



What's Cooking

All Eatable

CS 412 : Introduction to Machine Learning

He Huang, Ye Liu, Lichao Sun
Zhu Wang, Congying Xia, Fan Zhu

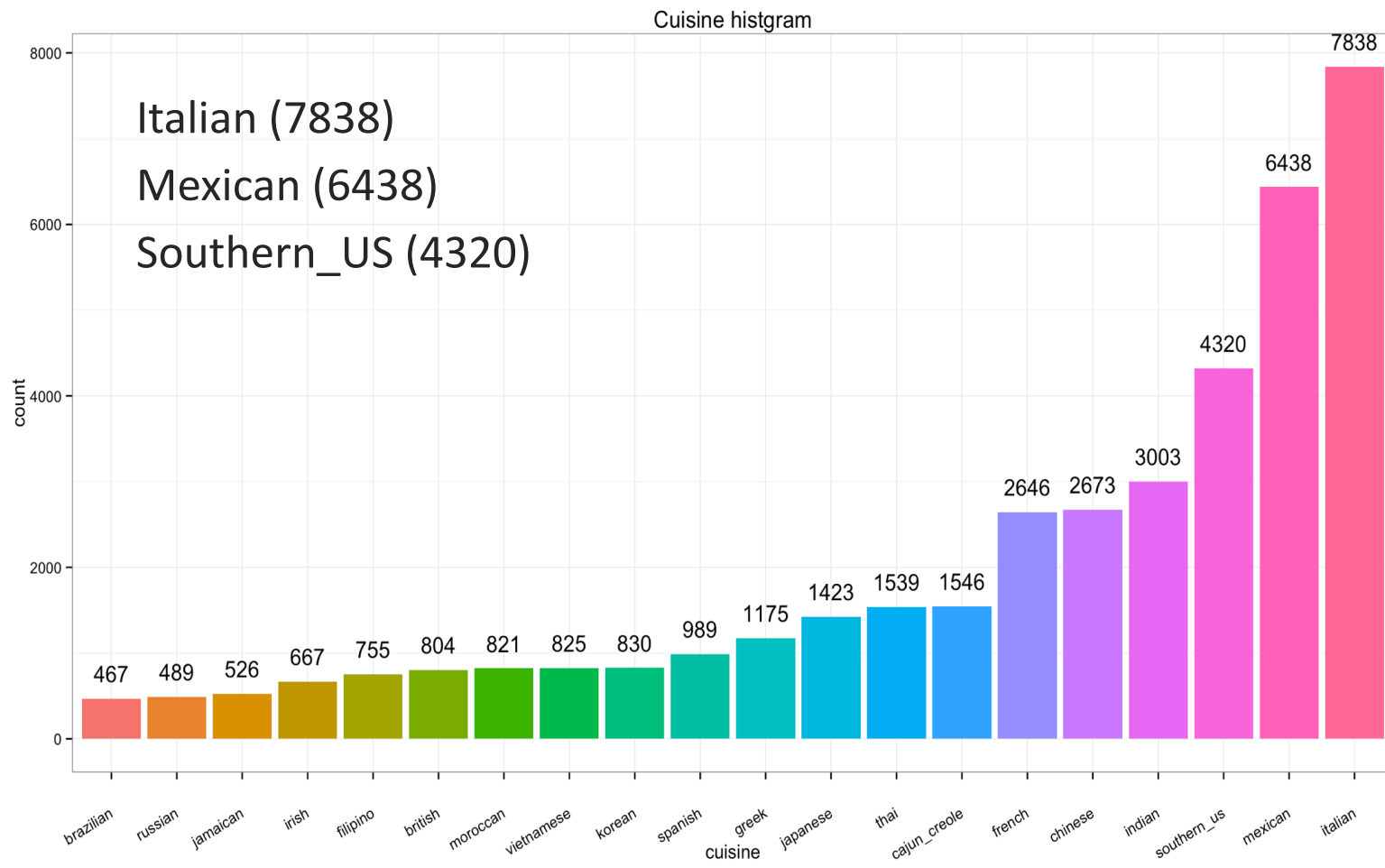
Apr 27, 2017

Project Description

- Kaggle
- Goal
 - use recipe ingredients to predict the cuisine
- Data
 - Training data
 - 39774 recipes and 20 kinds of cuisines
 - Recipe id
 - type of cuisine
 - list of ingredients of each
 - Testing data
 - Recipe id
 - list of ingredients of each

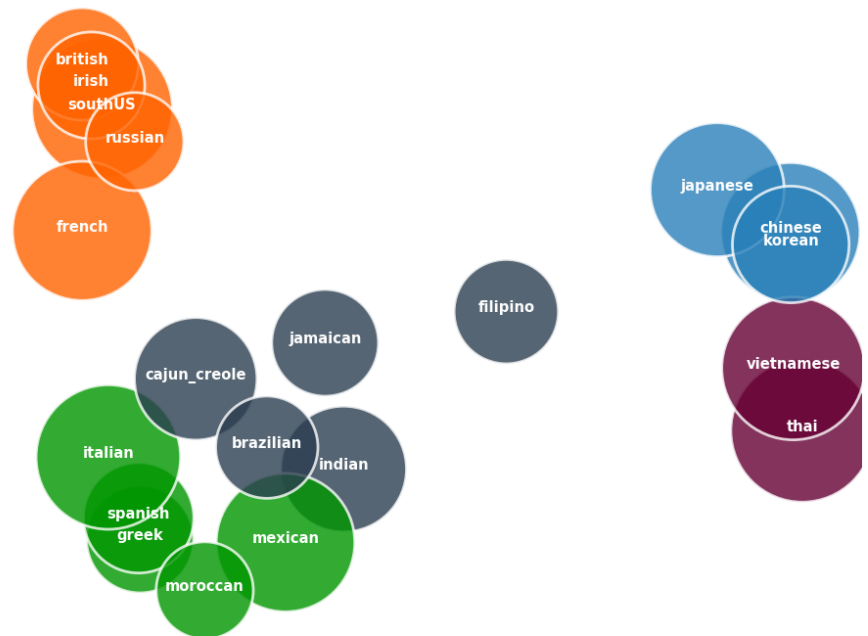


Introduction



Introduction

- Using K-means Clusters to visualize training dataset
 - Tf-idf : each term in gradients and cuisine as documents
 - PCA: reduced to 2-Dimensions
 - K-means cluster: $k = 5$
 - Size of bubbles: Jaccard Similarity(one v.s the other cuisines in its cluster)



Introduction

Ingredients frequency more than 1000



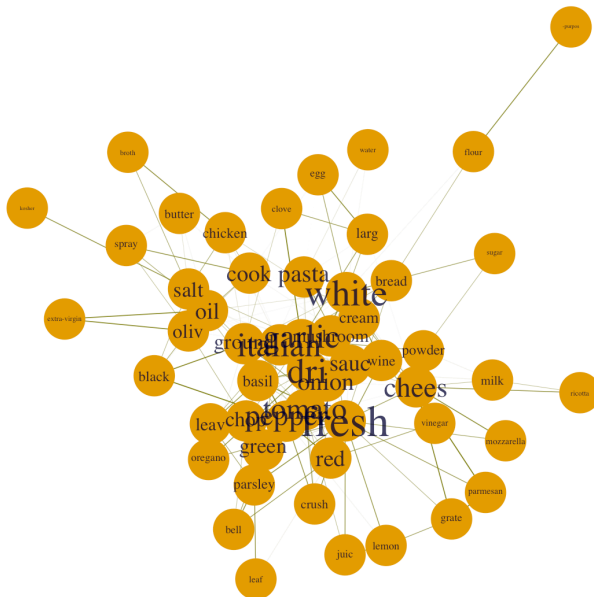
Introduction

➤ Data representation

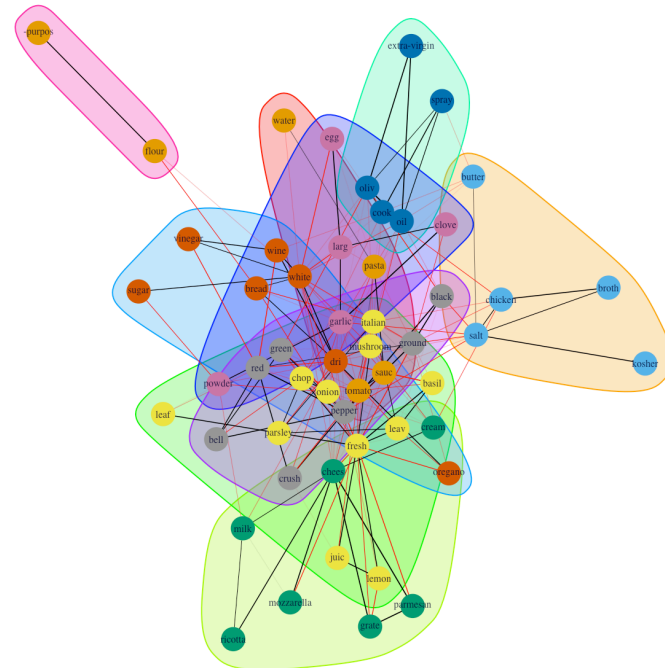
➤ Focus on *Italian* cuisine to show its ingredient network

➤ After removing stop-words, numbers and stemming

Italian ingredients network

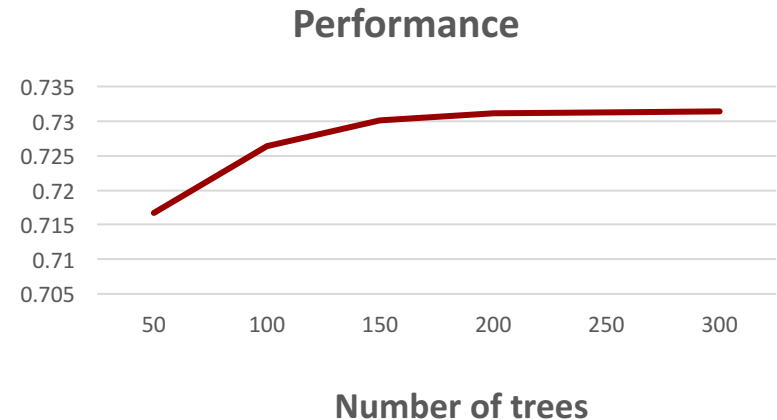


modularity community for italian cuisine

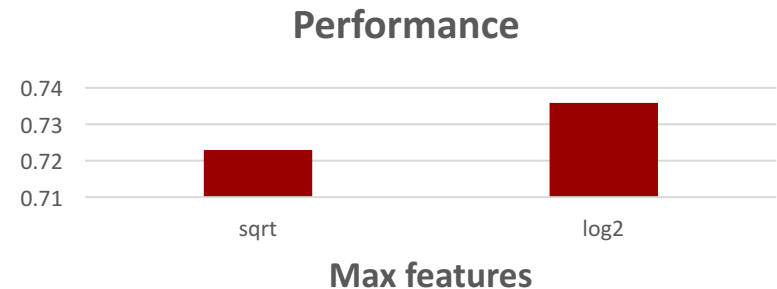


Random Forest

- Number of trees
 - Higher number of trees
 - Better performance
 - Lower speed



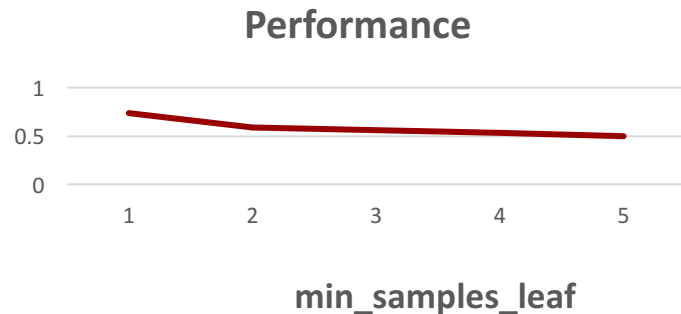
- Max_features
 - Sqrt -> $\sqrt{n_features}$
 - Log2 -> $\log_2(n_features)$
 - None -> $n_features$



Random Forest

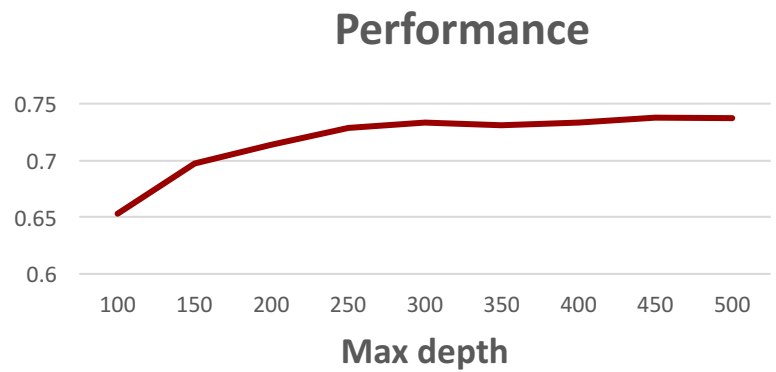
➤ Min samples leaf

- The minimum number of samples required to split an internal node

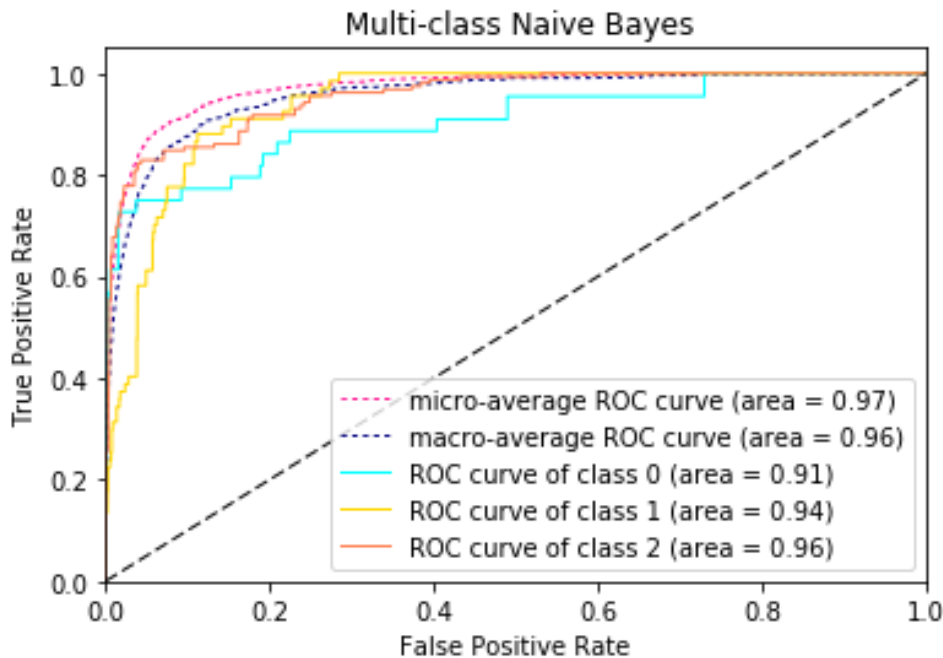


➤ max_depth

- The maximum depth of the tree.
- If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split sample



Naïve Bayes



Using 5-fold cross-validation to tune smoothing parameter α , we got $\alpha=0.125$ to build model.

There are 20 classes and the areas of ROC curves are around 0.89 to 0.99.

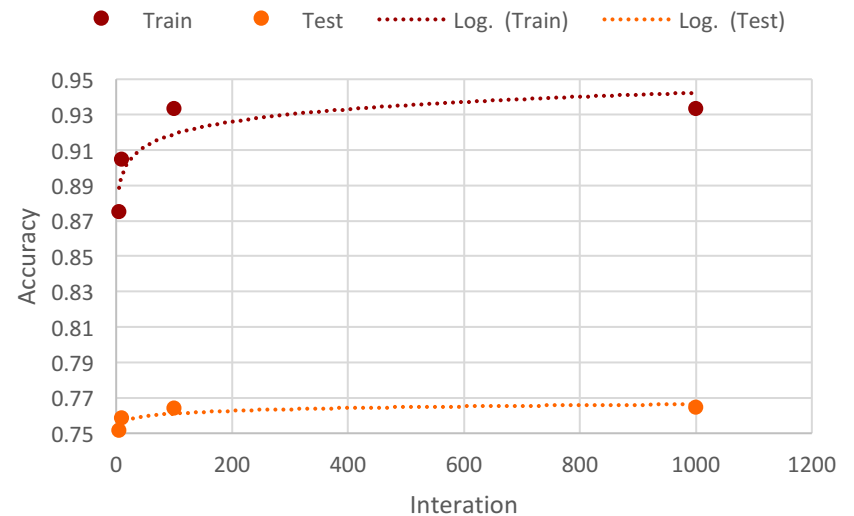
	Accuracy	F-score(micro)	F-score(macro)
Train	0.826601	0.826601	0.788229
Test	0.762946	0.762946	0.685015

Linear SVM

➤ max_iter

➤ The maximum number of iterations to be run

Max_Iter	Train	Test
5	0.874827153	0.751194368
10	0.904431175	0.758109127
100	0.933312382	0.764018104
1000	0.933123821	0.764395273



Linear SVM

➤ C: Penalty parameter C of the error term.

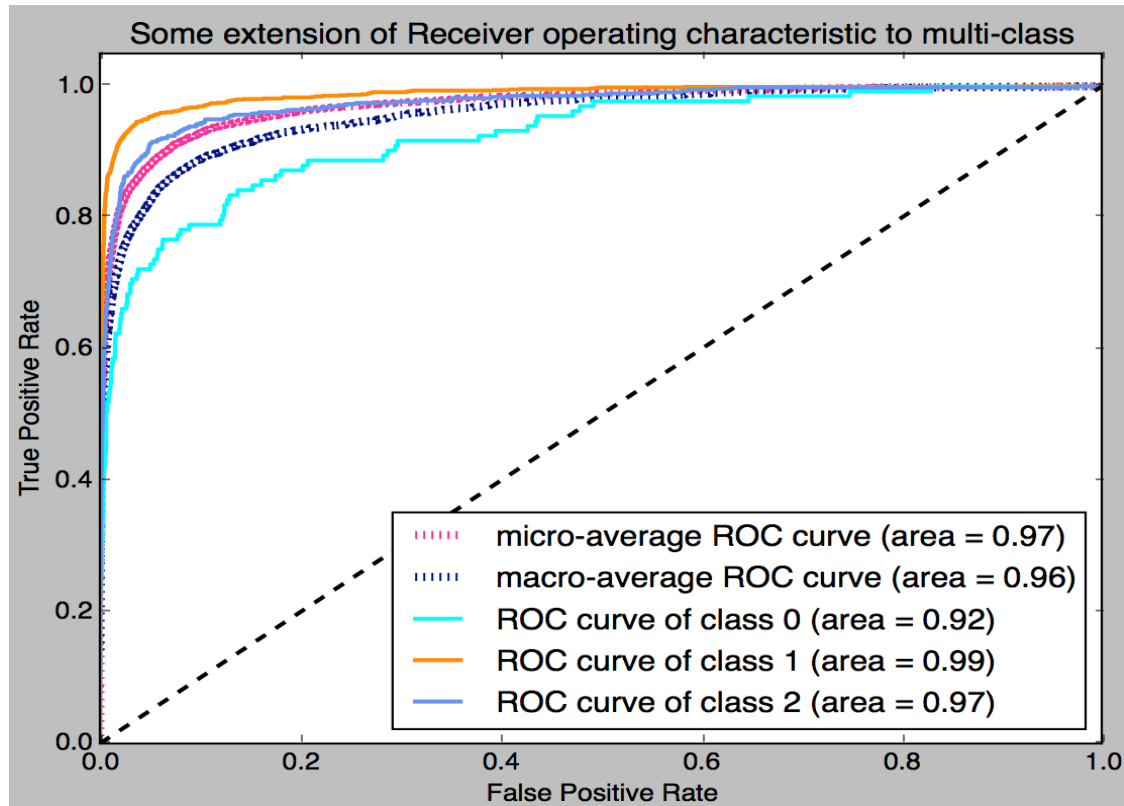
C	Train	Test
0.5	0.9221873	0.7738245
1	0.9331238	0.7643952
2	0.9403834	0.7523259

➤ Loss: 'hinge' or 'squared_hinge'

loss	Train	Test
hing	0.9085167	0.7725672
squared_hing	0.9331238	0.7643953


Linear SVM

➤ Roc Curve



LDA

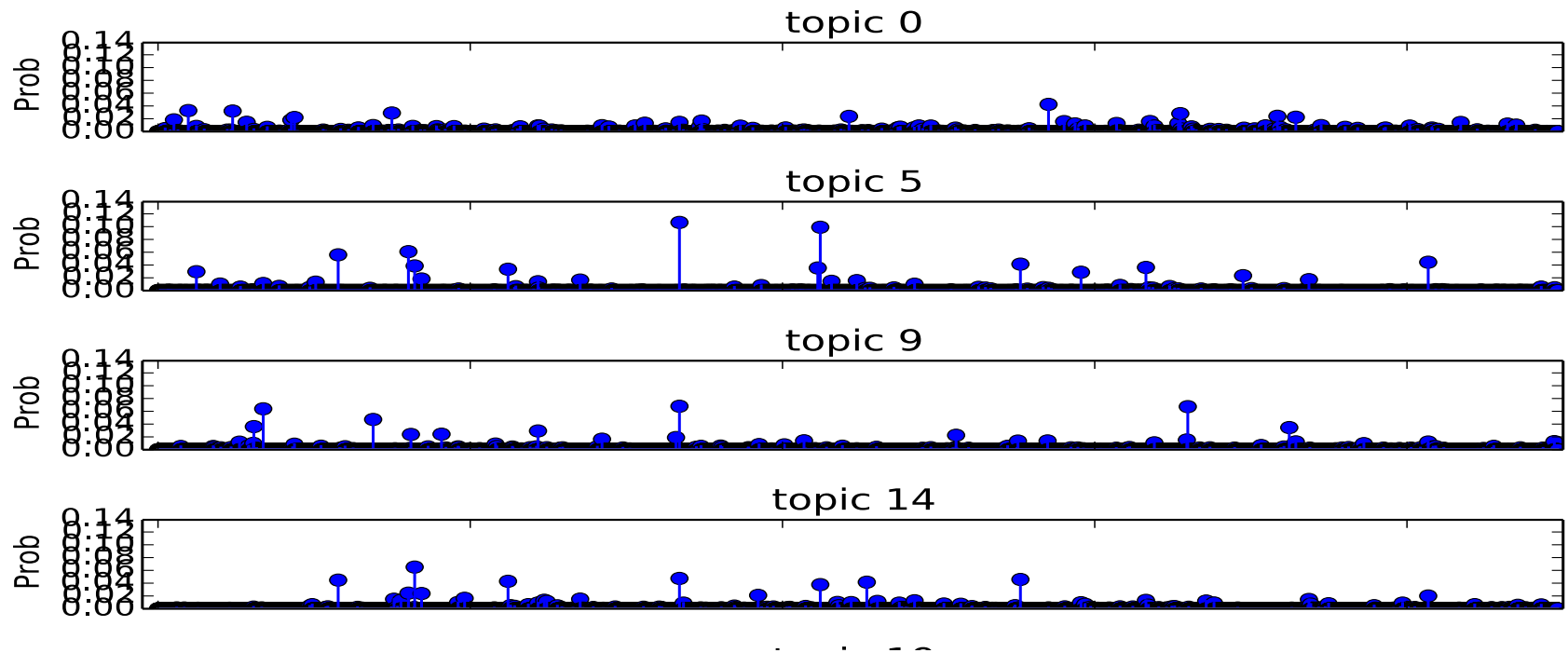
➤ Most probable words in four of the topics:



Topic 0	Topic 5	Topic 9	Topic 14
Sour cream	salt	salt	butter
Chili powder	All-purpose flour	Garlic cloves	salt
salsa	Baking powder	Cooking spray	eggs
Black beans	sugar	Olive oil	sugar
Flour tortillas	Large eggs	Chopped onion	milk

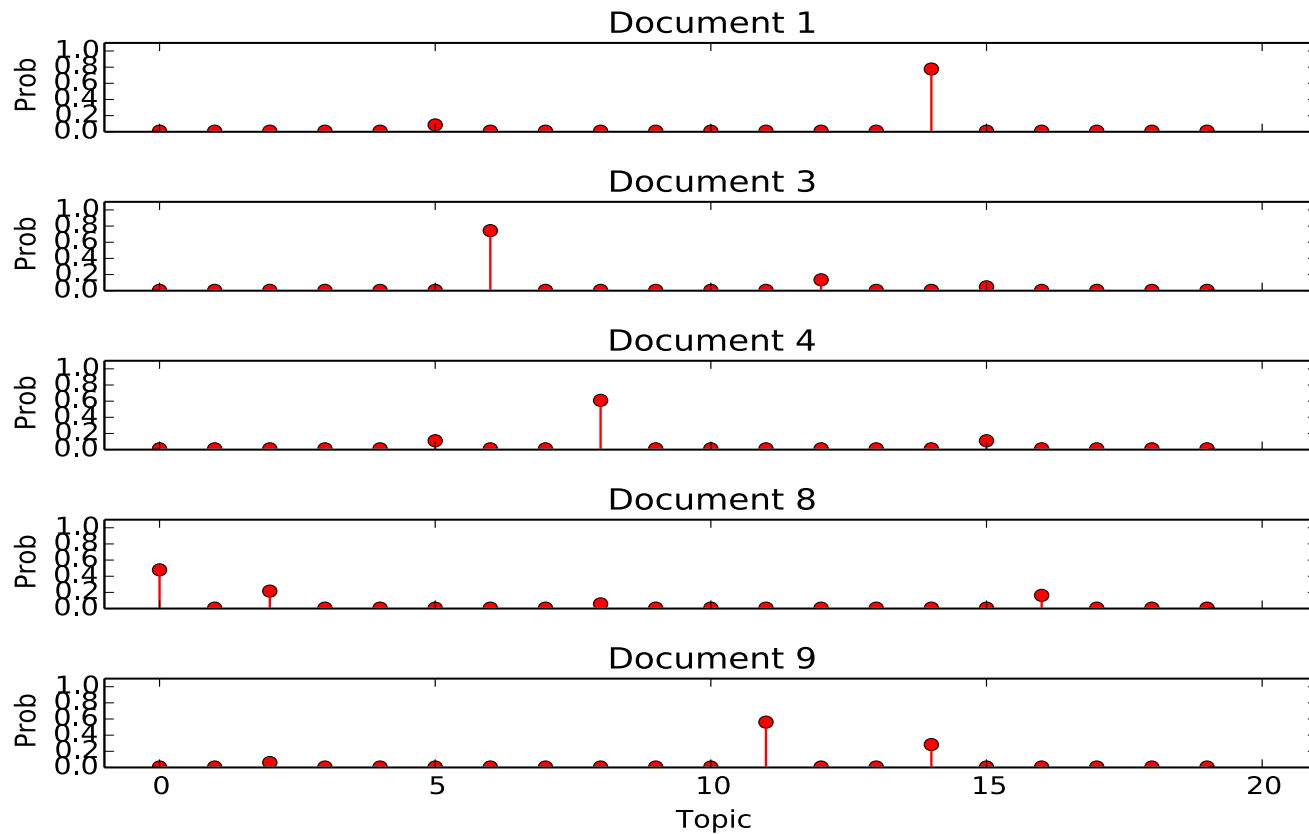
LDA

➤ Most probable words in four of the topics:



LDA

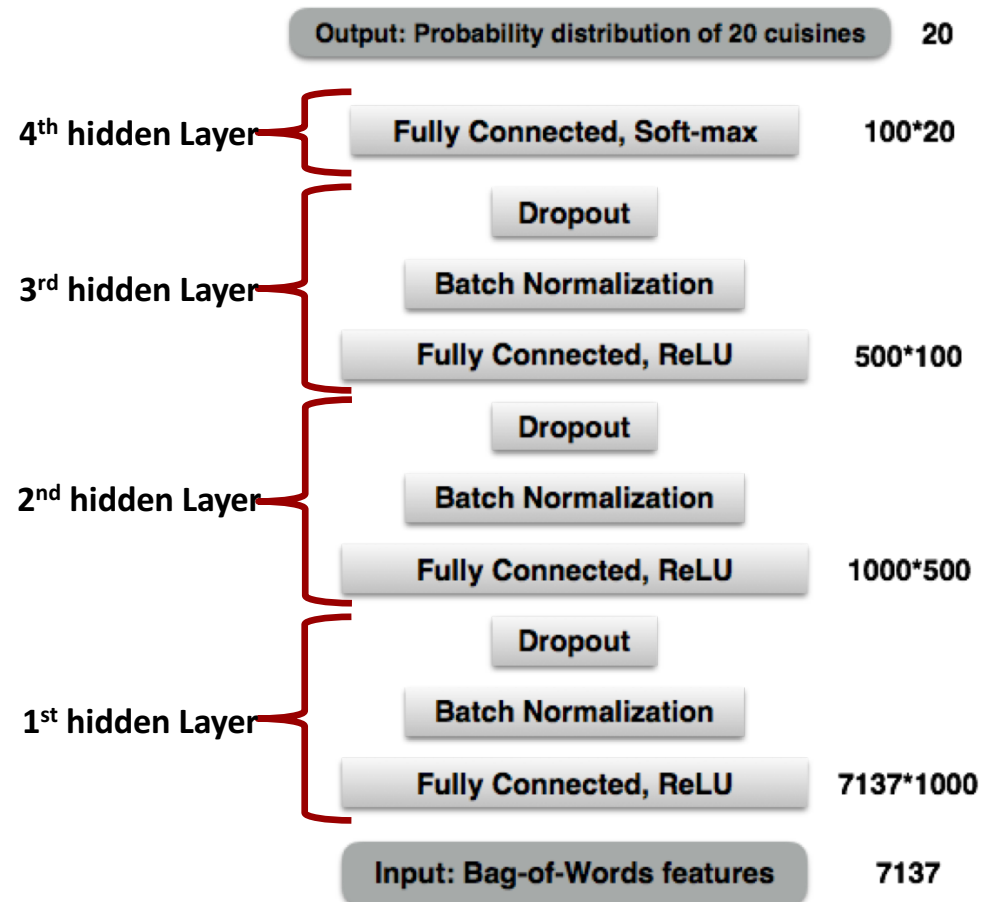
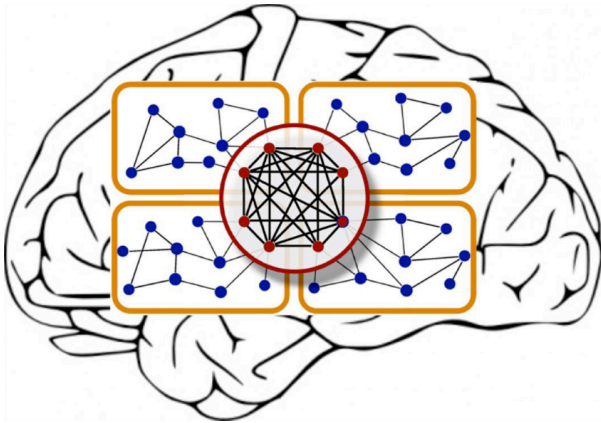
➤ Most probable topics in five of the docs:



LDA + KNN

➤ Unsupervised -> Supervised

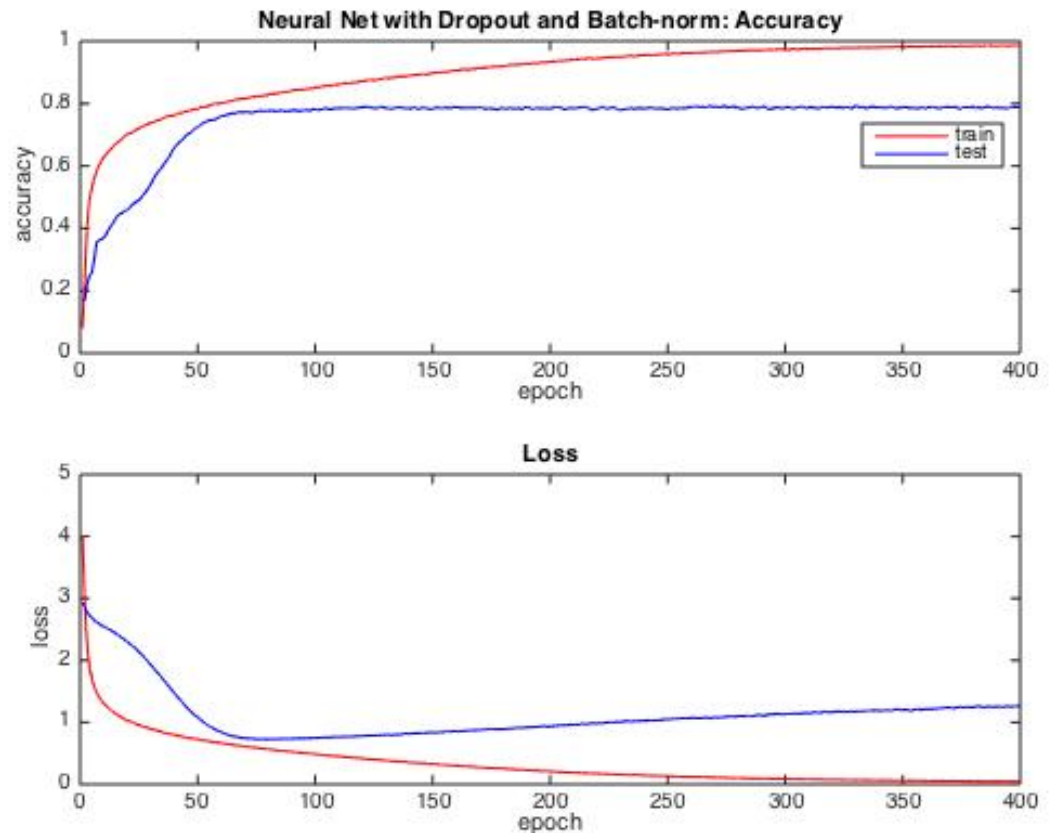
Neural Network



Results of Neural Network

Batch-norm with Dropout

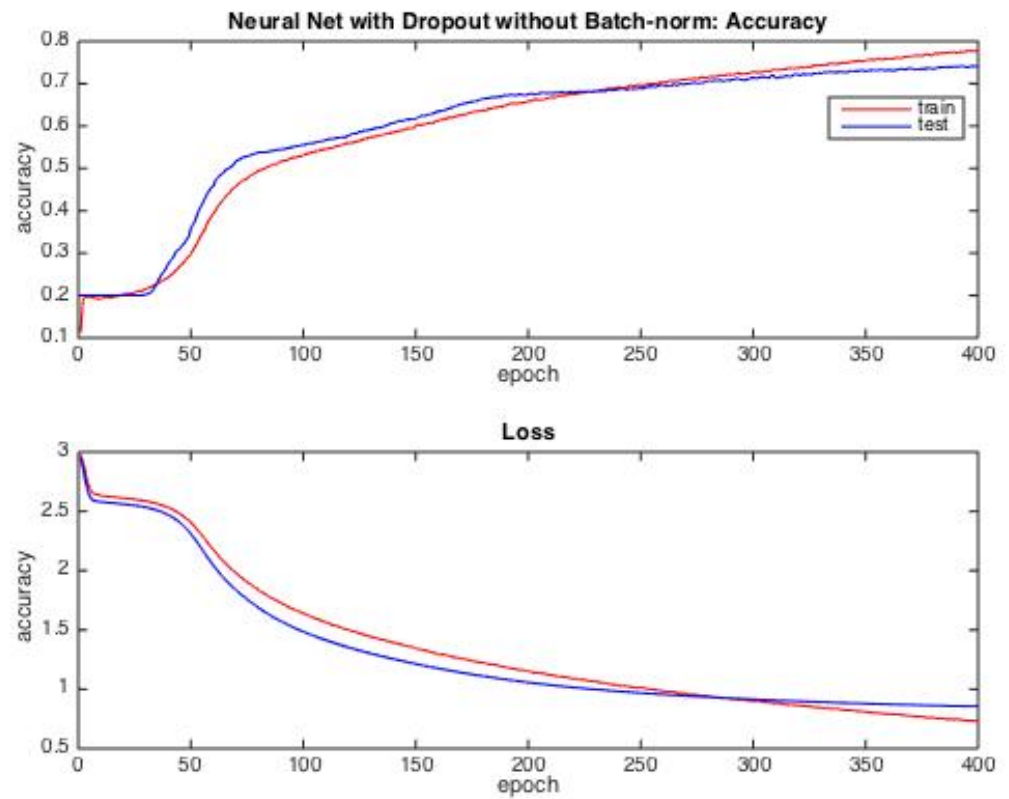
	Accuracy	Loss
Train	0.987010	0.043150
Test	0.789090	1.269093



Results of Neural Network

No Batch-norm but
with Dropout

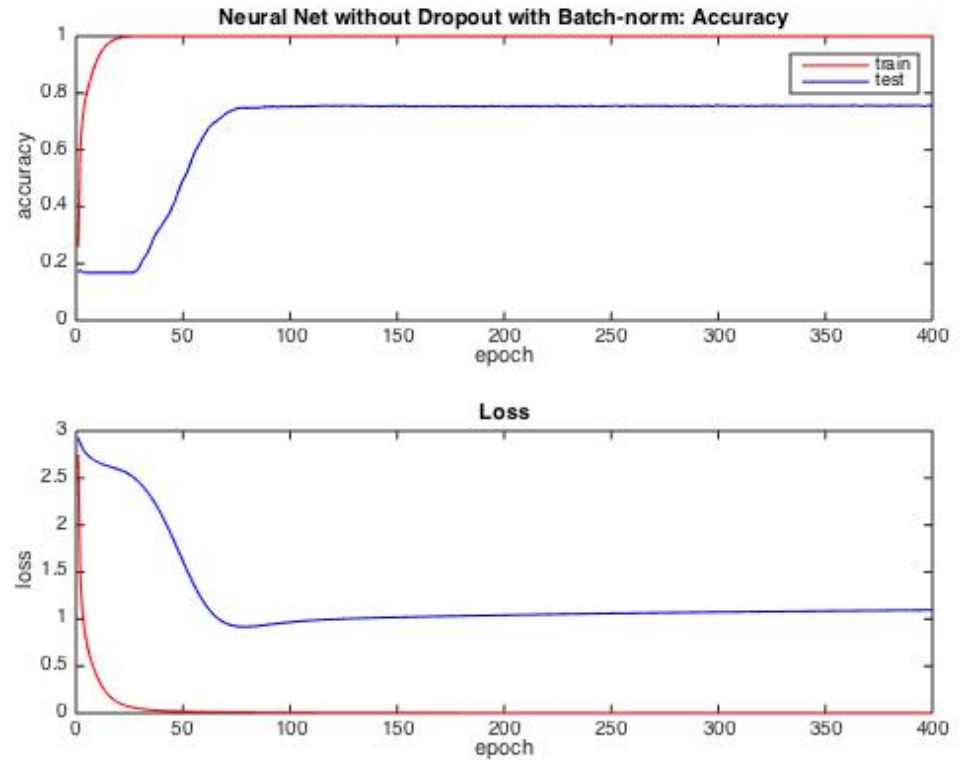
	Accuracy	Loss
Train	0.780003	0.726374
Test	0.741327	0.855012



Results of Neural Network

Batch-norm without Dropout

	Accuracy	Loss
Train	0.999609	0.001691
Test	0.754651	1.096316



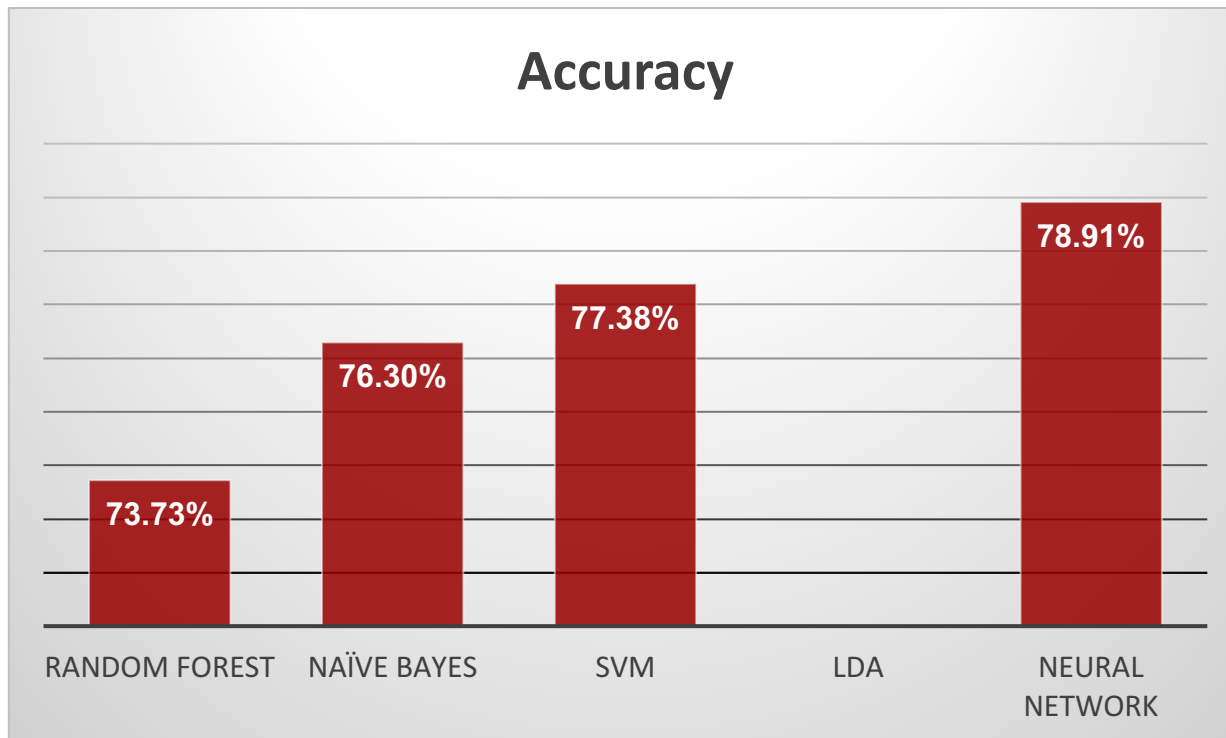
Comparison on the effects of Batch-norm and Dropout

Batch Normalization	Dropout	Train Accuracy	Test Accuracy
Yes	Yes	0.987010	0.789090
No	Yes	0.780003	0.741327
Yes	No	0.999609	0.754651
No	No	0.999497	0.746858

Batch-norm:
faster convergence

Dropout:
reduce over-fitting

Comparison



Fin

Thank you for your Attention