Linear Algebra & Matlab

10-701/15-781, Recitation

Jan 21, 2010

Ni Lao

- Part I Some Linear Algebra
  - Might be useful to HW2 and later courses

Basics

- We will use lower case letters for vectors, and upper case letters from matrixes. There elements are referred by x_i, $A_{i,j}$. Refer A's column vectors as A_j
- AB
  - still remember what is matrix multiplication?
- $A=A^T$
  - transpose and symmetric matrix
- $a \bullet b = a^T b$, $a \bullet b = |a|_2$
  - inner product, vectors are also matrixes
- $AA^{-1}=I$
  - Inverse and the identity matrix
- $tr(A)=diag(A)^T 1$
  - trace, and the diagonal of a matrix

Basis and Space

- $$span(x_1, x_2, x_3) = \{a_1 x_1 + a_2 x_2 + a_3 x_3 \mid a_i \in R)$$
  – the span of a set of vectors is a subspace in the R^d space, assuming x_i are vectors in R^d space

- *col(A)={x|x=Ab}*
  – A's column space is the span of A's column vectors

- row(A)={x|x=A^Tb}
  – A's rows space is the span of A's rows vectors

- Basis
  – A basis B of a space V is a linearly independent subset of V that spans (or generates) V

- e_i=(0,0,…1,..,0)
  – the standard basis

Unitary Matrix

- If $a \bullet b=0$, $|a|_2 != 0$, $|b|_2 != 0$,
  - then *a* and *b* are orthogonal

- If n-by-n matrix *A*, $A^T A=I$
  - then A is an unitary matrix
  - $|A_i|_2=1$ for any i
  - and $A_i \bullet A_j=0$, for $i != j$

- *If A is unitary, then A^T is also unitary*

Rank of a Matrix

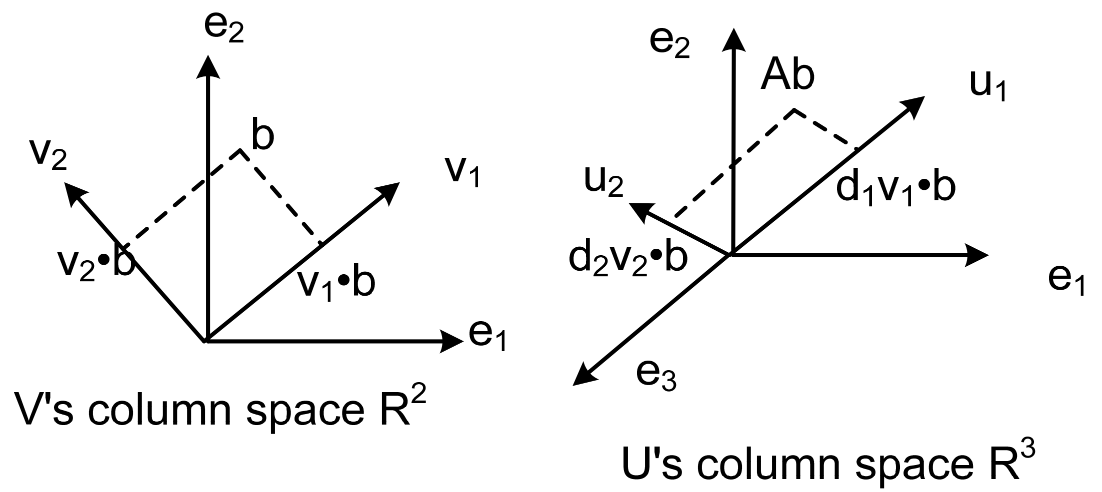
- rank(A) (the rank of a m-by-n matrix A) is
  - The maximal number of linearly independent columns
  - The maximal number of linearly independent rows
  - The dimension of col(A)
  - The dimension of row(A)

- If A is n by m, then
  - rank(A)<= min(m,n)
  - If n=rank(A), then A has full row rank
  - If m=rank(A), then A has full column rank

Singular Value Decomposition (SVD)

- Any matrix A can be decompose as $A=UDV^T$, where
  - where D is diagonal, with d=rank(A) non-zero elements
  - U and V are unitary matrices
  - The fist d rows of U are orthogonal basis for col(A)
  - The fist d rows of V are orthogonal basis for row(A)

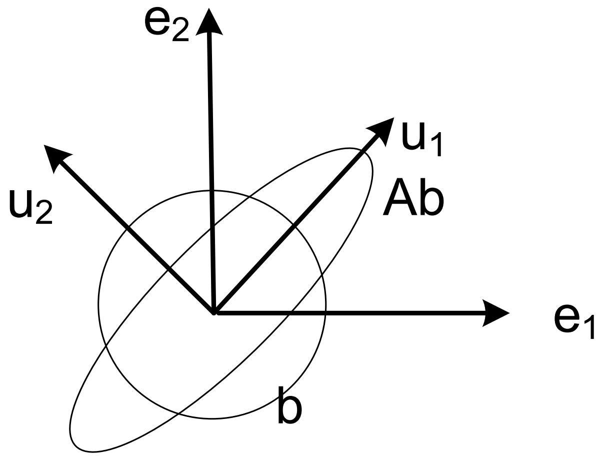
- Re-interpreting Ab
  - Decompose b by V basis
  - Scale it by diag(D)
  - Then map it to the space spanned by U basis



Eigen Value Decomposition

- Any symmetric matrix A can be decompose as $A=UDU^T$, where
  - where D is diagonal, with d=rank(A) non-zero elements
  - The fist d rows of U are orthogonal basis for col(A)=row(A)

- Re-interpreting Ab
  - first stretch b along the direction of u_1 by d_1 times
  - Then further stretch it along the direction of u_2 by d_2 times



U's column space R^2

Inversing a Low Rank Covariance Matrix

- In many applications (e.g. linear regression, Gaussian model) we need to calculate the inverse of covariance matrix $X^T X + \lambda I$
  - *where each row of X is* a data sample
  - *I is an identity matrix for regularization*

- If the number of feature is huge (e.g. each sample is an image, #sample n<<#feature d)
  - then X is an very wide and short matrix
  - inversing $X^T X + \lambda I$ becomes an problem
    - the complexity of matrix inversion is generally $O(n^3)$
    - Matlab can comfortably solve matrix with d=thousand, but not much more than that

Inversing a Low Rank Covariance Matrix

- With the help of SVD, we actually don't need to explicitly inverse $X^TX+\lambda I$
  - *Decompose* $X=UDV^T$
  - *Then* $X^TX+\lambda I=VDU^TUDV^T+\lambda I= V(D^2+\lambda I)V^T$
  - *Since* $V(D^2+\lambda I)V^TV(D^2+\lambda I)^{-1}V^T=I$
  - *We know that* $(X^TX+\lambda I)^{-1}= V(D^2+\lambda I)^{-1}V^T$
    - *Inversing a diagonal matrix* $D^2+\lambda I$ *is trivial*

- Part II Matlab
  - Might be useful to HW2

Matlab

- Very easy to do matrix manipulation in Matlab
- Available for installs by contacting help+@cs.cmu.edu
- If this is your first time using Matlab
  - Strongly suggest you go through the “Getting Started” part of Matlab help
  - Many useful basic syntax

Making Matrix

- A=[1 2 3; 4 5 6; 7 8 9]
- A=ones(m,n)
- A=zeros(m,n)
- A=eye(n)
- A=diag([1 2 3])

Referencing Matrix

- A(i,j)
  - reference a single element
- A(i,:), A(:,j)
  - reference a whole row/column
- b=1:3:100; A(b,:)
  - using vector as index
- b=diag(A)
  - reference the diagonal vector

Matrix Manipulation

- C=A';
  - transpose
- C=A+B; D=A*B;

- D=A^3
  - Equal to A*A*A
- x=A\b;   x=b/A
  - multiply the inverse of a matrix
- D=A.*B; D=A./B; D=A.\B; D=A.^3;
  - Point wise multiplication/division/power

Matrix Decomposition

- [U,S,V] = svd(X)
- [V,D] = eig(A)

- The End
- Thanks