Exercises
Data Mining: Learning from Large Data Sets
FS 2016

LAS Group, Institute for Machine Learning
Dept. of Computer Science, ETH Zürich
Prof. Dr. Andreas Krause
Web: `http://las.ethz.ch/teaching/dm-f16/`
Email questions to:
Carlos Cotrini, ccarlos@inf.ethz.ch

Series 2, Oct 14th, 2016 (Locality Sensitive Hashing)

It is not mandatory to submit solutions and sample solutions will be published after one week. If you choose to submit your solution, please send an e-mail from your `ethz.ch` address with subject `Exercise2` containing a PDF (LaTeX or scan) to `ccarlos@inf.ethz.ch` until Wednesday, Oct 19th 2016.

Problem 1 (Locality Sensitive Hashing for the Cosine Distance):

(a) In this question, you'll prove that random hyperplanes can be used for locality sensitive hashing of vectors $\mathbf{v} \in \mathbb{R}^n$ when using the cosine distance. In particular, consider the family of hash functions

$$\mathcal{H} = \{h(\mathbf{v}) = \text{sign}(\mathbf{w}^T\mathbf{v}) \text{ for some } \mathbf{w} \in \mathbb{R}^n \text{ s.t. } ||\mathbf{w}||_2 = 1\}$$

Every hash function in $\mathcal{H}$, when applied to some vector $\mathbf{v} \in \mathbb{R}^n$, computes the dot product of $\mathbf{v}$ with some unit length vector $\mathbf{w}$, and returns $+1$ if the product is positive, and -1 otherwise (we ignore the case $\mathbf{w}^T\mathbf{v} = 0$ here). The vector $\mathbf{w}$ is distributed uniformly on the unit sphere $\{\mathbf{w} \in \mathbb{R}^d \mid ||\mathbf{w}|| = 1\}$. Suppose that we pick a hash function uniformly at random from $\mathcal{H}$. Show that

$$Pr([h(\mathbf{u}) = h(\mathbf{v})]) = 1 - \text{angle}(\mathbf{u}, \mathbf{v})/\pi$$

where $\text{angle}(\mathbf{u}, \mathbf{v}) \in [0, \pi]$ denotes the angle (in radians) between two n-dimensional vectors.

(b) Write a Python script that will calculate the probability of two vectors having the same hash as a function of their cosine distance. More precisely, you should sample

- N pairs of random vectors in $\mathbb{R}^d$. You can for example use the uniform distribution on $[-1, 1]^d$, or a normal $\mathcal{N}(\mathbf{0}, I_{d\times d})$.
- M random hash functions in $\mathbb{R}^d$. Instead of using the family in the previous part, you can use sketches (vectors sampled uniformly from $\{-1, +1\}^d$), or you could try normally distributed vectors.

Then, for each of the N pairs you should compute the fraction of the M hash functions that agree on them. You are free to pick N and M and d as you wish. Create scatter plots that show your results. Do they relate to the result from the previous section? What happens when you change the dimension of the data?