Machine Learning Techniques

DATASCI 420

Lesson 04 Feature Selection

Feature Selection

- Process of selecting a subset of features that are good predictors of the target

- Useful for
  - Controlling complexity of model
  - Speed up model learning without reducing accuracy
  - Improve generalization capability

Model Selection vs Feature Selection

- Model selection includes selecting:
  - Model algorithm
  - Model algorithm hyperperameters
  - Features to be used to train the models

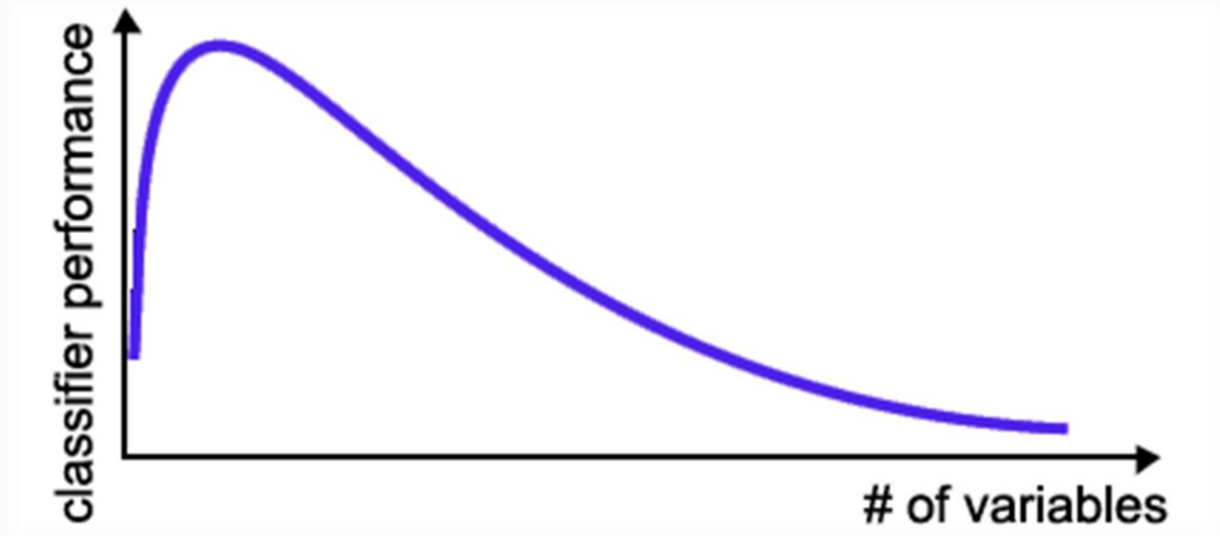
- Feature selection
  - Select features to be used to train the models

Why We Need Feature Selection?

Curse of Dimensionality

- The required number of samples (to achieve the same accuracy) grows exponentially with the number of variables!
- In practice: number of training examples is fixed!

  the classifier's performance will degrade for a large number of features!



In many cases the information lost by discarding variables is made up for by a more accurate mapping/sampling in the lower-dimensional space !

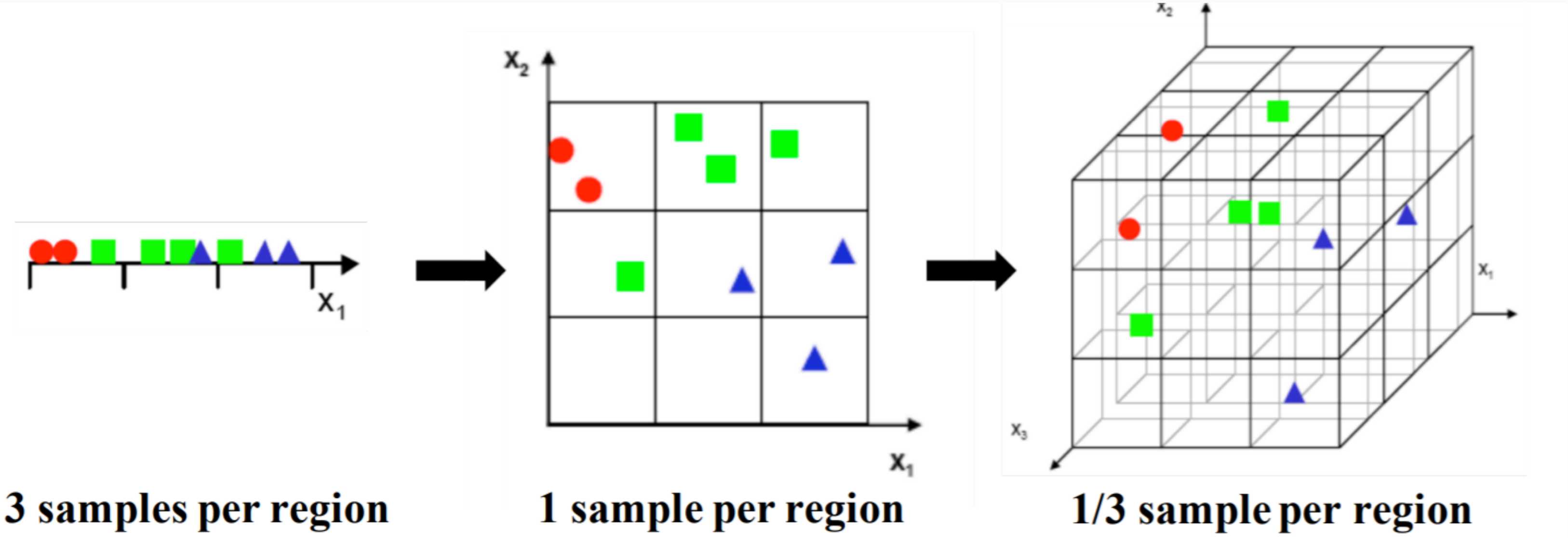
Problems of High-Dimensional Data

- High-dimensional data is often notorious to tackle due to the curse of dimensionality
  - Increase storage and running time
  - Overfit the machine learning models
  - Require more data

- The intrinsic dimension of data may be small
  - The number of genes responsible for a certain disease

Curse of Dimensionality – Required Samples

- Data sparsity becomes exponentially worse as feature dimension increases
  - Conventional distance metrics become ineffective
  - All points in the high-dimensional space look equally distant



http://nikhilbuduma.com/2015/03/10/the-curse-of-dimensionality/

Feature Selection, 3 types of methods

Filter Methods, select a subset of features before training a model, e.g.

- Correlation with target,
- Mutual Information between feature and target

- *Simple to implement, and have reasonable performance*

Wrapper Methods, search combination of feature space by training and evaluating model using a subset of features, e.g.

- Forward, backward, step-wise feature selection,
- Genetic algorithms.

- *Computationally expensive and prone to over-fitting*

Embedded Methods, feature subset is chosen as part of model training, e.g.

- LASSO (L-1) regression, Regularized decision trees, random forests

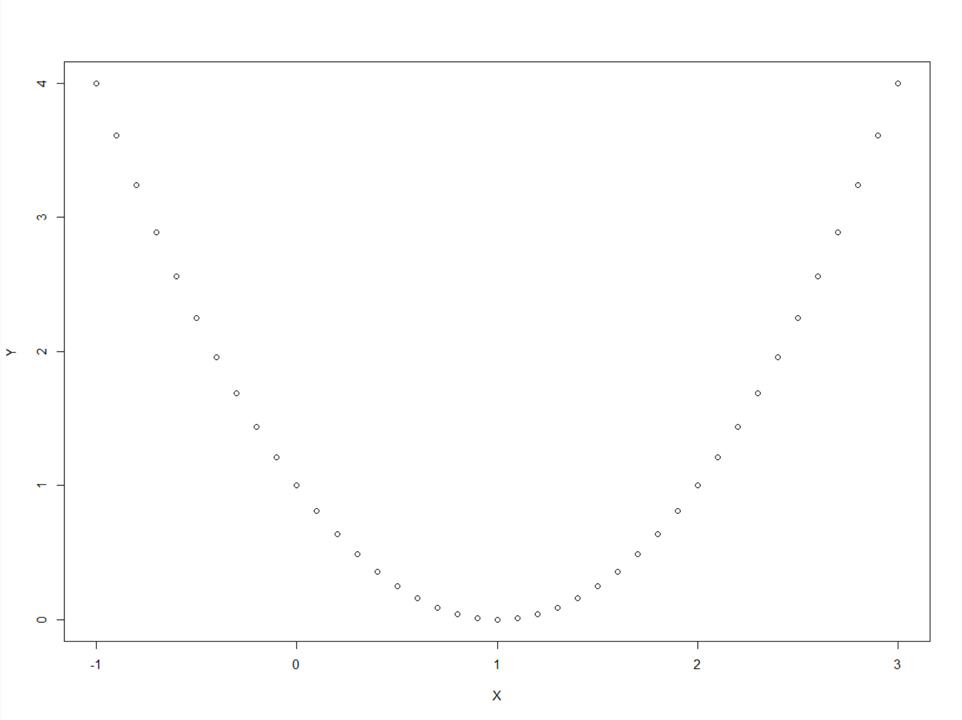
- *Typically robust to over-fitting, but has hyper parameters that will need to be fit using a validation data*

Filter-based Feature Selection

- Correlation with target variable
  - A good starting point
  - If Y is categorical variable (classification):
    - Use chi-square test to decide the correlation between each categorical X variable and Y variable
    - Use ANOVA test to decide the correlation between each numerical X variable and Y variable
  - If Y is continuous variable (regression):
    - Use ANOVA test to decide the correlation between each categorical X variable and Y variable
    - Use correlation between each numerical X variable and Y variable
  - *Alert*: If x1 and x2 are highly correlated, and x1 and Y are highly correlated, both x1 and x2 will be selected based on correlation with Y. Strong correlations in X will bring some challenge for some machine learning models, such as linear regression model.

Is Correlation Always a Good Choice?

- It makes sense for linear regression (logistic regression) model.
  - Since linear regression model only looks at linear relationship
- Does not make sense for nonlinear models such as tree-based models
- Cannot capture nonlinear relationship between X and Y



Mutual Information

- Captures Statistical Dependency between Two Variables
  - If two variables are statistically independent

$$\Pr(X, Y) = \Pr(X) \times \Pr(Y)$$

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p(x)\,p(y)}\right)$$

  - Estimate Pr(X) from observations by using a kernel function

$$\hat{f}(x) = \frac{1}{Nh\sqrt{2\pi}} \sum_{i=1}^{N} \exp\left(\frac{-(x-x_i)^2}{2h^2}\right).$$

Step-wise Model (Feature) Selection

- Forward:
  - Start with a model with only inception
  - Add one feature in the model at each step
  - At each step, the variable that can maximally reduce the residual sum of squares (RSS) is chosen as the feature to add in the model.
- Backward:
  - Start with a model with all features
  - Remove one feature from the model at each step
  - At each step, the variable that can minimally increase the residual sum of squares (RSS) is chosen as the feature to remove from the model.
- Both:
  - At each step, will check whether add a feature, or remove a feature

How to Select the Best Model (Feature Set)?

- Akaike information criterion (AIC)
  - k: number of coefficients to estimate in the model
  - L: likelihood of the training data based on the model

$$\mathrm{AIC} = 2k - 2\ln(\hat{L})$$

- Bayesian information criterion

$$\mathrm{BIC} = \ln(n)k - 2\ln(\hat{L}).$$

- Choose the model that has the minimal AIC or BIC
- AIC tends to choose a larger model than BIC
  - AIC has less penalty on the complexity of model (k) than BIC

Embedded Method

- Lasso (least absolute shrinkage and selection operator)

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{N} \|y - X\beta\|_2^2 \right\} \text{ subject to } \|\beta\|_1 \leq t.$$

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{N} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\}$$

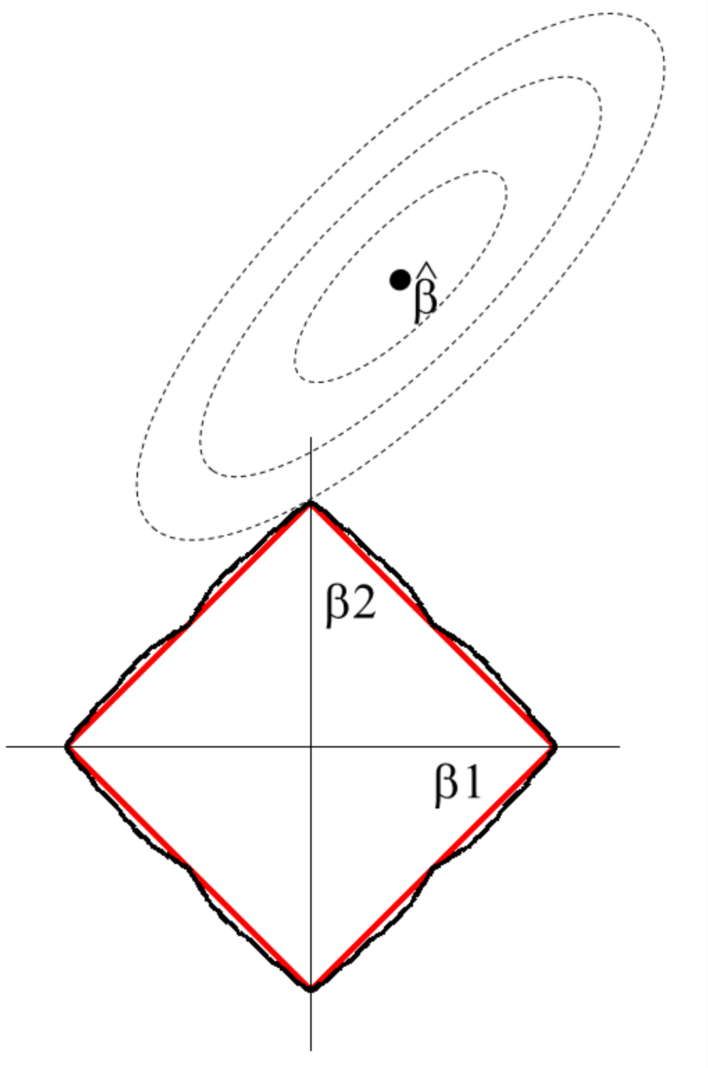
- Based on the second equation, we are penalizing on the complexity of the model (The sum of absolute values of the coefficients)

Why LASSO Can Select Features?

- Assuming only 2 X variables
- $\hat{\boldsymbol{\beta}}$ is the coefficient vector where there is no penalty
- Ellipsoid is the contour of MSE when coefficients change
- Very likely, some contour will meet with $|\beta_1| + |\beta_2| \leq t$

At the corner

- At the corner, the coefficients of some variables are set to 0
- These variables are de-selected



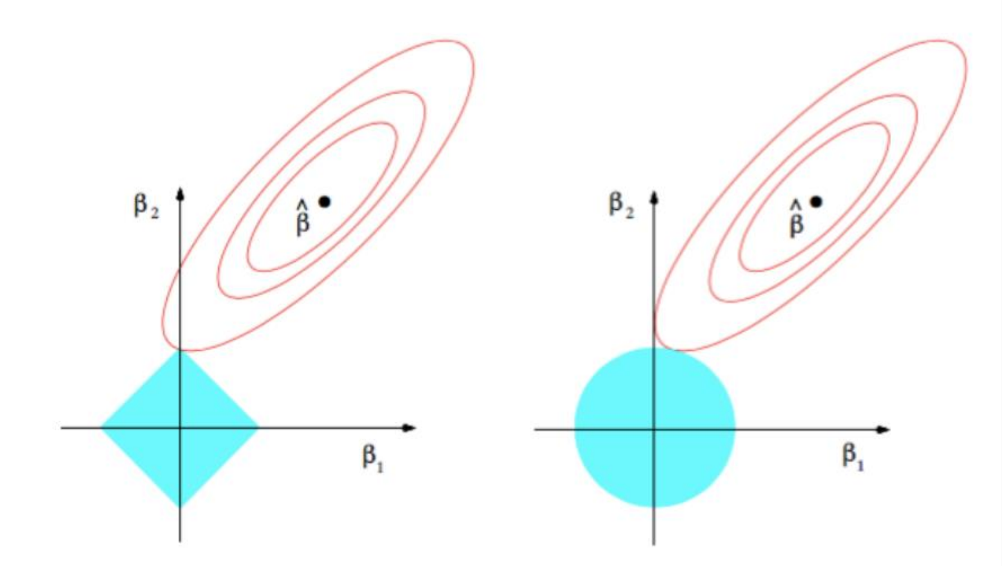
LASSO and Ridge Regression

- Ridge Regression

$$\text{minimize} \sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^\top \mathbf{z}_i)^2 \text{ s.t. } \sum_{j=1}^{p} \beta_j^2 \le t$$

$$\hat{\beta}_\lambda^{\text{ridge}} = (\mathbf{Z}^\top \mathbf{Z} + \lambda \mathbf{I}_p)^{-1} \mathbf{Z}^\top \mathbf{y}$$

- Ridge Regression can be helpful when Z is highly correlated
  - $(Z^\top Z)^{-1}$ does not exist, or is very sensitive to noise
  - $(Z^\top Z + \lambda I_p)$ is always inversible.
- But Ridge Regression just shrinks variables, it does not select variables



Feature Selection and Engineering

Optimality?

In theory the goal is to find an optimal set of features, one that maximizes the scoring function…

In real world applications this is usually not possible

- For most problems it is computationally intractable to search the whole space of possible feature subsets
- One usually has to settle for approximations of the optimal subset
- Most of the research in this area is devoted to finding efficient search-heuristics