Dimensionality Reduction and Feature Selection

Astrini Sie

asie@seattleu.edu

Week 5b

ECEGR4750 - Introduction to Machine Learning
Seattle University

October 19, 2023

Recap and Updates

- Lab Take Home Assignment due today at 11.59pm
- Office Hours
  - T, Th 12-1p at Bannan 224
  - W 7-9p via Zoom
  - F 9-9.45a via Zoom
- Zoom Link: `https://seattleu.zoom.us/j/7519782079?pwd=cnhCM2tPcHJKVWwxZVArS2VHSUNJZz09`
  - Meeting ID: 751 978 2079
  - Passcode: 22498122
- Pick your paper and paper presentation date by 11.59pm today or I will assign a slot for you!

- Evaluation Metrics
  - Benchmark metrics
  - Pay attention to real world applications and real world metrics!

Overview

1. The curse of dimensionality

2. Feature Selection
   - Filter Methods
   - Wrapper Methods
   - Embedded Methods

3. Dimensionality Reduction
   - Linear Algebra Based Methods
   - Deep Learning Based Methods

The curse of dimensionality

Figure: Redfin Listing

- of Bedrooms: 3
- of Beds Upper: 2
- of Bedrooms Main: 1
- of Full Baths: 2
- of Three Quarter Baths: 1
- of Full Baths Upper: 1
- of Full Baths Main: 1
- of Three Quarter Baths Upper: 1
- of Three Quarter Baths Main: 0
- of Bathtubs: 2
- of Showers: 1
- Has Fireplace
- of Fireplaces: 1
- Fireplace Features: Gas
- of Fireplaces Main: 1
- Has Heating
- Heating Information: 90%+ High Efficiency, Radiant
- Cooling Information: None
- Interior Features: Ceramic Tile, Concrete, Wall to Wall Carpet, Fireplace
- Appliances: Dishwasher, Dryer, Disposal, Refrigerator, Stove/Range, Washer
- Flooring: Ceramic Tile, Concrete, Engineered Hardwood, Carpet
- Appliances Included: Dishwasher, Dryer, Garbace Disposal, Refrigerator, Stove/Range, Washer
- Energy Source: Electric, Natural Gas
- Sq. Ft. Finished: 2,150
- Style Code: 12 - 2 Story
- Property Type: Residential
- Property Sub Type: Residential
- Has View
- Elementary School: Madrona Elementary
- Middle Or Junior High School: Meany Mid
- High School: Garfield High
- High School District: Seattle
- Living Area: 2,150
- Living Area Units: Square Feet
- Calculated Square Footage: 2150

The curse of dimensionality

Figure: Redfin Listing

- # of Bedrooms: 3
- # of Beds Upper: 2
- # of Bedrooms Main: 1
- # of Full Baths: 2
- # of Three Quarter Baths: 1
- # of Full Baths Upper: 1
- # of Full Baths Main: 1
- # of Three Quarter Baths Upper: 1
- # of Three Quarter Baths Main: 0
- # of Bathtubs: 2
- # of Showers: 1
- Has Fireplace
- # of Fireplaces: 1
- Fireplace Features: Gas
- # of Fireplaces Main: 1
- Has Heating
- Heating Information: 90%+ High Efficiency, Radiant
- Cooling Information: None
- Interior Features: Ceramic Tile, Concrete, Wall to Wall Carpet, Fireplace
- Appliances: Dishwasher, Dryer, Disposal, Refrigerator, Stove/Range, Washer
- Flooring: Ceramic Tile, Concrete, Engineered Hardwood, Carpet
- Appliances Included: Dishwasher, Dryer, Garbace Disposal, Refrigerator, Stove/Range, Washer
- Energy Source: Electric, Natural Gas
- Sq. Ft. Finished: 2,150
- Style Code: 12 - 2 Story
- Property Type: Residential
- Property Sub Type: Residential
- Has View
- Elementary School: Madrona Elementary
- Middle Or Junior High School: Meany Mid
- High School: Garfield High
- High School District: Seattle
- Living Area: 2,150
- Living Area Units: Square Feet
- Calculated Square Footage: 2150

The curse of dimensionality

The feature vector is sparse if many entries are zero.

What are the relevant dimension to make a prediction?

How do we find the best subset among all possible?

The curse of dimensionality

When the input space is large, we face some issues:

1. **Overfitting**. When we have more features and samples, the ability of the model to learn from the data and generalize to new unseen data diminishes.
2. **Occam's Razor**. Simpler models are better: easier / faster to train, smaller size.
3. **Garbage In Garbage Out**. Some features could be irrelevant, and including them in training would result in the model learning irrelevant information.

The curse of dimensionality

Dimensionality Reduction vs. Feature Selection

- **Feature selection.** Identify and remove unneeded, irrelevant and redundant attributes from data that do not contribute to the accuracy of a predictive model or may in fact decrease the accuracy of the model. (e.g. From 10 features, pick top 2)
  Reduces the complexity of the model, and a simpler model is simpler to understand and explain.
- **Dimensionality reduction.** Projecting features into a lower-dimensional feature space. Transforming features or combination of features into different forms or representations that are less complex than their original forms. (e.g. Encoding all lower body joint angles into one or two values)

Dimensionality Reduction Techniques

1. Start by using domain knowledge.
2. Normalize and standardize data.
3. Apply appropriate dimensionality reduction techniques

Some common techniques:

1. Feature Selection Methods
   - Filter methods
   - Wrapper methods
   - Embedded methods
2. Dimensionality Reduction (future lecture)
   - Linear Algebra Based (matrix factorization, SVD, PCA)
   - Manifold Learning
   - Autoencoder

Feature Selection

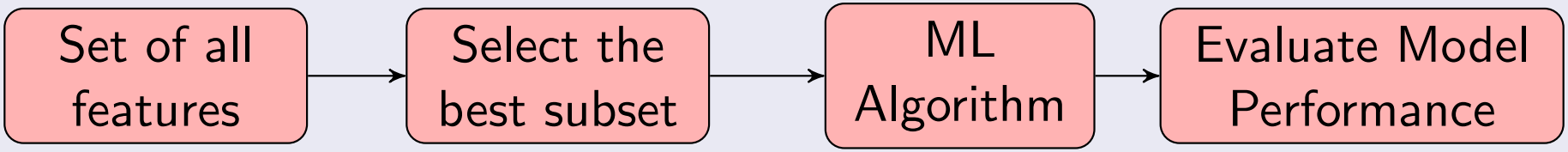
Filter Methods

Filter Methods

Feature Selection

Statistical methods to infer intrinsic feature attributes.

Filter Methods

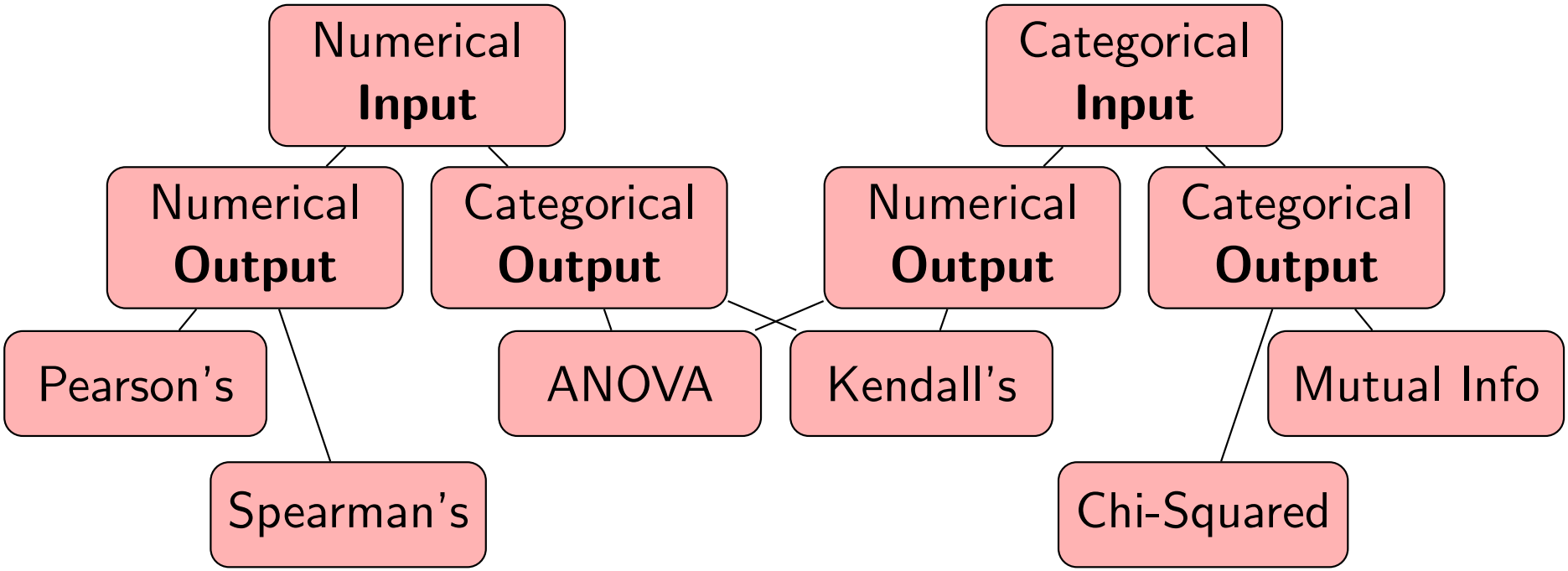


Simple, easy, and straightforward, but might not yield the best results.

Filter Methods: Correlation

Feature Selection

Uses statistical methods to measure **correlation** among features and outputs. The types of correlation to be evaluated depends on the types of inputs and outputs:



Filter Methods: Scoring of Each Feature

Feature Selection

1. **Fisher's Score**. Selects each feature independently according to their scores under Fisher criterion leading to a suboptimal set of features. Higher Fisher score is better.
2. **Variance Threshold**. Features with variance beyond a certain threshold are removed. Higher variance features are better as they likely contain more information.
3. **Mean Absolute Difference**.
4. **Dispersion Ration**. The Arithmetic Mean to the Geometric Mean of a given feature. Higher dispersion ration means a more relevant feature.
5. **Relief**. Measures the quality of features by randomly sampling an instance from the dataset and updating each feature

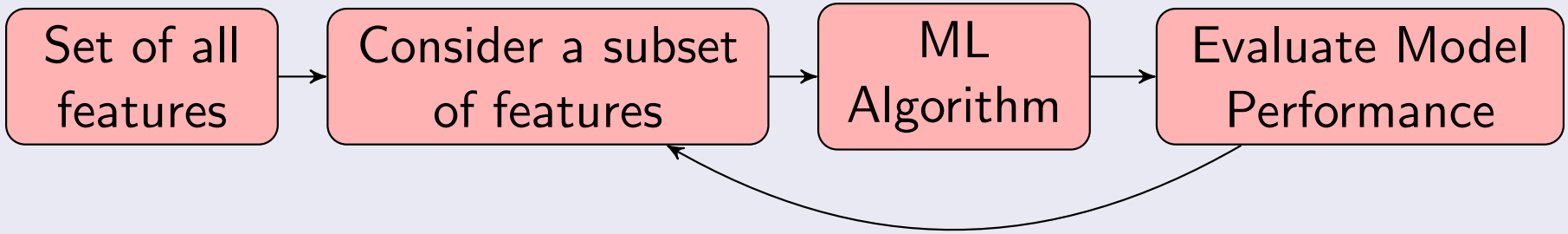
Wrapper Methods

Wrapper Methods

Feature Selection

Search problem where different combinations of features are prepared, evaluated, and compared to other combinations. The feature selection process is evaluated based on a specific machine learning algorithm.

Wrapper Methods



Outperform filter based approaches, but computational time can be costly especially when working with high dimensional datasets.

Wrapper Methods

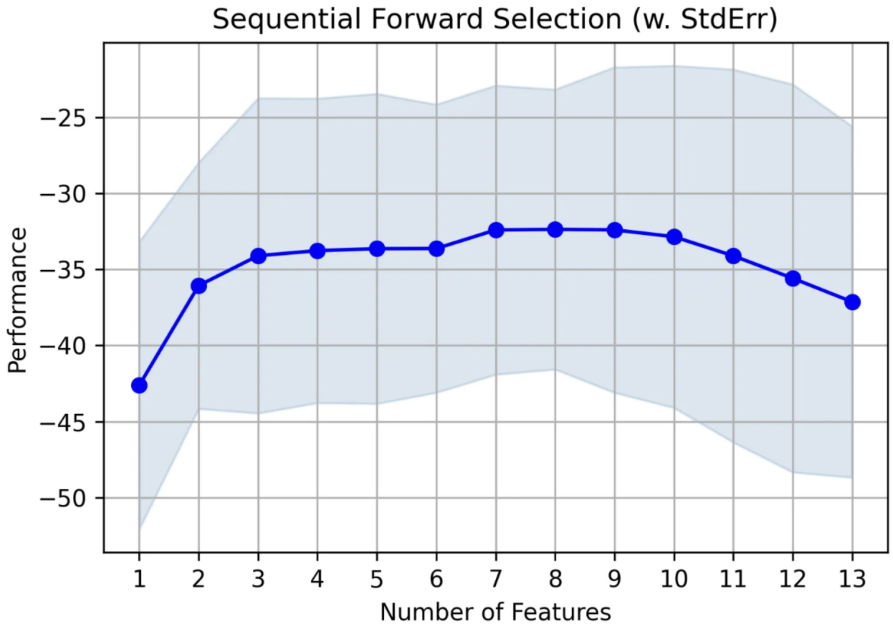


Figure taken from Towards Data Science

1. **Forward Selection**
   Start with empty feature set and iteratively add features to the set

Wrapper Methods

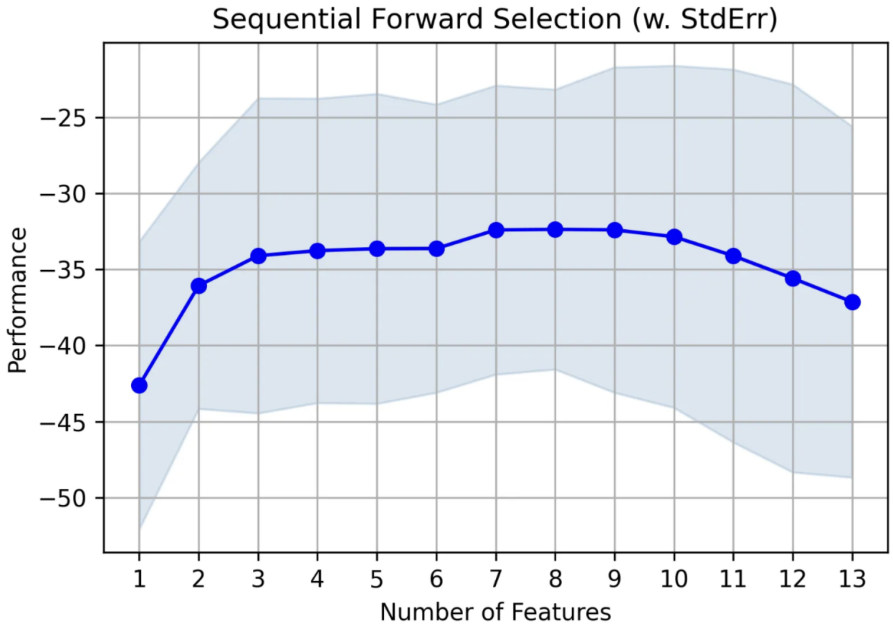


Figure taken from Towards Data Science

1. **Forward Selection**
   Start with empty feature set and iteratively add features to the set
   1. $T \leftarrow \emptyset$ where T : selected feature indices from $1, \ldots, d$
   2. For $j = 1, \ldots,$ do:
      Gradient descent
   3. $T \leftarrow T \cup j^*$

Wrapper Methods

1. **Forward Selection**
   Start with empty feature set and iteratively add features to the set
   1. $T \leftarrow \emptyset$ where T : selected feature indices from $1, \ldots, d$
   2. For $j = 1, \ldots,$ do:
      Gradient descent
   3. $T \leftarrow T \cup j^*$
2. **Backward Selection**
   Start with entire feature set and gradually eliminate features from the set.
3. **Bidirectional Elimination / Stepwise Selection**
   Uses both forward and backward selection techniques simultaneously.

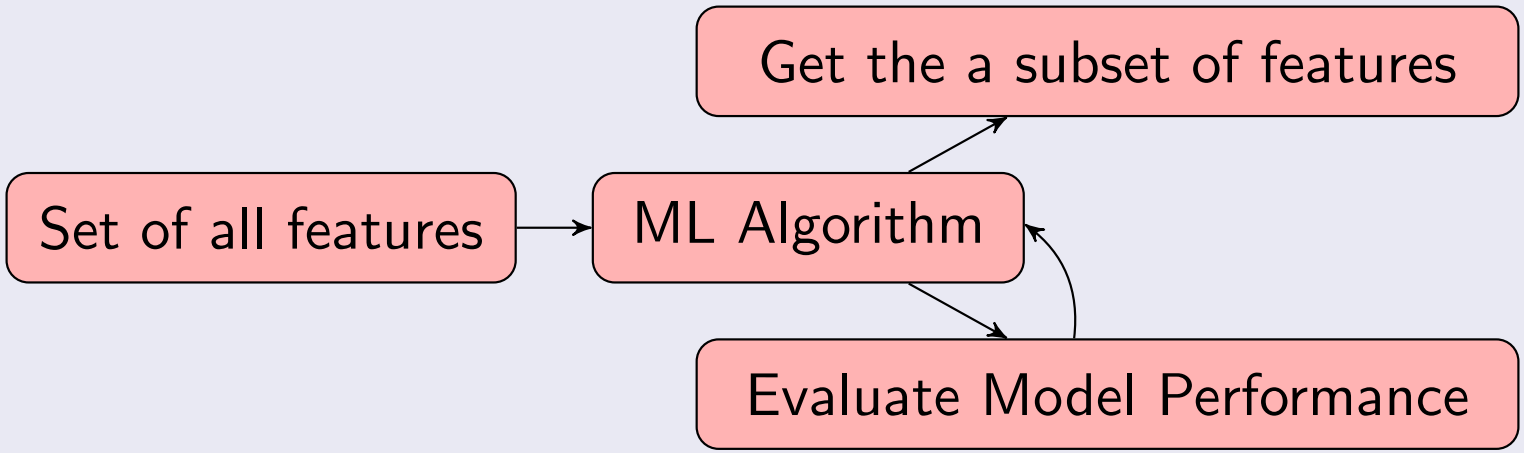
Wrapper Methods

4. **Exhaustive Feature Selection**
   Brute Force Approach. Compares performance of all possible feature subsets and chooses the best performing subset.
5. **Recursive Feature Elimination (RFE)**
   Greedy Optimization. Starts with the whole feature set and eliminates feature repeatedly depending on their relevance as judged by the learning algorithm

Embedded Methods

Feature selection algorithm is integrated as *part of the learning algorithm*. In the training process, the learning algorithm optimizes not only for the model parameters but also for the most important features.

Embedded Methods



More computationally effective than wrapper methods since there is no additional external feature selection procedure needed.

Embedded Methods: Regularization

Feature Selection

Applying penalty to coefficients of different parameters of a Machine Learning model to avoid overfitting. After the penalty, the coefficients can either be zero or very small.

Some examples of regularization methods:

1. L1 Regularization
2. L2 Regularization
3. ElasticNet

L1 Regularization (Lasso Regression)

- Penalizes the sum of absolute values of the coefficients
- Sets irrelevant features' coefficients to zero
- Might remove too many features in the model
- Gives a "sparse" solution

L1 Regularization Cost Function

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^{d} |\theta_j|$$

In the case of OLS or LMS, $\lambda = 0$ hence the blue term didn't exist.

*The cost function here is shown for the case of Linear Regression, but the L1 penalty term (in blue) can be implemented in other Cost Functions such as in Neural Networks.

L2 Regularization (Ridge Regression)

- Penalizes the sum of squared magnitudes of the coefficients
- Sets coefficients to be a very small value, but not zero
- Does not remove irrelevant features, only minimizes their impacts

L2 Regularization Cost Function

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^{d} \theta_j^2$$

In the case of OLS or LMS, $\lambda = 0$ hence the blue term didn't exist.

*The cost function here is shown for the case of Linear Regression, but the L2 penalty term (in blue) can be implemented in other Cost Functions such as in Neural Networks.

ElasticNet

- In Lasso regression, variable selection can be too dependent on data and thus unstable
- Combine the penalties of both lasso and ridge regression to get the best of both worlds

ElasticNet Cost Function

$$J(\theta) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda\left(\frac{1-\alpha}{2}\sum_{j=1}^{d}\theta_j^2 + \alpha\sum_{j=1}^{d}|\theta_j|\right)$$

In the case of OLS or LMS, $\lambda = 0$ hence the blue term didn't exist.

*The cost function here is shown for the case of Linear Regression, but the ElasticNet penalty term (in blue) can be implemented in other Cost Functions such as in Neural Networks.

Embedded Methods: Tree-Based Methods

Feature Selection

Methods in *Decision Tree* based algorithms such as **Random Forest**, **Gradient Boosting**, etc provides feature significance score for each feature as part of the tree-building process, which may be used to order features based on their importance.

Feature Selection Summary

Filter Methods	**Wrapper Methods**	**Embedded Methods**
Fast	High computation time for dataset with many features	Time complexity between Feature and Wrapper Methods
Less prone to overfitting	High change of overfitting because it involves training of modesl with different combinations of features	Used to reduce overfitting by penalizing coefficients of a model

https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/
https://www.kdnuggets.com/2023/06/advanced-feature-selection-techniques-machine-learning-models.html
https://medium.com/@creatrohit9/lasso-ridge-elastic-net-regression-a-complete-understanding-2021-b335d9e8ca3

References

Simon Du and Kevin Jamieson (2023)

CSE446 Machine Learning

University of Washington

Danny Butvinik

KDnuggets.com

Nate Rosidi (2023)

KDnuggets.com

Jason Brownlee

machinelearningmastery.com