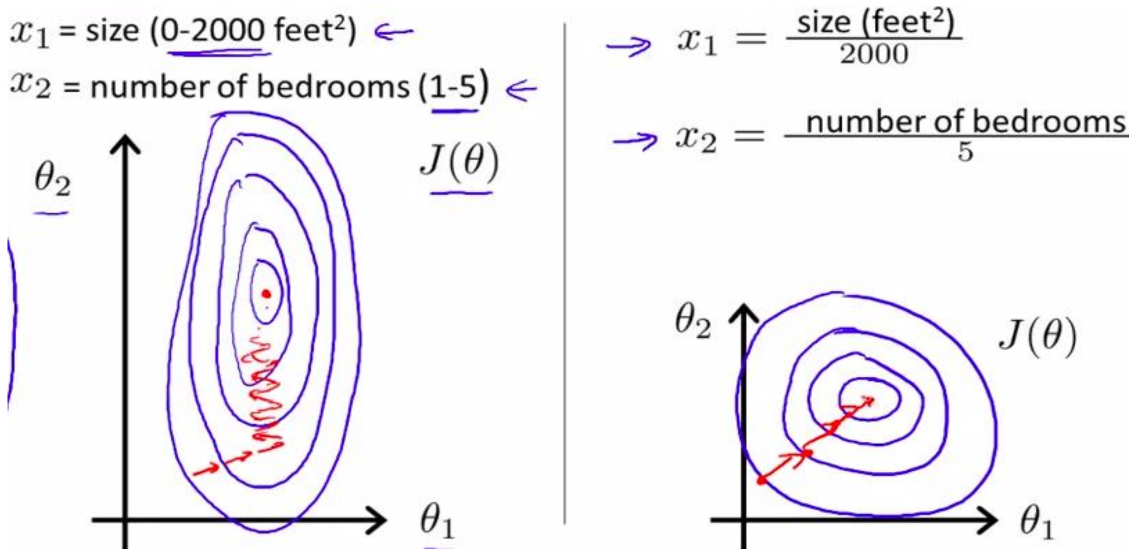

STANDARDIZATION
$\sigma = 1$
x
$\tilde{x}$
0
$\tilde{x} = \frac{x - \bar{x}}{\hat{\sigma}_x}$


MIN-MAX SCALING
x
$\tilde{x}$
0
1
$\tilde{x} = \frac{x - \min(x)}{\max(x) - \min(x)}$


L_2 NORMALIZATION
x_2
x_1
$\tilde{x}_2$
1
-1
1
-1
$\tilde{x}$
$\tilde{x} = \frac{x}{\|x\|_2}$

Pipeline

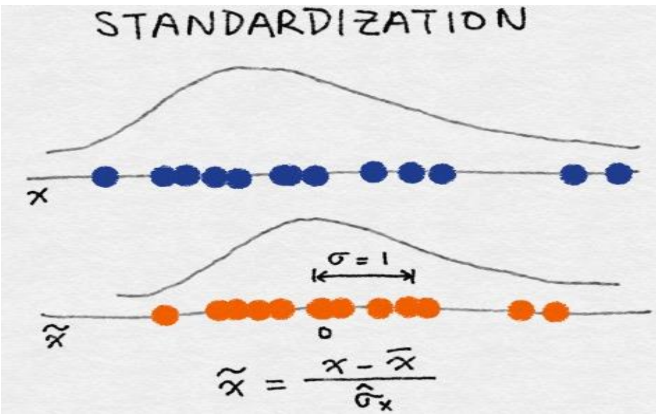


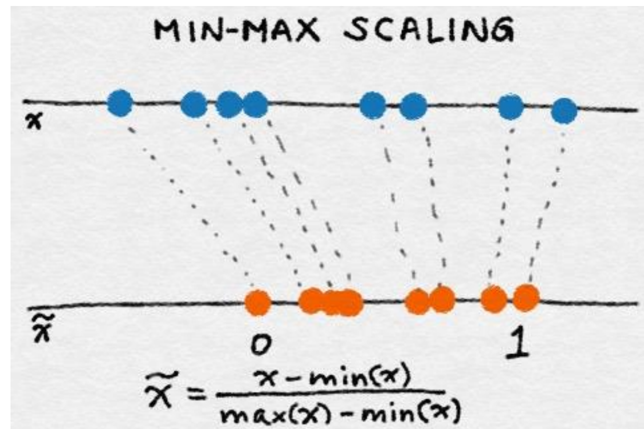


- KNN(KMeans) with Euclidean distance
- LR, SVM, Neural Networks(except Tree models) use gradient descent optimization
- LDA, PCA find direction of maximizing the variance

Pipeline

- Z-score or Min-Max ?





- No obvious answer, it depends on the application.
- Certain distance measures, PCA prefer Z-score over Min-Max scaling.
- Image processing, pixel intensities have to be normalized to fit within a certain range
- Neural Network algorithms require data that on a 0-1 scale.

Pipeline

- Feature Categories
- Feature Extraction & Construction
- Feature Cleaning
- Feature Processing
  - Normalization(z-score, min-max, Normalizer)
  - Encodings(one-hot, feature-hashing, bin-counting)
  - String Indexer([a, b, c] -> [1, 2, 3])

Pipeline

- Categories Feature Encodings

	one-hot	feature-hashing
space	O(n)	O(n)
time	O(kn) linear model	O(nm) linear/kernel
pros	1.Easiest to implement 2.Protentially most accurate 3.Feasible for online learning	1.Easy to implement 2.Makes model training cheaper 3.Easily adaptable to new categories 4.Easily handles rare categories 5.Feasisble for online learning
cons	1.Computationally inefficient 2.Does not adapt to growing categories 3.Not feasible for anything other than linear models 4.Require large-scale distributed optimization with truly large datasets	1.Only suitable for linear or kernelized models 2.Hashed features not interpretable 3.Mixed reports of accuracy

Pipeline

- Feature Categories
- Feature Extraction & Construction
- Feature Cleaning
- Feature Processing
- Feature Selection
  - Filter (Chi-square, Information Gain, Pearson, Variance Threshold)
  - Wrapper
  - Embedded(L1 norm, L2 norm, Elastic Net, Tree Model)

Pipeline

- L1 norm & L2 norm

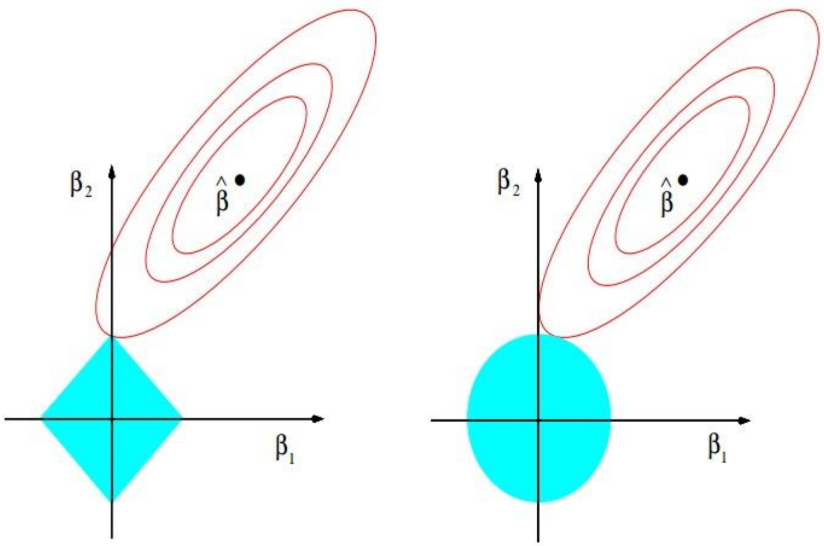


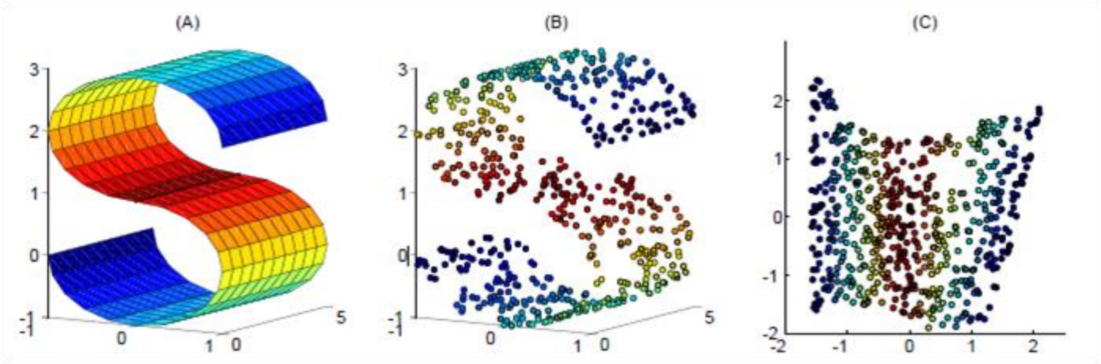
FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \le t$ *and* $\beta_1^2 + \beta_2^2 \le t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Pipeline

- Feature Categories
- Feature Extraction & Construction
- Feature Cleaning
- Feature Processing
- Feature Selection
- Dimensionality Reduction
  - Linearity
    - PCA, LDA
  - Non-Linearity:
    - Local Linear Embedding

Pipeline

- Non-Linearity: Local Linear Embedding



Pipeline

	PCA	LDA
similarity	1.Linear transformation tech. 2.Rely on the mean and covariance matrix	
difference	1.Unsupervised 2.Finds the directions of maximal variance 3.Mainly used for feature extraction 4.Can be formalized as Gaussian based ICA …	1.Supervised 2.Cares about class separability(between class variance is maximized, within class variance are minimized) 3.Mainly used for classification 4.Can be formalized as multinomial-Dirichlet based ICA …

Outline

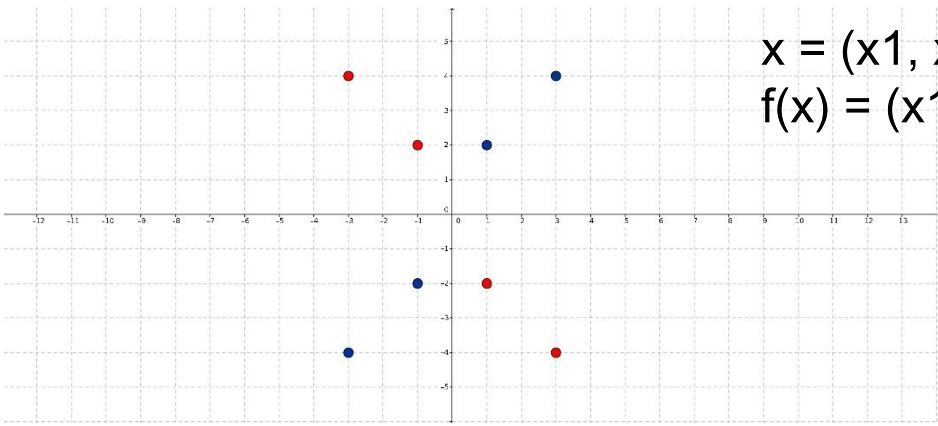
- Definition of Feature Engineering
- Feature Engineering Pipeline
- **Non-Linearity and Model Stacking**
- Feature Learning
- Application of Feature Engineering

Non-Linearity and Model Stacking

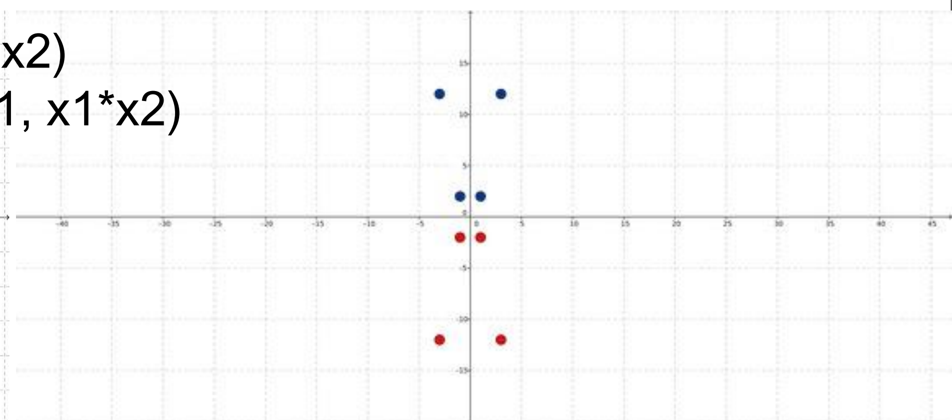
- Two methods to solve non-linearity

  1.Explicity use non-linear classifier, such as Tree Models, SVM(RBF kernel)

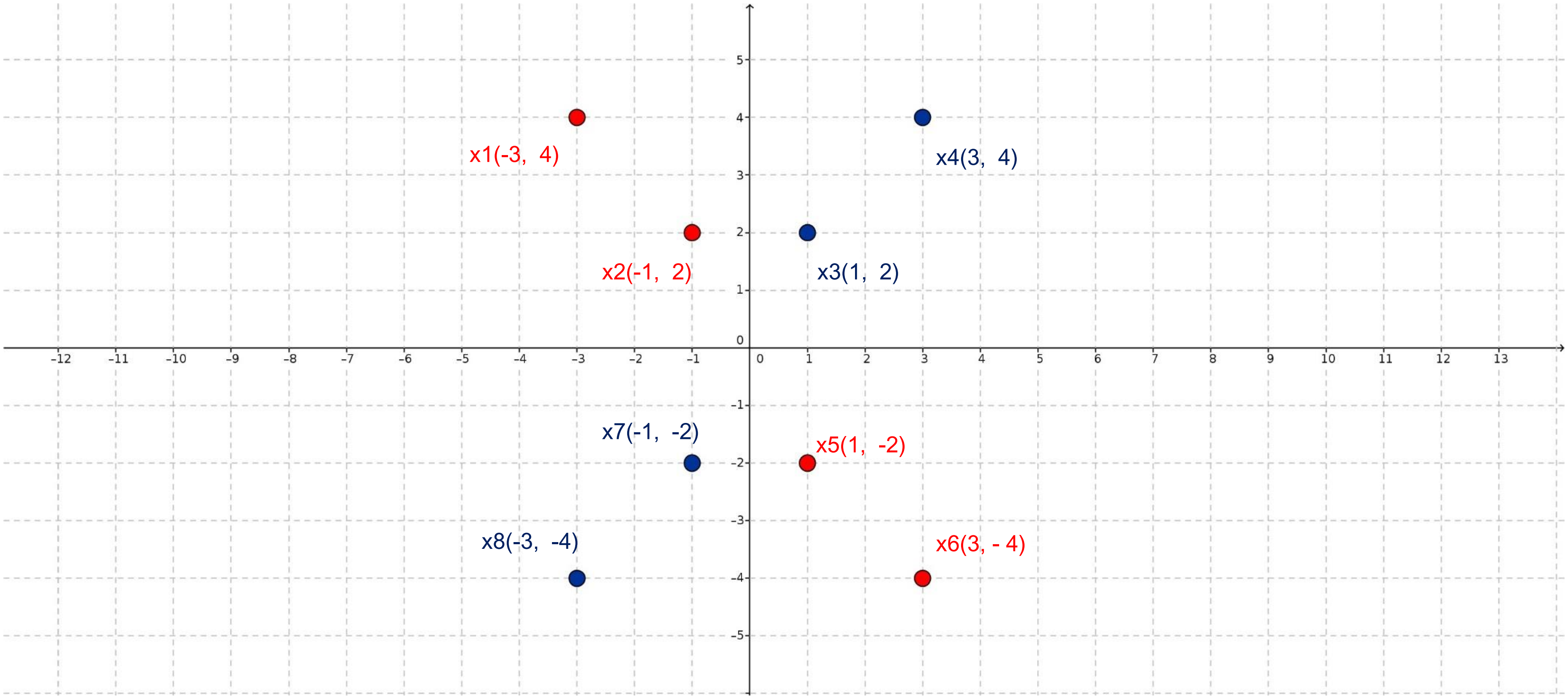
  2.Transform the input space to get a linear feature space



x = (x1, x2)
f(x) = (x1, x1*x2)

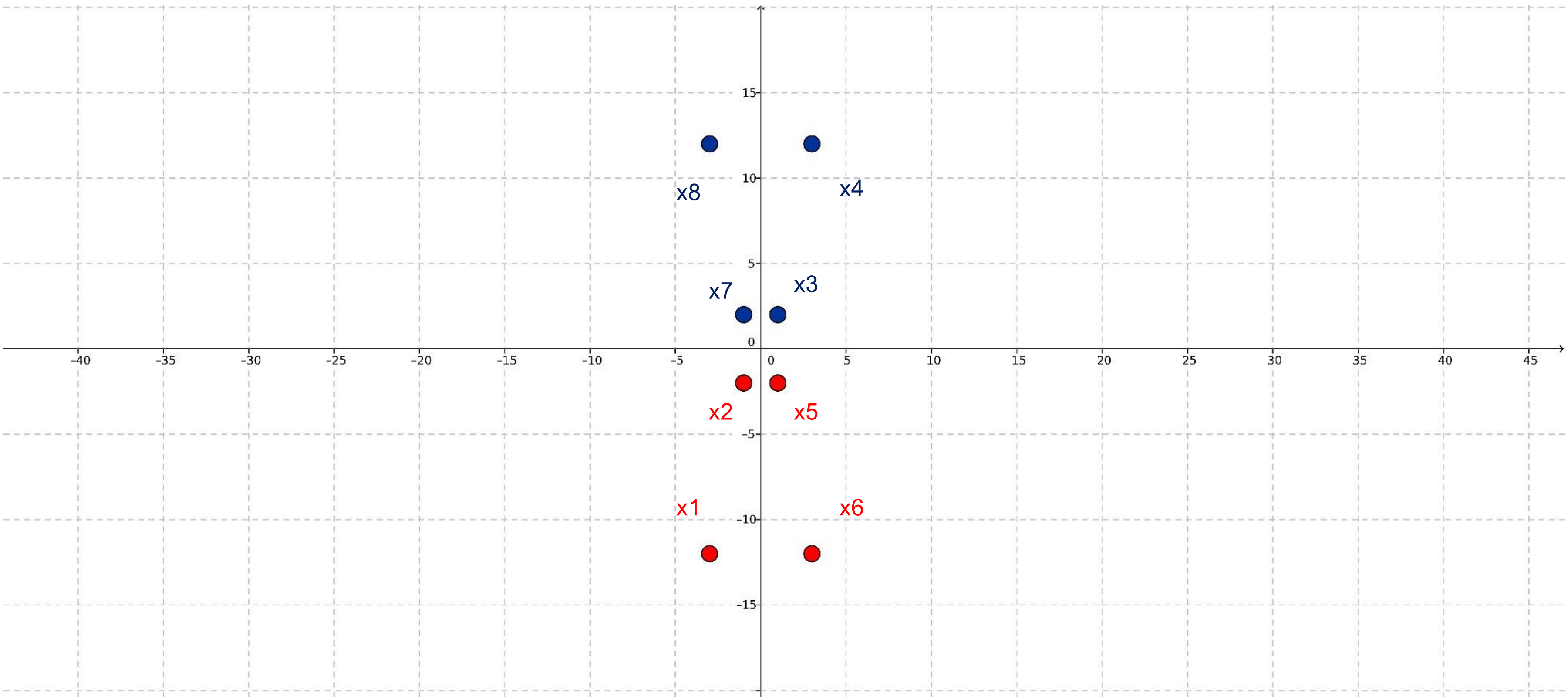


Non-Linearity and Model Stacking



Non-Linearity and Model Stacking

x = (x1, x2) => x' = (x1, x1*x2)

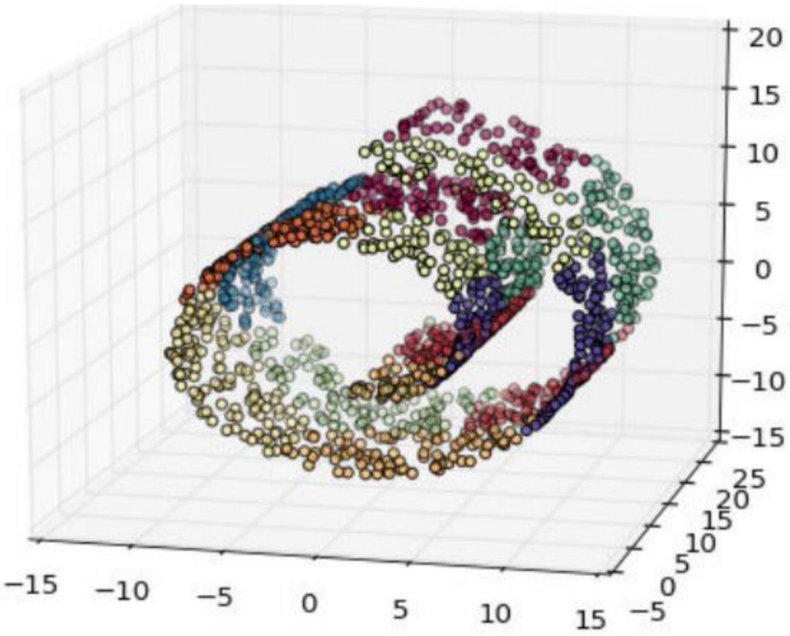


Non-Linearity and Model Stacking

- Non-Linearity Featurization
  - K-means featurization for classification(Supervised & Unsupervised)
- Model Stacking
  - Find a transformation for the given data such that the non linearity is removed.
  - Feature transformations with ensembles of trees(facebook paper)
  - Features generated by tree model(different training subset for training trees and meta-classifier(LR))

Non-Linearity and Model Stacking

- Non-Linearity Featurization
  - K-means featurization for classification(Supervised & Unsupervised)

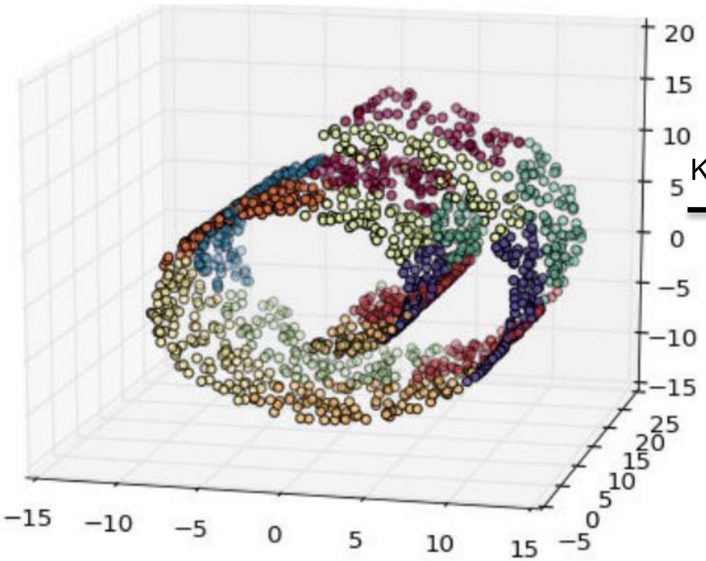


K-means on the Swiss roll with 10 clusters

K-means Featurization → (0, 0, 1, 0, 0, 0, 0, 0, 0, 0) or (0, 0, 0.7, 0, 0, 0, 0, 0, 0, 0)

Non-Linearity and Model Stacking

- Model Stacking



K-means on the Swiss roll with 10 clusters

K-means featurization → (0, 0, 1, 0, 0, 0, 0, 0, 0, 0) or (0, 0, 0.7, 0, 0, 0, 0, 0, 0, 0) → Logistic Regression

+

primitive feature vector (0.1, 0.3,…)

Non-Linearity and Model Stacking

- Model Stacking

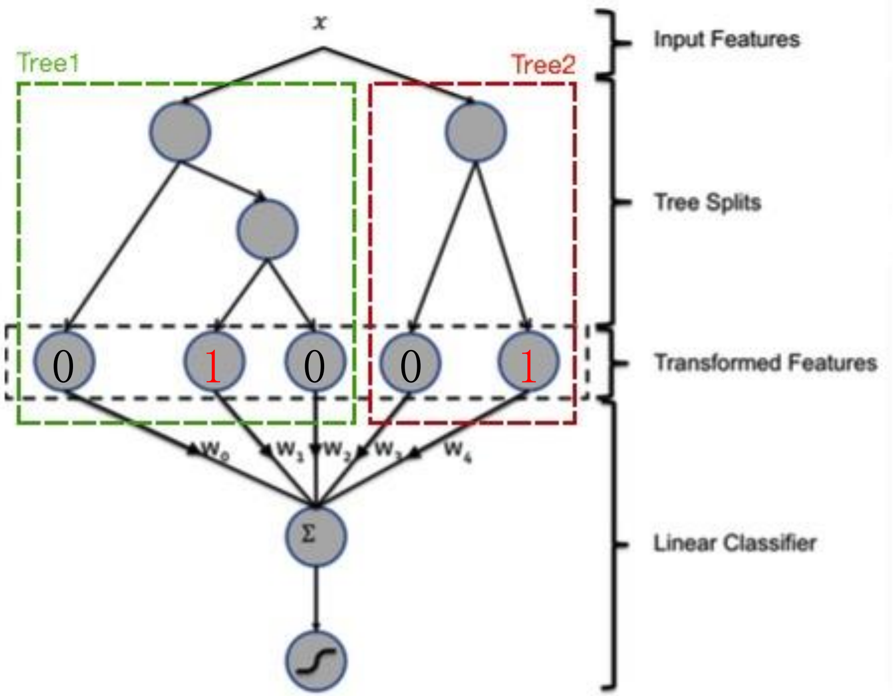
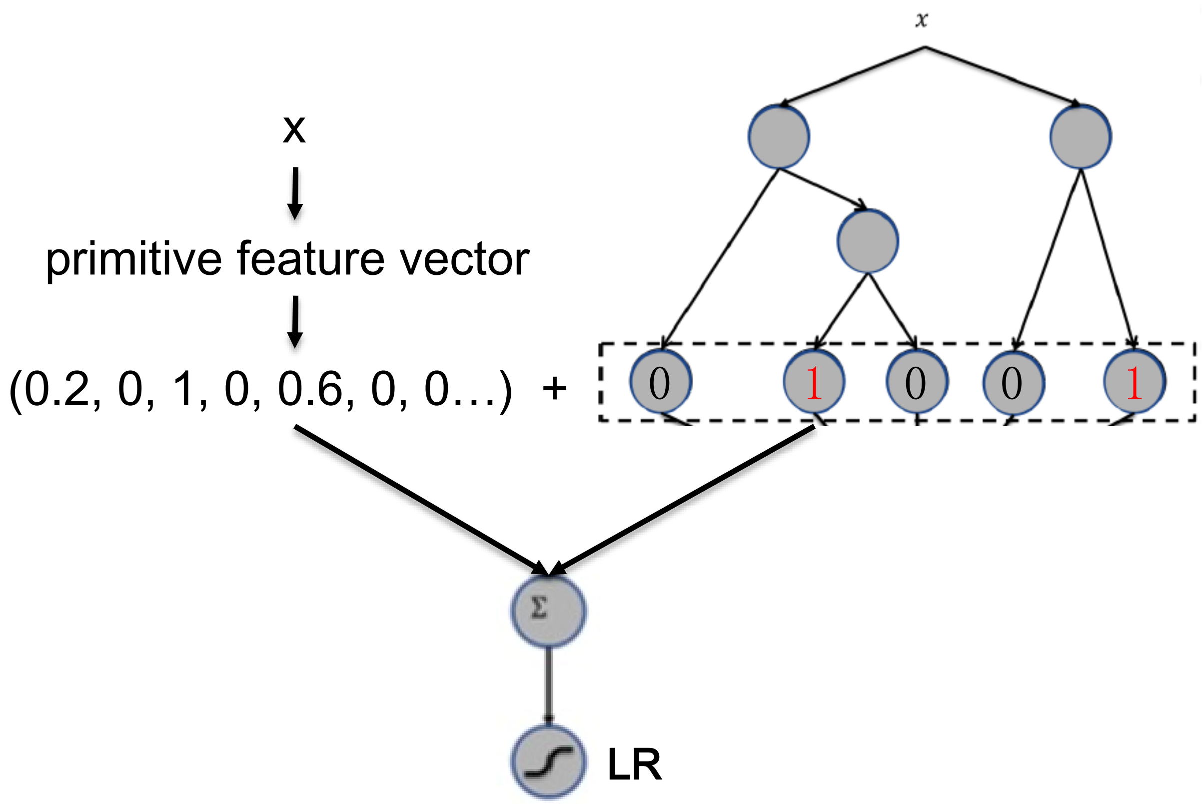


Figure 1: Hybrid model structure. Input features are transformed by means of boosted decision trees. The output of each individual tree is treated as a categorical input feature to a sparse linear classifier. Boosted decision trees prove to be very powerful feature transforms.

Non-Linearity and Model Stacking

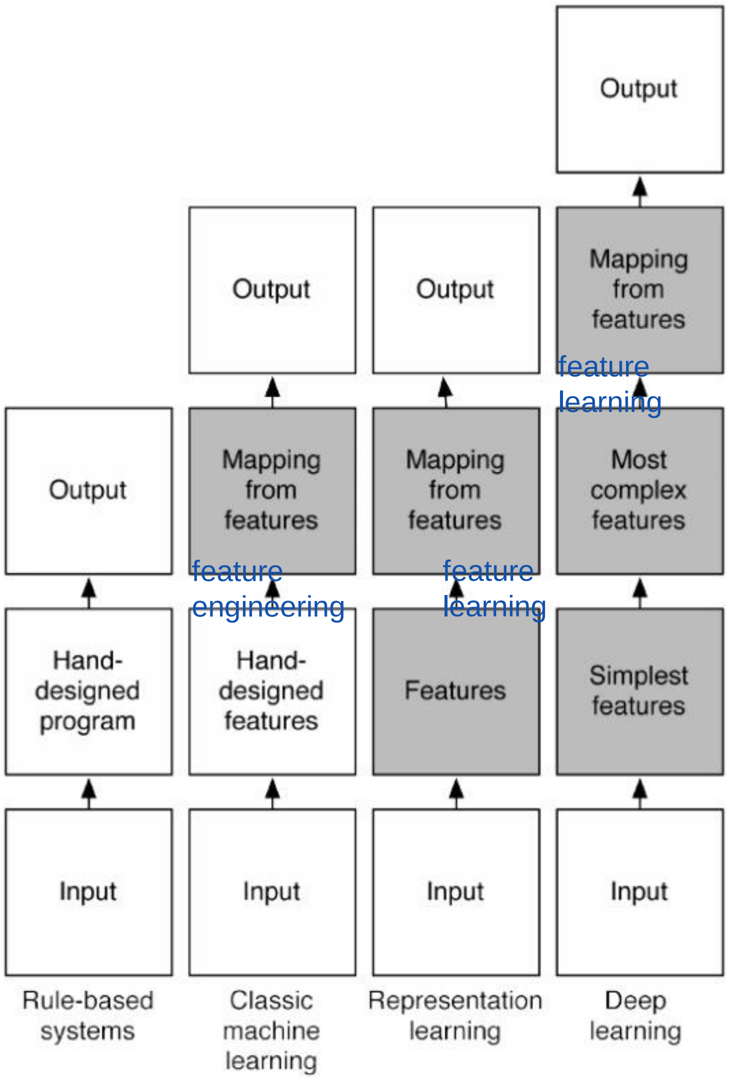
- Model Stacking



Outline

- Definition of Feature Engineering
- Feature Engineering Pipeline
- Non-Linearity Features and Model Stacking
- **Feature Learning**
- Application of Feature Engineering

Feature Learning



Feature Learning

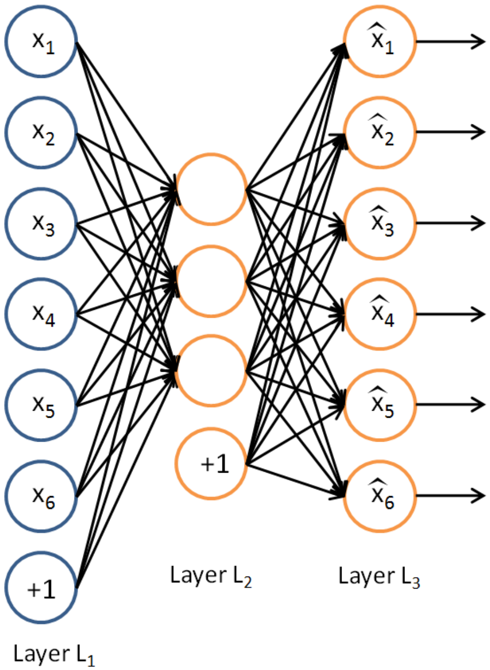
- Supervised Feature Learning

  (Learning features from labeled data)
  - Neural Networks
  - Dictionary learning
  - LDA
- Unsupervised Feature Learning

  (Learning features from unlabeled data)
  - K-Means, PCA, Local Linear Embedding
  - Autoencoder, Sparse coding, Self-Taught Learning
  - Generative Models

Feature Learning

- Autoencoder



$$J(W,b;x,y) = \frac{1}{2}\|h_{W,b}(x) - y\|^2$$

$h_{W,b}(x) = y \approx x$

- Sparse Autoencoder

Feature Learning

- Sparse coding

1.Find a set of basis vector ϕ_i

2.Represent X as a linear combination of these basis vectors ϕ_i, $\mathbf{x} = \sum_{i=1}^{k} a_i \phi_i$,k >> n

$$\mathbf{x} = \sum_{i=1}^{k} a_i \phi_i$$

$$\text{minimize}_{a_i^{(j)}, \phi_i} \quad \sum_{j=1}^{m} \left|\left| \mathbf{x}^{(j)} - \sum_{i=1}^{k} a_i^{(j)} \phi_i \right|\right|^2 + \lambda \sum_{i=1}^{k} S(a_i^{(j)})$$

$$\text{subject to} \qquad ||\phi_i||^2 \leq C, \forall i = 1, \ldots, k$$

L1 norm

$$\mathbf{x} = \sum_{i=1}^{k} a_i \phi_i$$

$$\text{minimize}_{a_i^{(j)}, \phi_i} \quad \sum_{j=1}^{m} \left|\left| \mathbf{x}^{(j)} - \sum_{i=1}^{k} a_i^{(j)} \phi_i \right|\right|^2 + \lambda \sum_{i=1}^{k} |a_i|_1$$

$$\text{subject to} \qquad ||\phi_i||^2 \leq C, \forall i = 1, \ldots, k$$

Outline

- Definition of Feature Engineering
- Feature Engineering Pipeline
- Non-Linearity Features and Model Stacking
- Feature Learning
- **Application of Feature Engineering**

Application

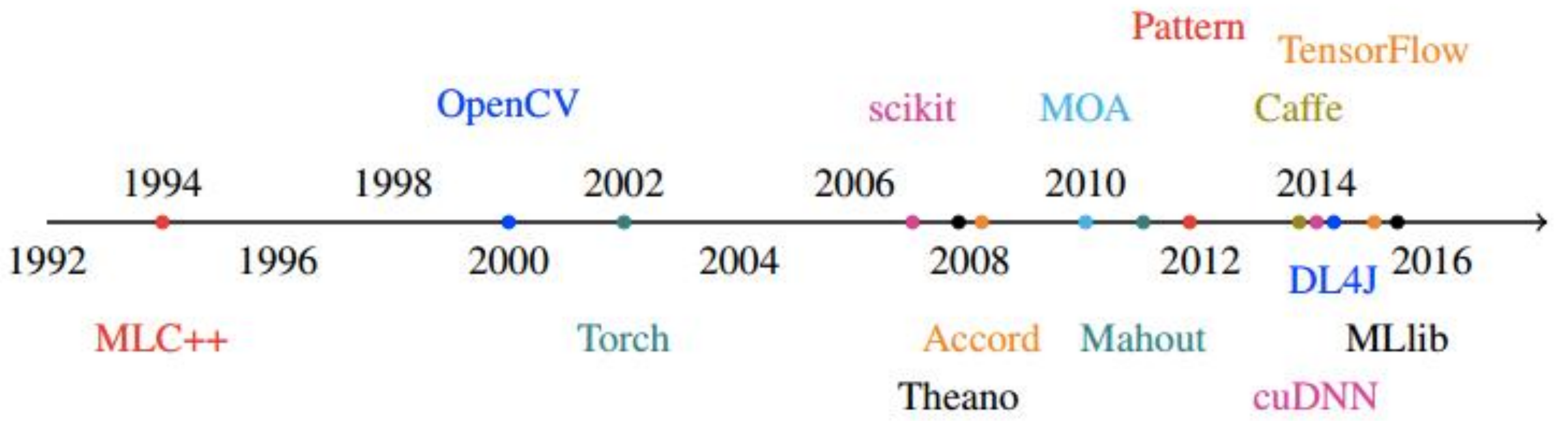


Fig. 1: A timeline showing the release of machine-learning libraries discussed in section I in the last 25 years.

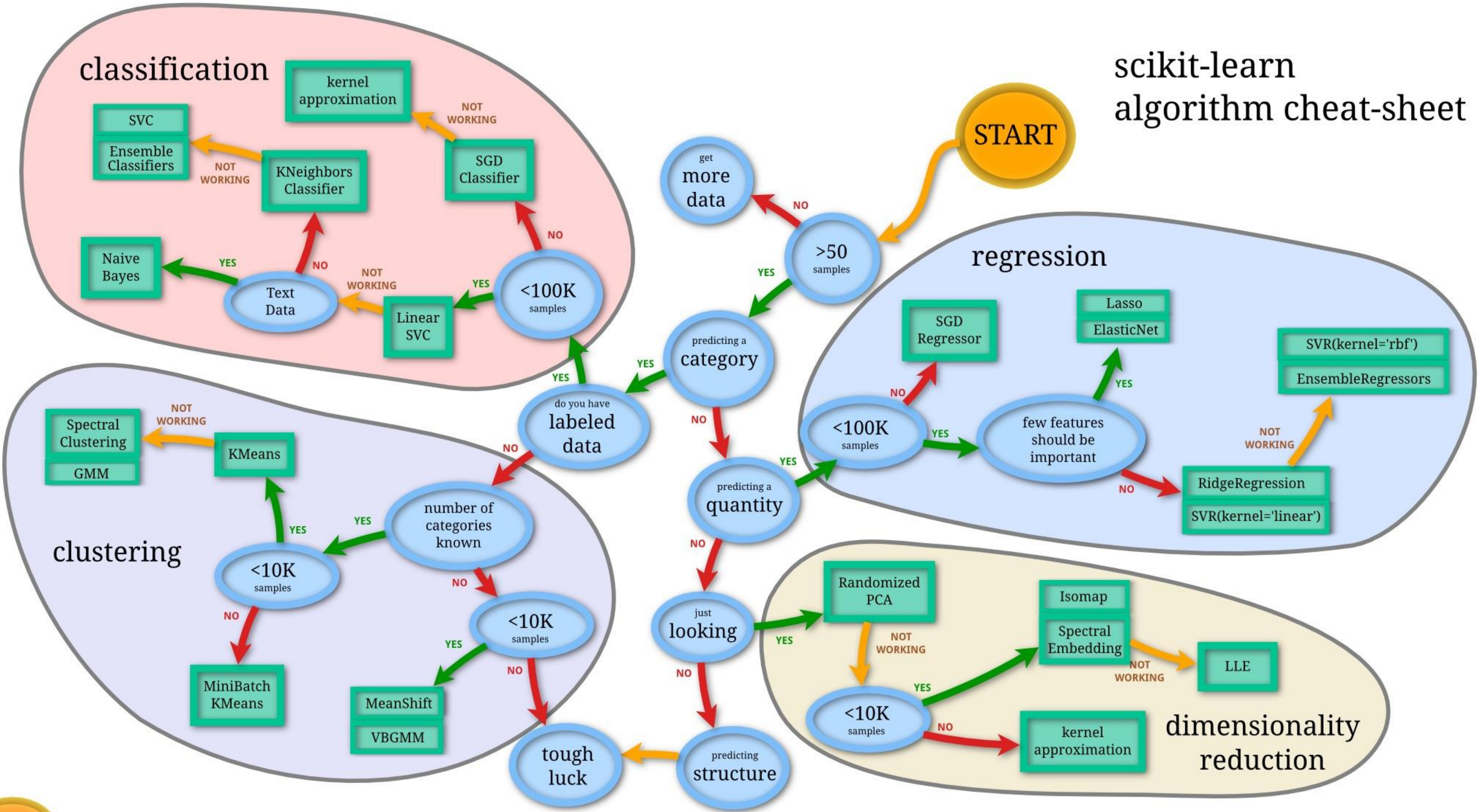
Application

Spark 1.6.2 Python 2.7	Spark ML	Python ML
DataFrame	Yes	Yes(Pandas)
ML Pipeline	Yes	Yes
XGBoost	Yes	Yes
MXNet	Yes	Yes
Word2Vec	Yes	Yes(Gensim)
Deep Learning	Deep Learning 4J	TensorFlow, Theano, Keras

Application

spark 1.6.2 sklearn 0.18	Spark ML	Scikit-Learn
Label	Binarizer / StringIndexer	LabelEncoder / LabelBinarizer
One-Hot	OneHotEncoder	OneHotEncoder
0-1	Binarizer	Binarization
Norm	Normalizer	Normalization
Z-score	StandarScaler	StandarScaler
Min-Max	MinMaxScaler	MinMaxScaler
Discretization	QuantileDiscretizer	No
Robust	No	RobustScaler
	IndexerToString	No
	VectorIndexer	No
	Bucketizer	No

Application



Back

Application

- Spark DataFrame VS Pandas DataFrame
- Scikit-Learn Feature Transformer VS Spark ML Feature Transformer
- Scikit-Learn Pipeline VS Spark ML Pipeline
- https://github.com/ustcqi/chuangxianghui/tree/master/src/main/scala
  http://stat-computing.org/dataexpo/2009/the-data.html

Question?

Thank You