Cuisine Prediction Based on Recipe

16組
洪立遠
余建遠
劉正康

Kaggle Competition Result

What's Cooking?
55 entries in team Li-Yuan Hung

Finished
44th/1388

Within Top 5%

Accuracy is 0.81516

Problem Definition


```
{
    "id": 22213,
    "ingredients": [
      "water",
      "vegetable oil",
      "wheat",
      "salt"
]
  }
```


"cuisine": "indian"

Statistics about Data

# of Train Data	39774
# of Test Data	9944
# of cuisine(class)	20
# of different ingredients	6714

Think of Recipe as a Document


```
{
    "id": 22213,
    "ingredients": [
      "water",
      "vegetable oil",
      "wheat",
      "salt"
]
  }
```


Bag of Words

["water","vegetable oil",
"wheat","salt"]

TF-IDF

TF-IDF

["water", "vegetable oil", "wheat", "salt"]

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Water_tf: 1/4

$$idf_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|}$$

Water_idf: log(39774/1836)

$$tf\_idf_{i,j} = tf_{i,j} \times idf_i$$

Vector Representation

- Each recipe is a vector of dimension 6714, value is taken by TF-IDF

1Nearest Neighbor with cosine similarity

- Find the most nearest train data with cosine similarity, predict the cuisine as the nearest train data.
- Accuracy is 0.66784

PCA and Random Forest

- Using PCA to do dimension reduction and then RandomForest
- With PCA is better than without PCA
- Accuracy is 0.69650

Xgboost

- Using xgboost(extrem gradient boosting)
- Accuracy is 0.70515
- A kaggle-winning weapon

Ingredient Split and Xgb

- Try split ingredients

- "vegetable oil".split() => ["vegetable", "oil"]
- dimension is reduced to 3589
- With xgboost, Accuracy is 0.78309

Add LDA Infomation

- Try LDA(Latent Dirichlet Allocation) to get 20 topics
- Add 20 dimension of word topic
- With xgboost, Accuracy is 0.80611

Grid Search to Blend RF and Xgb

- Use Grid Search to find the best parameters with Cross-Validation
- Parameters include max_depth, num_round, n_estimators
- Accuracy is 0.81516

Learned Lesson

- Just combine all you learn in this class and try it
- Cross-validation can give very close accuracy and be used to find good parameter
- Feature engineering is critical as long as your model is powerful enough and can find good parameter

Learned Lesson

- Near-area cuisines are easily mis-classified (ex : italian, frence)

Metric Learning

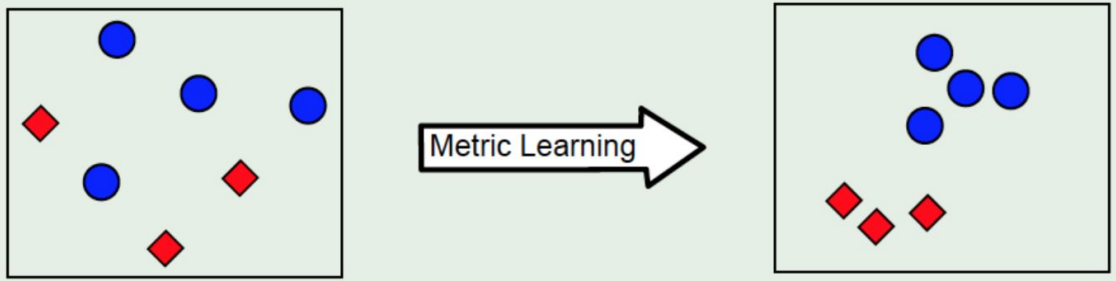
not finish running...

The notion of good metric is problem-dependent

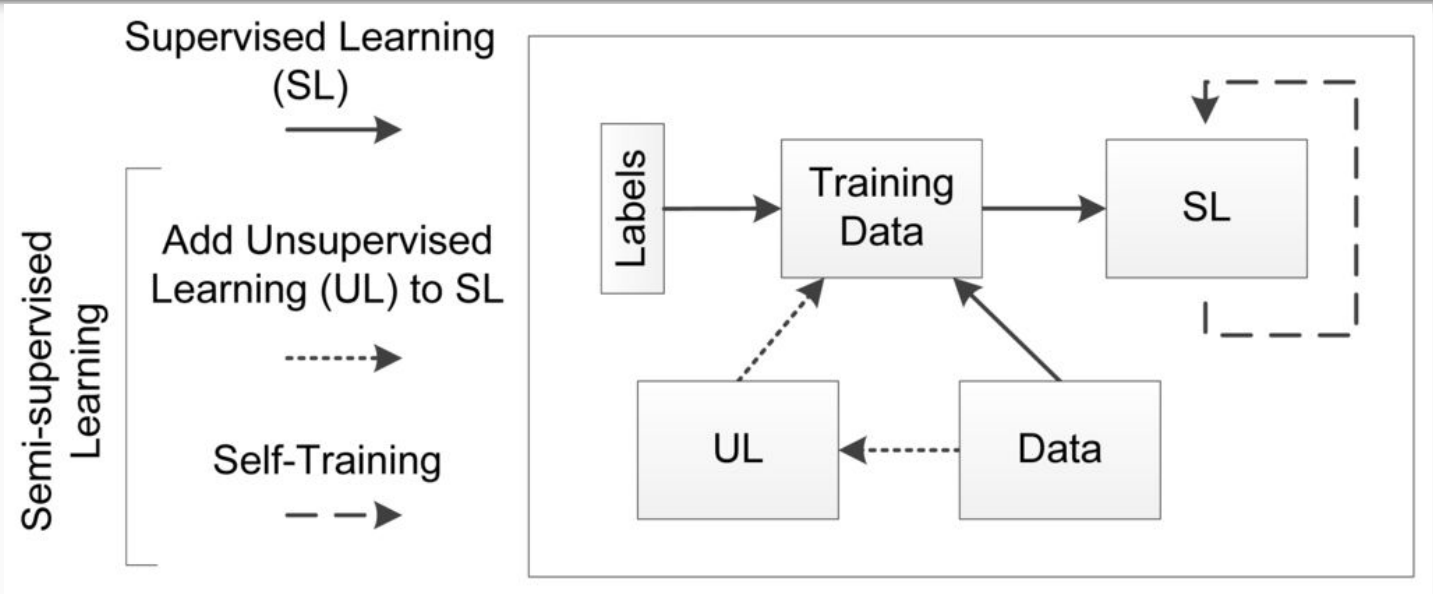
Each problem has its own semantic notion of similarity, which is often badly captured by standard metrics (e.g., Euclidean distance).

Solution: learn the metric from data

Basic idea: learn a metric that assigns small (resp. large) distance to pairs of examples that are semantically similar (resp. dissimilar).



Semi-Supervised Learning



Accuracy not better