INET 4061

Data Science I: Fundamentals

Lecture 6

Feature Selection, Model Selection, and Evaluation

Overview

- Feature/Variable Selection
- Model Selection
- Evaluation – Cross Validation
- Improve Models – Regularization
  - Ensembles

Feature/Variable Selection

- Dimensionality reduction
  - process of reducing the number of random variables to a set of principal variables.
  - divided into feature selection and feature extraction
- Feature selection
  - When input data to an algorithm is too large to be processed and variables are suspected to be redundant, variables can be transformed into a reduced set of features (called a feature vector)
  - Selected features are expected to contain relevant information from the input data, so that the desired ML task can be performed using the reduced feature subset instead of the complete data set.
- Feature extraction/creation
  - Creates derived variables (features) intended to be informative and non-redundant

https://en.wikipedia.org/wiki/Dimensionality_reduction
https://en.wikipedia.org/wiki/Feature_extraction

Dimension Reduction

Dimensionality reduction is used when you have many dimensions with *correlated* variance and want to reduce problem size by rotating data points into new orthogonal basis and taking only axes with largest variance.

Examples: http://scikit-learn.org/stable/modules/decomposition.html#

http://datascience.stackexchange.com/questions/1159/how-to-do-svd-and-pca-with-big-data

Why reduce dimensions?

- Avoid the curse of dimensionality
- Discover hidden correlations/topics
  - Words that commonly occur together
  - Word – vector of length 0 or 1
- Remove redundant and noisy features
  - Duplicate info contained in one or more other attributes (weight effect)
  - Not all words are useful
  - Irrelevant features or noise
  - No useful info for data mining task (confuse model)
- Interpretation and visualization
  - 2D or 3D
- Easier storage and processing
  - Reduce time and memory required

https://www.youtube.com/watch?v=SMvNwg9vraM

Curse of Dimensionality

- High-dimensional data
  - Data becomes sparser in highly-dimensional space, affecting algorithms designed for low-dimensional space
- Large number of features
  - Learning models tend to overfit
  - Performance degradation
    - Memory
    - Storage
    - Computation

https://arxiv.org/abs/1601.07996

Predictor Variable Techniques

- **PCA**: Principal Component Analysis
  - Linear mapping of the data to a lower-dimensional space so the explained variance in the low-dimensional representation is maximized
  - Nonlinear mapping
    - ex. kernel, cost function
- **SVD**: Support Vector Decomposition
  - factorization of A into the product of three matrices $A = UDV^H$ where the columns of U and V are orthonormal and the matrix D is diagonal with positive real entries.
- **Probabilistic models** – ex. Factor Analysis, LDA
- **Additive models** – ex. Non-Negative Matrix Factorization
- **Optimization** – ex. Dictionary Learning
- **Feature Hashing** –
  - https://alex.smola.org/papers/2009/Weinbergeretal09.pdf
- **Approximation Techniques**
- **…**

https://en.wikipedia.org/wiki/Singular_value_decomposition

Feature Creation

Feature Creation

- Original attributes not always the best representation
- Create new features that are more efficient/focused

- Techniques
  - Feature Extraction
    - Domain Specific
  - Feature Construction
    - Combine Features
      - ex. Aggregation
  - Map to New Space
    - Transforms entire data set
      - ex. Fourier transform

https://www.youtube.com/watch?v=Ah9XWzsB2mo

Create Features

- **Extracted data**
  - **Highlight hidden relationships**
  - **Create variables for difference in date, time, location**
  - **Create new ratio or proportion**
    - **Input/Output, productivity, efficiency, percentage**
  - **Create derived variable**
    - **transformation, binning**
      - **Log -** change the shape of distribution of the variable, reduce right skewness
      - **Cube root** – can be applied to zero and negative values
      - **Binning** – categorical values, percentile, frequency
  - **Create dummy variable**
    - convert categorical variable to numerical variable
- **Enrich data set**
  - **ex. geolocation associated features, weather**

https://www.analyticsvidhya.com/blog/2013/11/simple-manipulations-extract-data/
https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/

Examples – Create Features

- Polynomial Expansion – add interactions and powers
- http://www.dummies.com/programming/big-data/data-science/machine-learning-creating-features-data/

- Add Transformations
- https://courses.washington.edu/css490/2012.Winter/lecture_slides/05a_feature_creation_selection.pdf

- Add Temporal Features
- https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/create-features

- Same Algorithm – different data sets
- https://gallery.cortanaintelligence.com//Experiment/Regression-Demand-estimation-4

- Feature Hashing
- https://msdn.microsoft.com/library/azure/c9a82660-2d9c-411d-8122-4d9e0b3ce92a/

Feature Selection

Why Select a Feature Subset?

- Train machine learning algorithms faster
- Reduce model complexity
- Easier to interpret
- Can improve model accuracy
- Reduce overfitting
- Prepare cleaner, more understandable data

https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/

Feature Selection Techniques

- Brute force
  - Try all possible feature subsets as input to model
- Filter
  - Features are selected before input to model
- Iteration
  - Use an algorithm to find features
- Embedded
  - Feature selection occurs naturally as part of the model

https://www.youtube.com/watch?v=Ah9XWzsB2mo

Filter Method

- Usually done during preprocessing
  - Selection independent of machine learning algorithm
  - Select features based on statistical scores
    - correlation with the outcome variable
- Filter methods do not remove multicollinearity

Feature\Response	Continuous	Categorical
Continuous	Pearson's Correlation	LDA
Categorical	Anova	Chi-Square

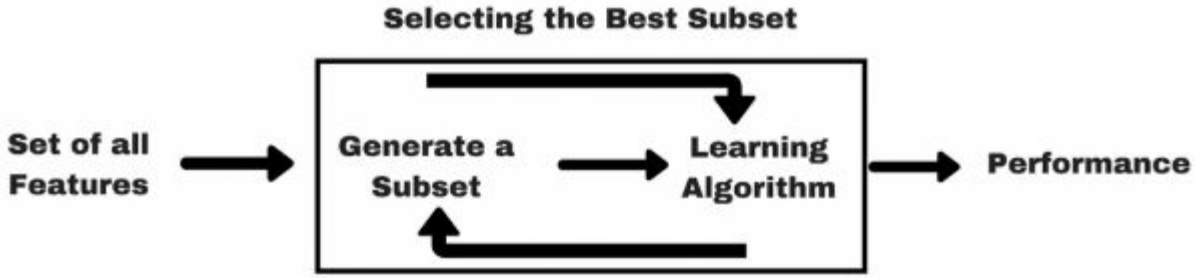
https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/

Filter Tests

- **Pearson's Correlation**
  - linear dependence between two continuous variables X and Y
- **LDA:** Linear Discriminant Analysis
  - linear combination of features that separates two or more classes of a categorical variable
- **ANOVA:** Analysis of Variance
  - statistical test whether the means of several groups are equal
- **Chi-Square:**
  - statistical test applied to groups of categorical features to evaluate the likelihood of correlation between groups based on their frequency distribution

https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/

Wrapper Method



- Train a model using a subset of features
- Iteratively decide to continue to add or remove features
- Computationally expensive
- Techniques
  - Forward Selection
  - Backward Elimination
  - Recursive Feature Elimination

https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/

Wrapper Techniques

- **Forward Selection:**
  - Iterative method starting with no features in the model.
  - Each iteration, add the feature that best improves the model until an addition of a new feature does not improve model performance.
- **Backward Elimination:**
  - Iterative method starting with all features
  - Each iteration, remove the least significant feature that improves the performance
  - Repeat until feature removal does not improve model performance.
- **Recursive Feature elimination:**
  - Greedy optimization algorithm
  - Iteratively creates models and remembers the best or the worst performing feature at each iteration.
  - Constructs the next model with the remaining features until all features are exhausted
  - Ranks features based on the order of their elimination.

https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/

Filter vs. Wrapper

- Filter methods measure relevance of features based on correlation with dependent variable
- Wrapper methods measure a subset of feature by training a model on features.

- Filter methods are much faster compared to wrapper methods as they do not involve training the models.
- Wrapper methods are computationally very expensive as well.

- Filter methods use statistical methods for evaluation of a subset of features.
- Wrapper methods use cross validation.

- Filter methods might fail to find the best subset of features in many occasions.
- Wrapper methods can always provide the best subset of features.

- Using the subset of features from the wrapper methods make the model more prone to overfitting as compared to using subset of features from the filter methods.

https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/

Wrapper Search

- Search space for d features is 2^d
  - Exhaustive search impractical when d is very large
  - Search strategies
    - ex. hill-climbing, best-first, branch-and-bound, genetic
    - Search space still very large for high-dimensional datasets
      - Wrapper methods seldom used in practice

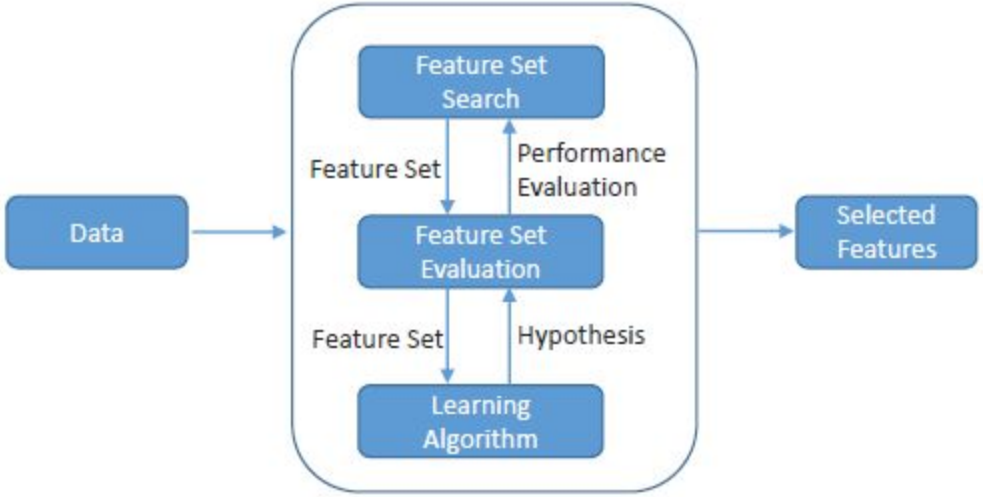
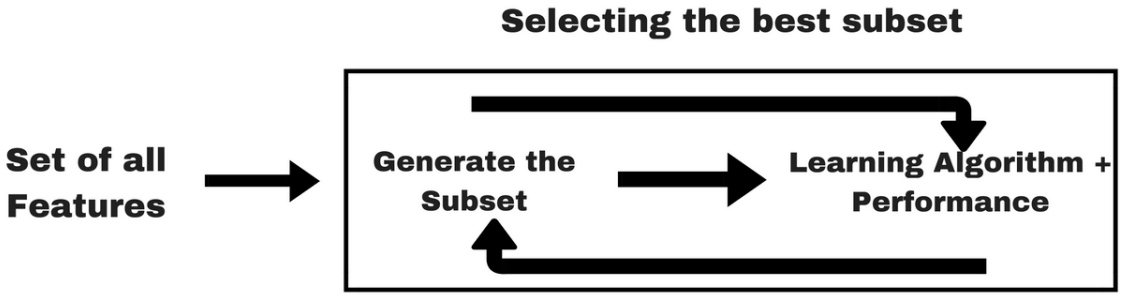


Figure 6: A general framework of wrapper feature selection methods.

https://arxiv.org/abs/1601.07996

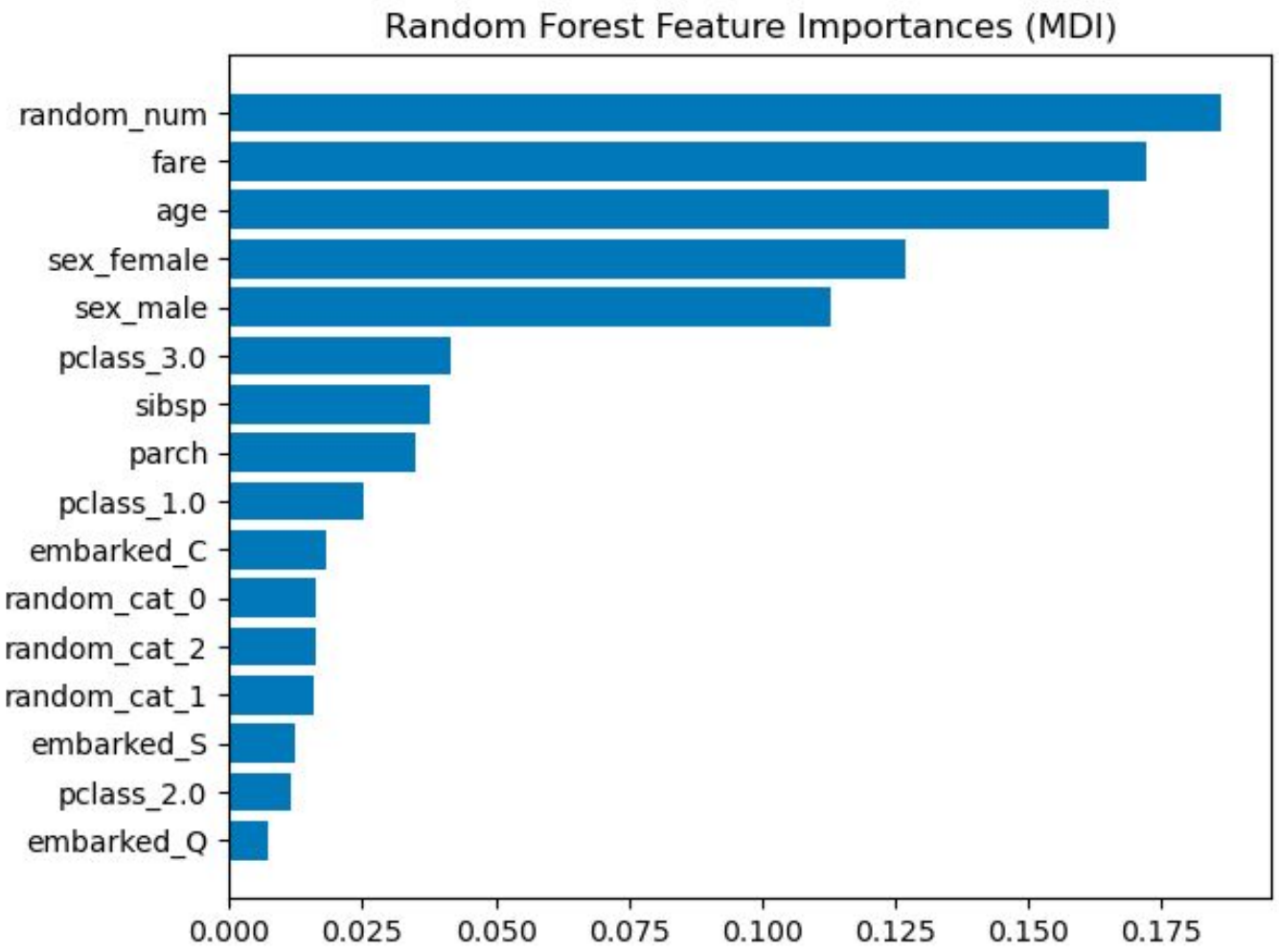
Embedded Method



- Algorithms with built-in feature selection methods
- Some techniques
  - Feature Importance
  - Lasso regression
    - L1 regularization adds a penalty equivalent to absolute value of the magnitude of coefficients.
  - Ridge regression
    - L2 regularization adds a penalty equivalent to square of the magnitude of coefficients.

https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/

Random Forest – Feature Importance



https://machinelearningmastery.com/calculate-feature-importance-with-python/

Embedded Usage

- Trade-off solution between filter and wrapper
  - Embed feature selection with model learning
    - More efficient than wrapper
  - Most widely used embedded methods are regularization models
    - Minimize fitting errors
    - Feature coefficients tend to zero

https://arxiv.org/abs/1601.07996

Model Selection and Evaluation

http://scikit-learn.org/stable/model_selection.html#model-selection

Overview

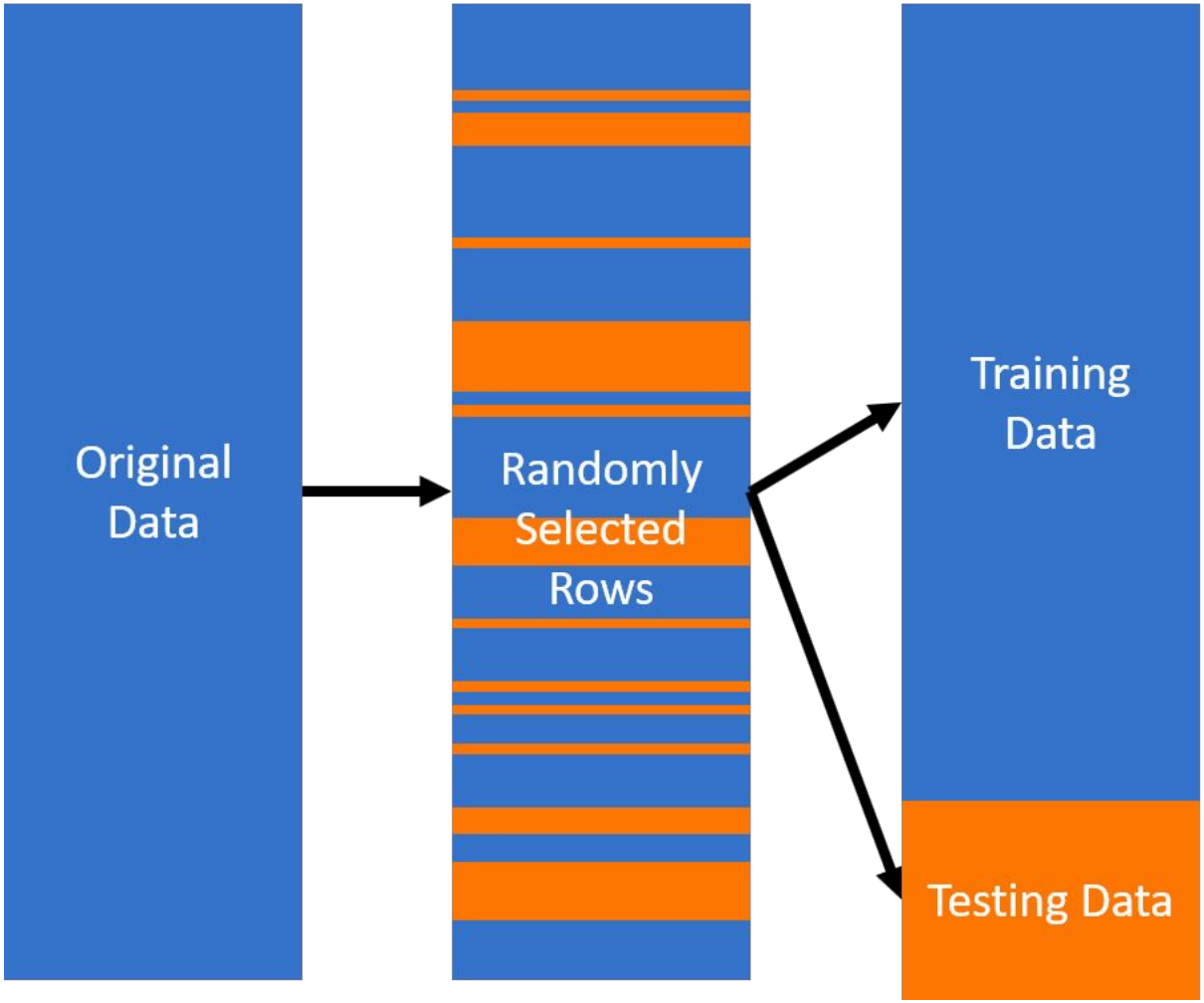
- Feature/Variable Selection
- Model Selection
- Evaluation – Cross Validation
- Improve Models – Regularization
  - Ensembles

Define the Problem

- **Step 1: What is the Problem?**
  - Describe the problem like you talking with a friend or colleague
  - State the problem as if you are a consultant presenting to a client
  - Decompose the problem
    - Set of data mining tasks
    - Data Sources
    - How to measure performance
    - Assumptions
  - Check for solutions to similar problems
- **Step 2: Why does the problem need to be solved?**
  - Significance of solution; solution benefits
  - How will the solution be used
- **Step 3: What is your detailed plan for solving the problem?**

https://machinelearningmastery.com/how-to-define-your-machine-learning-problem/

Splitting Data for Machine Learning



Train & Test

- Training Dataset and Test Dataset
  - Select a test set and a training set from transformed data
  - Train on training dataset
  - Evaluate against test set
  - A trained model is not exposed to the test dataset during training so that predictions made on the test dataset are designed to be indicative of the performance of the model in general.
  - Want to make sure the selection of your datasets are representative of the problem you are solving.

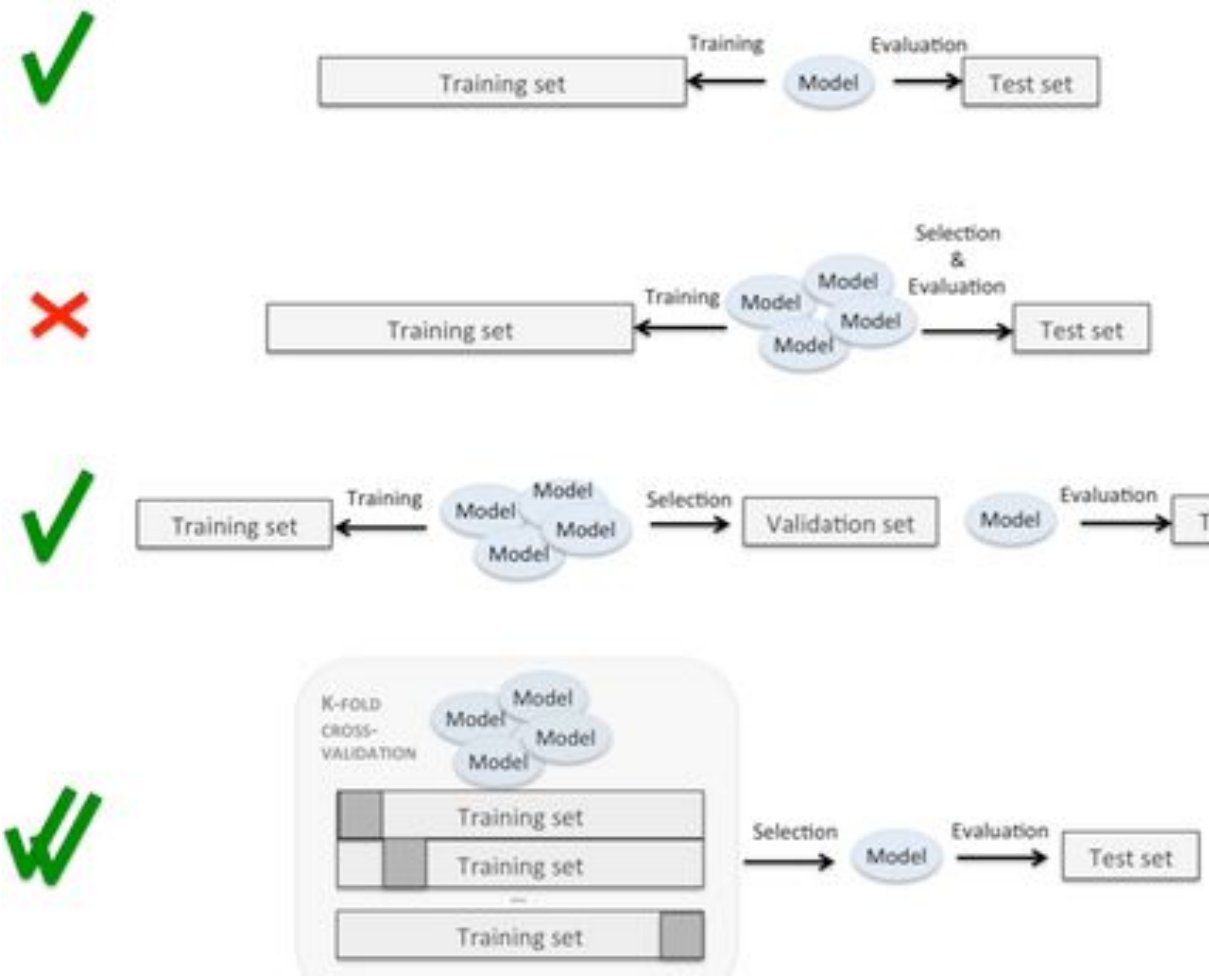
- Cross Validation
  - Use the entire transformed dataset to train and test an algorithm
  - Separate the dataset into a number of equally sized groups of instances (called folds
  - Repeat process so that each fold can be left out and act as the test dataset.
  - Average performance measures across all folds to estimate algorithm performance

https://machinelearningmastery.com/how-to-evaluate-machine-learning-algorithms/

Cross Validation

Evaluate Model

Test/Training Dataset Decisions



https://github.com/rasbt/python-machine-learning-book/blob/master/faq/evaluate-a-model.md

Holdout Method

- Data randomly partitioned into two sets
  - Training set (ex. 2/3); Test set (ex. 1/3)
  - Training set used to fit the model
  - Test set used to estimate prediction error for model selection

- Accumulate errors to provide a mean absolute test set error
- Evaluation may depend heavily on split
  - data points in the training set vs. data points in the test set
  - Evaluation may be significantly different, depending on the division

- Preferable to the residual method
- Can have a high variance

https://www.youtube.com/watch?v=SMvNwg9vraM
https://www.cs.cmu.edu/~schneide/tut5/node42.html

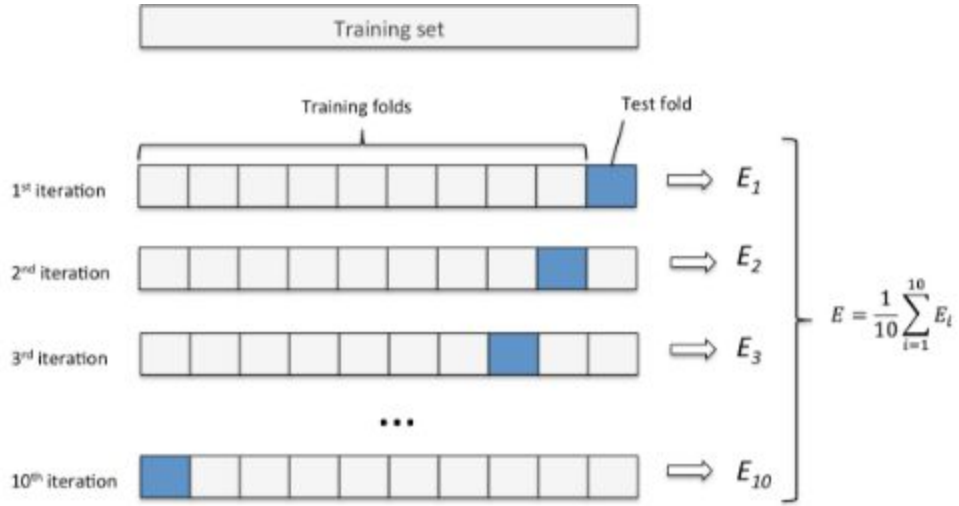
K-fold Cross Validation

- Divide dataset into *k* subsets, and repeat holdout method *k* times
- Each iteration, one of the *k* subsets is used as the test set and the other *k-1* subsets are put together to form a training set
- Then compute the average error across all *k* trials

- Advantages
  - Less impact of how data is divided
    - Every data point is in a test set exactly once
    - and is in a training set *k-1* times
  - Variance of the resulting estimate is reduced as *k* is increased.
- Disadvantage
  - training algorithm must be rerun from scratch *k* times
    - takes *k* times as much computation to make an evaluation
- Variant method
  - randomly divide the data into a test and training set *k* different times.
  - advantage
    - independently choose size of each test set and number of trials average over

https://www.cs.cmu.edu/~schneide/tut5/node42.html

K-fold Cross Validation Iterations



https://github.com/rasbt/python-machine-learning-book/blob/master/code/ch06/ch06.ipynb

What value should be chosen for K?

- K = N
  - cross-validation estimator is approximately unbiased for the true (expected) prediction error
  - but can have high variance because the N “training sets” are so similar
- K = 5
  - cross-validation has lower variance
  - but bias could be a problem, depending on how performance of the learning method varies with size of the training set
- If the learning curve has a steep slope at the given training set size, five or ten fold cross-validation will overestimate the true prediction error.
  - Whether this bias is a drawback in practice depends on the objective
- Five- or tenfold cross-validation are recommended as a good compromise

Fold k – Test Dataset

1. Divide the samples into K cross-validation folds (groups) at random.
2. For each fold k = 1, 2, . . . ,K
   - Find a subset of “good” predictors that show fairly strong (univariate) correlation with the class labels, using all of the samples except those in fold k.
   - Using just this subset of predictors, build a multivariate classifier, using all of the samples except those in fold k.
   - Use the classifier to predict the class labels for the samples in fold k

Regularization

Improve Model

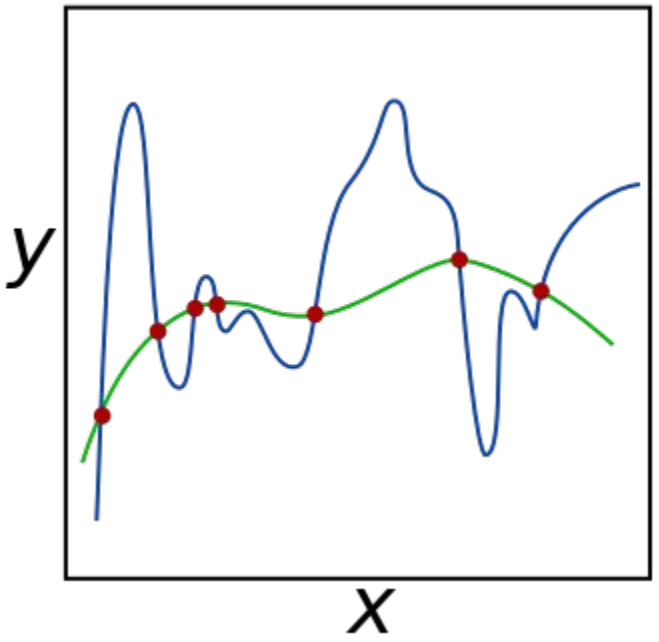
Chapter 6: Linear Model Selection and Regularization

Regularization

- Introduce additional information to solve an ill-posed problem or to prevent overfitting
- Classification
  - **Regularization term** is added to a loss function

$$\min_f \sum_{i=1}^{n} V(f(x_i), y_i) + \lambda R(f)$$

The green and blue functions both incur zero loss on the given data points. A learned model can be induced to prefer the green function, which may generalize better to more points drawn from the underlying unknown distribution by adjusting lambda, the weight of the regularization term.



https://en.wikipedia.org/wiki/Regularization_(mathematics)

Ridge Regression

- Use when independent variables are highly correlated
  - Multi-collinearity
    - Unbiased least squares estimates (OLS)
    - Large variances
      - observed value far from the true value.
- Add bias to regression estimates
  - reduces standard errors
  - shrinkage parameter λ

- First component
  - least square term
- lambda component
  - summation of β2 (beta- square) where β is the coefficient
  - added to least square term to shrink the parameter to low variance

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/

Ridge Regression Points

- Assumptions are the same as least squared regression
  - except normality is not to be assumed
- Shrinks coefficients values
  - but doesn't reaches zero, which suggests acts like no feature selection feature
- L2 regularization method
  - https://en.wikipedia.org/wiki/Regularization_(mathematics)
- Stability under rotation

https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/

Lasso Regression

- Lasso (Least Absolute Shrinkage and Selection Operator)
- Penalizes the absolute size of the regression coefficients
  - To reduce variability
  - To improve accuracy of linear regression models
- Differs from ridge regression in that absolute values are the penalty function instead of squares.
  - Some parameter estimates can be exactly zero.
  - The larger the penalty applied, the further estimates get shrunk towards absolute zero
    - Results in variable selection

https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/

Lasso Regression Points

- Assumptions are the same as least squared regression
  - except normality is not to be assumed
- Shrinks coefficients to zero (exactly zero)
  - feature selection
- L1 regularization method
- If group of predictors are highly correlated, lasso picks only one of them and shrinks the others to zero

https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/

ElasticNet Regression

- Hybrid of Lasso and Ridge Regression techniques
- Trained with L1 and L2 prior to regularizer
- Useful when there are multiple features that are correlated
  - Lasso is likely to pick one of these at random, while elastic-net is likely to pick both.

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\|y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1).$$

THIS CONCLUDES THE PRESENTATION.

Thank you.

UNIVERSITY OF MINNESOTA

Driven to Discover℠