Feature Selection

Definition

Restrict the set of predictors
to a subset
$Q \subset P = \{x_1, \ldots, x_m\}$

This is **not the same as dimensionality reduction**, but a special form of it.

Goals of Feature Selection

(For supervised learning)

1. **Solve problems caused by certain predictors and model interactions**
    - SVMs and Neural Networks are sensitive to irrelevant features
    - Linear/logistic regression is sensitive to **correlated** predictors
2. **Reduce model complexity**
    - Lowers cost of acquiring additional data
    - Reduces training time
    - Improves model maintainability

Misconception

Filtering out uninformative features **does not guarantee** better interpretability.

With few samples and many features:

- There are many locally optimal subsets.
- A locally optimal solution says nothing about **true feature importance**.

Even a globally optimal solution doesn't mean we have perfect data.

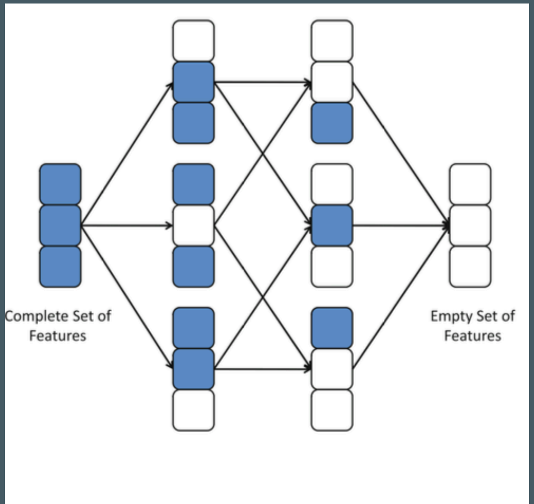
Central Questions

- How do I search for optimal feature subsets?
- How do I evaluate a given subset?
- Based on which principles do I add or remove features?

Combinatorics

With m features, there are 2^m combinations.

Example:
$m = 3 \Rightarrow 2^3 = 8$ subsets



Overview of Methods

Three major categories:

- **Intrinsic Methods**
- **Filters**
- **Wrappers**

Intrinsic Methods

Model identifies irrelevant features during training.

Examples:

- Decision trees: Unused features can be dropped
- Regularization: Coefficients are shrunk to zero (e.g., Lasso)

Filters vs. Wrappers

Filters:

- Select features before training
- Based on statistical tests
- Fast and simple

Wrappers:

- Train models repeatedly on different subsets
- Computationally intensive

Evaluation: Intrinsic Methods

Pros:

- Direct link to objective function
- Fast

Cons:

- Model-dependent
- Usually greedy selection

Evaluation: Filters

Pros:

- Identify individual predictor-target relationships well
- Easy to implement

Cons:

- Disconnected from model performance
- Usually evaluate features **independently**, missing interactions

Evaluation: Wrappers

Pros:

- Can test many combinations
- High potential to find optimal subsets

Cons:

- Very slow
- High-cost models (SVMs, NNs) often need FS the most
- High risk of overfitting

Filter Methods Deep Dive

Choose test based on variable types:

Categorical Predictor – Categorical Target

- Contingency table + χ^2 test
- If only two levels: Odds ratio

Categorical Predictor – Numeric Target

- ANOVA
- If two levels: t-test
- Area under ROC or PR curve (swapped roles)

Numeric Predictor – Categorical Target

- χ^2 test (swapped roles)
- Mutual Information Classifier

Numeric Predictor – Numeric Target

- Correlation
- Maximal Information Coefficient
- Generalized Linear Models

Mixed Predictor Types?

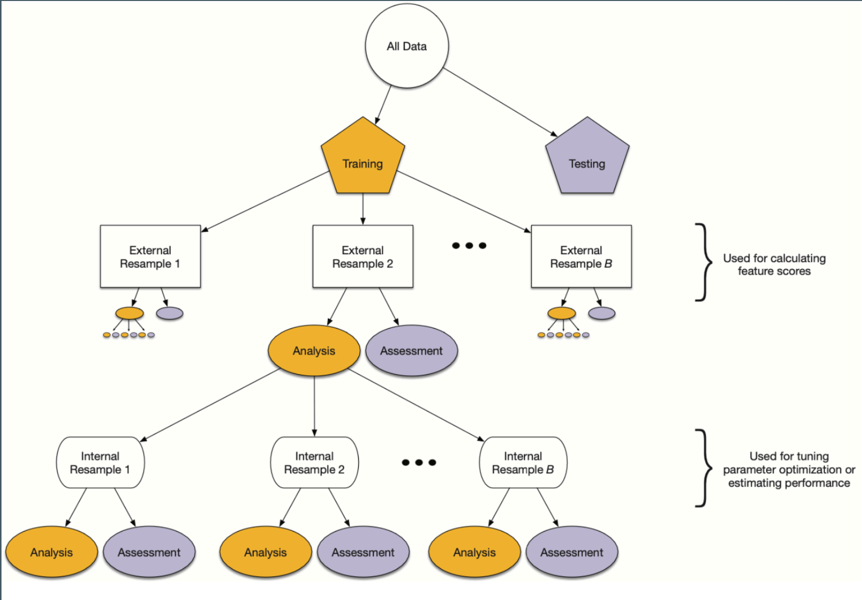
- Not always easy to compare (e.g., ROC AUC vs. t-test)
- Often all metrics are converted to **p-values**

But: p-values are prone to **false positives**

Avoiding False Positives

Resample a lot.

But: Hyperparameter tuning resampling must be done **after** feature selection



Exercise: Filters

Use `mutual_info_classif` from `sklearn.feature_selection`:

- Identify top 2 features for the Parkinson dataset
- Identify top 20% features using same method

Combined Methods

- Use an intrinsic model (e.g., linear model with L1 penalty) to drop irrelevant features
- Then train your final model on the reduced feature set

Exercise: Combined Methods

- Use `LinearSVC` with L1 penalty from `sklearn.svm`
- Use `.coef_` to extract top 10 features

Exercise: Combined Methods (II)

- Use `SelectFromModel` with `C=0.01`
- Use `.transform` to reduce the data

Exercise: Combined Methods (III)

Build a pipeline:

1. `SelectFromModel` with `LinearSVC`
2. Train a `RandomForestClassifier` on reduced data

How good is the confusion matrix on the test data?

Do not forget to standardize the test set.

Wrapper Methods Deep Dive

Advantage over `SelectFromModel`:

- Works for models without `.coef_` or `.feature_importances_`
- Uses **cross-validation score** to guide selection

Wrapper Hyperparameters

- Number of subsets to evaluate
- Size of subsets

Not implemented in scikit-learn by default.
More hyperparameters → more cross-validation → more training

Wrapper: Backward Selection

- Start with full feature set
- Remove one feature at a time based on CV score
- Stop when no improvement or desired number of features is reached

Exercise: Backward Wrapper

Use `SequentialFeatureSelector` from `sklearn.feature_selection`

- Reduce dataset to 150 features
- Use `RandomForestClassifier`

Stop execution once code runs – takes a long time

Wrapper: Forward Selection

- Start with empty feature set
- Add features one at a time based on CV score
- Stop when no improvement or enough features

Exercise: Forward Wrapper

- Use `SequentialFeatureSelector`
- Reduce to 2 features
- Use only 2 cross-validation folds
- Use `RandomForestClassifier`

Find out which attribute stores selected columns

Forward and Backward are Greedy

- Do not guarantee optimal subset
- May miss important features in early steps

Other Search Strategies

- **Exhaustive**: Try all subsets (impractical)
- **Stochastic Methods**: Simulated Annealing and Genetic Algorithms
    - Start with a random feature set and a fixed number of iterations
    - Randomly add new features or remove features
    - Accept changes based on a score with a tolerance