MTP SCEE 2019-20

Indian Institute of Technology Mandi

Dimensionality Reduction Using Feature Hashing

Suryakant Bhardwaj, B16117

Hrushikesh Sudam Sarode, B16032

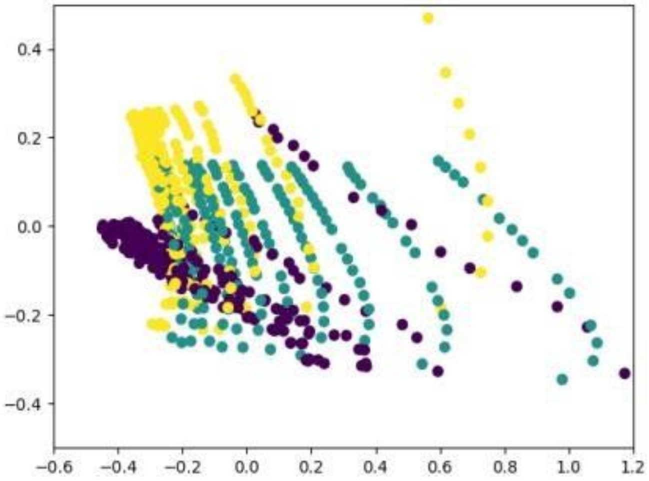
Under the guidance of

Dr. Rameshwar Pratap

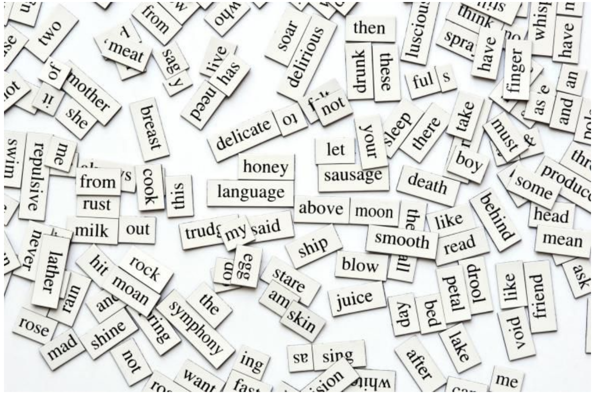
Why Dimensionality Reduction?

1. Collection of Large Data
2. Data size beats Computing Power
3. Vector Representation of Data
4. High Dimensionality of Vectors

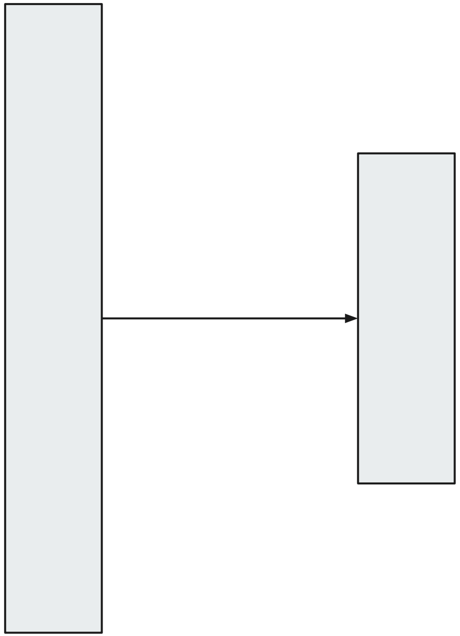
- Size of dataset depend on number of data points as well as on dimension of each data point.
- We can't control the number of data points involved for representing a real world entity model.
- But size of dataset can be reduced by reducing the dimension under which each data point is represented.



- Consider machine learning model which will check the similarity between two given documents.
- Requires large dataset for training that model and a dictionary containing list of attributes for which each data point is represented in vector form.
- Since dictionary can be too large may be of size 10000 to 100000.



- Dimension involved with each data point may become very high.
- This can be solved if somehow these data points (documents) can be represented in vector of much lower dimension.



Our Problem Statement

- *To achieve dimensionality reduction using feature hashing while preserving the similarity between the data objects.*

Our Methodology

- *Motive*

Reduce d dimensional Vector to N dimensions
Where N << d

$$<a, b> \cong <a', b'>$$

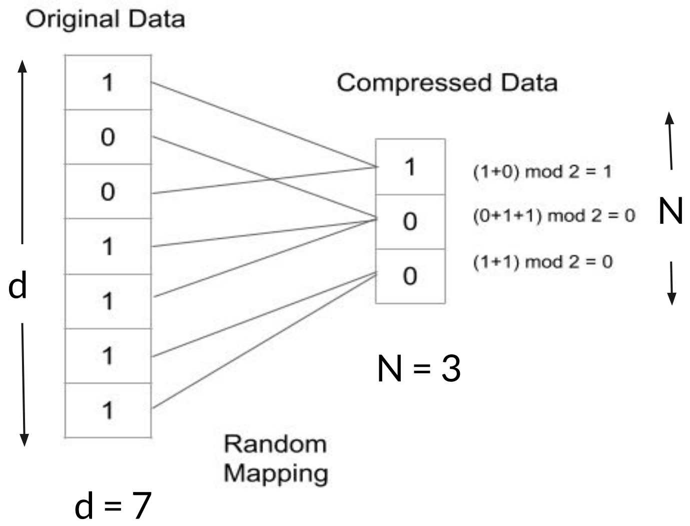
a, b **- N dimensional vectors**

Where , a', b' **- N dimensional vectors**

- *Compression scheme for Binary data*

- Sparse Data
- s - sparse data

$N = O(s^2)$



- Theorem Used for Binary Data

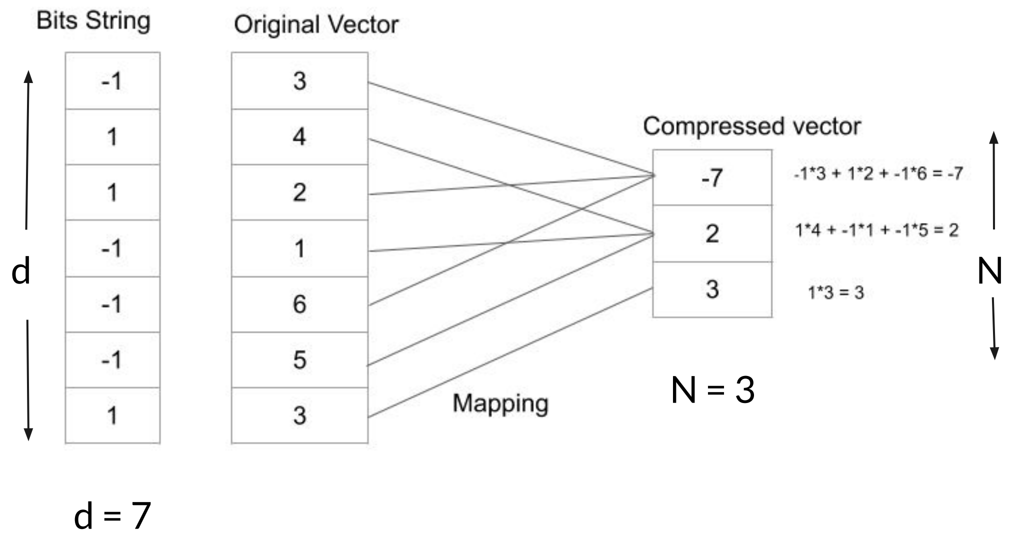
Theorem 1. *Consider a pair of binary vectors* $\mathbf{u_i}, \mathbf{u_j} \in \{0,1\}^d$ *such that the maximum number of* 1*s in any vector is at most* s*. If we set* $N = 10 \times s^2$*, and compress them into binary vectors* $\mathbf{u_i}', \mathbf{u_j}' \in \{0,1\}^N$ *via algorithm of [4], then the following holds with probability* 9/10

$$\langle \mathbf{u_i}, \mathbf{u_j} \rangle = \langle \mathbf{u_i}', \mathbf{u_j}' \rangle.$$

Source - Efficient Dimensionality Reduction for Sparse Binary Data

- Real Valued Data

- Random Mapping
- Random Bits



- Theorem Used for Real valued data

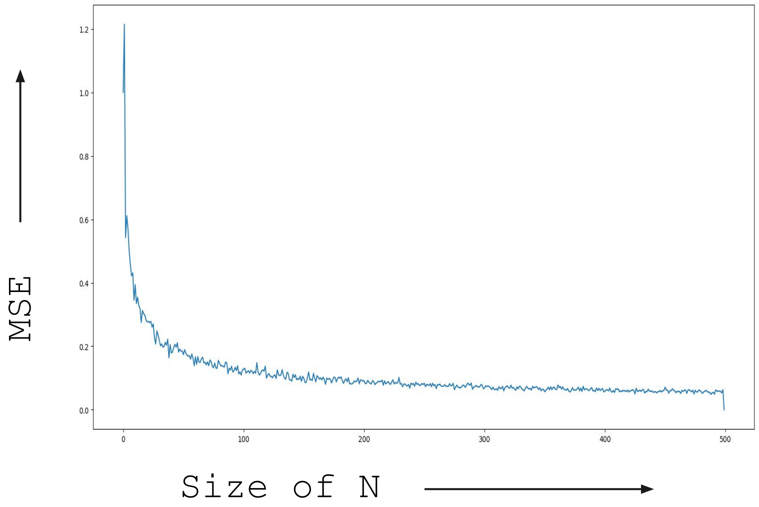
Consider a pair of normalised real valued vectors $\mathbf{u_i}, \mathbf{u_j} \in R^d$. *If we set* $N = 10/\epsilon^2$, *(where* ϵ *is the error tolerance, and* N *is the reduced dimension) and compress them into* $\mathbf{u_i}', \mathbf{u_j}' \in R^N$ *using the feature hashing algorithm of [1], then the following holds with probability* $9/10$

$$\langle \mathbf{u_i}, \mathbf{u_j} \rangle = \langle \mathbf{u_i}', \mathbf{u_j}' \rangle.$$

Source - feature hashing for large scale multitask learning

- *Defining N for Real Valued Data*

Trade off between Error and Storage



- *Why these compression scheme ?*

- ❑ Solves for high dimensional sparse data which is most common form of data now days.
- ❑ Independent of dimension and depends only on sparsity of data.
- ❑ Retain similarity between any 2 data points

Our Challenges

*–To come up with a data structure(code) which can maintain a random mapping while handling **insertions** and **deletions** for features in a growing dataset.*

- *Random Mapping*

- Used compression scheme demands highly random mapping .
- Highly random mapping is must for Uniform mapping of features from high dimension to lower dimension.

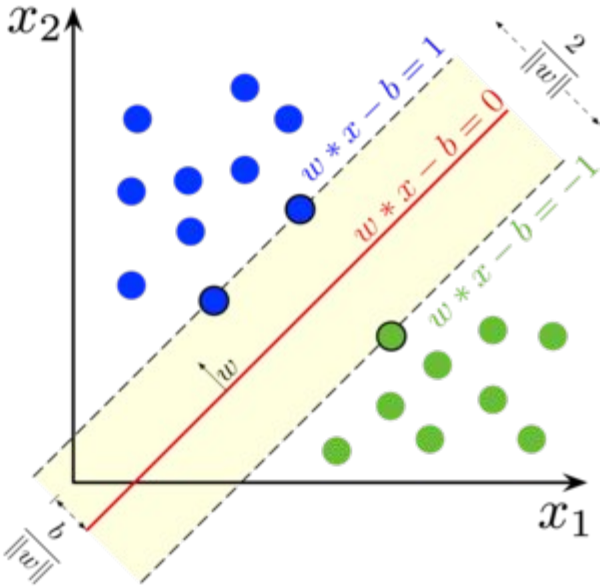
- Insertion and Deletion Handling

- Since dataset can be growing.
- New features can be added and deleted.
- Our code should be compatible with the proper feature insertion and deletion.
- Improper handling of mapping may result in risks of non uniform mapping.

Future Application

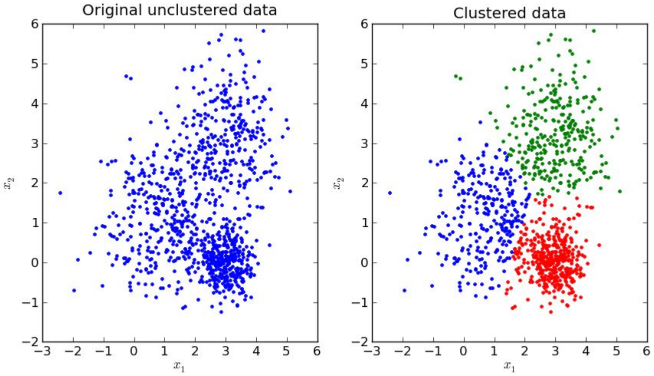
- *In classifier*

- Consider a d dimensional dataset for a Binary classification problem.
- Let any data point be represented as X.
- Let curve learned as a classifier be A as a d dimensional vector.
- For classification Inner product of A and X will be involved.

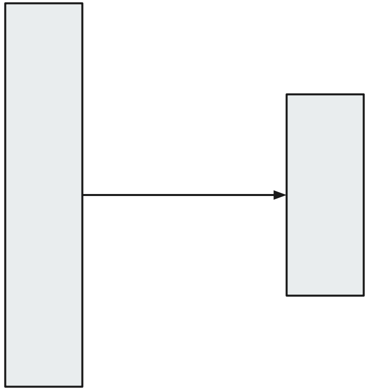


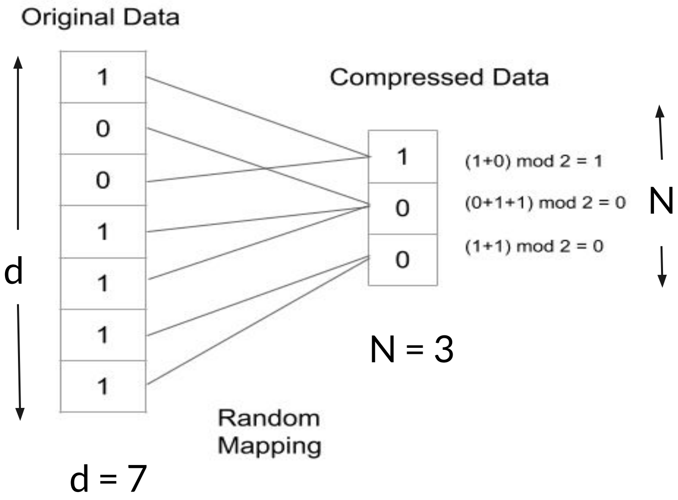
- *In clustering*

- Our compression scheme guarantee to maintain similarity between data points.
- Clustering Algorithms involves clustering of similar data objects.



- Thank You





Original Data
Compressed Data
1
0
0
1
1
1
1
d
1
0
0
(1+0) mod 2 = 1
(0+1+1) mod 2 = 0
(1+1) mod 2 = 0
N
N = 3
Random Mapping
d = 7