LING 506 - TOPICS IN COMPUTATIONAL LINGUISTICS

Introductory Machine Learning

Yan Tang

Department of Linguistics, UIUC

Week 8

Last week…

- Naïve Bayes classification
  - Bayes' theorem calculates posteriori probability for an observation in a class:

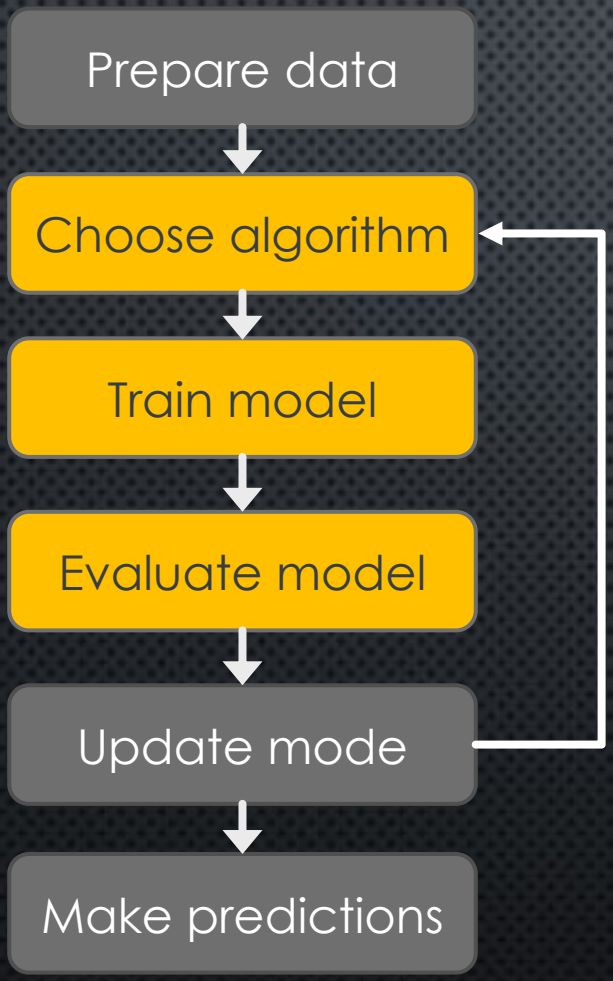
$$P(c|X) = \frac{P(X|c)P(c)}{P(X)}$$

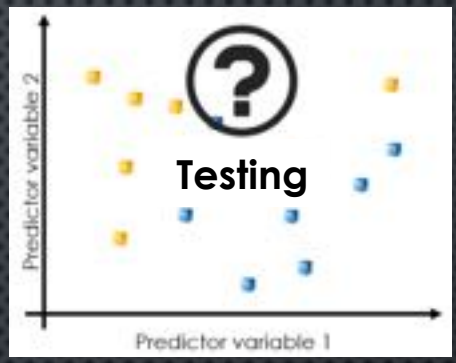
  - Assumption: features, $X = [x_1, \dots, x_n]$, are independent on each other, hence $P(X|c) = \prod_{i=1}^{n} P(x_i|c)$
- Naïve Bayes classifiers differ in the assumptions they make regarding the distribution of $P(x_i|c)$
- A good baseline model; few hyperparameters
- The basic assumption is too ideal

Last week…

- Support Vector Machines (SVM)
  - A discriminative algorithm to find decision boundaries between classes
  - Draws boundaries as far as possible from observations on the margin (i.e. support vectors); hyperparameter C
  - Non-linear problem: kernel tricks
  - Allows complex decision boundaries with limited number of features
  - Poor scalability

Model evaluation



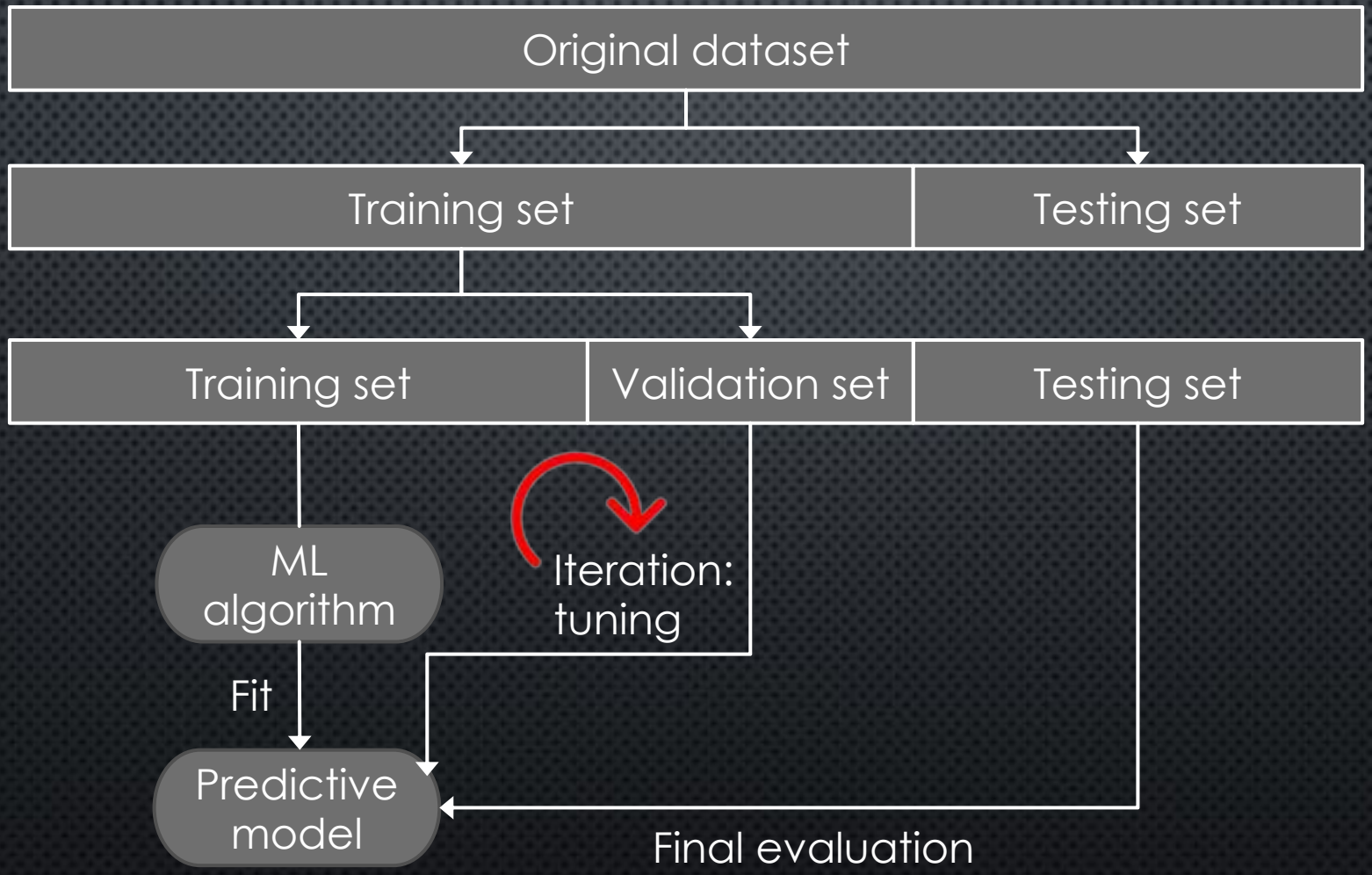


Method	Loss
KNN	0.283
Naïve Bayes	0.393
DTs	0.221
SVM	0.436

The holdout method

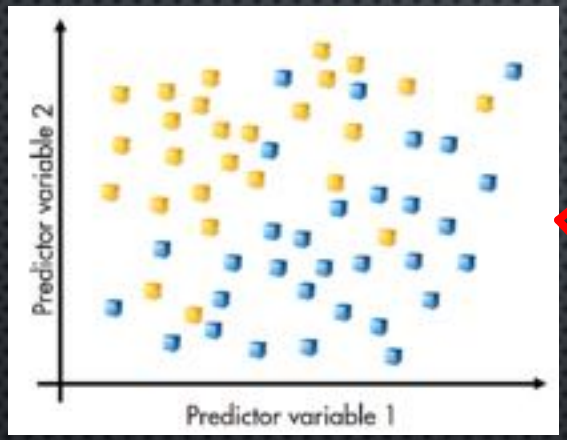
- The classic and popular approach for generalising performance of ML models
  - Splits the datasets into ***training*** and ***testing*** sets
- Model selection
  - Given a problem, select optimal values for parameters of a ML algorithm
  - Hyperparameter tuning
  - Issue of conducting model selection on testing data: overfitting
- A better holdout approach
  - ***Training*** set ⟹ ***training*** set and ***validation*** set

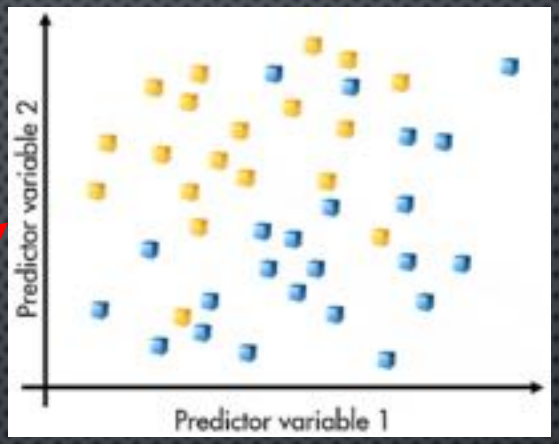
The holdout method



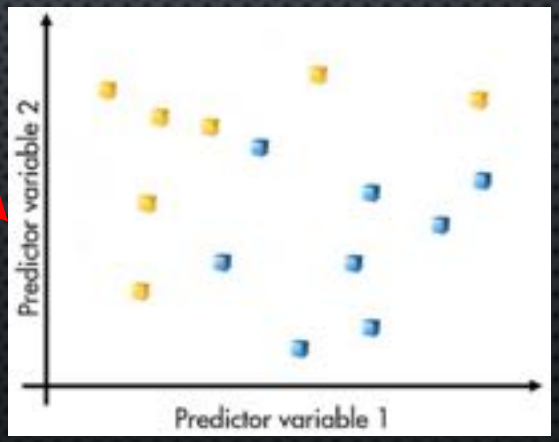
The holdout method: disadvantage

Original training set





Training set



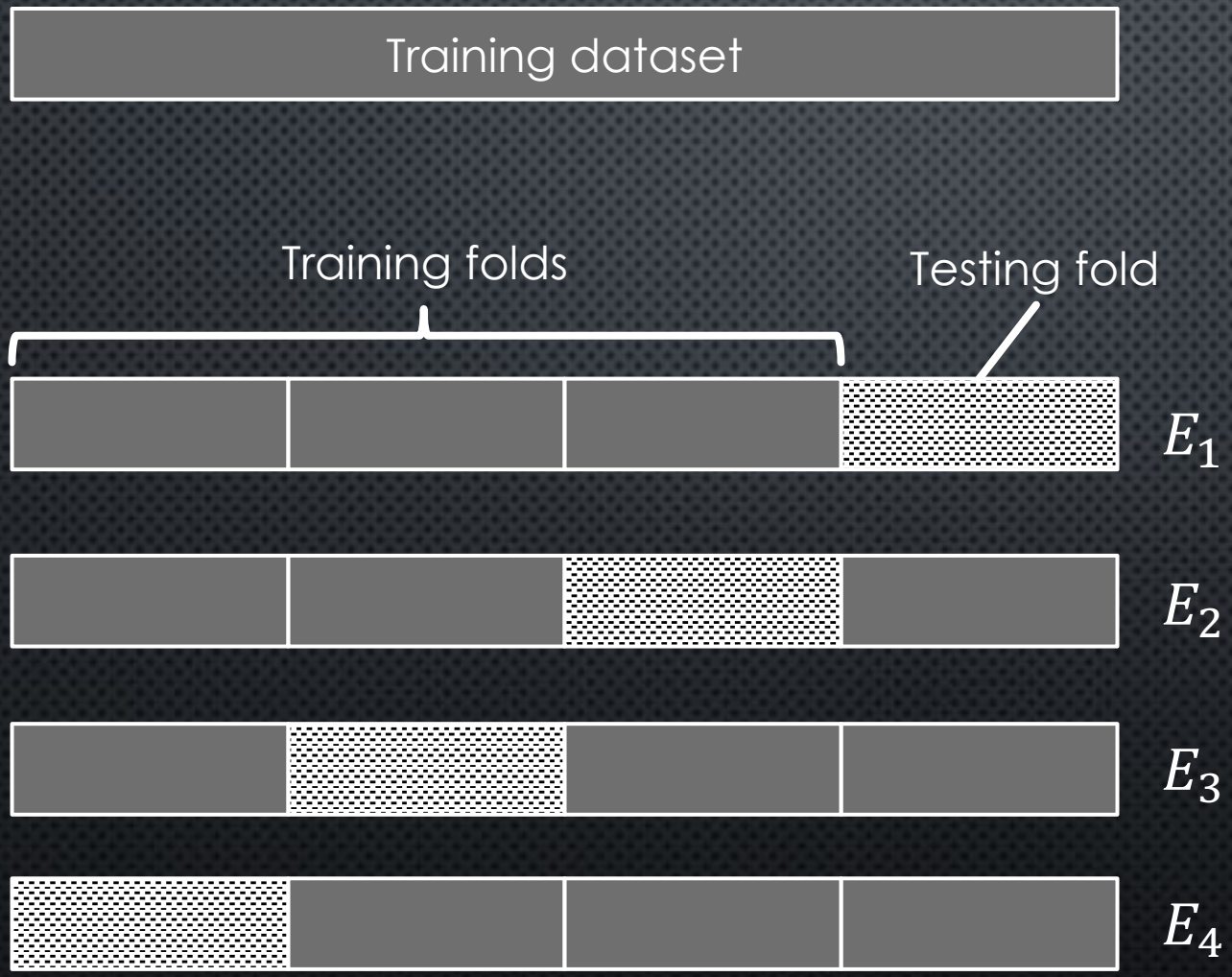
Validation set

- Performance may be dependent on how the datasets are partitioned

K-fold cross-validation

- Randomly splits the training data into k folds
  - Resampling without replacement
  - All the k subset have (approximately) same size
- During training:
  - $k - 1$ folds for training; 1 fold for validation
- Training is repeated k times, resulting in k models
  - Overall performance is the average across the models
  - Less sensitive to sub-partition

K-fold cross-validation



$$E = \frac{\sum_i^k E_i}{k}$$

K-fold cross-validation

- Use cross-validation to search for optimal values of hyperparameters
  - Train the model on the complete training data
- Rationale:
  - Providing more training samples to a ML algorithm usually leads to a more accurate and robust model

K-fold cross-validation: value for k

- How many folds should be used?
  - Empirically 10
- For small training sets:
  - Larger k : More training data, lower bias towards estimating the generalisation performance
  - Longer runtime of cross-validation
- For large training sets:
  - Smaller k: reduce the computational cost

K-fold cross-validation: advantages and disadvantages

- Advantages:
  - Each sample is used for training and testing only once, hence lower variance on model performance
  - It provides insights in how the model might perform in the worst and best scenarios with new data
  - Allows efficient use of data
- Disadvantage: computationally expensive
  - Approximately k times slower than doing single split of the data

Other cross-validation

- Stratified k-fold cross-validation
  - When the samples are ordered, e.g. [0, 0, 0, 1, 1, 1, 2, 2, 2]
  - It keeps the proportions between classes same in each fold as in the whole dataset
- Leave-one-out cross-validation
  - Each fold contains a single sample
  - May provide better estimate for small dataset; time consuming!
- Shuffle-split cross-validation
  - Allows for only engaging part of the data in each iteration
  - Suitable for sampling large datasets

Hyperparameter tuning

- Types of parameters in ML algorithm
  - Those learnt from training data, e.g. weights in neural networks and centers of Gaussians
  - Those of an algorithm that are optimised separately, e.g. K for KNN, C and γ for SVM, and node depth for Trees
- It is important to understand hyperparameters of a model
  - Type of parameter, e.g. Binary, categorical and numeric
  - Range of variable
  - Effect of each parameter
- Hyperparameter tuning is necessary for almost all models and problems

Grid search

- A brute-force exhaustive search paradigm
  - Specify a list of value for tuning hyperparameters
  - Train a model and evaluate the performance for each combination

- Examples:
  - For K in KNN, $K \in [1,2,3 \ldots N]$
  - E.g. For C and γ in SVM, $C \in [1,2,3 \ldots N]$ and $\gamma \in [0.01, 0.02, 0.03 \ldots Y]$

Simple grid search

- This can be implemented as "for" loops over N parameters for each combination:

```
split data into training and testing
C = a set of values
gamma = a set of values
for c in C:
    for g in gamma:
        initialise an algorithm with c and g
        fit the model on training data
        evaluate the model on testing data
        record the performance and current combination
optimal = the combination leading to the best performance
train model on training data with optimal
```

Grid search on validation set

```
split data into training data and testing data
split training into training2 and validation data
C = a set of values
gamma = a set of values
for c in C:
    for g in gamma:
        initialise an algorithm with c and g
        fit the model on training2 data
        evaluate the model on validation data
        record the performance and current combination
optimal = the combination leading to the best performance
train model on training data with optimal
```

Grid search with cross-validation

```
split data into training data and testing data
C = a set of values
gamma = a set of values
for c in C:
    for g in gamma:
        initialise an algorithm with c and g
        conduct cross-validation on training data for c and g
        compute the mean cross-validation score
        record the performance and current combination
optimal = the combination leading to the best performance
train model on training data with optimal
```

Randomised search

- What about when the number of hyperparameters and the search space are large?
- Randomised search
  - Only evaluates a given number of random combinations
  - May perform as well as Grid Search (Bergstra & Bengio, 2012)
- Advantages:
  - Allows for specifying number of iterations to control computing resources
  - Computationally efficient for large data set