Dimensionality Reduction

Jakramate Bootkrajang

10 Aug 2014

Random Projection

RP: Key Idea

- We have seen that the working of data mining algorithms depends in some way or other on the geometry of data – lengths of vectors, distances, angles.
- Hign dimensional geometry is very different from low dimensional geometry. It defeats our intuitions – we can see things in 2D and 3D but have no idea how things will look like in HD.

RP: So what happens in HD?

Concentration of norms: Generate points in 2D and up and measure their lengths.

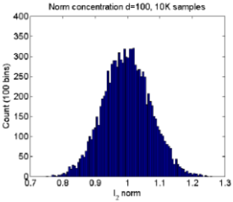


Figure: $d = 100$ norm concentration

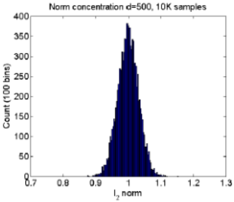


Figure: $d = 500$ norm concentration

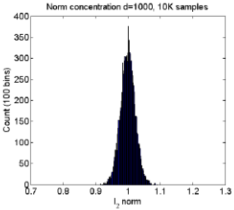
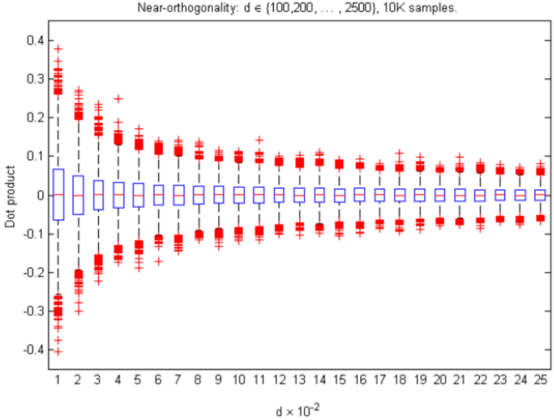


Figure: $d = 1000$ norm concentration

Credit: A.Kaban, CS-Bham

RP: So what happens in HD?

Near-ortholonality: Generate points in 2D and up and measure their angles.



RP: What is happening as d $\rightarrow \infty$

- We can see from the plots that
  - As m increases, any two random vectors end up being orthogonal to each other.
  - As m increases, any random vectors ends up having about the same length.

RP: Consequently

- When data has little structure (many features are indepednet of each other), then the ‘nearest neighbour’ is at about the same distance as the furthest one.
- When data does have structure, we can use a small collection of random vectors to project our data to lower dimension without losing much of the structure.

RP: Random projection

Theorem[Johnson and Lindenstrauss, 1984] Let $\epsilon \in (0,1)$. Let $N, k \in \mathbb{N}$ such that $k \geq C\epsilon^{-2} \log N$, for a large enough absolute constant C. Let $V \subseteq \mathbb{R}^d$ be a set of N points. Then there exists a linear mapping $R : \mathbb{R}^d \rightarrow \mathbb{R}^k$, such that for all $u, v \in V$:

$$(1-\epsilon)\|u-v\|^2_{\ell_2^d} \leq \|Ru - Rv\|^2_{\ell_2^k} \leq (1+\epsilon)\|u-v\|^2_{\ell_2^d}$$

- With high probability *random projection* satisfies JLL [Dasgupta & Gupta '02] (proof by Chernoff bounding).

This results can be used for large scale data mining, you just generate a kxm matrix where $k << d$ with entries i.i.d random from e.g. normal distribution, and pre-multiply your data with it to get k-dimensional data.