Singular Value Decomposition

Piyush Agarwal

IIT Bombay

12th March 2021

Contents

What is SVD?

Let A be a $m \times n$ matrix of rank r with all real entries then the singular value decomposition of $A = U\Sigma V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T$ where

U is a $m \times m$ orthonormal matrix, V is a $n \times n$ orthonormal matrix and Σ is a $m \times n$ diagonal matrix with all non-negative diagonal entries

The columns vectors of U or u_i are known as left singular vectors of A.
The columns vectors of V or v_i are known as right singular vectors of A.
The diagonal entries $\{\sigma_1, \sigma_2, \sigma_3, ...\}$ are known as singular values of A.
Note that $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq ... \geq \sigma_r \geq 0$.Rest of the singular values are zero.

Why is the SVD important?
Note that the SVD is applicable to all kind of matrices unlike the eigen value decomposition which is applicable only to square matrices
The way SVDs are defined any truncated SVD of rank a captures maximum information of the original matrix A as compared to any other matrix B of rank a.We'll talk more on it later.

More on u_i, v_i, σ_i

Let us work backwards and consider $A = U\Sigma V^T$.

Now, Consider $A^TA = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T\Sigma V^T \qquad [U^TU = I]$
$\implies A^TAV = V\Sigma^T\Sigma V^TV = V\Sigma^T\Sigma \quad [V^TV = I]$
Now Σ is a diagonal $m \times n$ matrix and so $\Sigma^T\Sigma$ is a $n \times n$ diagonal matrix .
$\implies A^TAv_i = \sigma_i^2 v_i$
Hence the right singular vectors(v_i) of A are eigen vectors of A^TA and singular values of A, $\sigma_i = \sqrt{\lambda_i}, \qquad \lambda_i \in \lambda(A^TA)$.

Note that A^TA is a semi-positive definite symmetric matrix and hence all eigen values, $\lambda(A^TA) \geq 0$

Continued

Similarly, consider $AA^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T$
$\implies AA^T U = U\Sigma\Sigma^T$, and hence we have that
$AA^T u_i = \sigma_i^2 u_i$
The left singular vectors, u_i of A are the eigen vectors of AA^T.

Few More Direct Relations between A, u_i, v_i and σ_i

$A = U\Sigma V^T \implies AV = U\Sigma \implies Av_i = \sigma_i u_i$
$A^T = V\Sigma^T U^T \implies A^T U = V\Sigma^T \implies A^T u_i = \sigma_i v_i$

Now, These relations will lead to the proof of the Singular Value Decomposition for any matrix A.

Proof of SVD

Theorem

Let A be a matrix of order $m \times n$ of rank r, then there exists an orthonormal matrix U of order $m \times m$, a diagonal matrix Σ with all non-negative entries($\sigma_1 \geq \sigma_2 \geq \sigma_3 ...$) of order $m \times n$ and an orthonormal matrix V of order $n \times n$ such that $A = U\Sigma V^T$

Proof: Consider AA^T. AA^T being symmetric positive semi-definite, has an orthonormal basis of eigen vectors v_i with all eigen values $\lambda_i \geq 0$.
Set $\sigma_i = \sqrt{\lambda_i}$ and assume that $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq ... \geq \sigma_r > 0$ (if not change the order)
Now, Set the columns of V as v_i.
Define $u_i = \frac{Av_i}{\sigma_i}$ for $i \in \{1, 2, 3, .., r\}$ The set $\{u_1, u_2, u_3, .., u_k\}$ forms an orthonormal set as $\sigma_i \sigma_j u_j^T u_i = v_j^T A^T A v_i = \lambda_i v_j^T v_i$
Complete the set $\{u_1, u_2, u_3, .., u_r\}$ to an orthonormal basis in $\mathbb{R}^m$
Now, Since $Av_i = \sigma_i u_i \quad \forall i$, We have $A = U\Sigma V^T$

Geometrical Perspective

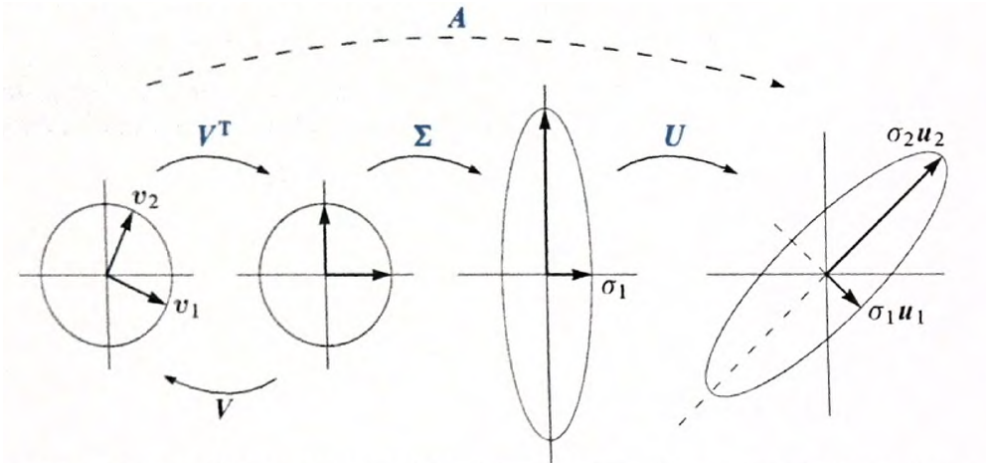


Figure: Visualisation of the SVD in 2 dimensions

$A = U\Sigma V^T$ and the three operations they depict are:
Rotation-Stretching-Rotation

Norms and Best Fitting Subspaces

Let A be a matrix of order $m \times n$, $\quad A = [a_{ij}],\ 1 \leq i \leq m, 1 \leq j \leq n$

Frobenius norm of A is defined as $||A||_F = \sqrt{\sum_{i,j} a_{ij}^2}$

The two norm of A is defined as $||A||_2 = max(\frac{||Ax||}{||x||}) = max_{||x||=1}||Ax||$

Note that $||x|| = \sqrt{x_1^2 + x_2^2 + .. + x_n^2}$, is the usual norm on vectors.

We have some very interesting results concerning the Singular Value Decomposition of A and the above two norms which also connect with best fitting subspaces.

Continued

Let A be a matrix of order $n \times d$ to be geometrically interpreted as n points in a d dimensional space.
What we are interested in is a k-dimensional subspace which is a best approximation to the n points in the d space. Call the rows of A: a_i and let us consider $k = 1$. Let v be the best fitting 1-dimensional subspace. Then we have to maximize $\sum_{i=1}^{n}(a_i.v)^2 = ||Av||^2 = ||A||_2$, as $||v|| = 1$

Claim is $||A||_2 = \sigma_1$
Proof: Consider the columns vectors of V (this forms an orthnonormal basis for $\mathbb{R}^n$), $\{v_1, v_2, .., v_n\}$. Any vector $x \in \mathbb{R}^n$ can be written as
$x = c_1 v_1 + c_2 v_2 + .. + c_n v_n$
$||Ax|| = ||c_1 Av_1 + c_2 Av_2 + .. + c_n Av_n|| = ||c_1 \sigma_1 u_1 + c_2 \sigma_2 u_2 + .. + c_k \sigma_k u_k||$
$\implies ||A||_2 = max_{||x=1||}||Ax|| = \sigma_1$ for $x = v_1$
Note that for all vectors v perpendicular to say $v_1, v_2, .., v_l$, $max||Av|| = \sigma_{l+1}$ given $||v|| = 1$. This is achieved for $v = v_{l+1}$. Proof follows from above.

Continued

Now, this is true for any k, i.e. the best fitting k dimensional subspace is formed by $\{v_1, v_2, .., v_k\}$

Proof is by induction.
We have already shown it for $k = 1$

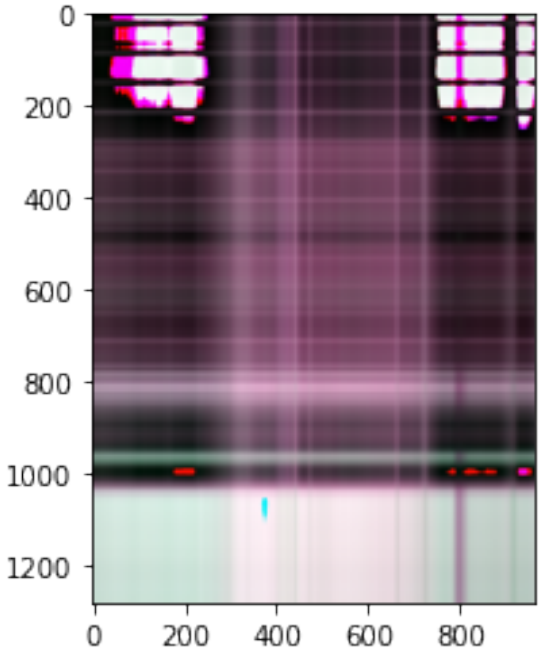
Let us suppose it to be true for k_0 and we'll show it for $k_0 + 1$
Let $\{w_1, w_2, .., w_{k_0+1}\}$ be a $k_0 + 1$-dimensional subspace such that w_{k_0+1} is perpendicular to all the vectors $v_1, v_2, .., v_{k_0}$
Then from the previous slide we have that $||Aw_{k_0+1}||^2 \leq ||Av_{k_0+1}||^2$.
Also from induction we have that
$||Aw_1||^2 + ||Aw_2||^2 + ... + ||Aw_{k_0}||^2 \leq ||Av_1||^2 + ||Av_2||^2 + .. + ||Av_{k_0}||^2$
Combining both we have the claim.

Frobenius Norm

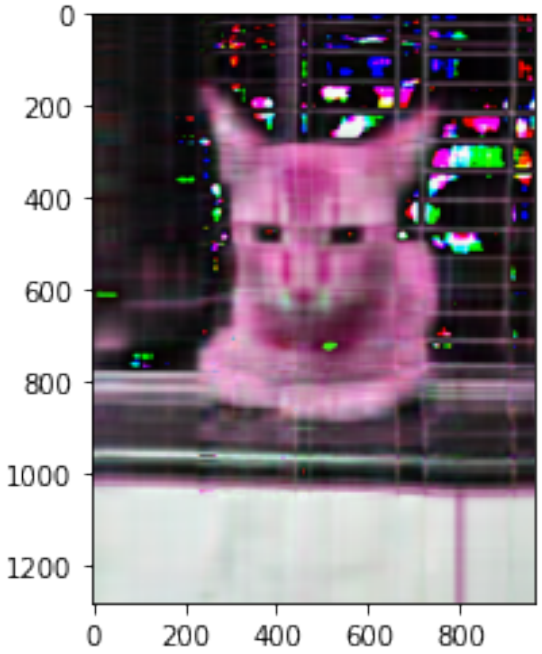
Let $A \in \mathbb{R}^n \times \mathbb{R}^d = \sum_{i=1}^{r} \sigma_i u_i v_i^T$
For $k < r$, define $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$. Then for any matrix B of rank $k, ||A - A_k||_F \leq ||A - B||_F$
Proof:
Consider $||A - B||_F = ||U\Sigma V^T - B||_F = ||\Sigma - U^T B V||_F$. Call $U^T B V = N$
$\implies ||\Sigma - N||^2 = \sum |\sigma_i - N_{ii}|^2 + \sum |N_{i,j}|^2$
Hence, to achieve min. N should be such that $N_{ii} = \sigma_i$ for $i = 1, 2, .., k$ and $N_{i,j} = 0$ otherwise. This forces $B = A_k$

Applications

Image Compression

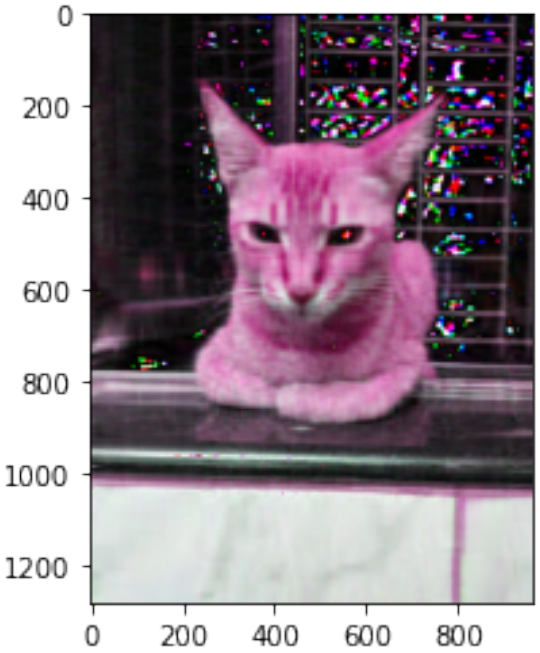


(a) no.comp=1 , variance=0.80

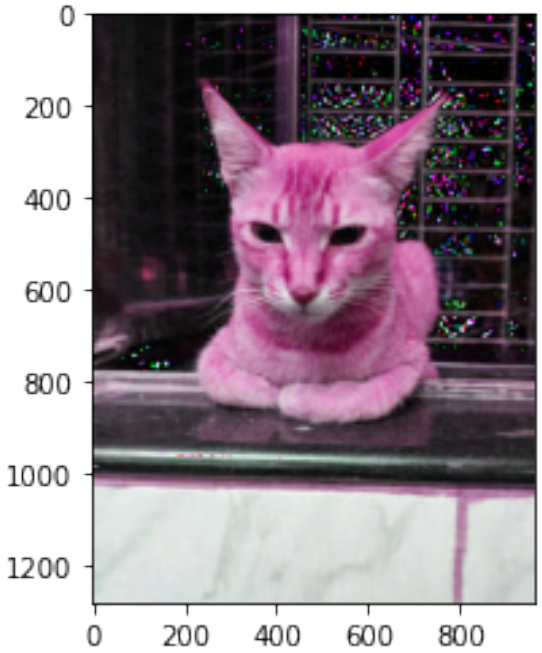


(b) no.comp=10,variance=0.96

continued

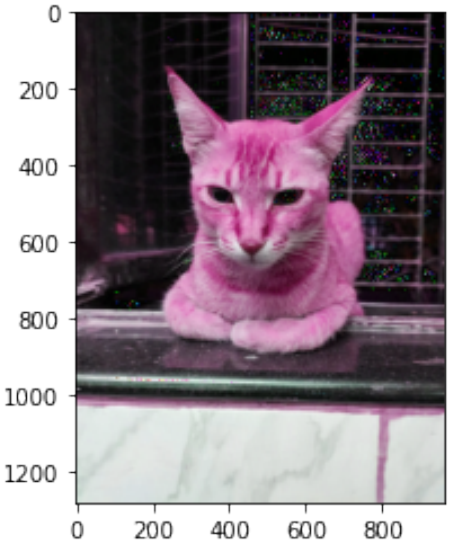


(a) no.comp=30 , variance=0.988

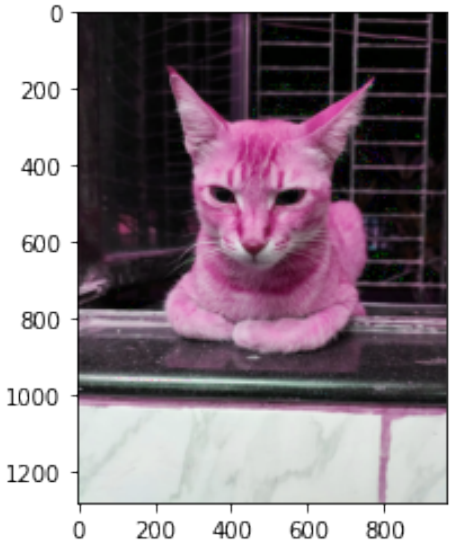


(b) no.comp=60,variance=0.994

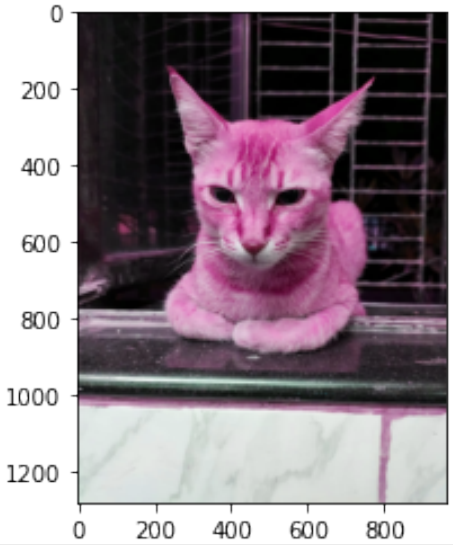
continued



(a) no.comp=120 , variance=0.998



(b) no.comp=240,variance=0.9996



Continued

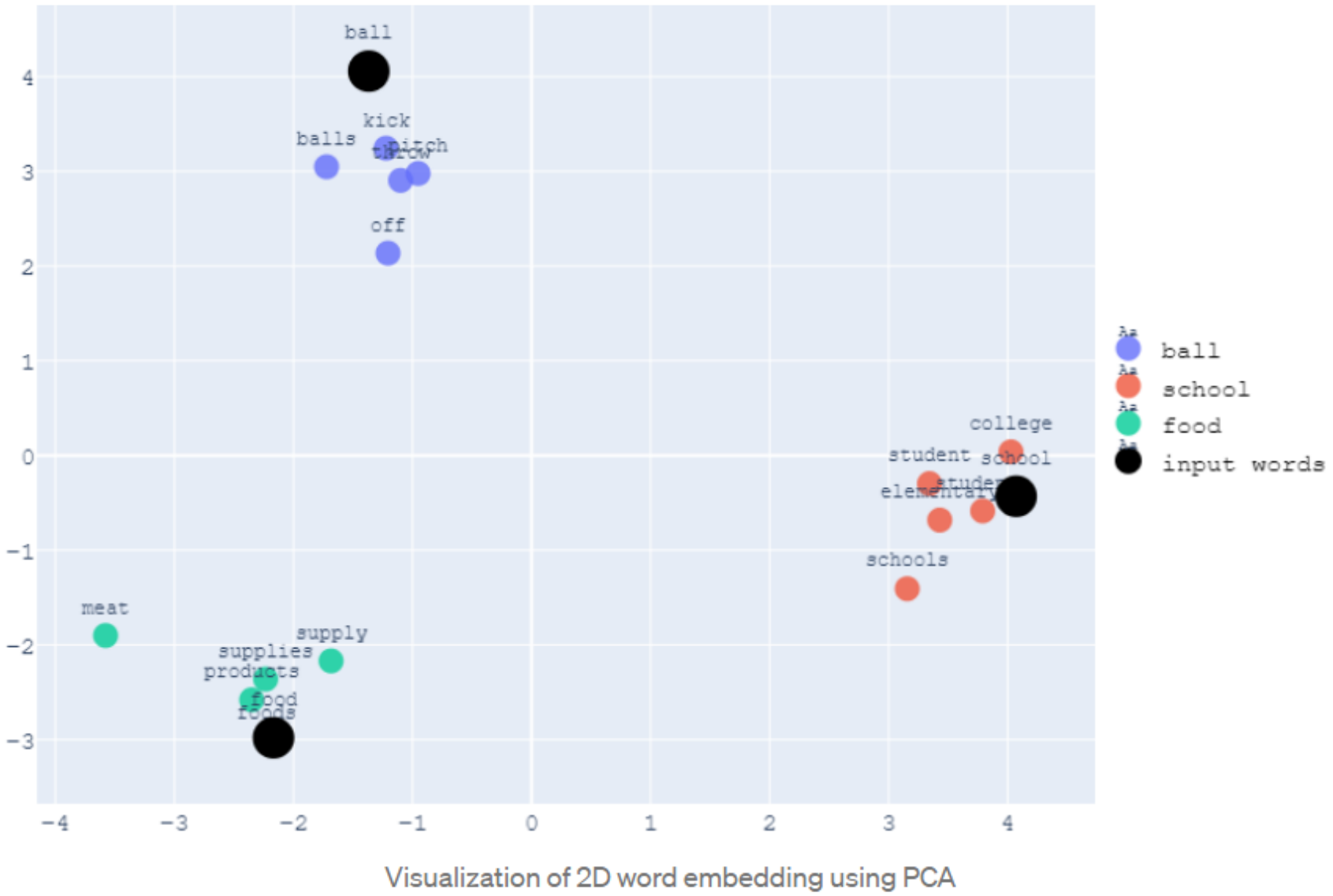
A Quick Analysis of Images.

The full image requires a matrix of size $1280 * 960 * 3$ which is about $3.6 * 10^6$.
Whereas The images presented here are in this order:

1. $(1280 + 960) * 3 = 6720$
2. $6720 * 10 = 67200$
3. $6720 * 30 = 201,600$
4. $6720 * 60 = 403,200$
5. $6720 * 120 = 806,400$
6. $6720 * 240 = 1,612,800$ approx(1.6 million)

It is very easily seen that the images even from 60 components itself becomes very similar to the original image and the data used is quite less.

Word Embeddings: Visualisation



References

- 'Singular Value Decomposition of Matrices' by Prof.Jugal K. Verma
- Foundations of Data Science by Avrin Blum, John Hopcroft and Ravindran Kannan
- Gilbert Strang: Introduction to Linear Algebra
- First Picture taken from Introduction to Linear Algebra by Gilbert Strang
- All pictures of the cat printed on google colab using matplotlib
- last picture of word embeddings and pca taken from the article "Visualizing Word Embeddings" by Ruben Winawastan in Towards Data Science

Markov Chains and Random Walks

Piyush Agarwal

IIT Bombay

16th April 2021

Contents

- Exposition to Markov Chains and Random Walks
- Sampling Methods
- Comparison to Electrical Networks
- Applications
  - World Wide Web

Markov Chains

Markov Chains are defined as random process having a discrete state space and satisfying the Markov Property.

Markov Property also known as memorylessness:
$P(x_{t+1}|x_t, x_{t-1}, x_{t-2}, .., x_1) = P(x_{t+1}|x_t)$
Also interpreted as "All past information stored in the current state"

Here we only talk Markov Chains which are Homogeneous:

- Homogeneous: probability of transition between two states is independent of time. $P(x_t = s|x_{t-1} = s')$ is same for all t

Random Walks

What is a Random Walk?

Given a graph G, start from any vertex v. Then from v chose any adjacent vertex at random and continue.

The random sequence of vertices $v, v_1, v_2, ..$ constitutes the random walk on the graph G. Random Walks have the Markov Property as the probability of moving from vertex x to y is independent of how the random walk reached x.

$p(y|x) = p(y|x, v_{n-1}, v_{n-2}, .., v_1)$ where v_i are the past vertices.

Define P to be the transition probability matrix i.e. $P = [p_{ij}]$ where p_{ij} is the probability of moving from vertex i to j.
Then if $p(t)$ is the probability distribution at step t, we have
$p(t+1)_x = \sum_y p_{yx} p(t)_y$

Continued

if $p(t) = [p(t)_{v_1}, p(t)_{v_2}, ..,]$ is written as a row vector we have $p(t+1) = p(t)P$

We define the stationary probability distribution of a random walk as the probability π such that $\pi P = \pi$

It can be shown that if our graph G is strongly connected we have that our random walk achieves the stationary probability distribution π that is unique and independent of the starting vertex v.

We prove a small lemma and show a few general cases.

Continued

Lemma

If our prob. density a satisfies $a_x p_{xy} = a_y p_{yx}$ then $a = \pi$.
Proof: $a_x p_{xy} = a_y p_{yx} = \sum_y a_x p_{xy} = \sum_y a_y p_{yx} \implies a_x = \sum_y a_y p_{yx}$
$\implies a = aP$ Hence $a = \pi$

For a undirected unweighted graph $G(v, E)$ with $|E| = m$
Define $\pi_x = \frac{d_x}{2m}$ and $p_{xy} = \frac{1}{d_x}$ d_x : Degree of x
Now $\pi_x p_{xy} = \frac{1}{2m} = \pi_y p_{yx}$. Hence π is the stationary probability distribution.

For an undirected graphs with weighted edges w_{xy}.
Define $p_{xy} = \frac{w_{xy}}{\sum_y w_{xy}}$ and $\pi_x = \frac{\sum_y w_{xy}}{\sum_{x,y} w_{xy}}$
Here as well, $p_{xy}\pi_x = p_{yx}\pi_y$

Monte Carlo Simulations via Random Walks

Given a multivariate probability distribution, $\boldsymbol{p}$, Random Walks can be used to sample points as per $\boldsymbol{p}$.
A graph G is constructed with vertices same as the support of $\boldsymbol{p}$ and Transition probability P can designed such that the stationary probability π is same as $\boldsymbol{p}$.

There are two algorithms which serve this purpose.

- Metropolois Hasting Algorithm
- Gibbs Sampling Formula

We'll state the two algorithms. [These work when the number of states is finite but too large to compute all cases. like exponential number of states]

Metropolis Hasting Algorithm

A state x with $\boldsymbol{p}(x) \neq 0$ is of the form $(x_1, x_2, .., x_d)$ with each $x_i \in \{0, 1, 2, .., N\}$. The graph G can be considered as a lattice Graph and the Random Walk is carried out as follows:

Let r be the maximum degree in the graph G. At any state i, choose a neighbour j with prob. $\frac{1}{r}$. Now, if $p_j > p_i$, move to j.
Otherwise,move to j with prob. $\frac{p_j}{p_i}$ and stay back at i with $1 - \frac{p_j}{p_i}$.

Hence $p_{ij} = \frac{1}{r} min(1, \frac{p_j}{p_i})$

$$p_{ii} = 1 - \sum_{j \neq i} p_{ij}$$

Now, $p_i p_{ij} = \frac{p_i}{r} min(1, \frac{p_j}{p_i}) = \frac{1}{r} min(p_i, p_j) = \frac{p_j}{r} min(1, \frac{p_i}{p_j}) = p_j p_{ji}$

By a Previous Lemma, we have that the stationary prob. distribution is same as $\boldsymbol{p}$

Gibbs Sampling Algorithm

Let $\boldsymbol{p}(x)$, $x = (x_1, x_2, .., x_d)$ be the target distribution. By the Gibbs Sampling Algorithm, all points which differ in only one coordinate are connected by an edge. Hence, the graph in the case of the Gibbs Sampling is a sort of lattice with cliques along each coordinate line.

Let $\boldsymbol{x} = (x_1, .., x_d)$ and $\boldsymbol{y} = (y_1, y_2, .., y_d)$ be two adjacent vertices and WLOG, assume $x_1 \neq y_1$.
Define $p_{\boldsymbol{xy}} = \frac{1}{d} p(y_1 | x_2, x_3, .., x_d) = \frac{1}{d} \frac{p(\boldsymbol{y})}{p(x_2, x_3, .., x_d)}$

Also $\quad p_{\boldsymbol{yx}} = \frac{1}{d} p(x_1 | y_2, y_3, .., y_d) = \frac{1}{d} \frac{p(\boldsymbol{x})}{p(x_2, x_3, .., x_d)}$

Now, $p_{\boldsymbol{xy}} p(\boldsymbol{x}) = \frac{1}{d} \frac{p(\boldsymbol{x}) p(\boldsymbol{y})}{p(x_2, x_3, .., x_d)} = p(\boldsymbol{y}) p_{\boldsymbol{yx}}$

Again, from the same lemma we have that the stationary distribution is same as $\boldsymbol{p}$.
Note that $\frac{1}{d}$ comes from $\sum_{\boldsymbol{y}} p_{\boldsymbol{xy}} = 1$

Random Walks and Electrical Networks

Given an electrical network one can convert it into a Markov Chain with a fixed transition probability and stationary probability.
Also given a Markov Chain one can convert it into an equivalent network.
[I think there would be certain restrictions on the Markov Chains]
Suppose G be an electrical network with r_{xy} being the resistance between x, y. Work with $c_{xy} = \frac{1}{r_{xy}}$ to carry out the conversion.

Define for the random walk, $p_{xy} = \frac{c_{xy}}{c_x}$ where $\quad c_x = \sum_y c_{xy}$

$$\pi_x = \frac{c_x}{c_0} \quad \text{where} \quad c_0 = \sum_x c_x$$

Now, $\pi_x p_{xy} = \frac{c_x}{c_0}\frac{c_{xy}}{c_x} = \frac{c_{xy}}{c_0} = \frac{c_{yx}}{c_0} = \frac{c_y}{c_0}\frac{c_{yx}}{c_0} = \pi_y p_{yx}$

And, $\sum_x \pi_x = 1$

Hence from a previous lemma, we have that π is the stationary distribution for a random walk.

Continued

Now, given a Markov Chain G we need to covert it to an equivalent electrical network. The Markov Chain will have certain restrictions:

- There cannot be self loops.
- The graph has to be undirected.
- There is also an issue of scale.

A valid Markov Chain gives an entire class of electrical networks.

We have $c_{xy} = p_{xy}c_x = p_{xy}\pi_x c_0$ But, given just the chain, there is no way of discerning c_0, Hence, can consider a standard system with $c_0 = 1$

Analogies between Random Walks and Electrical Current

Electrical Networks	Random Walks
Voltage at point x, v_x	Probability of Random Walk originating at x and reaching a before b
current in edge xy, i_{xy}	Net Frequency of Random Walks from a to b using the edge xy
$\frac{c_{eff}}{c_a}$	escape probability, p_{esc}

Note that:

- In the First Case $v_a = 1$ and $v_b = 0$
- c_{eff} is the effective conductance between a and b and p_{esc} is the proabaility with which a random walk starting at a reaches b before returning back to a

Properties of Random Walks

Hitting Time

The Expected Time taken for a Random Walk to move from vertex x to vertex y is defined as the hitting time h_{xy}. More generally, h_x can be defined as the expected time taken to reach x from a given probability distribution.

Note that the hitting time:

- is not symmetric. $h_{xy} \neq h_{yx}$
- can arbitrarily increase or decrease on changing the number of edges.

We'll find out the hitting time for some general graphs. [All the graphs have n vertices unless otherwise mentioned] [Linearity of Expectation comes in very handy]

- Clique- $h_{j,n} = \frac{1}{n-1} + \frac{1}{n-1}\sum_{i=1,i\neq j}^{n-1} 1 + h_{i,n} \quad \forall j = \{1, 2, 3, ., n-1\}$
  Solving these set of $n-1$ equations gives $h_{j,n} = n-1$

Continued

- Path- $(1, 2, 3, .., n)$ We proceed recursively to obtain $h_{i,n}$
  Clearly, $h_{n,n} = 0$
  $h_{1,n} = 1 + h_{2,n}$ and $h_{2,n} = \frac{1+h_{1,n}}{2} + \frac{1+h_{3,n}}{2} \implies h_{2,n} = 3 + h_{3,n}$
  Similarly, $h_{3,n} = \frac{1+h_{2,n}}{2} + \frac{1+h_{4,n}}{2} \implies h_{3,n} = 5 + h_{4,n}$
  Hence, we have $h_{i,n} = (2i - 1) + h_{i+1,n} \forall i \in \{1, 2, 3, .., n - 1\}$
  Solving, this gives
  $h_{i,n} = \sum_{j=i}^{n-1} 2j - 1 = (n - i)(n + i - 2) \implies h_{1,n} = (n - 1)^2$

- Cycle (1,2,3,..,n,1) Using similar ideas to find $h_{1,n}$
  We have $h_{i,n} = 1 + \frac{h_{i-1,n}+h_{i+1,n}}{2} \forall i \in \{1, 2, 3, .., n - 1\}$ where $h_{0,n} = h_{n,n} = 0$ On Solving, we have $h_{i,n} = i + \frac{i}{i+1} h_{i+1,n}$
  This gives, $h_{n-1,n} = n - 1$ By Symmetry $h_{1,n} = h_{n-1,n} = n - 1$

Page Rank Algorithm

- The Page Rank Algorithm employs Random Walks to rank the web pages in order and display them to the user.
- In the most basic version, the Algorithm ranks the web pages based on the number of hyper text links directed to them by other important web pages.
- Essentially, the Web can be seen as a digraph with web pages acting as vertices and a hyper text link on a web page to the other web page acting as a directed edge of the graph.
- Start with any probability distribution on the digraph and carry out a random walk. The stationary probability $\boldsymbol{p}$ is the ranking of the web pages.

Continued

Clearly,there are some problems associated with such a random walk:

- The Random Walk may get stuck in a part of the graph which has no outgoing edges.
- There my be web pages or connected components which do not have any incoming edges. These web pages are then ranked by the walk.

These problems are dealt with by introducing a restart condition.

Suppose the Random Walk is at vertex u:

- With prob r, the random walk moves to any vertex v in the graph G.
- With prob $1 - r$, the walk moves to a vertex adjacent to u

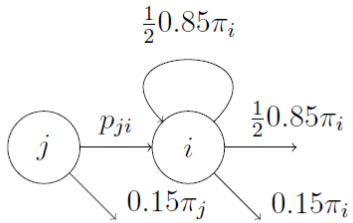


Figure: Page Rank with r=0.15

References

- Foundations of Data Science by Avril Blum, Ravindra Kanan, John Hopcroft
- CPSC 340: Data Mining Machine Learning https://www.cs.ubc.ca/ schmidtm/Courses/LecturesOnML/pageRank.pd
- Markov chain - Wikipedia https://en.wikipedia.org/wiki/Markov_chain