Feature Selection in Text

Applied Text Mining

Dr. Maryam Movahedifar

4-6 December 2024

University of Bremen, Germany
movahedm@uni-bremen.de

Universität Bremen

DATA SCIENCE CENTER

Outline

Cross-Validation Method

Introduction to Feature Selection

Feature Selection Methods

Filter Method

Wrapper Method

Embedded Method

Principal Component Analysis (PCA)

Conclusion

Cross-Validation Method

Data Splitting and Cross-Validation

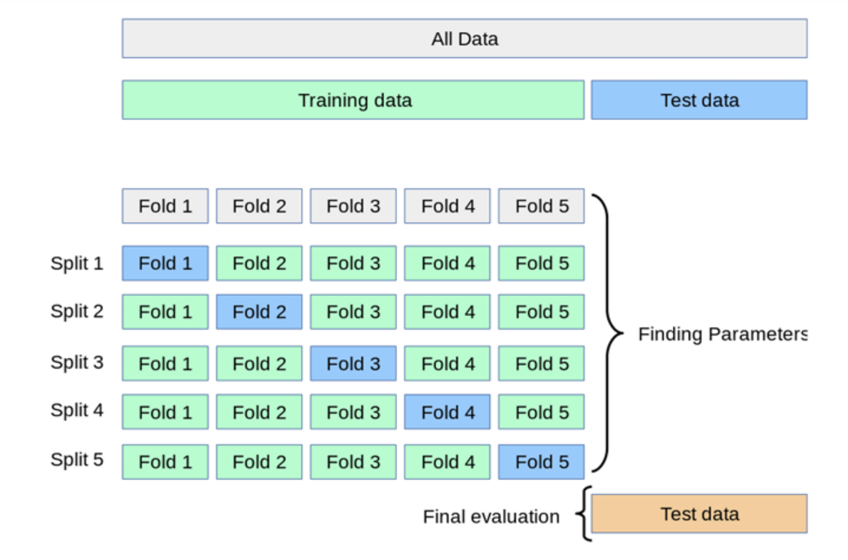
In supervised learning, data is divided into three main sets to train and evaluate models effectively:

- Training Set: Used to train the model, fitting the parameters based on the input data.
- Validation Set: A subset used to tune the hyperparameters of the model (e.g., architecture or learning rate).
- Test Set: Reserved for the final evaluation of the model's generalization ability, after training and validation.

Additionally, K-fold cross-validation is a common technique to assess model performance:

- The data is divided into K subsets (or folds).
- The model is trained on K-1 subsets and tested on the remaining one subset.
- This process is repeated for each of the K subsets, and the final performance metric is averaged over all K iterations.

Cross-Validation: K-Fold



All Data

Training data | Test data

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

Finding Parameters

Final evaluation { Test data

Introduction to Feature Selection

What is Feature Selection?

Feature selection involves identifying the most relevant features (variables) in a dataset for building predictive models.

- Reduces the dimensionality of the data.
- Improves model interpretability.
- Reduces overfitting by eliminating irrelevant features.

Why Feature Selection Matters

- High-dimensional text data can lead to computational inefficiency and overfitting.
- Selecting the right features improves model accuracy and performance.
- Helps in understanding the importance of specific features in prediction.

Feature Selection Example

Imagine you have a dataset with 10,000 fields (features), and you want to build a classifier to predict something, such as whether an email is spam.

- Goal: Cut down the number of features to a manageable size before applying machine learning.
- You need to reduce the 10,000 fields to 1,000 features.
- Question: Which 1,000 features should you choose?

This process of selecting the most relevant features is called **Feature Selection**.

Why Does Accuracy Reduce with More Features?

Adding more features to a model might seem beneficial, but it can often lead to reduced accuracy. This happens even when the original, important features remain in the model.

- Suppose the best feature set has 20 features.
- Adding 5 more features can unexpectedly reduce accuracy.
- But why? You still have the original 20 features!

Key Issue: Additional features often introduce noise.

Reasons Accuracy Reduces with More Features

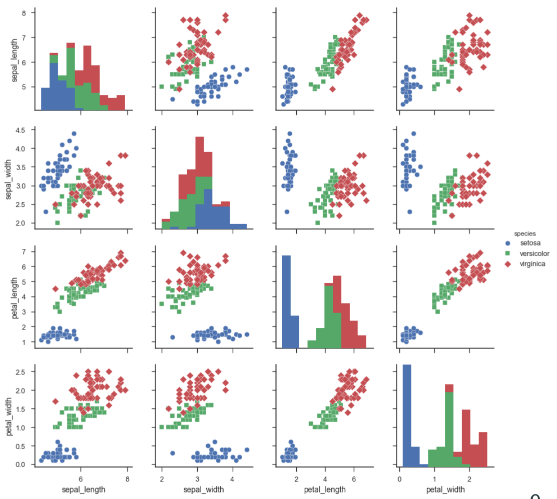
Adding too many features introduces complexity and noise, which can harm the model's ability to generalize.

- **Noise and Spurious Correlations:**
    - Additional features can create random patterns specific to the training data.
    - These patterns may not generalize well to the test data.
- **Increased Model Complexity:**
    - More features require learning more parameters (e.g., neural network weights or decision tree nodes).
    - A larger search space makes optimization more challenging.
- **Overfitting Risk:**
    - The model may fit the noise in the training data rather than the true patterns.

Feature Selection Example

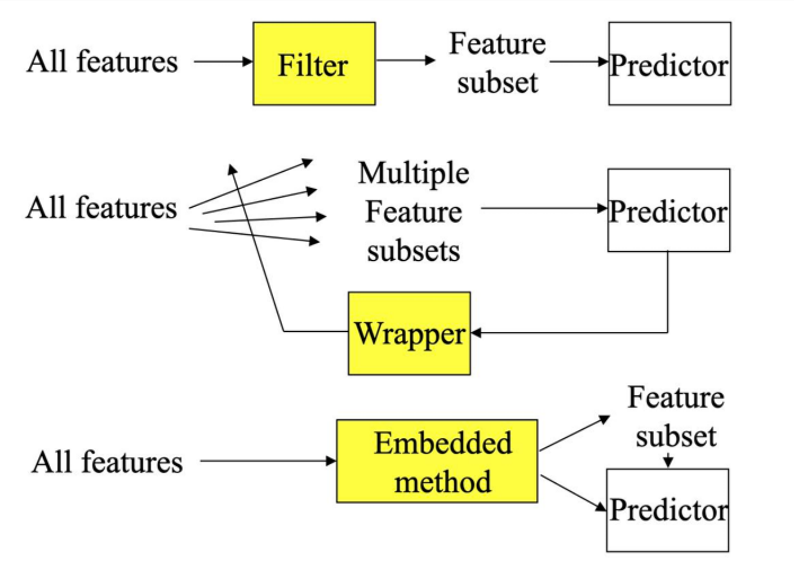
Feature selection involves identifying which features are most informative for distinguishing between classes in a dataset.

- Each feature pair is plotted against the others to show how they separate the three species (setosa, versicolor, virginica).
- Some features clearly separate the classes (e.g., petal length and petal width), while others provide less distinction (e.g., sepal length and sepal width).
- Feature selection retains only those features that significantly improve classification accuracy.

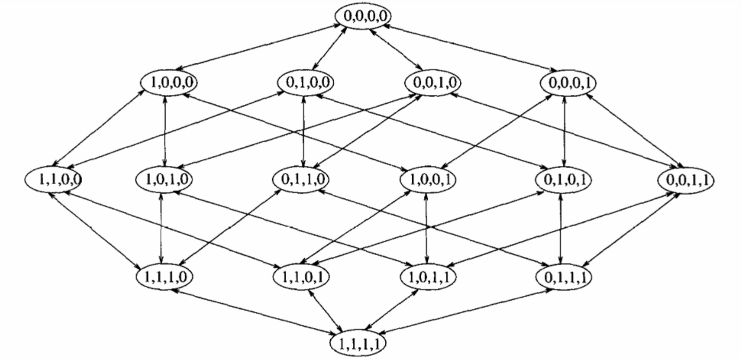


Feature Selection Methods

Feature Selection Methods Overview



Feature Subset Selection: State Space Search



For a dataset with N features, there are 2^N possible subsets of features. Each state represents a feature subset, and the nodes in the search space are connected based on the addition or deletion of a single feature. The search space is too large to exhaustively search for all possible subsets when N is large. Therefore, heuristic search methods are used to guide the search towards the optimal feature subset based on evaluation.

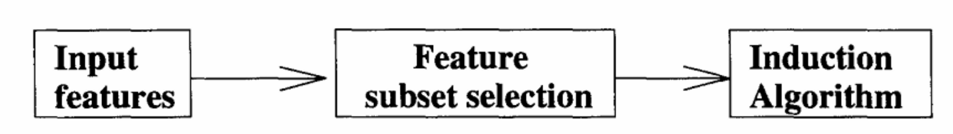
Filter Method

Filter-Based Feature Selection

Filter-based feature selection methods evaluate the importance of features independently from the predictive model using statistical tests. These methods are computationally efficient, making them suitable for large datasets. Below are some key metrics used in filter-based methods:

- Gini Index: Measures the purity of a feature split.
- Chi-Square (χ^2) Statistics: Evaluates the dependency between features and target categories.
- Mutual Information: Quantifies the relatedness between a feature and a class.
- Odds Ratio: Compares the likelihood of a term appearing with a class to its absence.
- Document Frequency: Helps eliminate rare terms that do not contribute to accurate predictions.

Visual Representation of Filter-Based Feature Selection



Key Concept: All features are evaluated independently of the predictive model to identify the most relevant feature subsets.

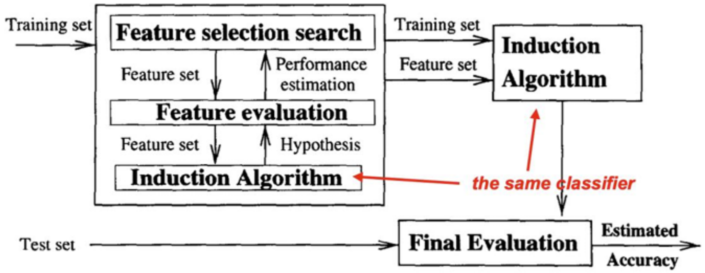
Wrapper Method

Wrapper Methods

Wrapper methods aim to find the best subset of features tailored for a specific classification method. These methods evaluate feature subsets by training a learning algorithm on each subset and selecting the one with the best performance.

- Optimizes for a specific learning algorithm.
- Features subsets are tested by training models and evaluating performance on validation data.
- Example: Recursive Feature Elimination (RFE).
- Computationally expensive due to testing many subsets of features.
- Impractical for large feature sets or text classification due to the NP-complete problem.

Wrapper Approach to Feature Subset Selection



R. Kohavi, G.H. John/Artificial Intelligence 97 (1997) 273-324

The wrapper approach treats the induction algorithm as a "black box" to evaluate different feature subsets. The feature selection algorithm searches for the best subset by evaluating the performance of the model on each subset. The subset that results in the highest evaluation is chosen for the final model, which is then tested on an independent test set.

Embedded Method

Embedded Method

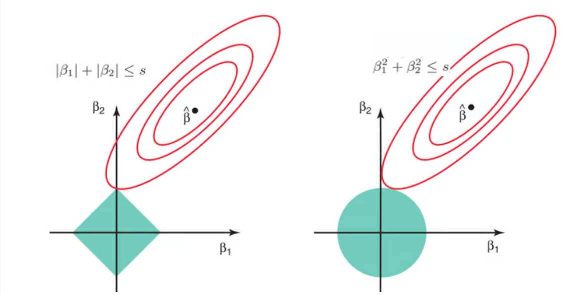
Overview: Many learning algorithms are cast into the minimization of a regularized functional:

$$\min_{\alpha} \hat{F}(\alpha, \sigma) = \min_{\alpha} \sum_{k=1}^{n} L(f(\alpha, \sigma \circ x_k), y_k) + \Omega(\alpha)$$

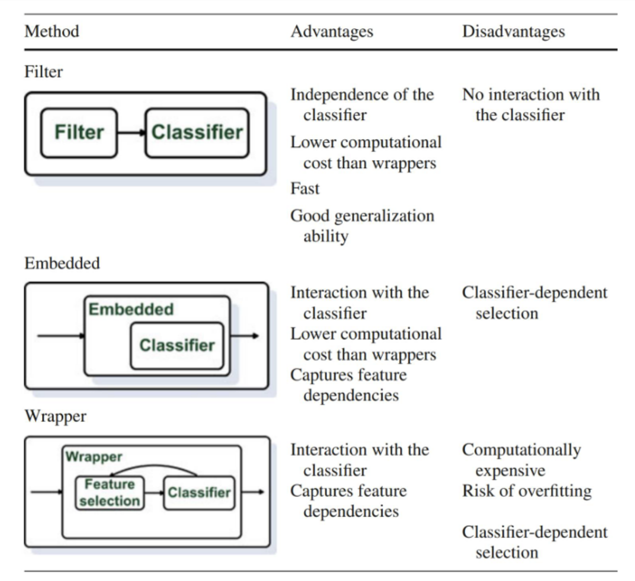
- Feature selection occurs during model training.
- Lasso regression (L1 regularization) shrinks some coefficients to zero, performing feature selection.
- Ridge regression (L2 regularization) shrinks coefficients but does not set them to zero.
- More computationally efficient than wrapper methods.

Lasso vs Ridge Regression

- Lasso Regression (L1 Regularization):
    - Shrinks some coefficients to zero.
    - Performs feature selection by removing less important features.
- Ridge Regression (L2 Regularization):
    - Shrinks coefficients but does not set any to zero.
    - Keeps all features in the model, though with reduced weights.



Comparison of Feature Selection Methods



Filter
Classifier
Embedded
Classifier
Wrapper
Feature selection
Classifier

Method	Advantages	Disadvantages
Filter	Independence of the classifier Lower computational cost than wrappers Fast Good generalization ability	No interaction with the classifier
Embedded	Interaction with the classifier Lower computational cost than wrappers Captures feature dependencies	Classifier-dependent selection
Wrapper	Interaction with the classifier Captures feature dependencies	Computationally expensive Risk of overfitting Classifier-dependent selection

Principal Component Analysis (PCA)

Introduction to PCA

What is PCA?

- Dimensionality Reduction: PCA reduces the number of features in the data.
- Linear Transformation: PCA combines n features into p components where $p < n$.
- Unsupervised Approach: It does not consider the output labels.
- Effective for Linear Correlations: Works well when data have linear relationships.

How PCA Works

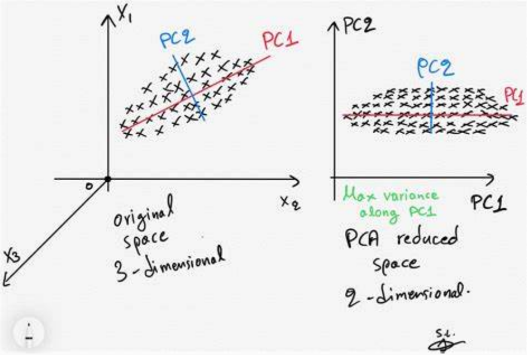
Steps in PCA:

- Calculate the covariance matrix to identify relationships between features.
- Compute the eigenvectors and eigenvalues to determine principal components.
- Project the data onto the principal components.
- Select the top p components to reduce dimensions.

Feature Selection vs. Feature Reduction

Overview: Handling high-dimensional data can be done through feature selection or feature reduction.

- Feature Selection: Retains the most important features.
    - Example methods: Wrappers (Forward Selection), Filters (Chi-Square), Embedded (Lasso).
- Feature Reduction: Combines original features into a smaller set of new features.
    - Example methods: PCA, LDA (Linear Discriminant Analysis).



Evaluation Metrics Overview

After training and validating a model, it is crucial to evaluate its performance using appropriate metrics. Common evaluation metrics include:

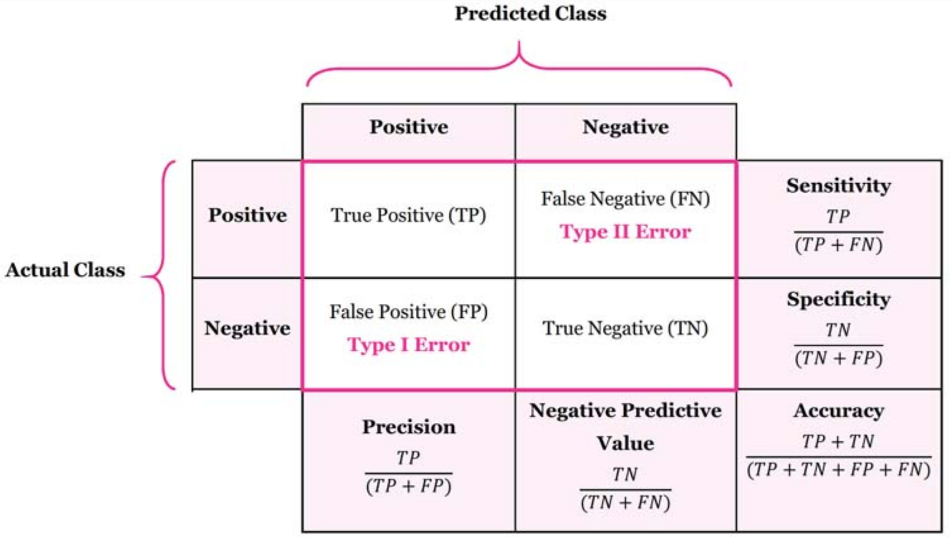
- Accuracy: The ratio of correctly predicted instances to the total instances.
- Precision: Of the predicted positives, how many are actually correct?
- Recall: Of the actual positives, how many were correctly predicted?
- F1-Score: A harmonic mean of precision and recall, balancing both metrics.

Confusion Matrix

A Confusion Matrix is a table used to evaluate the performance of a classification model by comparing actual vs. predicted labels:

- **True Positives (TP):** Correctly predicted positive instances.
- **False Positives (FP):** Incorrectly predicted positive instances.
- **True Negatives (TN):** Correctly predicted negative instances.
- **False Negatives (FN):** Incorrectly predicted negative instances.

Confusion Matrix Example



		Predicted Class		
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Conclusion

Conclusion

- Feature selection is crucial in text mining for building efficient and interpretable models.
- Various methods (filter, wrapper, embedded) serve different purposes.
- Apply feature selection based on your data and task requirements.

Programming

Practical 3