Tutorial: Feature Selection

Cheat Sheet and Practice Problems

ES335 - Machine Learning
IIT Gandhinagar

July 23, 2025

1 Summary from Slides

1.1 Baseline Models

Before applying sophisticated feature selection techniques, it's important to establish simple baseline models to compare against:

- **Mean Model**: $\hat{y} = \text{mean of training set}$
- **Median Model**: $\hat{y} = \text{median of training set}$
- **Mode Model**: $\hat{y} = \text{mode of training set}$
- **Random Model**: $\hat{y} \sim \text{Uniform}(\min(\text{training set}), \max(\text{training set}))$

These baselines help establish whether more complex feature selection methods provide meaningful improvements.

1.2 The Feature Selection Problem

When selecting the best subset of features from d available features, exhaustive enumeration considers all possible feature combinations. Each feature can either be included or excluded, leading to a binary choice table:

Feature_1	Feature_2	$\cdots$	Feature_d
True	False	$\cdots$	False
False	True	$\cdots$	False
True	True	$\cdots$	False
$\vdots$	$\vdots$	$\vdots$	$\vdots$
True	True	$\cdots$	True

This results in 2^d possible feature combinations, making exhaustive enumeration computationally prohibitive for large d.

1.3 Stepwise Forward Selection (SFS)

Forward selection is a greedy algorithm that starts with an empty feature set and iteratively adds the best feature:

Algorithm:

1. Initialize: $F = \{\}$ (empty feature set)
2. For $i = 1$ to K (desired number of features):

(a) $F_i = \text{argmin}_{\text{feature} \notin F} \text{Loss}(F \cup \{\text{feature}\})$

(b) $F = F \cup \{F_i\}$

Where Loss(features) denotes the loss incurred by the model trained with the specified features.

California Housing Example: The algorithm was applied to the California Housing Dataset to predict median selling price. Results showed:

Iteration	Added Feature	MSE
1	Median Income of block	0.97
2	Avg. number of rooms in the block	0.63
3	Latitude	0.65
4	Longitude	0.66

This example demonstrates that after the first two features, additional features provide minimal improvement or even degrade performance.

1.4 Stepwise Backward Selection (SBS)

Backward selection operates in the opposite direction of forward selection:

- Start with all features
- Iteratively remove the feature whose removal causes the least increase in loss
- Continue until desired number of features is reached

1.5 Time Complexity Analysis

Both forward and backward selection have $O(d^2)$ time complexity where d is the number of features.

For forward selection, the number of evaluations is:

$$\begin{aligned}
\text{Total evaluations} &= d + (d-1) + (d-2) + \cdots + 1 && (1)\\
&= \sum_{i=1}^{d} i && (2)\\
&= \frac{d(d+1)}{2} && (3)\\
&= O(d^2) && (4)
\end{aligned}$$

This quadratic complexity makes forward and backward selection much more tractable than exhaustive enumeration's $O(2^d)$ complexity.

2 Practice Problems

Exercise 1: Baseline Model Selection

For a regression dataset with training targets $y = [2.1, 3.5, 1.8, 4.2, 2.7, 3.1, 2.9, 3.8, 2.4, 3.6]$:

(a) Calculate the mean model prediction

(b) Calculate the median model prediction

(c) If the test set has targets $y_{test} = [2.8, 3.2, 2.5]$, compute the MSE for both baseline models

(d) Which baseline performs better on the test set?

Exercise 2: Exhaustive Search Complexity

You are working with different sized feature sets. Calculate the number of model evaluations required for exhaustive feature selection:

(a) Dataset with 5 features

(b) Dataset with 10 features

(c) Dataset with 20 features

(d) If each evaluation takes 2 seconds, how long would exhaustive selection take for each case?

(e) At what point does exhaustive selection become impractical (assume 1 day = reasonable time limit)?

Exercise 3: Forward Selection Algorithm Trace

Given a dataset with features $\{x_1, x_2, x_3, x_4\}$ and their individual performance when used alone:

- Feature x_1: MSE = 3.2
- Feature x_2: MSE = 2.1
- Feature x_3: MSE = 4.5
- Feature x_4: MSE = 2.8

Second iteration MSE values (adding to best single feature):

- $\{x_2, x_1\}$: MSE = 1.5
- $\{x_2, x_3\}$: MSE = 2.0
- $\{x_2, x_4\}$: MSE = 1.8

Show the complete forward selection trace for 2 iterations, explaining your feature choices at each step.

Exercise 4: Time Complexity Derivation

Derive the time complexity for backward selection:

(a) Starting with d features, how many models are evaluated in the first iteration?

(b) How many models in the second iteration?

(c) Write the general formula for total evaluations over all iterations

(d) Show that this leads to $O(d^2)$ complexity

(e) Compare the exact number of evaluations between forward and backward selection for $d = 6$

Exercise 5: California Housing Analysis

Based on the California Housing example from the slides:

Iteration	Added Feature	MSE
1	Median Income	0.97
2	Avg. rooms	0.63
3	Latitude	0.65
4	Longitude	0.66

(a) Why did the MSE increase from iteration 2 to 3?

(b) At which iteration should feature selection stop? Justify your answer

(c) What does this suggest about the importance of geographic features vs. economic features?

(d) How would you modify the stopping criterion to prevent overfitting?

Exercise 6: Forward vs Backward Selection Comparison

Consider a dataset with 4 features where forward selection gives the order $x_2 \rightarrow x_1 \rightarrow x_4 \rightarrow x_3$ and backward selection removes features in order $x_3 \rightarrow x_4 \rightarrow x_1 \rightarrow x_2$.

(a) Are these results consistent? Explain why or why not

(b) Which features appear to be most important according to both methods?

(c) In what scenarios might forward and backward selection give different rankings?

(d) Given computational constraints, which method would you choose and why?

Exercise 7: Greedy Algorithm Limitations

Forward selection is a greedy algorithm that makes locally optimal choices.

(a) Construct a simple example where forward selection fails to find the globally optimal feature subset

(b) Explain why this happens in terms of feature interactions

(c) What are the advantages of using greedy approaches despite this limitation?

(d) How does the $O(d^2)$ complexity compare to exhaustive search $O(2^d)$ for $d = \{5, 10, 15, 20\}$?

Exercise 8: Feature Interaction Effects

Consider features x_1, x_2, and x_3 with the following performance:

- Individual: MSE(x_1) = 5.0, MSE(x_2) = 4.5, MSE(x_3) = 6.0
- Pairs: MSE(x_1, x_2) = 4.2, MSE(x_1, x_3) = 3.0, MSE(x_2, x_3) = 4.1
- All three: MSE(x_1, x_2, x_3) = 2.8

(a) Trace through forward selection step by step

(b) Which feature combination would exhaustive search find as optimal?

(c) Does forward selection find the optimal solution in this case?

(d) What does this reveal about feature interactions?

Exercise 9: Stopping Criteria Design

Design appropriate stopping criteria for forward selection in different scenarios:

(a) **Scenario 1**: Limited computational budget - can only evaluate 20 models

(b) **Scenario 2**: Performance-based - stop when improvement is less than 5%

(c) **Scenario 3**: Cross-validation based - prevent overfitting on training set

(d) For each scenario, write the modified algorithm and explain the trade-offs

(e) Which stopping criterion would be most appropriate for the California Housing example?

Exercise 10: Algorithm Implementation

Implement the forward selection algorithm in pseudocode:

(a) Write detailed pseudocode including input parameters and return values

(b) Add appropriate error checking and edge cases

(c) Include provisions for different stopping criteria

(d) Modify your algorithm to track and return the MSE at each iteration

(e) How would you parallelize the feature evaluation step?

Exercise 11: Computational Scaling Analysis

Analyze how forward selection scales with dataset size:
Given:

- Dataset with n samples and d features
- Each model training takes $O(nd^2)$ time (assuming linear regression)
- Forward selection evaluates $O(d^2)$ models total

(a) What is the overall time complexity of forward selection?

(b) How does this compare to training a single model with all features?

(c) For $n = 10,000$ and $d = 50$, estimate the computational overhead

(d) At what ratio of n to d does forward selection become impractical?

Exercise 12: Real-World Application Design

Design a feature selection strategy for a real-world housing price prediction problem:
Dataset: 50,000 houses with features including:

- 20 continuous features (area, age, rooms, etc.)
- 15 categorical features (neighborhood, style, etc.)
- 10 derived features (price per sq ft, etc.)

(a) Would you use forward or backward selection? Justify your choice

(b) Design an appropriate baseline model for comparison

(c) What stopping criteria would you implement?

(d) How would you handle categorical features in your selection process?

(e) Outline a validation strategy to ensure reliable feature selection

Exercise 13: Advanced Complexity Analysis

Compare the theoretical and practical complexity of different approaches:

(a) For $d = 25$ features, calculate the exact number of model evaluations for:

- Exhaustive search
- Forward selection
- Backward selection

(b) If you can evaluate 1000 models per hour, how long would each method take?

(c) At what value of d does forward selection become faster than exhaustive search by a factor of 100?

(d) Derive the "break-even" point where $2^d = \frac{d(d+1)}{2}$

Exercise 14: Advanced Feature Selection Scenarios

Analyze challenging scenarios for stepwise selection:
Scenario A: Highly correlated features where x_1 and x_2 provide similar information **Scenario B**: Features that are only useful in combination (XOR-type relationships) **Scenario C**: Noisy features that occasionally appear useful due to random correlations

(a) For each scenario, predict how forward selection would behave

(b) Design synthetic datasets to test these scenarios

(c) What modifications to the basic algorithm could help handle these cases?

(d) How would cross-validation help identify and mitigate these issues?