Feature Construction and Selection

CS4780/5780 – Introduction to Machine Learning

Thorsten Joachims
Cornell University

Creating New Features

- Input Features:
  - Contain all information, but often not good for learning from directly.
  - Example: GRE, GPA, Country → PhD Admit
- Feature Construction:
  - Create new features that make it easier to learn from.
  - Example:
    - One-hot encoding of Country: "Germany" → $(0,0,1,0, \dots, 0)$
    - Percentile Features of GRE: GRE → $(\mathrm{X}_{GRE}^{10\%}, \dots, \mathrm{X}_{GRE}^{99\%})$
    - Percentile Features of GPA: GPA → $(\mathrm{X}_{GRE}^{10\%}, \dots, \mathrm{X}_{GRE}^{99\%})$
    - Pairwise features: Country x $(\mathrm{X}_{GRE}^{10\%}, \dots, \mathrm{X}_{GRE}^{99\%})$

Feature Selection

- Idea: Prune away irrelevant features to avoid overfitting.
- Approaches
  - Regularization and Margins
    - L2-Norm $\|\vec{w}\|_2$: Good when many features are relevant
    - L1-Norm $\|\vec{w}\|_1$: Good when only small subset of features is relevant
  - Feature scoring
  - Forward/Backward Selection

Feature Scoring

- Idea: Find features that are informative itself.
- Procedure
  - Sort features $X_1 \dots X_N$ by
    - $InformationGain(X_j, Y)$
    - $\text{Chi}^2(X_j, Y)$
    - $\text{ErrorReduction}(X_j, Y)$
    - Etc.
  - Pick top k feature and use those for learning
  - Determine best value of k via validation set / cross-validation.

Forward Selection

- Idea: Keep adding features that improve performance.
- Procedure
  - $\text{Avail} = \{X_1, \ldots, X_N\}$
  - $\text{Chosen} = \emptyset$
  - REPEAT
    - For $X_j \in Avail$
      - Train learner with features Chosen $\cup \{X_j\} \rightarrow h_j$
    - Find h_j with best validation set performance and add that X_j to Chosen. Remove that X_j from Avail.
  - UNTIL Avail = $\emptyset$
  - Pick h_j with best validation set performance overall.

Backward Selection

- Idea: Keep removing features that improve performance.
- Procedure
  - $\text{Chosen} = \{X_1, \ldots, X_N\}$
  - REPEAT
    - For $X_j \in Chosen$
      - Train learner with features Chosen $- \{X_j\} \rightarrow h_j$
    - Find h_i with best validation set performance and remove that X_j from Chosen.
  - UNTIL Chosen = Ø
  - Pick h_j with best validation set performance overall.

Summary

- Be creative in transforming and combining features → make learning easier for algorithm.
- Remove features that do not provide information to avoid overfitting.