NLP7-8780 Project: **Latent Semantic Analysis for Clustering Genomic Sequences**

Quang Tran

Spring 2017

1 Introduction

Latent Semantic Analysis (LSA) is a technique in natural language processing, in particular distributional semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. In the field of information retrieval, LSA techniques are used to identify text documents with related underlying concepts. In this project, we extend these approaches to infer the biological structure through which a collection of organisms are related; we would like to identify and partition evolutionarily similar genomic sequences.

2 Background

2.1 Genomic K-mer Sequences

The term k-mer typically refers to all possible subsequences of length k that are contained in a larger sequence. For instance, we can say ATGCCCATTA is a k-mer of length 10 or a "10-mer". And this "10-mer" contains the 4-mers: ATGC, TGCC, GCCC, CCCA, CCAT, CATT, and ATTA. Since the nucletotides in DNA contain four different bases: A, T, G, and C, these 4-mers above are just a small subset of the $4^k = 4^4 = 256$ possible k-mers.

2.2 Bag-of-k-mers Model

A DNA sequences can be hypothesized by some statistical presentation of the k-mers from which it was comprised. LSA techniques are based on the bag-of-words model for prepresenting text documents By treating k-mers as the words, we modify the original model to bag-of-k-mers model in genetic language.

3 Approach

3.1 Latent Semantic Analysis

Latent Semantic Analysis refers to a theory and standard collection of techniques that attempt to identify similar documents in a text corpus based on their underlying concepts or knowledge content. In [1], LSA has mainly four steps generally:

- Formation of a term-document matrix
- Transformation/modified weighting of term-document matrix
- Dimensionality reduction
- Clustering of documents in the reduced space

Our objective is to modify these techniques to determine evolutionary similar sequences. a DNA sequence would be characterized by some statistical interpretation of the k-mers fro which it was comprised. We consider a sequence as an unordered collection of distinct k-mers, a "bag-of-k-mers". Since LSA works on the bag-of-words model for representing text documents, we are able to adjust the technique by treating k-mers as the words in our genetic language. We replace the notion of term-document matrices with k-mer sequence matrices.

3.2 Clustering Genomic Sequences

Formation of a k-mer-sequence matrix X is first constructed. Based on literature, a k-mer size of 7 would be recommended. Choosing k-mer size of 7 is a good trade-off, lowering the k-mer size reduces the information, i.e. lower resolution. However, k-mer size higher than 7 increases the resolution but also increase memory usage as well as computational expenditure. Each genomic sequence has k-mer frequency counts $4^7 = 16,384$ elements long.

Next step is important: dimensionality reduction. ”Non-negative matrix factorization” algorithm[2] has been used, since the toolbox is available.

4 Experiments

The code is written in Python, available at https://github.com/Coaxecva/COMP8780-Natural-Lang-Processng/tree/master/proj. Experiments is run on a single core of Intel i7-6700 CPU running @ 3.40GHz.

4.1 Genomic Data

The 16s rRNA sequences were taken from the Green Genes Database [1], which contains 116 sequences (Table 1) with sizes varying between 1,323 to 1,656bps .

4.2 Method Evaluation

Mutual information is an information-theoretic measure of how similar two joint distributions are. In the context of clustering, the mutual information between two clusterings T and C is defined as

$$MI(T,C) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(i,j)\log\frac{P(i,j)}{P(i)P(j)}$$

where $P(i,j)$ is the probability that a read belongs to both T_i and C_j; $P(i)$ is the probability that a read belongs to T_i; $P(j)$ is the probability that a sequence belongs to C_j. The Adjusted Mutual Information (**AMI**) [3] of two clusterings is an adjustment of mutual information to account for chance and is defined as follows:

$$AMI(T,C) = \frac{MI(T,C) - E(MI(T,C))}{\max(H(T),H(C)) - E(MI(T,C))}$$

where $E(MI(T,C))$ is the expected mutual information of two random clusterings and $H(T)$ is the entropy of the clustering T. An AMI value of 0 occurs when the mutual information between C and T is equal to that the mutual information of two random clusterings, whereas a value of 1 occurs when C and T are identical.

[1] http://greengenes.lbl.gov/cgi-bin/nph-index.cgi

Synechococcaceae	228054, 844608, 178780, 198479, 187280, 179180, 175058, 176884
Pelagibacteraceae	228057, 234102, 234685, 1121497, 767731, 230047, 330751, 317400, 347564, 352714, 234168, 231859, 232604, 233538, 573838, 136068, 585338, 4474077, 1121583, 4342576, 4382430, 4486293,
Mycobacterium	73627, 785154, 581446, 177511, 245190,
Staphylococcus	378462, 398771, 445143, 394166, 406264, 391797, 374752, 497126,
kestanbolensis	89370, 582313, 300272, 264371,
chondroitinus	38076, 704873, 769793, 793197, 740402, 621373, 684620, 724916,
Streptococcus	3415982, 851646, 173018, 164074, 162908, 170228, 4303137, 4303136, 4314992, 4303128, 4314995, 4314996, 4314991, 4314993, 4303135, 4314989, 4438695, 172949, 4328619, 166394, 163283, 165134, 163168, 4353727, 4485108,
terrigena	545968, 6928, 66690, 160747, 2589453, 1608765, 4419254, 4453999, 66691, 730466, 164688, 1119663, 4387805, 4450540, 3796497, 1117588, 4418736, 781355, 282831, 287170, 954826, 760306, 1075753, 901360, 905842, 642760, 636768, 793966, 1051447, 721579, 888868, 652357, 1058757, 761572, 1093772, 760622

Table 1: 116 16S rRNA sequences

Rand Index is a common measure in classification problems, where the measure takes into account directly the number of correctly and incorrectly classified items.

$$RI(T, C) = \frac{2(a+b)}{n(n-1)}$$

where a is the number of pairs of reads that are in the same cluster in T and C; and b is the number of pairs of reads that are in different clusters in T and C. The Adjusted Rand Index (**ARI**) is the same the Rand Index, but corrected for chance, when the index of two random clusterings is not a constant value [4]. An ARI value of 0 occurs when two C and T are independent, whereas a value of 1 means C and T are identical.

In addition to AMI and ARI, we also considered two complementary metrics, introduced by [5]: **homogeneity** and **completeness**. A clustering is homogenous if each cluster C_j contains only reads that come from some bacterium T_i. A clustering is complete if all sequences that belong to any group of bacteria T_i are placed into some cluster C_j. These two metrics are opposing in that it is often hard to achieve high scores on both homogeneity and completeness. A few examples might help understand this intuition:

- $T = C$ if and only if both homogeneity are completeness scores are 1. T being identical to C only occurs when sequences in each T_i are placed in exactly one C_j, and all sequences in each C_j comes only from one T_i.
- Suppose $T = \{\{r_1, r_2\}, \{r_3, r_4\}\}$ and $C = \{\{r_1, r_2, r_3, r_4\}\}$. Then, the completeness score is 1, because all sequences that belong to T_1 (and respectively to T_2) are placed in the same cluster in C. On the other hand, the homogeneity score is 0, because reads in the only cluster in C come from different bacteria in T.
- Suppose $T = \{\{r_1, r_2\}, \{r_3, r_4\}\}$ and $C = \{\{r_1, r_3\},\ \{r_2, r_4\}\}$. Then, both completeness and homogeneity scores are 0.

5 Implementation

To compute k-mer sequence matrix, we use three following functions. Using $k = 7$, we have $4^7 = 16384$ dimensional vector for each sequence. With 116 sequences, we have a matrix $116 * 16384$.

- KmerMaker(k, alphabet): create all possible k-mers with a given k

- KmerLister(dna, k): list all k-mers of a given sequence
- KmerCounter(dna_kmers, all_possible_kmers): compute the counts of all possible k-mers with list of given k-mers

In this study, we happen to know the number of distinct groups in the dataset. Knowing this information, we can use any clustering methods to predict on the reduced data. We used K-mean clustering [2].

Algorithm 1 KmerMaker(k, alphabet):

```
# Make K-mers list with a given k, and alphabet
alphabet_length ← len(alphabet)
# Recursive Call
return_value = []
for each kmer in KmerMaker(k-1, alphabet) do
    for each i_letter in range(0, alphabet_length) do
        return_value ← (kmer + alphabet[i_letter])
    end for
end for
return return_value
```

Algorithm 2 KmerLister(dna, k):

```
# List all k-mers with given sequence dna, and k
return_value = []
for each x in range(len(dna)+1-k) do
    return_value ← (dna[x:x+k])
end for
return return_value
```

6 Results

To run the program, we can use following command:

```
$python LSAKmerCounter.py < input − fasta > < k >
```

where $input - fasta$ contains multiple sequences we want to cluster (in FASTA format), and k is the value of length of k-mers.

K-mean clustering was used by following parameters:

[2] http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

Algorithm 3 KmerCounter(dna_kmers, all_possible_kmers):

```
# Compute one row of sequence-kmer-matrix
return_value = [0] * len(all_possible_kmers)
for each v in dna_kmers do
    index ← all_possible_kmers.index(v)
    return_value[index] += 1
end for
return return_value
```

AMI:	0.87569501105
ARI:	0.809573096635712
Homogeneity:	0.96370311876
Completeness:	0.89065347830
V-measure:	0.92573945716

Table 2: K-mean clustering results

kmeans = KMeans(n_clusters=8, init='k-means++', max_iter=100, n_init=1, verbose=False)

7 Discussion

We showed that LSA can be used to reduce high dimensional feature spaces of short genomic sequences (16S rRNA). We might be able to extend this approach to allow for the clusterings of longer sequence - whole genomes. However, different species from bacteria have more than one sequence in their genomes. Therefore, in the scope of k-mer sequence matrix, we only apply this approach to bacteria or viruses, whose genomes are single sequences.

In this project, we examined several k-mer sizes from 3 to 10, and we chose a k-mer size of 7*bp*, which results in k-mer sequence array of 16,384 elements. Since the data of 16S rRNA contains short sequences, the number of elements when using small k-mer size is able to distinguish two samples. As sequences increase, larger k-mer sizes should be investigated.

References

[1] S. T. Dumais, "Latent semantic analysis," *Annual review of information science and technology*, vol. 38, no. 1, pp. 188–230, 2004.

[2] Y. Li and A. Ngom, “Non-negative matrix and tensor factorization based classification of clinical microarray gene expression data,” in *Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on*, pp. 438–443, IEEE, 2010.

[3] N. X. Vinh, J. Epps, and J. Bailey, “Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance,” *Journal of Machine Learning Research*, vol. 11, no. Oct, pp. 2837–2854, 2010.

[4] L. Hubert and P. Arabie, “Comparing partitions,” *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[5] A. Rosenberg and J. Hirschberg, “V-measure: A conditional entropy-based external cluster evaluation measure.,” in *EMNLP-CoNLL*, vol. 7, pp. 410–420, 2007.

[6] V. Rus, “Lecture slides for natural language processing course nlp7-8780,” 2017.

[7] J. H. Martin and D. Jurafsky, “Speech and language processing,” *International Edition*, vol. 710, 2000.

[8] T. K. Landauer, P. W. Foltz, and D. Laham, “An introduction to latent semantic analysis,” *Discourse processes*, vol. 25, no. 2-3, pp. 259–284, 1998.