ML2012 Final Project

TAs: Wei-Yuan Shen, Han-Jay Yang and Ching-Pei Lee
Instructor: Hsuan-Tien Lin

2012.12.3

Outline

- Data Introduction
- Evaluation Criterion
- Possible Directions
- Practical Issue

Data Introduction

The data sets origin from our validation set blending process in the track 2 of KDDCUP2012.

The track 2 of KDDCUP2012

- Task: predict click-through rate of ads on search engine.
- Data: 155,750,158 training instances, over 10 GB data sets.
- Goal: Maximize AUC among those instances.
- Difficulties: Huge data sets and feature extraction.
- Key to our success:
    - Explore useful features from the data.
    - Exploit diverse set of model.
    - Use blending to enhance the diversity, and boost the performance.

Data Introduction

Validation set blending

1. Validation Set(V): sample 1/11 instances from train set.

Data Introduction

Validation set blending

1. Validation Set(V): sample 1/11 instances from train set.
2. Training several models on the rest 10/11 instances.

Data Introduction

Validation set blending

1. Validation Set(V): sample 1/11 instances from train set.
2. Training several models on the rest 10/11 instances.
3. Split V into sub-training(V1) and sub-testing(V2) sets.

Data Introduction

Validation set blending

1. Validation Set(V): sample 1/11 instances from train set.
2. Training several models on the rest 10/11 instances.
3. Split V into sub-training(V1) and sub-testing(V2) sets.
4. Use models in step 2 to get predictions on both V and test set.

Data Introduction

Validation set blending

1. Validation Set(V): sample 1/11 instances from train set.
2. Training several models on the rest 10/11 instances.
3. Split V into sub-training(V1) and sub-testing(V2) sets.
4. Use models in step 2 to get predictions on both V and test set.
5. Create features of V1,V2 and testing data sets for validation set blending, including the predictions of models in step 2 and some optional extra features.

Data Introduction

Validation set blending

1. Validation Set(V): sample 1/11 instances from train set.
2. Training several models on the rest 10/11 instances.
3. Split V into sub-training(V1) and sub-testing(V2) sets.
4. Use models in step 2 to get predictions on both V and test set.
5. Create features of V1,V2 and testing data sets for validation set blending, including the predictions of models in step 2 and some optional extra features.
6. Treat V1 as the new training data and V2 as the new validation data, then do training to predict on the test set.

Data Introduction

Validation set blending(cont.)

Benefits:

- Validation set blending works when single models have enough diversity.
- The training size is much smaller than training for single models, we can try more complicated algorithms and feature engineering.
- We get about 1% improvement in the last week of the competition.

Data sets of final project

- 40,000 training examples, and 50,000 test ones.
- Binary label and each example contains 71 features.
- All training and testing examples are sampled from our validation set(V) of track2 of KDDCUP2012.
- The features including 45 single model predictions and 26 numerical features we extract from the raw data.

The ROC Curve

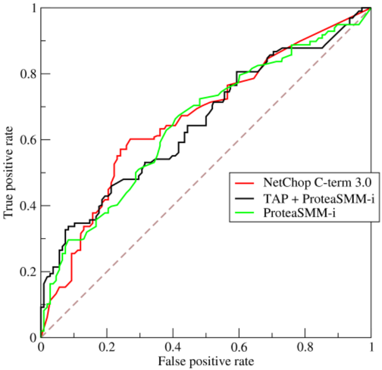
Receiver Operating Characteristic

		actual value		
		p	**n**	**total**
prediction outcome	**p'**	True Positive	False Positive	P'
	n'	False Negative	True Negative	N'
total		P	N	

- **True Positive Rate** = TP / P
- **False Positive Rate** = FP / N

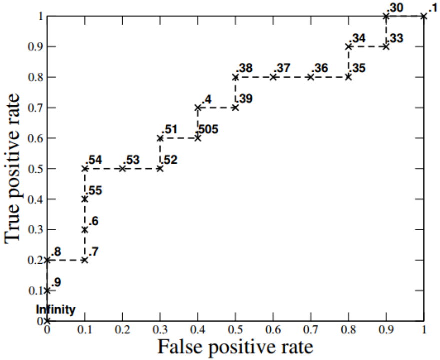
The ROC Curve

Receiver Operating Characteristic



- Each point on the curve correspond to an (TP,FP) pair.
- Imagine as we incline to report more positive instances, both TP and FP increases.

Typical Ranking Scenario & ROC Curve



Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

Area Under Curve (AUC)

- Defined as the area under ROC curve.

Area Under Curve (AUC)

- Defined as the area under ROC curve.
- Characteristics:
  - Equal to the $\mathbf{P}(\text{Rank}(I^{+})$ ¿ $\text{Rank}(I^{-}))$
  - Equal to the proportion of "corrected-ranked pair" among all pairs.
  - Measure how well your training model rank positive instances (higher), in a sense.

Calculation of AUC

- Equal to the proportion of “corrected-ranked pair” among all pairs.
- Given a sorted list, we can count the number of “corrected-ranked pair” in O(n).
  - For each Negative item, (accumulately) count how many instances are before it.

The Challenges

- What you know so far:
  - How to do (binary) classification.
  - How to do linear / logistic regression.
- The challenge:
  - Ranking: output is a sorted list.
    - Bipartite ranking: instance is either positive or negative.
  - Missing values.

The Bipartite Ranking Problem

- "Ranking": give "score" to each instance
  - Similar as in a **regression** problem.
  - But the binary label in training data could be a problem.
- Want to rank positive instance before negative ones.
  - Not that different with a **classification** problem.
- Thus, possible strategies:
  - "Score": use regression techniques.
  - "Pairwise Comparison": transform to the binary classification problem over pair of examples: $F : (\mathbf{x}, \mathbf{x'}) \rightarrow y$, which measures if $\mathbf{x}$ is "better" than $\mathbf{x'}$.
  - Any way you can turn a classification prediction into a confidence measure.

The Bipartite Ranking Problem

Few things to note, though:

- Handle ties with caution. Try to break ties if possible.
- As typical bipartite ranking problems, the samples could be **unbalanced**.
- Be sure to use AUC to measure your performance. (that's including your validation performance)

Handling Missing Data

- Random values.
- Average values.
- Special label '?' ..?
- Most "likely" values.
  - Look for similar sample?
  - Predict the missing value?
- Use your imagination.

Practical Issue

1. Data Pre-Processing
    - Target normalization
    - Feature normalization
    - Feature engineering
2. Parameter Selection
    - Depends on your data
    - Overfitting and Under fitting
    - Model type selection
    - Tradeoff between training time and performance
    - Stopping criteria: error tolerance
3. Accelerate the whole training procedure
    - Training time v.s. Loading time
    - Local disk v.s. NFS
    - Parallelization
    - Parameter selection

Questions?