Fast approximate text document clustering using Compressive Sampling

Laurence A. F. Park

School of Computing and Mathematics
University of Western Sydney, Australia
lapark@scm.uws.edu.au
http://www.scm.uws.edu.au/~lapark

ECML PKDD 2011

Clustering in high dimensional spaces

- Clustering complexity depends on the number of objects and features in the data set.
- The data set sizes obtained are increasing in size due to the advancement in technology.
- Text documents databases exhibit the characteristic of high-dimensional, sparse data.
- In order to cluster large document sets, we must be able to reduce the data to a manageable form.
- Compressive sampling is a feature sampling method that guarantees no loss of information. provided that sufficient samples are taken.

Therefore, we should be able to cluster a small number of sampled features and maintain the clustering accuracy.

Outline

Compressive Sampling

Compressive sampling theory shows us that given:

- a vector space
- a linear transform Ψ from a sparse space to the vector space,
- a linear transform Φ incoherent to Ψ

we are able sample a small number of features of the Φ space and also provide perfect reconstruction of the original vector space.

Compressive sampling has shown to be very useful for images where Ψ is a wavelet transform and Φ is a noiselet transform.

Compressive sampling and reconstruction

To obtain the samples $\vec{\xi}$:

$$\vec{\xi} = \Phi \vec{y} \tag{1}$$

To recover the data $\vec{x}$:

$$\vec{x} = \min \|\vec{z}\|_1, \quad \text{s.t. } \vec{\xi} = \Phi \Psi \vec{z} \tag{2}$$

- $\vec{y} = \Psi \vec{x}$ is the original signal
- $\vec{x}$ is sparse,
- Φ is the sampling function that is incoherent to Ψ,
- $\vec{\xi}$ is the set of samples, and
- $\| \cdot \|_1$ is the l_1 norm.

Compressive sampling for document vectors

We must choose Ψ as the mapping from a sparse feature space containing $\vec{x}$ to our data feature space containing $\vec{y}$:

$$\vec{y} = \Psi\vec{x} \tag{3}$$

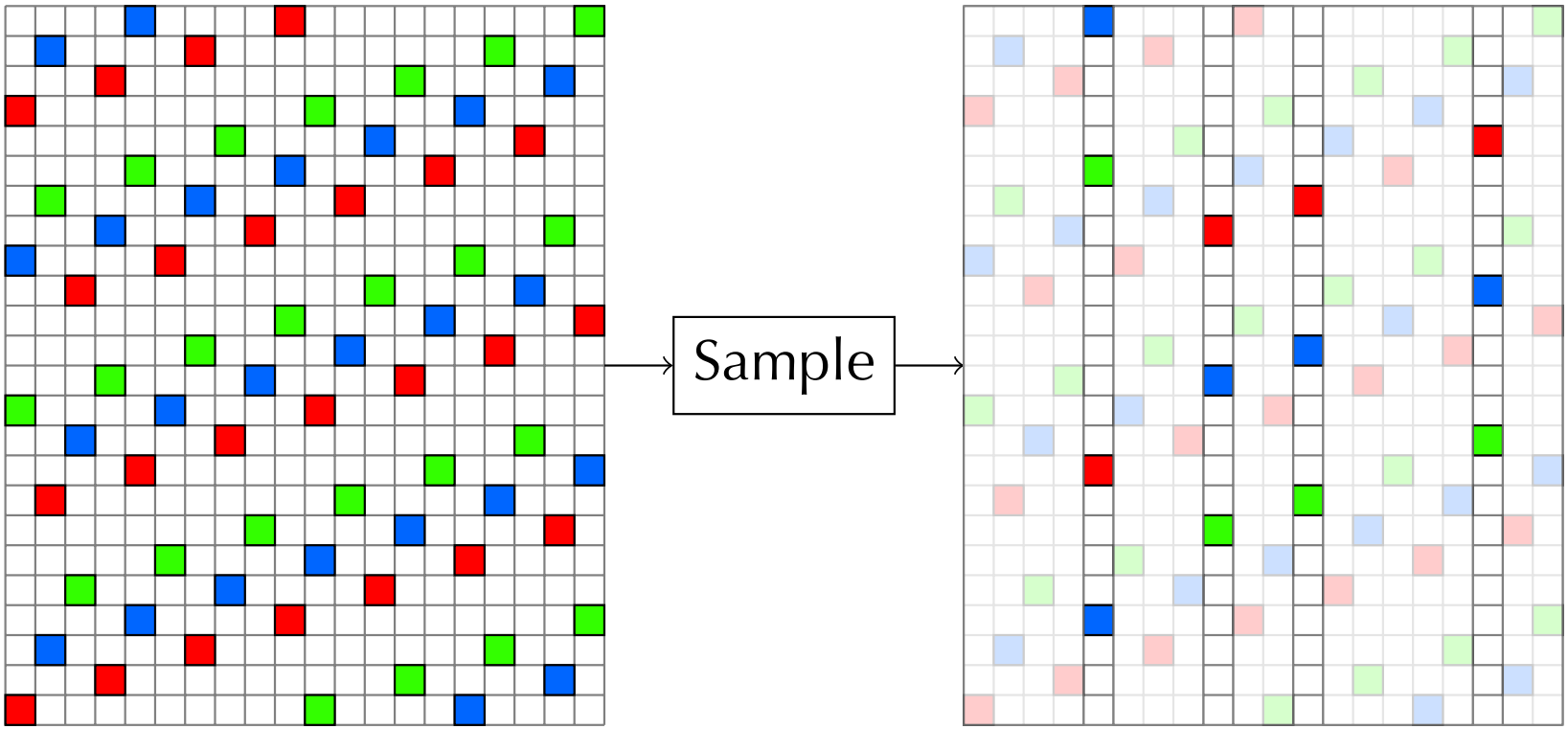
Since document vectors $\vec{y}$ are already sparse, we can set Ψ to the identity matrix I.

We must then choose the sampling matrix Φ to be incoherent with Ψ.

The Fourier transform matrix is maximally incoherent with respect to the identity matrix.

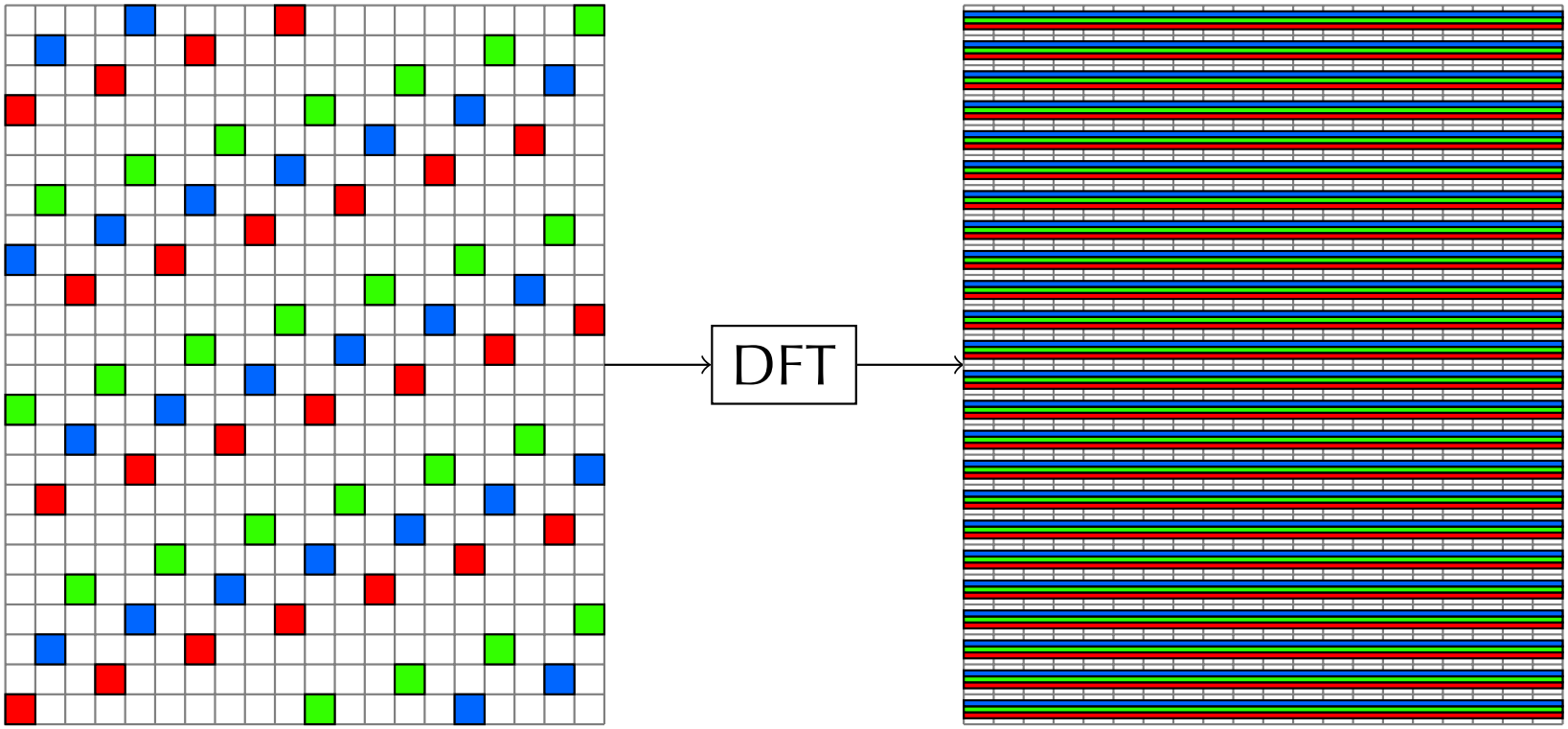
Therefore, we take a sample using the DFT basis as our sampling matrix.

Sampling the feature space



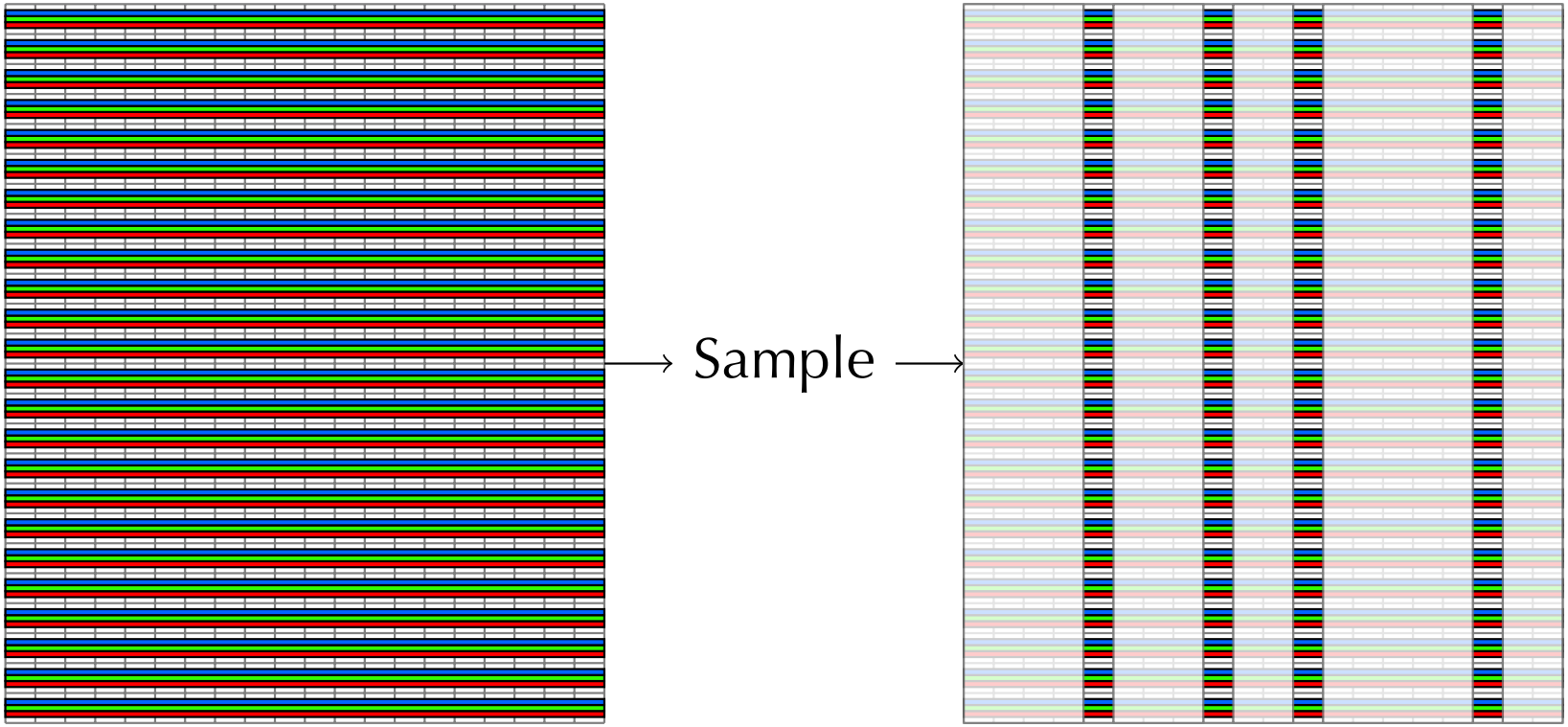
We can sample the features to reduce the scale of the clustering problem.
Unfortunately, if data is sparse, we lose most of the information.

Spreading out the sparse data



We use a transformation that is incoherent to the sparse domain (in our case, the Discrete Fourier Transform).

Sampling the spread data



We can now randomly sample a set of features. Each of the features contain a portion of the spread sparse features.

Outline

Clustering using the sample space

To cluster using compressive sampling:

1. Sample the features of the vector space using the sampling function to obtain $\vec{\xi}_i$, where $\vec{\xi}_i = \Phi \vec{y}_i$.
2. Cluster the sampled space $\{\vec{\xi}_1 \ldots \vec{\xi}_M\}$.
3. For each vector $\vec{\xi}_i$ in cluster $\mathcal{C}_m$, reconstruct the original vector $\vec{x}_i$,
4. Compute each cluster definition in the unsampled space based on the cluster vectors $\{\vec{x}_1, \ldots, \vec{x}_M\}$.

To obtain the cluster number associated to each vector, we then assign each unsampled vector based on the unsampled cluster definitions.

Therefore, we require an initial pass over the sparse data set to map the vectors into the sample space, and then finish with two passes over the data set to refine the clusters.

Radial k-means

We compute the radial k-means clusters using the iterative process:

$$\vec{c}_m \leftarrow \sum_{i \in \mathcal{C}_m} \vec{x}_i \text{ for } m \in \{1, \ldots, C\}$$

$$\mathcal{C}_m \leftarrow \{x_i | m = \underset{m}{\text{argmin}} \left(\vec{c}_m \angle \vec{x}_i\right)\} \text{ for } m \in \{1, \ldots, C\}$$

where $\mathcal{C}_m$ is the set of vectors associated to cluster m.
The angle between any two complex vectors $\vec{x}_i$ and $\vec{x}_j$ is defined as:

$$\cos\left(\vec{x}_i \angle \vec{x}_j\right) = \frac{\text{Re}(\langle \vec{x}_i, \vec{x}_j \rangle)}{\|\vec{x}_i\|_2 \|\vec{x}_j\|_2}$$

where $\|\vec{x}_i\|_2 = \left(\sum_j |x_{i,j}|^2\right)^{1/2}$ and $\text{Re}(a + ib) = a$ for real values a and b.

Real approximation

If we sample the sparse document vectors using a small subset of the Fourier basis, we obtain a sample space in the complex domain.

Every complex number is equivalent to two real numbers ($z = a + ib$)

Therefore, we will also examine the use of the Discrete Cosine Transform (DCT) as a real approximation for the sampling matrix.

Outline

Small Document Collection

SMART Collection (`ftp://ftp.cs.cornell.edu/pub/smart`):

- Contains four document sets (CRAN, CACM, CISI and MED).
- Each set contains 1398, 3204, 1460, and 1033 documents respectively, totalling $7,095$ documents.
- The collection contains $14,523$ unique terms.
- We ignored all terms that appears only once in the collection. These terms do not contribute to the clustering. This reduced the number of terms from $14,523$ to $7,866$.

Task

Cluster each document in the collection into its document set.

Due to the randomness in the experiment, we ran the compressive clustering algorithms 10 times.

Radial k-means

Accuracy				Time (sec)
CRAN	CACM	CISI	MED	
0.9709	0.9408	0.8798	0.9778	409.82

Radial k-means on DFT samples

DFT Feat	Mean Accuracy				Time (sec)
	CRAN	CACM	CICI	MED	
1024	0.9599	0.9196	0.8528	0.9620	74.11
512	0.9258	0.9403	0.8531	0.8944	36.30
256	0.9688	0.9328	0.8607	0.9638	28.49
128	0.8882	0.7117	0.6093	0.5981	24.66
64	0.8219	0.8498	0.7652	0.7184	25.70
32	0.6430	0.7304	0.5976	0.4092	34.93
16	0.7091	0.5815	0.4403	0.3331	24.95
No Samples	0.9709	0.9408	0.8798	0.9778	409.82

Trend

As the number of sampled features increases, the mean accuracy increases and the time increases.

Radial k-means on DFT samples

DFT Feat	Accuracy Standard Deviation			
	CRAN	CACM	CICI	MED
1024	0.0367	0.0670	0.0822	0.0408
512	0.1229	0.0037	0.0660	0.2071
256	0.0028	0.0148	0.0248	0.0132
128	0.1345	0.2112	0.2425	0.3925
64	0.1573	0.1079	0.0912	0.2704
32	0.1844	0.1683	0.1337	0.2357
16	0.1492	0.1412	0.1018	0.1982

Trend

As the number of sampled features increases, the accuracy standard deviation decreases.

Outline

Large Document Collection

TREC Disk 2 (`http://trec.nist.gov/data/docs_eng.html`):

- Contains four document sets (AP, FR, WSJ and ZIFF).
- Each set contains 79,919, 19,860, 74,520, and 56,920 documents respectively, totalling to 231,219 documents.
- The collection contains 208,932 unique terms.
- We ignored all terms that appears only once in the collection. These terms do not contribute to the clustering. This reduced the number of terms from 208,932 to 108,734.

Task

Cluster each document in the collection into its document set.

Due to the randomness in the experiment, we ran the compressive clustering algorithms 10 times.

Clustering on sparse data

Accuracy				Time (sec)
AP	FR	WSJ	ZIFF	
0.6780	0.9962	0.5969	0.8603	181734

The radial k-means algorithm takes 50.48 hours to converge.

Large scale results

Transform	Features	Accuracy			
		AP	FR	WSJ	ZIFF
DFT	128	0.6704	0.9959	0.5821	0.8617
DCT	256	0.6839	0.9964	0.6134	0.8627

Transform	Features	Computation Time (sec)			
		Pass 1	k-means	Pass 2	Pass 3
DFT	128	3626	458	2398	2334
DCT	256	3785	1130	2412	2342

The radial k-means algorithm takes 2.45 hours to converge using DFT sampling and 2.69 hours using DCT sampling.

Outline

Conclusions

- Large scale clustering requires intense computation and large storage.
- Compressive sampling allows us to take advantage of sparsity.
- We are able to obtain a fast approximate clustering by using the sampled data.

Fast approximate text document clustering using Compressive Sampling

Laurence A. F. Park

School of Computing and Mathematics
University of Western Sydney, Australia
lapark@scm.uws.edu.au
http://www.scm.uws.edu.au/~lapark

ECML PKDD 2011

Related work

Clustering using compressive sampling can be thought of as a type of Locality-Sensitive Hashing (LSH) or Random Projection (RP), where the hashing/projection function is defined as the DFT or DCT (independent of the data).

Note that we are able to invert the projection, which is not possible using LSH or RP.

Radial k-means on DCT samples

DCT Feat	Mean Accuracy				Time (sec)
	CRAN	CACM	CICI	MED	
1024	0.9494	0.8895	0.8022	0.9280	65.10_{*}
512	$0.9139^{\dagger}_{*}$	0.9207	0.8450	0.8977	48.93_{*}
256	$0.8752^{\dagger}$	0.8989_{*}	$0.7804^{\dagger}$	$0.8006^{\dagger}$	31.54_{*}
128	$0.7423^{\dagger}$	0.7662	0.5652	0.4937	$35.33^{\dagger}_{*}$
64	0.7163	0.7442	$0.5867^{\dagger}$	0.5109	29.78
32	0.5669	$0.5437^{\dagger}$	$0.4197^{\dagger}$	0.3207	27.14
16	$0.4694^{\dagger}$	0.5413	0.4074	0.3019	25.95
No Samples	0.9709	0.9408	0.8798	0.9778	409.82

- $\dagger$ denotes a significant difference when compared to the same result using the DFT.
- $*$ denotes a significant difference when compared to the associated result when using the DFT with half the number of features.

Radial k-means on DCT samples

DCT Feat	Accuracy Standard Deviation			
	CRAN	CACM	CICI	MED
1024	0.0454	0.1102	0.1640	0.0961
512	0.0699	0.0500	0.0669	0.0972
256	0.1586	0.0952	0.1198	0.2903
128	0.1986	0.1730	0.2540	0.3653
64	0.1975	0.1190	0.2045	0.3012
32	0.1508	0.1626	0.1518	0.2227
16	0.0930	0.1615	0.2737	0.1638