CS155 Recitation: Linear Algebra

Fabian Boemer

January 14, 2016

Outline

20 Least Squares Problem

Linear Space

A vector space over a field $\mathbb{F}$ is a set V with two operations:

- $+ : V \times V \to V$
- $- : V \times V \to V$

satisfying the following axioms

- $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
- $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
- zero vector $\mathbf{0}$ such that $\mathbf{v} + \mathbf{0} = v$
- additive inverse $-\mathbf{v}$ such that $\mathbf{v} + (-\mathbf{v}) = 0$
- associativity $a(b\mathbf{v}) = (ab)\mathbf{v}$
- identity element $1\mathbf{v} = \mathbf{v}$
- distributivity $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$
- distributivity $(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$

Linear Dependence and Independence

- A set of vectors $S = \{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}\}$ is *linearly dependent* if $\exists \alpha_1, \ldots, \alpha_n$ such that $\alpha_1 \mathbf{v_1} + \ldots + \alpha_n \mathbf{v_n} = 0$.
- A set of vectors $S = \{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}\}$ is *linearly independent* if it is not linearly dependent.
- The *span* of a set S is $\text{span}(S) = \left\{ \sum_{i=1}^{k} \alpha_i \mathbf{v_i} \middle| k \in \mathbb{N}, v_i \in S, \alpha_i \in \mathbb{F} \right\}$.
- A set S is a *basis* for a vector space V if S is linearly independent and spans V.
  - Every $\mathbf{v} \in V$ can be uniquely written as $\sum_{i=1}^{n} \alpha_i \mathbf{v_i}$ where $\mathbf{v_i}$ are the basis elements of V.
- The *dimension* n of a (finite-dimensional) linear space V is the length of any basis for V.

Linear map

- Let V and W be vector spaces over the same field $\mathbb{F}$. $L : V \to W$ is a linear map if, $\forall u, v \in V, \alpha \in \mathbb{F}$,
  - $L(u + v) = L(u) + L(v)$
  - $L(\alpha u) + \alpha L(u)$
- In this course, we let $\mathbb{F} = \mathbb{R}$.
- Any linear map L can be completely determined by its action on the basis $\{v_1, \ldots, v_n\}$ for V as linear combinations on the basis $\{w_1, \ldots, w_n\}$ for W.
  - Usually, $v_i = e_i = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$, with 1 at i'th location.

Matrix and Vectors

- A *matrix* $M^{m\times n}$ is a rectangular array of numbers (which specifices the action of a linear operator on the basis elements of $\mathbb{R}^n$).

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

- A *column vector* $x \in \mathbb{R}^n$:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Matrix Multiplication

- If $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, then $C = AB \in \mathbb{R}^{m \times p}$ with

$$C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}$$

Properties of Matrix Multiplication:

- Associativity $(AB)C = A(BC)$
- Distributive: $A(B + C) = AB + AC$
- Non-commutative (in general): $AB \neq BA$.

Operators and properties

Transpose: if $A \in \mathbb{R}^{m\times n}$, then $A^T \in \mathbb{R}^{n\times m} : (A^T)_{ij} = A_{ji}$.

Properties:

- $(A^T)^T = A$
- $(AB)^T = B^T A^T$
- $(A+B)^T = A^T + B^T$

Trace: if $A \in \mathbb{R}^{m\times n}$, then $\mathrm{tr}(A) = \sum_{i=1}^{n} A_{ii}$

Properties:

- $\mathrm{tr}(A^T = \mathrm{tr}(A)$
- $\mathrm{tr}(A+B) = \mathrm{tr}(A) + \mathrm{tr}(B)$
- $\mathrm{tr}(\lambda A) = \lambda\,\mathrm{tr}(A)$
- If AB is square, $\mathrm{tr}(AB) = \mathrm{tr}(BA)$.

Special types of matrices

- Identity matrix: $I = I_n \in \mathbb{R}^{n \times n}$
  $$\forall A \in \mathbb{R}^{n \times n} : AI_n = I_m A = A$$
- Diagonal matrix: $D = \text{diag}(d_1, d_2, \ldots, d_n)$:
  $$D_{ij} = \begin{cases} d_i & j = i \\ 0 & \text{otherwise} \end{cases}$$
- Symmetric matrices: $A \in \mathbb{R}^{n \times n}$ is symmetric if $A = A^T$.
- Orthogonal matrices: $U \in \mathbb{R}^{n \times n}$ is orthogonal if $UU^T = I = U^T U$.

Vector Norms

A *norm* of a vector space V is a function $\|\cdot\| : V \to \mathbb{R}^+$ such that:

- $\|x\| = 0 \iff x = 0$
- $\|\alpha x\| = |\alpha| \cdot \|x\|$
- $\|x + y\| \leq \|x\| + \|y\|$

The norm of a vector is a measure of its "length" or "magnitude". The most common is Euclidean (ℓ_2) norm.

- ℓ_p norm: $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$
- ℓ_2 norm: $\|x\|_2 = \left(\sum_{i=1}^n |x_i|^2\right)^{1/2}$
  used in ridge regression $\|y - X\beta\|^2 + \lambda\|\beta\|_2^2$
- ℓ_1 norm: $\|x\|_1 = \sum_{i=1}^n |x_i|$.
  used in lasso regression $\|y - X\beta\|^2 + \lambda\|\beta\|_1$
- ℓ_∞ norm: $\|x\|_\infty = \max_i |x_i|$.

Rank of a Matrix

- If $A \in \mathbb{R}^{m \times n}$, then $\text{rank}(A) = \dim(\text{span}(\text{cols}(A))$ is the maximum number of linearly independent columns (or rows)
- Properties
  - $\text{rank}(A) = \text{rank}(A^T)$
  - $\text{rank}(A) \leq \min\{m, n\}$
  - $\text{rank}(AB) \leq \min\{\text{rank}(A), \text{rank}(B)\}$
  - $\text{rank}(A + B) \leq \text{rank}(A) + \text{rank}(B)$

Inverse of a Matrix

- If $A \in \mathbb{R}^{n \times n}$ is *invertible* if $\exists B \in \mathbb{R}^{n \times n}$ such that $AB = I_n = BA$. B is the *inverse* of A, and written $B = A^{-1}$
- Properties (if A^{-1} exists)
  - $(A^{-1})^{-1} = A$
  - $(AB)^{-1} = B^{-1}A^{-1}$
  - $(A^{-1})^T = (A^T)^{-1}$
  - The inverse of an orthogonal matrix is its transpose

Eigenvalues and Eigenvectors

- $A \in \mathbb{R}^{n \times n}, \lambda \in \mathbb{F}$ is an *eigenvalue* of A with corresponding *eigenvector* $x \in \mathbb{F}^n (x \neq 0)$ if $Ax = \lambda x$.
- Every finite-dimensional complex-valued ($\mathbb{F} = \mathbb{C}$) linear operator has an eigenvalue.
- Properties
  - $\text{tr}(A) = \sum_{i=1}^{n} \lambda_i$
  - $\det(A) = \prod_{i=1}^{n} \lambda_i$
  - $\text{rank}(A) = |\{1 \leq i \leq n | \lambda_i \neq 0\}|$

Determinant

- Determinant: $\det(A) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma_i}$
- Properties:
  - $\det(I) = 1$
  - $\det(\lambda A) = \lambda \det(A)$
  - $\det(A^T) = \det(A)$
  - $\det(AB) = \det(A)\det(B)$
  - $\det(A) \neq 0 \iff A$ is invertible.
  - If A invertible, then $\det(A^{-1}) = \det(A)^{-1}$

Invertible Matrix Theorem

Invertible Matrix Theorem. Let $A \in \mathbb{R}^{n \times n}$. The following are equivalent:

- A is invertible
- nullspace $\{x | Ax = 0\} = \{0\}$
- The columns of A form a linearly independent set
- The columns of A span $\mathbb{R}^n$.
- $Ax = b$ has at least one solution for each $b \in \mathbb{R}^n$.
- A^T is invertible.
- $\det A \neq 0$

Singular Value Decomposition

- For $A \in \mathbb{R}^{n \times n}$, the *singular values* $\sigma(A) = \sqrt{\lambda(A^T A)}$.
- For $A \in \mathbb{R}^{m \times n}$, the *singular value decomposition (SVD)* is a factorization $A = U \Sigma V^T$ where U and V are orthogonal, Σ is diagonal, with $\Sigma_{ii} = \sigma_i(A)$, the i'th largest singular value of A.

Gradient

Let $f : \mathbb{R}^{m \times n} \to \mathbb{R}$. Then, given a matrix $A \in \mathbb{R}^{m \times n}$, the *gradient* $\nabla : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ of f is:

$$\nabla_A f(A) = \begin{bmatrix} \frac{\partial f(A)}{\partial A_{11}} & \frac{\partial f(A)}{\partial A_{12}} & \cdots & \frac{\partial f(A)}{\partial A_{1n}} \\ \frac{\partial f(A)}{\partial A_{21}} & \frac{\partial f(A)}{\partial A_{22}} & \cdots & \frac{\partial f(A)}{\partial A_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(A)}{\partial A_{m1}} & \frac{\partial f(A)}{\partial A_{m2}} & \cdots & \frac{\partial f(A)}{\partial A_{mn}} \end{bmatrix}$$

In particular, if A is the vector $x \in \mathbb{R}^n$,

$$\nabla_x f(x) = \begin{bmatrix} \frac{\partial f(A)}{\partial x_1} \\ \frac{\partial f(A)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}$$

The Hessian

Let $f : \mathbb{R}^n \to \mathbb{R}$. Then, given a vector $x \in \mathbb{R}^n$, the *Hessian* $\nabla^2 : \mathbb{R}^n \to \mathbb{R}^n$ of f is:

$$\nabla_x^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1\,\partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1\,\partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2\,\partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2\,\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n\,\partial x_1} & \frac{\partial^2 f(x)}{\partial x_n\,\partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}.$$

More Derivatives

- $\frac{\partial x^T a}{\partial x} = \frac{\partial a^T x}{\partial x} = a$
- $\frac{\partial a^T X b}{\partial X} = ab^T$
- $\frac{\partial a^T X a}{\partial X} = ba^T$
- $\frac{\partial a^T X a}{\partial X} = \frac{\partial a^T X^T a}{\partial X} = aa^T$

Least Squares Problem

Solve the following minimization problem:

$$\text{minimize } \|Ax - b\|_2^2$$

Note that

$$\begin{aligned}\|Ax - b\|_2^2 &= (Ax - b)^T(Ax - b) \\ &= x^T A^T Ax - 2b^T Ax + b^T b\end{aligned}$$

Least Squares Problem

Solve the following minimization problem:

$$\text{minimize } \|Ax - b\|_2^2$$

Note that

$$\begin{aligned}\|Ax - b\|_2^2 &= (Ax - b)^T(Ax - b) \\ &= x^T A^T Ax - 2b^T Ax + b^T b\end{aligned}$$

Taking the gradient with respect to x (and using the properties above):

$$\begin{aligned}\nabla_x(x^T A^T Ax - 2b^T Ax + b^T b) &= \nabla_x x^T A^T Ax - \nabla_x 2b^T Ax + \nabla_x b^T b \\ &= 2A^T Ax - 2A^T b\end{aligned}$$

Setting this to zero and solving for x yields the Moore-Penrose Pseudoinverse:

$$x = (A^T A)^{-1} A^T b$$

Moral of the Story

MATLAB, Numpy are optimized for fast matrix operations. Use matrix operations whenever possible, instead of nested 'for' loops.

```
N = 10000; d = 5; nu = 1
X = np.random.random((N, d));
w = np.random.random((d, 1));
y = np.random.random((N, 1));
```

How to compute $dw = -2 \cdot nu \cdot X^T \cdot (y - X \cdot w)$?

For loops:

```
y_minus_x_dot_w = [0 for i in range(N)]
dw = [0 for i in range(d)]
for i in range(N):
        dot = 0
        for j in range(d):
                dot += X[i][j] * w[j]
        y_minus_x_dot_w[i] = y[i] - dot
for i in range(d):
        dot = 0
        for j in range(N):
                dot += X[j][i] * y_minus_x_dot_w[j]
        dw[i] = -2 * nu * dot[0]
```

For loops:

```
y_minus_x_dot_w = [0 for i in range(N)]
dw = [0 for i in range(d)]
for i in range(N):
        dot = 0
        for j in range(d):
                dot += X[i][j] * w[j]
        y_minus_x_dot_w[i] = y[i] - dot
for i in range(d):
        dot = 0
        for j in range(N):
                dot += X[j][i] * y_minus_x_dot_w[j]
        dw[i] = -2 * nu * dot[0]
```

0.30 seconds.

Matrix operations:

```
dw = -2 * nu * X.T.dot((y - X.dot(w)))
```

For loops:

```
y_minus_x_dot_w = [0 for i in range(N)]
dw = [0 for i in range(d)]
for i in range(N):
        dot = 0
        for j in range(d):
                dot += X[i][j] * w[j]
        y_minus_x_dot_w[i] = y[i] - dot
for i in range(d):
        dot = 0
        for j in range(N):
                dot += X[j][i] * y_minus_x_dot_w[j]
        dw[i] = -2 * nu * dot[0]
```

0.30 seconds.

Matrix operations:

```
dw = -2 * nu * X.T.dot((y - X.dot(w)))
```

0.00006 seconds