15-859: Algorithms for Big Data
Recitation 1 – Preliminaries

September 8, 2017

Outline

- Linear algebra – geometric interpretation
- Probability – inequalities and bounds
- Some interesting stuff

Linear Algebra

Vectors, vector spaces, matrices, SVD

Vectors

- $\mathbf{x} = (x_1, x_2, \dots, x_d) \in R^d$ (each x_i is a component)
  - A point in d-dimensional space
- Norm or magnitude $\|\mathbf{x}\| = (x^T x)^{1/2} = (x_1{}^2 + x_2{}^2 + \dots + x_d{}^2)^{½}$
  - Length of the vector (Pythagorean theorem)
- Zero vector (norm zero), unit vector (norm one)
- Inner product $<\mathbf{x}, \mathbf{y}> = x_1 y_1 + \dots x_d y_d$
  - Result is a scalar
  - $\|\mathbf{x}\| = (<\mathbf{x}, \mathbf{x}>)^{1/2}$
  - $<\mathbf{x}, \mathbf{y}> = 0$ implies $\mathbf{x} \perp \mathbf{y}$

Vector spaces

- Space where vectors live
- Formally, a collection of vectors which is closed under linear combination
  - If {**x**, **y**} are in the space, so is a**x**+b**y** for any scalars a, b $\in$ R
  - Should always contain zero vector
- Examples: {0}, R^d, the line x = 3y in R^2

Span and basis

- A set of vectors is said to span a vector space if one can write any vector in the vector space as a linear combination of the set
- $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ span the space $\{\sum a_i\mathbf{x}_i \mid a_i \in R\}$
- This set is called the basis set
- Examples
  - The vectors $\{(0,1), (1,0)\}$ span R^2
  - $\{(1, 1)\}$ spans $x=y$ which is a subspace of R^2
  - The vector $\{(0,1), (0,1), (1,1)\}$ also span R^2

Linear independence and orthonormality

- Linear independence – a notion to remove redundancy in the basis
  - $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ are linearly independent iff the only solution to $\sum a_i\mathbf{x}_i = 0$ is $a_1 = a_2 = \ldots = a_n = 0$.
  - Cannot express any vector $\mathbf{x}_i$ as a linear combination of the others
- Dimensionality of a vector space is the maximum number of linearly independent basis vectors
- Orthonormal basis
  - $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ is orthonormal basis if $< \mathbf{x}_i , \mathbf{x}_j >= 1$ if i=j and 0 otherwise
  - Coordinate axes for the vector space
- Example: The basis {(0, 1), (1,1)} for R^2 is linear independent but not orthonormal.

Matrices

- Operator which transforms vectors from one vector space to another
  - $\mathbf{y} = A\mathbf{x}$
- The operator is linear, that is

$$A\,(a\mathbf{x} + b\mathbf{y}) = a(A\mathbf{x}) + b\,(A\mathbf{y})$$

- The result of applying the operator is a linear combination of the column vectors
  - Thus, $A\mathbf{x} = \mathbf{b}$ has an exact solution iff $\mathbf{b}$ is in the column space of A
- Eigen vectors of A are the special vectors are the special vectors $\mathbf{x}$ which satisfy

$$A\mathbf{x} = \lambda\mathbf{x} \text{ for some } \lambda$$

  - λ is called the eigen value and $\mathbf{x}$ is the eigen vector

- How do we visualize the transformation geometrically?

Visualizing the matrix operator – special cases

- Identity matrix
  - Square matrix with diagonal elements 1 and non-diagonal elements 0
  - The transformed vector A**x** is same **x**
- Diagonal matrix
  - Square matrix with non-diagonal elements 0
  - i^{th} component in A**x** is a scaled version of x_i (scaling = A_{ii})
- Orthonormal (or rotation) matrix
  - Matrix whose columns $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ are such that $< \mathbf{a}_i , \mathbf{a}_j >= 1$ if i=j and 0 otherwise. That is, $A^T A = I$
  - Rotates the vector
  - Preserves norms $\|A\mathbf{x}\| = \|\mathbf{x}\|$ (why?)

General case – Singular Value Decomposition

- We have a rectangular matrix $A \in R^{m \times n}$
- It can be decomposed as

$$A = UDV^T$$

- U and V are orthonormal, i.e., $U^TU = V^TV = I$ and D is a diagonal matrix containing singular values
  - Number of non-zero diagonal elements in D = rank of A
- Provides a nice way to understand the operator A
  - Rotation in n-dimensional space, scaling, rotation in m-dimensional space
- Can be computed in $O(\min\{mn^2, m^2n\})$ time (or better using fast matrix multiplication)

Computation of SVD

- Let m>n, i.e., A is a skinny matrix. How to compute SVD of A in $O(mn^2)$ time?
- Step 1: Compute A^TA in $O(mn^2)$ time.
- Step 2: Get eigenvalue decomposition of A^TA in $O(n^3)$ or better. Why do this?
  - If the SVD of A is UDV^T, then $A^TA = VDU^TUDV^T$
  - That is, the eigenvalues of AA^T are the square of the singular values of A and the eigenvectors are the right singular space
- Step 3: $U = AVD^{-1}$ in $O(mn^2)$ time.

Example problem 1

- If singular values of $A \in R^{nxn}$ all lie in [a, b], prove that

$$a\|\mathbf{x}\| \leq \|A\mathbf{x}\| \leq b\|\mathbf{x}\|$$

Solution:

- Let $A = UDV^T$
- $\|A\mathbf{x}\| = \|UDV^T\mathbf{x}\|$
- Let $\mathbf{y} = V^T\mathbf{x}$. (note: $\|\mathbf{y}\| = \|\mathbf{x}\|$)
  - We can do this because we prove this for every **x**
- $\|A\mathbf{x}\| = \|UD\mathbf{y}\| = \|D\mathbf{y}\|$
- As singular values lie in [a, b], $a\|\mathbf{y}\| \leq \|D\mathbf{y}\| \leq b\|\mathbf{y}\|$

Example problem 2

- Prove that Frobenius norm of a matrix ($\|A\|_F = (\sum_i \sum_j A_{ij}^2)^{1/2}$) is always greater than or equal to the operator norm ($\|A\|_2 = \sup_{\mathbf{x}} \|A\mathbf{x}\|/\|\mathbf{x}\|$). Solution:

Solution:

- Let $\mathbf{x} = \Sigma_j c_j \mathbf{e}_j$ for coefficients $c_1, .. c_d$
- Let $\|\mathbf{x}\|_2 = 1$. Then, $\Sigma_j |c_j|^2 = 1$
- $\|A\mathbf{x}\|_2^2 = \| \Sigma_j c_j A\mathbf{e}_j \|_2^2$
- By triangle inequality, this is $\leq (\Sigma_j |c_j| \|A\mathbf{e}_j\|_2)^2$
- Which is $\leq (\Sigma_j |c_j|^2)(\|A\mathbf{e}_j\|_2^2)$ by Cauchy-Schwarz inequality
- Which is $\|A\mathbf{e}_j\|_2^2 = \|A\|_F$

Probability

Useful inequalities

Expectation and variance

- Let X be a random variable
- Expectation $E[X] = \sum_j P(X=j).j$ (discrete)
- Variance $Var[X] = E[(X-E[X])^2] = E[X^2] – E[X]^2$
- In general, k^{th} order moment is $E[|X-E[X]|^k]$

Markov inequality

- For a non-negative random variable X and non-negative t,

$$\Pr[X \geq t] \leq E[X]/t$$

Proof:

- We'll show for continuous r.v, but proof is similar for discrete r.v
- $E[X] = \int_0^\infty x\, p(x)\, dx = \int_0^t x\, p(x)\, dx + \int_t^\infty x\, p(x)\, dx$
- $E[X] \leq \int_t^\infty x\, p(x)\, dx$
- $\leq t \,.\, \int_t^\infty p(x)\, dx = t \,.\, \Pr[X \geq t]$

Chebyshev inequality

- Let $\mu = E[X]$ and $\sigma^2 = Var[X]$. Then,

$$\Pr[|X-\mu| \geq t] \leq \sigma^2/t^2$$

Proof:

- $\Pr[|X-\mu| \geq t] = \Pr[|X-\mu|^2 \geq t^2]$
- By Markov inequality, $\Pr[|X-\mu|^2 \geq t^2] \leq E[|X-\mu|^2]/ t^2 = \sigma^2/t^2$

Chernoff bound

- For independent random variables $X_1, X_2, .. X_n$, with $X = \sum_i X_i$

$$\Pr[X \geq a] \leq \min_{t\geq 0} e^{-ta} \prod_i E[e^{tX_i}]$$

$$\Pr[X \leq a] \leq \min_{t\geq 0} e^{ta} \prod_i E[e^{-tX_i}]$$

Proof:

- Key idea: Apply Markov inequality on e^{tX}
- $\Pr[X \geq a] = \Pr[e^{tX} \geq e^{ta}] \leq e^{-ta} E[e^{tX}]$
- By independence, this is $e^{-ta} \prod_i E[e^{tX_i}]$
- This is true for every positive t, so take infimum to get the best bound

Chernoff bound (i.i.d Bernoulli)

- For independent Bernoulli random variables $X_1, X_2, .. X_n$ each having probability p of being equal to 1, if X is the sum $\sum_i X_i$,

$$\Pr(X > (1+\delta)\mu) < \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}$$

- A more useful but loose bound is:

$$\Pr(X \geq (1+\delta)\mu) \leq e^{-\frac{\delta^2 \mu}{3}}, \qquad 0 \leq \delta \leq 1$$

Interesting stuff

Just one this time…

Hadamard matrix–vector product in O(nlogn)

- Let H_k be the Hadamard matrix with 2^k rows and columns
- Observe that $H_k = \begin{bmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{bmatrix}$
- Let $\mathbf{x}$ be $(\mathbf{x}_u, \mathbf{x}_l)$ – the upper and lower parts contain n/2 entries each
- Then, $H_k\ \mathbf{x} = \begin{bmatrix} H_{k-1}\ \mathbf{x}_u + H_{k-1}\ \mathbf{x}_l \\ H_{k-1}\ \mathbf{x}_u - H_{k-1}\ \mathbf{x}_l \end{bmatrix}$
- Once $H_{k-1}\ \mathbf{x}_l$ and $H_{k-1}\ \mathbf{x}_u$ have been computed in T(n/2) time, we perform O(n) element wise addition/subtraction to solve the original problem
- Thus, T(n) = 2T(n/2) + O(n) which gives O(n log n) time complexity