Numerical Algorithms and Optimization

CS-2363-1
Monsoon 2024

Kahaan Shah

December 2, 2024

Contents

Chapter 1

Introduction

Lecture 1

1.1 Course Information

26 Aug. 11:50

1.1.1 Books

- Golun and Van Loan for Linear Algebra
- Trefthan (Numerical Linear Algebra)

1.2 Grading

- Midterm+Final (equal-ish weightage)
- 6-7 Assignments
- No tests

1.3 Floating Point Representation

1.3.1 Representation

We have $x = \pm d_0.d_1 \dots d_{n-1} \cdot \beta^e$ where $0 \leq d_i < \beta$ and $m < e < M$ where $m = -M$ usually. So we have $x = d_0 + \sum_i^{n-1} d_i \beta^{-i}$. This is normalised if $d_0 \neq 0$ or $d_0 = \dots d_{n-1} = 0$.

This is just scientific notation!!

1.3.2 Properties for the representation

1. We require that $fl(x) = x(1+\delta)$ and we want a bound on δ independent of x. So we have $|\delta| \leq \frac{1}{2}\beta^{1-n}$ for rounding and $\beta^{1-n} \leq \delta < 0$ for chopping. For example 5.73 with a δ of 0.005.
2. $x \circ y = (x \circ y)(1+\delta)$ So the machine epsilon is μ is such that $|\delta| \leq \mu$. The definition of this is such that $fl(1+\mu) > 1$. So in words it is the smallest floating point number we can represent that when added to 1 is greater than 1.

1.3.3 Precision

Consider x^{2^n}. We can do $x_1 = x_0^2, x_2 = x_1^2 \dots x_n = x_{n-1}^2$. But on a floating point machine we have $\hat{x}_1 = x_0^2(1+\delta_1), \hat{x}_2 = \hat{x}_1^2(1+\delta_2) \dots \hat{x}_n = \hat{x}_{n-1}^2(1+\delta_n)$. But this means we have $\hat{x}_n = x_0^{2n}(1+\delta_1)^{2^{n-1}}(1+\delta_2)^{2^{n-2}} \cdots = x_0^{2n}(1+\eta)^{2n} = (x_0(1+\eta))^{2n}$ by IVT for some $\eta < \mu$.

We need to be careful about the degree of precision we have when doing operations like subtraction. We have the condition number of $f(x)$ given by $c(f(x)) = \max |\frac{f(x)-f(x^\star)}{f(x)} / \frac{x-x^\star}{x}| \approx \max \frac{xf'(x)}{f(x)}$ if continuous and differentiable. If this is very large then not good.

Read what every computer scientist should know

Chapter 2

Polynomials

Lecture 2

28 Aug. 11:50

Example. Consider $M = 2$, $n = 3$, $m = -1$. We get $\frac{1}{2}$ from 1.00×2^{-1} and then $\frac{5}{8}$ from 1.01×2^{-1} and so no but eventually we only get $4, 5, 6, 7$ with no decimals in between. So we see the loss of precision.

2.1 Representation

Definition 2.1.1 (Polynomial). Consider polynomial $p(x) = a_0 + a_1 x + \cdots + a_n x^n$. Degree $\leq n$ and exactly n if $a_n \neq 0$.

Definition 2.1.2 (Centred Form). We have the centred form of the polynomial as $a_0 + a_1(x - c) + \cdots + a_n(x - c)^n$.

Definition 2.1.3 (Newton Form). We have the Newton form as $p(x) = a_0 + a_1(x - c_1) + a_2(x - c_1)(x - c_2) + \cdots + a_n(x - c_1)(x - c_2)\dots(x - c_n)$.

Remark. A grouping on Newton's Form is given by $p(x) = a_0 + (x - c_1)(a_1 + (x - c_2)(a_2 + (x - c_3)(\dots)))$

Example. Consider $p(5000) = \frac{1}{3}$ and $p(5001) = -\frac{2}{3}$ represented by $p(x) = 5000.3 - x$. But now we get $p(5000) = 0.3$ and $p(5001) = -0.7$. We have the centred form as $p(x) = 0.33333 - (x - 5000)$.

2.2 Evaluation and Interpolation

We have Newton's Evaluation for a given z for polynomial p as follows:

Algorithm 2.1: Newton's Evaluation

Data: $p(x) = a_0 + a_1(x - c_1) + \dots a_n(x - c_1)\dots(x - c_n)$

$a'_n \leftarrow a_n$

for $i \leftarrow n; n - 1; n \dots 0$ **do**

  $a'_i \leftarrow a_i + (z - c_{i+1})a'_{i+1}$

Result: $p(z) = a'_0$

This is $O(n)$ for multiplication and addition and is actually the same as doing polynomial division by $(x - z)$ as we will see.

Theorem 2.2.1 (Theorem 1). Let $p(x)$ be any polynomial with distinct zeros $z_0 \dots z_k$. Then $p(x)$ can be written as $p(x) = (x - z_0) \dots (x - z_k)r(x)$ for some polynomial $r(x)$.

Proof. Take $p(x) = p(z) + (x - z)r(x)$ (from Newton's Evaluation). Now choose $z = z_0$. So we have $p(x) = (x - z_0)r(x)$. This implies that $r(x)$ must vanish at $z_1 \dots z_k$ (i.e. they are zeros of $r(x)$) so we simply repeat on $r(x)$. ■

Theorem 2.2.2 (Theorem 2). Let $p(x)$ and $q(x)$ be polynomials of degree $\leq k$. Then if $p(x)$ and $q(x)$ agree at $k + 1$ points then $p(x) = q(x)$.

Proof. If $d(x) = p(x) - q(x)$ then apply theorem 1 to $d(x)$. We can write $d(x) = (x - z_0) \dots (x - z_k)r(x)$. If $r(x) = c_0 + c_1x + \dots + c_mx^m$ then $k \geq \deg d(x) = k + 1 + m$ which is a contradiction. ■

When we go from coefficients to values it is evaluation, and going from values to coefficients is interpolation. We have the matrix $\begin{bmatrix} y_0 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_0 & \dots & x_0^n \\ x_1 & \dots & x_1^n \\ \vdots & \dots & \vdots \\ x_n & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix}$ as the evaluation matrix. The inverse gives the interpolation.

2.3 Polynomial Multiplication

2.3.1 Basic Idea

Consider $A(x) = a_0 + a_1x + \dots + a_dx^d$ and $B(x) = b_0 + b_1x + \dots + b_dx^d$. We want to find have $C(x) = A(x)B(x) = c_0 + c_1x + \dots + c_{2d}x^{2d}$ where $c_k = \sum_{k=0}^{2d} \sum_{i=0}^{k} a_ib_{k-i}$. Solving this sum is $O(d^2)$.

Algorithm 2.2: Polynomial Multiplication

Data: $A(x) = a_0 + \dots + a_dx^d$
Data: $B(x) = b_0 + \dots + b_dx^d$
1 Choose suitable $x_0 \dots x_n$ where $n \geq 2d + 1$ `// Works because of Theorem 2.2.2 with n power of 2`
2 Evaluate all $A(x_i), B(x_i)$
3 **for** $i \in \{0, \dots, n\}$ **do**
4 $\quad C(x_i) = A(x_i)B(x_i)$
5 $[c_0, \dots, c_n] \leftarrow$ **Interpolate**$([x_0, \dots x_n], [C(x_0), \dots, C(x_n)])$
Result: $C(x) = c_0 + c_1x + \dots + c_{2d}x^{2d}$

Lecture 3

2 Sep. 11:50

Example. Consider $A(x) = 1 + .7x + 5x^2 + 6x^3 + 2x^4 + x^4$. We can rewrite this as $A(x) = A_e(x^2) + xA_o(x^2)$.

If we have $A(x) = A_e(x^2) + xA_0(x^2)$ with points as $\pm x_0, \dots, \pm x_{\frac{n}{2}-1}$. So now we have $A(-x) = A_e(x^2) - xA_o(x^2)$. This doesn't work because the x_i cannot be plus-minus paired when squared.

2.3.2 Complex Numbers

Consider $z = a + ib = re^{i\theta}$. We have $(r_1, \theta_1) \times (r_2, \theta_2) = (r_1, r_2, \theta_1 + \theta_2)$. Similarly $-z = (r, \theta + \pi)$. We want roots of unity where:

1. They are $\pm$ paired so $\omega^{\frac{n}{2}+j} = -\omega^j$ where $\omega = e^{i\frac{2\pi}{n}}$.
2. Squaring n roots of unity gives us $\frac{n}{2}$nd roots of unity.

2.3.3 Evaluation Algorithm

Algorithm 2.3: Evaluation Algorithm for unity

Data: $A(x)$
`FFT(`A, ω`)`
if $\omega = 1$ **then** base case
  return $A(1)$
else split and combine
  Split A into A_e and A_o
  FFT(A_e, ω^2) to eval at even powers
  FFT(A_o, ω^2) to eval at odd powers
  for $j = 0 \dots n-1$ **do** combine
    $A(\omega^j) = A_e(\omega^{2j}) + \omega^j A_o(\omega^{2j})$
Result: $A(\omega^0), \dots, A(\omega^{n-1})$

In our case the evaluation matrix is now $M_n(\omega) \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ 1 & \dots & \dots & \dots & \dots \\ 1 & \omega^{n+1} & \omega^{2(n+1)} & \dots & \omega^{(n+1)(n-1)} \end{bmatrix}$. We then see that $M_n(\omega)^{-1} = \frac{1}{n} M_n(\omega^{-1})$. So we see that the interpolation becomes an evaluation at **FFT**$(C, \omega^{-1}$). The above algorithm is $O(n \log n)$

2.3.4 Interpolation

We have interpolating polynomial $p_n(x) = A_o + A_1(x - x_0) \dots A_n(x - x_0) \dots (x - x_{n-1})$, $q_k(x) = A_0 + A_1(x - x_0) + A_k(x - x_0) \dots (x - x_{k-1})$. We then have $p_n(x). = q_k + (x - x_0) \dots (x - x_k) r(x)$. We know that $q_k(x)$ is already an interpolating polynomial for $x_0 \dots x_k$ because $p_n(x)$ is an interpolating polynomial. We can rewrite $p_n(x) = f[x_0] + f[x_1 x_2](x - x_0) + \dots + f[x_0 \dots x_n](x - x_0) \dots (x - x_{n-1})$ where $f[x_0] = p_n(x_0)$, $f[x_1 x_2] = \frac{p_n(x_1) - p_n(x_0)}{x_1 - x_0}$ and so on. This is newton's divided difference recurrence.

Lecture 4

2.4 Root finding

4 Sep. 11:50

2.4.1 Regula Falsi

In univariate we just want $f(X) = 0$. The naive method is to do a binary search with invariant $f(a_n) f(b_n) = 0$. This give 1 bit of information per iteration. This is called regular falsi. We have the formulas $w = \frac{f(b_n) a_n - f(a_n) b_n}{f(b_n) - f(a_n)}$ and we check $f(a_n) f(w) < 0$. This does not guarantee that the interval necessarily becomes very small.

Derive this formula

2.4.2 Secant Method

We have another method called the secant method. We start with x_0, x_1 and compute $x_{n+2} = \frac{f(x_{n+1}) x_n - f(x_n) x_{n+1}}{f(x_{n+1}) - f(x_n)}$. This may not converge. We only apply this for monotonic functions, not oscillating. We also need a minus sign here, which may be catastrophic. We can equivalently have $x_{n+2} = x_n - f(x_n) \frac{x_{n+1} - x_n}{f(x_{n+1}) - f(x_n)} \approx x_n - \frac{f(x_n)}{f'(x_n)}$.

2.4.3 Fixed Point Iteration

Consider $x_{n+1} = g(x_n)$. We can convert all invariants to an iteration of this type.

Definition 2.4.1 (Fixed point). η is a fixed point of $g(\cdot)$ if $g(\eta) = \eta$.

Example. Consider $f(x) = x^2 - x - 2$ and we want its root (which is $x = 2$). We want a fixed point iteration for this root. Some options are $g(x) = x^2 - 2$ or $g(x) = \sqrt{2+x}$, $g(x) = 1 + \frac{2}{x}$ or $g(x) = x - \frac{x^2-x-2}{m}$.

We essentially now want to try and iterate over $g(x) = x - \frac{f(x)}{f'(x)}$ (from the secant method) and try to find its fixed point. If we find the fixed point then we have found the root of $f(x)$.

We can make some assumptions to assure convergence:

1. There is an interval $I = [a, b]$ such that $x \in I \Rightarrow g \in I$.
2. g is a continuous function
3. g is differentiable and $|g'(x)| < 1$ in I

Theorem 2.4.1 (Fixed point theorem). Given the above assumptions, then $\exists \eta \in I$ such that $g(\eta) = \eta$ and converges from within I .

Proof. If the point is at a or b we are done. Otherwise consider $g(a) > a$ and $g(b) < b$ and consider $h(x) = g(x) - x$ which has to be 0 at some point by IVT. The convergence of this depends on the slope. For the error at each step we have we have $e_{n+1} = \eta - x_{n+1} = g(\eta) - g(x_n) = g'(\eta_n)e_n$ for some η_n. So we have $e_{n+1} = g'(\eta_n)e_n$. This is why we need the third condition which then guarantees convergence. ■

Corollary 2.4.1. If $|g'(\eta)| < 1$ then $\exists \epsilon$ such that if $I = [n - \epsilon, n + \epsilon]$ for which FPI converges.

We can write our earlier formulation for the error at each step as $e_{n+1} = -g'(\eta)e_n - \frac{1}{2}g'(\tau_n)e_n^2$ for some τ_n. If we have $g'(\eta) = 0$ then the second term dominates and this term gives us a quadratic equation which gives us a quicker convergence. Newton's iteration has $g'(\eta) = 0$ so we have a fast convergence.

Example. Take the equations $z^n = 1$. Start on the complex plane, and pick a random point. Apply the newton iteration and color accordingly. We see the basin of attraction, and this is the newton fractal.

Chapter 3

Linear Algebra

Lecture 5

Consider $\mathbf{a}_i \in \mathbb{R}^m$ and scalers $x_i, \dots, x_n \in \mathbb{R}$. 9 Sep. 11:50

3.1 Bases

Definition 3.1.1 (Linear Combination). $\sum_{j=1}^{n} x_j a_j$ is a linear combination.

Definition 3.1.2 (Linear Dependence). $\mathbf{a}_1 \dots \mathbf{a}_n$ are linearly dependent if $\sum_j x_j \mathbf{a}_j = 0$ has a non-trivial solution. This can be written as $\mathbf{0} = \mathbf{A}\mathbf{x}$ has a non-trivial solution.

Theorem 3.1.1 (Linear Dependence). If $\mathbf{a}_1 \dots \mathbf{a}_n$ are linearly dependent if at least one of them is a linear combination of the others.

Proof. Suppose $\mathbf{a}_k$ is a linear combination of the others. We can write out this linear combination and subtract $\mathbf{a}_k$. We then have a non-trivial solution. ■

Corollary 3.1.1. $\mathbf{a}_i \dots \mathbf{a}_n$ are linearly dependent if and only if at least one of them is a linear combination of the others.

Definition 3.1.3 (Linear Independence). If not linearly dependent then $\mathbf{a}_i$ are linearly independent.

Definition 3.1.4 (Basis). If $\mathbf{a}_1, \dots, \mathbf{a}_n$ are linearly independent and every element in B can be written as an linear combination of the $\mathbf{a}$s then $[\mathbf{a}]$ is a basis for B.

Corollary 3.1.2. Given $\mathbf{b} \in B$ the coefficients x_i in $\sum_{j=1}^{n} x_j \mathbf{a}_j = b$ are uniquely determined. If $\mathbf{b} = \mathbf{A}\mathbf{x}$ with linearly independent columns of $\mathbf{A}$ that admit a solution, then the solution is unique.

Proof. If we can write $\mathbf{b}$ as a linear combination then we can rewrite $b. = \sum_j x_j \mathbf{a}_j = \sum_j x'_j \mathbf{a}_j$. ■

Corollary 3.1.3. If $\mathbf{a}_1, \dots, \mathbf{a}_n$ is a basis for B and $\mathbf{a}'_1, \dots, \mathbf{a}'_m$ is another basis for B then $n = m$.

Proof. Consider $\mathbf{a}'_1, \dots, \mathbf{a}_n$ where $\mathbf{a}_k$ is a linear combination of the others. Replace this with $\mathbf{a}'_2$. Eventually we will have replaced all the $\mathbf{a}_i$ with $\mathbf{a}'_i$. ■

Corollary 3.1.4. Any linearly independent set can be extended to a basis.

3.2 Inner Product Space

Consider the cosine law $a^2 = b^2 + c^2 - 2bc\cos\theta$. We write $\mathbf{b} = \sum_j b_j \mathbf{e}_j$ where $[\mathbf{e}]$ is the standard basis. By taking the sums as $b_1\mathbf{e}_1 + \|\sum_{j=2} b_j\mathbf{e}_j\|$ out we can write $\|\mathbf{b}\|^2 = \sum_i^n b_i^2$.

Definition 3.2.1 (Transpose). We have $\mathbf{b}^\top\mathbf{c} = \sum_i b_i c_i$.

Corollary 3.2.1. We have $\|\mathbf{b}\|^2 = \mathbf{b}^\top\mathbf{b}$.

Corollary 3.2.2. $\mathbf{b}^\top\mathbf{c} = \|\mathbf{b}\|\|\mathbf{c}\|\cos\theta$.

3.3 Projection

Consider a vector $\mathbf{c}$ and a vector $\mathbf{b}$. The projector $\mathbf{P}$ is such that $\mathbf{Pb} = \mathbf{p}$ where $\mathbf{p}$ is the projection of $\mathbf{b}$ onto $\mathbf{c}$. We have $\mathbf{p} = \mathbf{c}x = \frac{\mathbf{c}\mathbf{c}^\top}{\mathbf{c}^\top\mathbf{c}}\mathbf{b}$.

3.4 Gram-Schmidt

Consider $\mathbf{A} = \begin{bmatrix}\mathbf{a}_1 & \dots & \mathbf{a}_n\end{bmatrix}$ with $\mathbf{a}_i \in \mathbb{R}^m$.

Algorithm 3.1: Gram-Schmidt Process

Data: $\mathbf{a}_1, \dots, \mathbf{a}_n$

$r = 0$

for *j=1 ...n* **do** Generate orthogonal vectors

  $\mathbf{a}'_j = \mathbf{a}_j - \sum_{k=1}^r (\mathbf{q}_k^\top\mathbf{a}_j)\mathbf{q}_k$

  if $\|\mathbf{a}'_j \neq 0\|$ **then** Store orthogonal vector

    $r = r + 1$

    $\mathbf{q}_r = \frac{\mathbf{a}'_j}{\|\mathbf{a}'_j\|}$

Result: We have span of $\mathbf{q}_1 \dots \mathbf{q}_r$ = span of $\mathbf{a}_1 \dots \mathbf{a}_n$ and $\mathbf{q}_i$s are all orthogonal to each other

3.5 Perpendiculars

Definition 3.5.1 (Perp). We have $\mathbf{A}^\perp$ as the set of vectors in $\mathbb{R}^m$ that are orthogonal to the column space of $\mathbf{A}$.

Corollary 3.5.1. $\dim(\mathbf{A}) + \dim(\mathbf{A}^\perp) = m$.

Corollary 3.5.2. We have $\text{range}(\mathbf{A}^\top)^\perp = \text{null}(\mathbf{A})$ and $\text{range}(\mathbf{A})^\perp = \text{null}(\mathbf{A}^\top)$.

Lecture 6

10 Sep. 12:00

3.6 Echelon Forms

Consider the leading entry in the ith row as k_i. The variables corresponding to columns with no leading entries are $u_1 \dots u_h$ where $n-r = h$. So we have $\mathbf{Ax} = 0$ as $x_{k_1} + \sum_{j=1}^r c_{1j}u_j = 0 \dots x_{k_r} + \sum_{j=1}^r c_{rj}u_j = 0$. If $\mathbf{Ax} = \mathbf{b}$ admits a solution and has a non-trivial nullspace then there are $\mathbf{y}$ such that $\mathbf{Ay} = 0$ so $\mathbf{A}(\mathbf{x}+\mathbf{y}) = 0$ so there are infinite solutions.

3.7 Orthogonality

Note. The terms orthogonal and orthonormal are used interchangeably below. When we say orthogonal we mean orthonormal unless otherwise specified.

We now have orthogonal vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ as orthogonal (orthonormal) vectors. We then have points as $\mathbf{p} = \sum_i q_i \mathbf{v}_i$ where $q_i = \mathbf{v}_i^\top \mathbf{p}$.

Corollary 3.7.1. A square orthogonal matrix $\mathbf{V}$ has inverse $\mathbf{V}^{-1} = \mathbf{V}^\top$.

Proof. This is easy to see since the dot product of each vector with itself is the norm which is 1 for a unit vector, and the dot product with every other vector must be 0 since they are orthogonal to each other. ■

Corollary 3.7.2. An orthogonal matrix does not change the norm of a vector.

Proof. $\|\mathbf{V}\mathbf{x}\|^2 = \mathbf{x}^\top \mathbf{V}^\top \mathbf{V}\mathbf{x} = \mathbf{x}^\top \mathbf{x} = \|x\|^2$ ■

Theorem 3.7.1 (Orthogonal Projections)**.** Let $\mathbf{U}$ be an orthogonal matrix. Then $\mathbf{U}\mathbf{U}^\top$ is a projector onto range($\mathbf{U}$). So, it projects a point $\mathbf{b}$ onto range of $\mathbf{U}$. Further if $\mathbf{p}$ is the projection then $\mathbf{b} - \mathbf{p}$ is perpendicular to the range of $\mathbf{U}$.

Proof. Consider $\|\mathbf{b}-\mathbf{p}\|^2 = \mathbf{b}^\top\mathbf{b}+\mathbf{p}^\top\mathbf{p}-2\mathbf{b}^\top\mathbf{p} = \mathbf{b}^\top\mathbf{b}+\mathbf{x}^\top\mathbf{x}-2\mathbf{b}^\top\mathbf{U}\mathbf{x}$ where $\mathbf{p} = \mathbf{U}\mathbf{x}$. Taking the derivative equal to 0 we have $2\mathbf{x} - 2\mathbf{U}^\top\mathbf{b} = 0 \Rightarrow \mathbf{x} = \mathbf{U}^\top\mathbf{b} \Rightarrow \mathbf{U}\mathbf{x} = \mathbf{U}\mathbf{U}^\top\mathbf{b}$. So the best projection is $\mathbf{p} = \mathbf{U}\mathbf{U}^\top\mathbf{b}$. Now consider $\mathbf{U}^\top(\mathbf{b}-\mathbf{p}) = \mathbf{U}^\top(\mathbf{b}-\mathbf{U}\mathbf{U}^\top\mathbf{b}) = \mathbf{U}^\top\mathbf{b}-\mathbf{U}^\top\mathbf{U}\mathbf{U}^\top\mathbf{b} = 0$ so we see that it is perpendicular. ■

Lecture 7

3.8 Norms

11 Sep. 11:50

Definition 3.8.1 (Norm)**.** The norm must satisfy these properties:

1. $\|\mathbf{x}\| \geq 0$ and $\|\mathbf{x}\| = 0 \Leftrightarrow \mathbf{x} = 0$
2. $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$
3. $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$

Definition 3.8.2 (1-Norm)**.** We have $\|\mathbf{x}\|_1 = \sum_i |x_i|$

Definition 3.8.3 (2-Norm)**.** We have $\|\mathbf{x}\|_2^2 = \mathbf{x}^\top\mathbf{x} = \sum_i x_i^2$ so $\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$

Definition 3.8.4 (Infinite Norm)**.** We have $\|\mathbf{x}\|_\infty = \max_i |x_i|$

Definition 3.8.5 (p-norm)**.** In general $|\mathbf{x}|_p = (\sum_i |x_i|^p)^{\frac{1}{p}}$

Definition 3.8.6 (Matrix Norm)**.** $\|\mathbf{A}\| = \sup_{\mathbf{x}:\|\mathbf{x}\|=1} \|\mathbf{A}\mathbf{x}\|$.

The largest stretch it can do to a unit vector.

Definition 3.8.7 (Frobeneus Norm)**.** We ave $\|\mathbf{A}\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$

3.9 Singular Value Decomposition

Theorem 3.9.1 (Singular Value Decomposition). If $\mathbf{A} \in \mathbb{R}^{m\times n}$ then $\exists \mathbf{U} \in \mathbb{R}^{m\times m}, \mathbf{V} \in \mathbb{R}^{n\times n}, \mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ and $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \dots \sigma_p)$ where $p = \min(m, n)$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$ such that $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ and $\mathbf{U}^\top\mathbf{A}\mathbf{V} = \mathbf{\Sigma}$. $\mathbf{U}$ and $\mathbf{V}$ are orthogonal.

Proof. Let $\|\mathbf{A}\|_2 = \sigma_1$ so $\sigma_1\mathbf{u}_1 = \mathbf{A}\mathbf{v}_1$. Expand so we have $\mathbf{U}_1 = \begin{bmatrix}\mathbf{u}_1 & \overline{\mathbf{u}}_2 & \overline{\mathbf{u}}_3 & \dots & \overline{\mathbf{u}}_1\end{bmatrix} \in \mathbb{R}^{m\times m}$ and $\mathbf{V}_1 = \begin{bmatrix}\mathbf{v}_1 & \overline{\mathbf{v}}_2 & \dots & \overline{\mathbf{v}}_n\end{bmatrix} \in \mathbb{R}^{n\times n}$. Now consider $\mathbf{S}_1 = \mathbf{U}_1^\top\mathbf{A}\mathbf{V}_1 = \begin{bmatrix}\sigma_1 & \mathbf{w}^\top \\ \mathbf{0} & \mathbf{A}_1\end{bmatrix}$. Now consider $\frac{1}{\sqrt{\sigma_1^2+\mathbf{w}^\top\mathbf{w}}}\mathbf{S}_1\begin{bmatrix}\sigma_1 \\ \mathbf{w}\end{bmatrix} = \frac{1}{\sqrt{\sigma_1^2+\mathbf{w}^\top\mathbf{w}}}\begin{bmatrix}\sigma_1^2+\mathbf{w}^\top\mathbf{w} \\ \mathbf{A}_1\mathbf{w}\end{bmatrix}$. Now we see that $\mathbf{w}$ must be $\mathbf{0}$ because $\mathbf{S}_1$ can have at most stretch of σ_1^2. Now repeat the argument with $\mathbf{A}_1$ so we have $\mathbf{U}_2^\top\mathbf{A}_1\mathbf{V}_2 = \begin{bmatrix}\sigma_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2\end{bmatrix}$ and consider $\begin{bmatrix}1 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_2^\top\end{bmatrix}\mathbf{U}_1^\top\mathbf{A}\mathbf{V}_1\begin{bmatrix}1 & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2\end{bmatrix} = \begin{bmatrix}\sigma_1 & 0 & 0 \\ 0 & \sigma 2 & 0 \\ 0 & 0 & \mathbf{A}_2\end{bmatrix}$. Induct on this to get the singular value decomposition. ■

Remark. We see that if $m > n$ (which is often the case in CS) thn the remaining $m - n$ rows of $\mathbf{\Sigma}$ are all 0. So we only need the first n columns of $\mathbf{U}$, we call this $\mathbf{U}_r$ and the remaining portion as $\widetilde{\mathbf{U}}_r$. We can take this further by only considering the first r columns where $r = \text{rank}(\mathbf{A}) = \text{rank}(\mathbf{\Sigma})$. Similarly we only need the first r rows of $\mathbf{V}^\top$ called $\mathbf{V}_r^\top$ and are left with $\widetilde{\mathbf{V}}_r^\top$. This is called the **skinny SVD**.

Remark. We see that $\text{range}(\mathbf{A}) = \langle\mathbf{U}_r\rangle$ and $\text{null}(\mathbf{A}) = \langle\widetilde{\mathbf{V}}_r\rangle$. Similarly, we have $\text{range}(\mathbf{A})^\perp = \text{null}(\mathbf{A}^\top) = \langle\widetilde{\mathbf{U}}_r\rangle$ and $\text{range}(\mathbf{A}^\top) = \text{null}(\mathbf{A})^\perp = \langle\mathbf{V}_r\rangle$.

Corollary 3.9.1. The projector onto $\text{range}(\mathbf{A})$ is $\mathbf{U}_r\mathbf{U}_r^\top$. The projector onto $\text{null}(\mathbf{A})$ is $\widetilde{\mathbf{V}}_r\widetilde{\mathbf{V}}^\top$. Similarly $\widetilde{\mathbf{U}}_r\widetilde{\mathbf{U}}_r^\top$ projects onto $\text{range}(\mathbf{A})^\perp = \text{null}(\mathbf{A}^\top)$ and $\mathbf{V}_r\mathbf{V}_r^\top$ projects on to $\text{null}(\mathbf{A})^\perp = \text{range}(\mathbf{A}^\top)$.

Remark. Consider $\mathbf{A}\mathbf{x} = \mathbf{b}$ where $\mathbf{b} \notin \text{range}(\mathbf{A})$. We then can do $\mathbf{U}_r\mathbf{U}_r^\top\mathbf{b}$ to find the projection of $\mathbf{b}$ onto the range space of $\mathbf{A}$ solving the least squares problem.

Remark. Consider a $A \in \mathbb{R}^{3\times 2}$. We decompose into $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. The first action of $\mathbf{A}$ is to express the input vectors in the basis of $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^2$ by rotation. Then with $\mathbf{\Sigma}$ we scale to $\sigma_1\mathbf{v}_1$ and $\sigma_2\mathbf{v}_2$. This also adds a new $\mathbf{v}_3$ which is 0 and orthogonal to $\mathbf{v}_1$ and $\mathbf{v}_2$. Finally with $\mathbf{U}$ we rotate out of our first plane.

Lecture 8

16 Sep. 11:50

Corollary 3.9.2. $\mathbf{A} \in \mathbb{R}^{m\times n}$ can be written as a sum of rank 1 matrices.

Remark. We have $\mathbf{A} = \sum_{i=1}^r \sigma_i\mathbf{u}_i\mathbf{v}_i^\top$ where $r = \text{rank}(\mathbf{A})$.

Definition 3.9.1 (Epsilon Rank). $\text{rank}(\mathbf{A}, \epsilon) = \min_{\|\mathbf{A}-\mathbf{B}\|<\epsilon} \text{rank}(\mathbf{B})$

Corollary 3.9.3. Consider $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. Then let $\mathbf{A}_k = \sum_{i=1}^k \sigma_i\mathbf{u}_i\mathbf{v}_i^\top$ then $\|\mathbf{A} - \mathbf{A}_k\|_2 = \min_{\text{rank}(\mathbf{B})=k}\|\mathbf{A} - \mathbf{B}\|_2 = \sigma_{k+1}$ where $\mathbf{B} \in \mathbb{R}^{m\times n}$.

Proof. Consider $\mathbf{U}^\top \mathbf{A}_k \mathbf{V} = \text{diag}(\sigma_1 \dots \sigma_k) \Rightarrow \text{rank}(\mathbf{A}_k) = k$. We also have $\mathbf{U}^\top(\mathbf{A} - \mathbf{A}_k)\mathbf{V} = \text{diag}(0, \dots, \sigma_{k+1}, \dots, \sigma_r, \dots, 0) \Rightarrow \|\mathbf{A} - \mathbf{A}_k\|_2 = \sigma_{k+1}$. Now consider $\mathbf{B}$ such that $\text{rank}(\mathbf{B}) = k$. So $\mathbf{B}$ has a $n-k$ null space and $\text{null}(\mathbf{B}) = \text{span}\langle \mathbf{x}_1, \dots, \mathbf{x}_{n-k} \rangle$ for some $\mathbf{x}$s. Now consider $\text{span}\langle \mathbf{x}_1, \dots \mathbf{x}_{n-k} \rangle \cap \text{span}\langle \mathbf{v}_1 \dots \mathbf{v}_{k+1} \rangle$. Now let $\mathbf{z}$ be a unit vector in the intersection. We have $\mathbf{Bz} = 0$ because it is in the null space and $\mathbf{Az} = \sum_{i=1}^{k+1} \sigma_i (\mathbf{v}_i^\top \mathbf{z}) \mathbf{u}_i$. So now we have $\|\mathbf{A} - \mathbf{B}\|_2^2 \geq \|(\mathbf{A} - \mathbf{B})\mathbf{z}\|_2^2 = \|\mathbf{Az}\|_2^2 \geq \sigma_{k+1}^2$. ■

Lecture 9

3.10 Regression

18 Sep. 11:50

Remark. We now have $\mathbf{Ax} = \mathbf{b}$ as $\mathbf{U\Sigma V}^\top \mathbf{x} = \mathbf{b}$ which gives us $\mathbf{\Sigma}(\mathbf{V}^\top \mathbf{x}) = \mathbf{U}^\top \mathbf{b}$. So finally we have $\mathbf{\Sigma x}' = \mathbf{b}'$. We have $\mathbf{x}_{\text{LS}} = \mathbf{Vx}'$ which is $\min \|.\|_2$ solution to $\|\mathbf{Ax} - \mathbf{b}\|_2$. So we can reduce systems to a diagonal echelon form.

Remark. The regression problem is how close you can bring $\mathbf{b}$ to the range of $\mathbf{A}$. The solution to this is given by $\mathbf{x}'_i = \frac{\mathbf{b}'_i}{\sigma_i}$ for $i = 1, \dots, r$ and 0 for $i = r+1, \dots, n$. We then have $\mathbf{x} = \mathbf{Vx}'$.

Remark. A related problem is minimize $\|\mathbf{Ax}\|_2$ subject to $\|\mathbf{x}\| = 1$. The solution to this problem is given by the last column of $\mathbf{V}$.

Remark. Consider $\mathbf{a}$. We have the variance $\sigma_{\mathbf{a}}^2 = \langle a_i a_i \rangle_i$ and $\sigma_{\mathbf{b}}^2 = \langle b_i b_i \rangle_i$. We then have the covariance $\sigma_{\mathbf{ab}}^2 = \langle a_i b_i \rangle_i = \frac{1}{n-1} \mathbf{ab}^\top$. So if we have $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_n \end{bmatrix} \in \mathbb{R}^{m \times n}$. We have $S_{\mathbf{X}} = \frac{1}{n-1} \mathbf{XX}^\top$ which is the covariance matrix. We now want a transformation $\mathbf{Y} = \mathbf{PX}$ such that $S_{\mathbf{Y}}$ is diagonal i.e. every dimension is correlated only with itself.

We now have $S_{\mathbf{Y}} = \frac{1}{n-1} \mathbf{YY}^\top = \frac{1}{n-1} \mathbf{PXX}^\top \mathbf{P}^\top = \frac{1}{n-1} \mathbf{P} S_{\mathbf{X}} \mathbf{P}^\top$. Since $S_{\mathbf{X}}$ is symmetric we know that it can be decomposed as $\mathbf{EDE}^\top$. So we know that we are looking for $\mathbf{P} = \mathbf{E}^\top$.

We have assumed that that the whole phenomenon is linear. We have also assumed an underlying Gaussian distribution.

Example. Consider the matrix of faces ϕ. We then take $\frac{1}{n} \phi \phi^\top$. We then diagonalise this and take some basis vectors γ_is.

Lecture 10

23 Sep. 11:50

Corollary 3.10.1. The solution to minimizing $\|\mathbf{Ax}\|$ subject to $\|\mathbf{x}\| = 1$ is given by the last column of $\mathbf{V}$.

Proof. We have the equivalent problem of minimize $\|\mathbf{U\Sigma V}^\top \mathbf{x}\|$ which is equivalent to minimizing $\|\mathbf{\Sigma V}^\top\|$ with respect to $\|\mathbf{V}^\top \mathbf{x}\| = 1$ since $\mathbf{U}$ and $\mathbf{V}^\top$ are rotation matrices and do not change the norm. This is then equivalent to minimizing $\|\mathbf{\Sigma y}\|$ with respect to $\|\mathbf{y}\| = 1$ where $\mathbf{y} = \mathbf{V}^\top \mathbf{x}$. So we would want to choose $\mathbf{y} = (0, \dots, 0, 1)^\top$ which corresponds to the smallest singular value in $\mathbf{\Sigma}$. In this case we are simply choosing the last column of $\mathbf{V}$ since $\mathbf{x} = \mathbf{Vy}$. ■

Exercise. Minimize $\|\mathbf{Ax}\|$ subject to $\|\mathbf{x}\| = 1$ and $\|\mathbf{Cx} = 0\|$.

Answer. We have $\mathbf{C}\mathbf{C}_\perp = 0$ where $\mathbf{C}_\perp = \langle \mathbf{v}_{r+1}, \ldots, \mathbf{v}_n \rangle$. Any $\mathbf{x}$ for which $\mathbf{C}\mathbf{x} = 0$ can be written as $\mathbf{C}_\perp \mathbf{x}' = \mathbf{x}$. So we have $\|\mathbf{C}_\perp \mathbf{x}'\| = 1$. This is $\|\mathbf{x}'\| = 1$ since $\mathbf{C}_\perp$ does not change the norm. We then have minimizing $\|\mathbf{A}\mathbf{C}_\perp \mathbf{x}'\|$ with respect to $\|\mathbf{x}'\| = 1$. We can then use Corollary 3.10.1 to solve for $\mathbf{x}'$ and recover $\mathbf{x} = \mathbf{C}_\perp x'$. ⊛

Exercise. Minimize $\|\mathbf{A}\mathbf{x}\|$ subject to $\|\mathbf{x}\| = 1$ and $\mathbf{x} \in \text{range}(\mathbf{G})$.

Answer. Any $\mathbf{x} \in \text{range}(\mathbf{G})$ can be written as $\mathbf{U}_r \mathbf{x}' = \mathbf{x}$. We see that $\|\mathbf{x}\| = \|\mathbf{U}_r \mathbf{x}'\| = \|x'\|$ constrained to 1. So we have $\mathbf{A}\mathbf{U}_r \mathbf{x}'$ with constraint $\mathbf{x}' = 1$. We can solve this and then recover $\mathbf{x} = \mathbf{U}_r \mathbf{x}'$. ⊛

3.11 Projectors

Definition 3.11.1 (Projector). $\mathbf{P}$ is a projector if $\mathbf{P}^2 = \mathbf{P}$. It projects a vector $\mathbf{v}$ to the range space of $\mathbf{P}$ by giving a point $\mathbf{P}\mathbf{v}$.

Corollary 3.11.1. $\mathbf{v} - \mathbf{P}\mathbf{v} \in \text{null}(\mathbf{P})$.

Proof. $\mathbf{P}(\mathbf{v} - \mathbf{P}\mathbf{v}) = \mathbf{P}\mathbf{v} - \mathbf{P}\mathbf{P}\mathbf{v} = \mathbf{P}\mathbf{v} - \mathbf{P}\mathbf{v} = 0$. ■

Corollary 3.11.2. If $\mathbf{P}$ is a projector then so is $\mathbf{I} - \mathbf{P}$.

Proof. $(\mathbf{I} - \mathbf{P})^2 = \mathbf{I} - 2\mathbf{P} + \mathbf{P} = \mathbf{I} - \mathbf{P}$. ■

Corollary 3.11.3. $\text{range}(\mathbf{I} - \mathbf{P}) = \text{null}(\mathbf{P})$.

Proof. Consider $\mathbf{v} \in \text{null}(P) \Rightarrow \mathbf{v} - \mathbf{P}\mathbf{v} = (\mathbf{I} - \mathbf{P})\mathbf{v} \in \text{range}(\mathbf{I} - \mathbf{P})$. ■

Corollary 3.11.4. $\text{range}(\mathbf{P}) \cap \text{null}(\mathbf{P}) = \{0\}$

Proof. We see that $\text{range}(\mathbf{P}) = \text{null}(\mathbf{I} - \mathbf{P})$. So $(\mathbf{I} - \mathbf{P})\mathbf{v} = 0$ and $\mathbf{P}\mathbf{v} = 0$. So $\mathbf{P}\mathbf{v} = \mathbf{v}$ and $\mathbf{P}\mathbf{v} = 0$ therefore $\mathbf{v} = 0$. ■

Definition 3.11.2 (Complementary Projector). We call $\mathbf{I} - \mathbf{P}$ the complementary projector of $\mathbf{P}$.

Definition 3.11.3 (Orthogonal Projector). $\mathbf{P}$ is an orthogonal projector if $\text{range}(\mathbf{P})$ is orthogonal to $\text{null}(\mathbf{P})$.

Corollary 3.11.5. $\mathbf{P}$ is an orthogonal projector if and only if $\mathbf{P} = \mathbf{P}^\top$.

Proof. First consider if $\mathbf{P} = \mathbf{P}^\top$. Now consider $\mathbf{v}_1 \in \text{range}(\mathbf{P}) = \mathbf{P}\mathbf{x}$. Similarly, consider $\mathbf{v}_2 \in \text{null}(\mathbf{P})$. We then have $\mathbf{v}_1^\top \mathbf{v}_2 = \mathbf{x}^\top \mathbf{P}^\top \mathbf{v}_2 = \mathbf{x}^\top \mathbf{P}\mathbf{v}_2 = \mathbf{x}^\top 0 = 0$ so they are orthogonal. Now in the opposite direction consider an ortogonal basis of the space $\mathbb{R}^{m \times m}$ given by $\mathbf{Q}$ where $\mathbf{q}_1, \ldots \mathbf{q}_n$ form the range of $\mathbf{P}$ and $\mathbf{q}_{n+1}, \ldots, \mathbf{q}_m$ form the nullspace. We then have $\mathbf{P}\mathbf{Q} = \mathbf{Q}\mathbf{\Lambda}$ where $\mathbf{L}$ is a diagonal matrix with n 1s and $m-n$ 0s on the diagonal. We then multiply on the right with $\mathbf{Q}^\top$ giving up $\mathbf{P}\mathbf{Q}\mathbf{Q}^\top = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$. So we have $\mathbf{P}^\top = (\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top)^\top = (\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top) = \mathbf{P}$. ■

Remark. This is the same as saying for a set of orthogonal vectors $\mathbf{q}_1, \ldots, \mathbf{q}_m$ for some n we have $\mathbf{q}_1, \ldots, \mathbf{q}_n$ where $\mathbf{P}\mathbf{q}_j = \mathbf{q}_j$ and for $j > n$ we have $\mathbf{P}\mathbf{q}_j = 0$. We then have $\mathbf{P}\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \ldots & \mathbf{q}_n & 0 & \ldots & 0 \end{bmatrix}$ and $\mathbf{Q}^\top \mathbf{P}\mathbf{Q}$ is a diagonal matrix with n 1s on the diagonal. This is the same as the eigenvalue decomposition.

Lecture 11

24 Sep. 13:30

Remark. Suppose we have a projector with an arbitrary basis $\mathbf{S} = \langle \mathbf{a}_1, \dots, \mathbf{a}_n \rangle$. We then want to minimise $\|\mathbf{Ax} - \mathbf{b}\|_2$. This can be given by $\mathbf{A}^\top(\mathbf{y} - \mathbf{v}) = 0$. Since $\mathbf{y}$ is in the range of $\mathbf{A}$ there is an $\mathbf{x}$ such that $\mathbf{A}^\top(\mathbf{Ax} - \mathbf{v}) = 0 \Rightarrow \mathbf{x} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{v}$, which is called the Moore-Penrose pseudoinverse. So the projector is given by $\mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top$.

Remark. Consider $\langle \mathbf{a}_1 \rangle \subseteq \langle \mathbf{a}_1, \mathbf{a}_2 \rangle$ and so on. We want an orthogonal, relationship preserving basis. We create vectors using gram-schmidt given by $\mathbf{a}_1 = r_{11}\mathbf{q}_1, \mathbf{a}_2 = r_{12}\mathbf{q}_1 + r_{22}\mathbf{q}_2$ and so on. So we finally have $\mathbf{A} = \mathbf{QR}$ where $\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1 & \dots & \mathbf{q}_n \end{bmatrix}$ and $\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & & r_{nn} \end{bmatrix}$ which is the QR-factorization. When $\mathbf{Q}$ is not square it is the reduced factorization represented by $\hat{\mathbf{Q}}$.

Remark. We have the projector onto the range of $\mathbf{A}$ as $\hat{\mathbf{Q}}\hat{\mathbf{Q}}^\top$. We then have $\hat{\mathbf{Q}}\mathbf{Rx} = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^\top\mathbf{b}$. So we have $\mathbf{x} = \mathbf{R}^{-1}\hat{\mathbf{Q}}^\top\mathbf{b}$ meaning $\mathbf{R}^{-1}\hat{\mathbf{Q}}^\top = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top$ is the pseudoinverse.

Lecture 12

25 Sep. 11:50

3.12 Recapping Least Squares

We have three methods to solve least squares, i.e. minimising $\|\mathbf{Ax} - \mathbf{b}\|_2$:

1. The normal equations where $\mathbf{x} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}$. We can Cholesky factorise $\mathbf{A}^\top\mathbf{A} = \mathbf{R}^\top\mathbf{R}$ where $\mathbf{R}^\top\mathbf{w} = \mathbf{y}$, $\mathbf{Rx} = \mathbf{w}$ and $\mathbf{y} = \mathbf{A}^\top\mathbf{b}$. So we have the orthogonal projector $\mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top$. This is $O(n^3)$.

2. The QR-factorization where $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$. We use Gram-Schimdt to get the factorization and have the orthogonal projector $\hat{\mathbf{Q}}\hat{\mathbf{Q}}^\top$. We then have $\hat{\mathbf{Q}}\hat{\mathbf{R}} = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^\top\mathbf{ab} \Rightarrow \hat{\mathbf{R}}\mathbf{x} = \hat{\mathbf{Q}}^\top\mathbf{b}$ and $\hat{\mathbf{R}}^{-1}\hat{\mathbf{Q}}^\top$ is the pseudoinverse. This is $O(n^3)$.

3. SVD where $\mathbf{A} = \mathbf{\Sigma V}^\top$. We then have $\mathbf{\Sigma V}^\top\mathbf{x} = \hat{\mathbf{U}}\hat{\mathbf{U}}^\top\mathbf{b}$. We can write this as $\mathbf{\Sigma x}' = \mathbf{b}'$. In this case the solutions are $x'_i = \frac{b'_i}{\sigma_i}$ and then recovering $\mathbf{x} = \mathbf{Vx}'$. In this case we can easily solve where $\mathbf{A}$ is not full rank.

3.13 Some factorizations

3.13.1 QR-Factorization

Remark. We have a modified Gram-Schmidt MGS where we use $(j-1)$ projectors of rank $(m-1)$. We create $\mathbf{P}_j = P_{\perp\mathbf{q}_1} + P_{\perp\mathbf{q}_2} + \dots + P_{\perp\mathbf{q}_{j-1}}$ where $P_{\perp\mathbf{q}_i} = (\mathbf{I} - \mathbf{q}_1\mathbf{q}_1^\top)$.

Remark. We can also look at the QR-factorization as a series of operations $\mathbf{Q}_j$ which convert $\mathbf{A}$ to $\mathbf{R}$. So $\mathbf{Q}_j$ is characterised by $\begin{bmatrix} \mathbf{I_{j-1}} & 0 \\ 0 & \mathbf{F} \end{bmatrix}$ because we want to leave the first $j-1$ row and columns untouched, converting the jth column to 0s below the pivot using $\mathbf{F}_k$. So we want $\mathbf{F}_k\mathbf{x} = \|\mathbf{x}\|\mathbf{e}_1$. Imagine a triangle of these two vectors with the difference $\mathbf{v}$. We have $\|\mathbf{x}\| = \|\mathbf{x}\|\|\mathbf{e}_j\|$ to align $\mathbf{x}$ to the $\mathbf{e}_j$ access. The perpendicular bisector is then $\mathbf{H}$ and the projector onto this space is given by $\mathbf{F}_k = \mathbf{I} - \frac{2\mathbf{vv}^\top}{\mathbf{v}^\top\mathbf{v}}$. So we see that $\mathbf{F}_k\mathbf{x} = \mathbf{x} - 2\frac{\mathbf{vv}^\top}{\mathbf{v}^\top\mathbf{v}}\mathbf{x}$.

Lecture 13

30 Sep. 11:50

Remark. We look at the sign of x_j (the pivot position) and construct $\mathbf{v} = \text{sign}(x_j)\|\mathbf{x}\|\mathbf{e}_j + \mathbf{x}$ to hope to correct the larger angle. This leads to better stability in the subtraction.

Algorithm 3.2: QR Factorization

Data: $\mathbf{A}_{m\times n}$

for $k = 1 \dots m$ **do**

- $\mathbf{x} \leftarrow \mathbf{A}[k:m, k]$
- $\mathbf{v} \leftarrow \text{sign}(x_j)\|\mathbf{x}\|\mathbf{e}_1 + \mathbf{x}$
- $\mathbf{v}_k \leftarrow \frac{\mathbf{v}_k}{\|\mathbf{v}_k\|_2}$
- $\mathbf{A}[k:m, k:n] = A[k:m, k:n] - 2\mathbf{v}_k(\mathbf{v}_k^\top \mathbf{A}[k:m, k:n])$

Result: $\mathbf{A}$ is now $\mathbf{R}$ in the QR-factorization

3.13.2 LU-factorization

Remark. We have $\mathbf{A}_{m\times n}$. We want the LU decomposition so we can contruct for example $\begin{bmatrix} 1 & 0 \\ -\tau & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{bmatrix} x_1 \\ 0 \end{bmatrix}$. Here we have $\tau = \frac{x_2}{x_1}$. Similarly for τ we have $\tau_i = \frac{x_i}{x_k}$ for $i = k+1 \dots n$, which is called a Gauss multiplier. We then have $\mathbf{M}_k = \mathbf{I} - \tau\mathbf{e}_k^\top$ called a Gauss matrix. We then have $\mathbf{M}_n \dots \mathbf{M}_1\mathbf{A} = \mathbf{U}$ and $\mathbf{L}^{-1} = \mathbf{M}_n \dots \mathbf{M}_1$.

Theorem 3.13.1 (LU Decomposition). Every $\mathbf{A} \in \mathbb{R}^{n\times n}$ has a LU decomposition where $\mathbf{L}$ is a unit lower triangular $\mathbf{U}$ is upper triangular and $\det(\mathbf{A}) = u_{11}u_{22}\dots u_{nn}$. This decomposition is unique.

Algorithm 3.3: LU Decomposition

Data: $\mathbf{A}_{n\times n}$

while $k < n$ *and* $\mathbf{A}(k,k) \neq 0$ **do**

- rows $\leftarrow k+1 : n$
- $\tau^k = \mathbf{A}(\text{rows}, k)$
- $\mathbf{A}(\text{rows}, k) \leftarrow \frac{\mathbf{A}(\text{rows},k)}{\mathbf{A}(k,k)}$
- $\mathbf{A}(\text{rows}, \text{rows}) \leftarrow \mathbf{A}(\text{rows}, \text{rows}) - \mathbf{A}(\text{rows}, k)\mathbf{A}(k, \text{rows})$
- $k \leftarrow k+1$

Lecture 14

3.14 Eigenvalues and Eigenvectors

14 Oct. 11:50

Definition 3.14.1 (Eigenvalues). For a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ if we have $\mathbf{Ax} = \lambda\mathbf{x}$ then λ is an eigenvalue with corresponding eigenvector $\mathbf{x}$.

Remark. If $\mathbf{A}$ is diagonizable then we have $\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1} \Rightarrow \mathbf{AX} = \mathbf{\Lambda X}$ where the entries of the diagonal are eigenvalues and the coulmns of $\mathbf{X}$ are eigenvectors.

Remark. We can characterise eigenvalues as having an invariant subspace E_λ corresponding to every λ. So $\dim(E_\lambda) = \dim(\text{null}(\lambda\mathbf{I} - \mathbf{A})) = \text{gm}(\lambda)$.

Theorem 3.14.1 (Characteristic polynomial). λ is an eiegnvalue if and only if $\det(\lambda\mathbf{I} - \mathbf{A}) = 0$ so they are the roots of the polynomial $p_A(z) = \det(z\mathbf{I} - \mathbf{A})$.

Theorem 3.14.2 (Decomposition). if $\mathbf{A}$ has eigenvalue decomposition $\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}$ then $\mathbf{A}$ and $\mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}$ have the same characteristic polynomials, evaluations, geometric multiplicity and algebraic multiplicity.

Corollary 3.14.1. We have $\text{am}(\lambda) \geq \text{gm}(\lambda)$.

Proof. Let $\hat{\mathbf{Q}} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \dots & \mathbf{q}_n \end{bmatrix}$ be a orthogonal representation of E_λ. Extend this to a small basis $\mathbf{Q}$. We then have $B = \mathbf{Q}\mathbf{A}\mathbf{Q}^\star = \begin{bmatrix} \lambda\mathbf{I}_{n\times n} & \mathbf{C} \\ 0 & \mathbf{D}_{(m-n)\times(m-n)} \end{bmatrix}$. So we have $\det(z\mathbf{I} - \mathbf{B}) = \det(z\mathbf{I} - \lambda\mathbf{I}) \det(z\mathbf{I} - \mathbf{D})$ meaning $\text{am}(\lambda) \geq \text{gm}(\lambda)$. ■

Corollary 3.14.2. A matrix is diagonizable if and only if $\forall\lambda(\text{am}(\lambda) = \text{gm}(\lambda))$.

Proof. If it is diagnoizable then $\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}$ then $\mathbf{A}$ and $\mathbf{\Lambda}$ are similar and have the same characteristics therefore $\mathbf{A}$ is non-defective. In the opposite direction if we have $\text{am}(\lambda) = \text{gm}(\lambda)$, meaning we can construct n linearly independent eigenvectors for $\mathbf{X}$ and it is invertible since it a full rank square matrix. ■

Definition 3.14.2 (Unitary diagnoizable). We say $\mathbf{A}$ is unitary diagonizable if $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\star$.

Theorem 3.14.3 (Schur Decomposition). For any $\mathbf{A} \in \mathbb{R}^{m\times m}$ we have $\mathbf{A} = \mathbf{Q}\mathbf{T}\mathbf{Q}^\star$ will always exist where $\mathbf{Q}$ is unitary and $\mathbf{T}$ is upper triangular.

Proof. Take the first vector $\mathbf{u}_1 = \mathbf{a}_1$ and extend it to an orthogonal basis $\mathbf{U}$. We have $\mathbf{U}^\star\mathbf{A}\mathbf{U} = \begin{bmatrix} \lambda & \mathbf{C} \\ 0 & \mathbf{D} \end{bmatrix}$. We now take $\mathbf{D} = \mathbf{V}\mathbf{T}\mathbf{V}^\star$. We can then recover $\mathbf{Q} = \mathbf{U}\begin{bmatrix} 1 & 0 \\ 0 & \mathbf{V} \end{bmatrix}$ and attach λ. ■

Lecture 15

3.14.1 Summary

16 Oct. 11:50

We have the following properties:

1. $\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}$ exists if and only if $\mathbf{A}$ is not defective.
2. If $\mathbf{A}$ is symmetric/hermitian then it is unitary diagoniazable $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\star$ with real eigenvalues.
3. The Schur decomposition always exists $\mathbf{A} = \mathbf{Q}\mathbf{T}\mathbf{Q}^\star$ where $\mathbf{T}$ is upper diagonal.

3.15 Eigenvector Algorithms

3.15.1 Phase 1

The difficulty is that we must do a householder operation $\mathbf{Q}_1^\star\mathbf{A}\mathbf{Q}_1$. If we do a regular diagonalization then the post multiplication will revert those entries. So we allow a subdiagonal entry as well, which allows us to not affect the first row. For asymmetric matrix this gives us tridiagonal and for a other matrices this gives us upper hessenberg matrix.

Remark. Essentially we are doing a householder, but on A_{12} followed by A_{21} to leave elements untouched. In a symetric matrix, this gives us 0s in rows and columns.

3.15.2 Phase 2

Definition 3.15.1 (Raleigh Quotient). We have $r(\mathbf{x}) = \frac{\mathbf{x}^\top \mathbf{A}\mathbf{x}}{\mathbf{x}^\top \mathbf{x}}$.

Corollary 3.15.1. We have as $x \to$ eigenvector $\hat{\mathbf{x}}$ we have $r(x) \to \hat{\lambda_{\mathbf{x}}}$

Remark. For real symmetric matrices we have $\mathbf{x} = a_1\mathbf{q}_1 \dots a_m\mathbf{q}_m$. We then have $r(\mathbf{x}) = \frac{a_1^2\lambda_1 + \dots a_m^2\lambda_m}{a_1^2 + \dots + a_m^2}$. So if $\mathbf{x} \to \mathbf{q}_1$ then $|r(\mathbf{x}) - \lambda| = O(\|\mathbf{x} - (\pm\mathbf{q}_i)\|^2)$.

We now have the power iteration as follows:

Algorithm 3.4: Power Iteration

Data: $\mathbf{v}^{(0)}$ such that $\|\mathbf{v}^{(0)}\| = 1$
for $k = 1, 2, \dots$ **do**
  $\mathbf{w} \leftarrow \mathbf{A}\mathbf{v}^{(k-1)}$
  $\mathbf{v}^{(k)} \leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|}$
  $\lambda^{(k)} \leftarrow {\mathbf{v}^{(k)}}^\top \mathbf{A}\mathbf{v}^{(k)}$

Remark. We have $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ then $(\mathbf{A} - \mu\mathbf{I})\mathbf{x} = (\lambda - \mu)\mathbf{x}$ and $(\mathbf{A} - \mu\mathbf{I})^{-1}\mathbf{x} = (\lambda - \mu)^{-1}\mathbf{x}$. This gives us the eigenvalue closest to μ.

Corollary 3.15.2. If $|\lambda_J - \mu| < |\lambda_K - \mu| \le |\lambda_j - \mu|$ and $\mathbf{q}_J^\top \mathbf{v}^{(0)} \neq 0$ for $j \neq J$ then $\|\mathbf{v}^{(k)} - (\pm q_J)\| = O(\frac{|\lambda_J - \mu|}{|\lambda_K - \mu|})^k$.

Remark. So far we have not used the raleigh quotient.

Algorithm 3.5: RQ Iteration

Data: $\mathbf{v}^{(0)}$ such that $\|\mathbf{v}^{(0)}\| = 1$
${\lambda^{(0)}} \leftarrow {\mathbf{v}^{(0)}}^\top \mathbf{A}\mathbf{v}^{(0)}$ **for** $k = 1, 2, \dots$ **do**
  Solve $(\mathbf{A} - \lambda^{(k-1)}\mathbf{I})\mathbf{w} = \mathbf{v}^{(k-1)}$
  $\mathbf{v}^{(k)} \leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|}$
  $\lambda^{(k)} \leftarrow {\mathbf{v}^{(k)}}^\top \mathbf{A}\mathbf{v}^{(k)}$

Corollary 3.15.3. We now have $\|\mathbf{v}^{(k)} - (\pm\mathbf{q})\| = O(\|\mathbf{v}^{(k-1)} - (\pm\mathbf{q})\|^3)$.

Lecture 16

Consider the following assumptions: 21 Oct. 11:50

1. $|\lambda_1| > |\lambda_2| > \dots > |\lambda_{n+1}| \ge \dots \ge |\lambda_m|$
2. For $\hat{\mathbf{A}}^\top \mathbf{V}^{(0)}$ all principal submatrices are non-singular where $\hat{\mathbf{Q}}$ are the first n eigenvectors.

Corollary 3.15.4. If both the assumptions hold then as $k \to \infty$ in simultaneous iteration, the columns of $\hat{\mathbf{Q}}^{(k)}$ converge to $\begin{bmatrix}\mathbf{q}_1 & \dots & \mathbf{q}_n\end{bmatrix}$.

Proof. We know that the $\mathbf{v}_j^0$ can always be written as a combination of the eigenvectors. So $\mathbf{A}^k\mathbf{v}_j^{(0)}$ can be written as $\lambda_1^k\mathbf{a}_{1j}\mathbf{q}_1 + \cdots + \lambda_m^k a_{mj}\mathbf{q}_m$. So we have $\mathbf{V}^{(k)} = \mathbf{A}^k\mathbf{V}^{(0)} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top\mathbf{V}^{(0)} + O(|\frac{\lambda_{n+1}}{\lambda_n}|^k) = (\mathbf{Q}\mathbf{\Lambda}^k + O(|\frac{\lambda_{n+1}}{\lambda_n}|^k))\mathbf{Q}^\top\mathbf{V}^{(0)}$ ■

Remark. The problem here is that we may see the principal eigenvector dominate and converge too fast.

Algorithm 3.6: Nomralised Simultaneous Iteration

Data: $\mathbf{Q}^{(0)}$
Data: $\mathbf{A}$
for $k = 1, 2, \ldots$ **do**
  $\mathbf{Z} \leftarrow \mathbf{A}\overline{\mathbf{Q}}^{(k-1)}$
  Solve factorization $\overline{\mathbf{Q}}^{(k)}\mathbf{R}^{(k)} = \mathbf{Z}$
Result: $\overline{\mathbf{Q}}^{(k)}$

We implement the above algorithm with the QR iteration:

Algorithm 3.7: QR Simultaneous Iteration

Data: $\mathbf{A}$
$\mathbf{A}^{(0)} \leftarrow \mathbf{A}$
for $k = 1, 2, \ldots$ **do**
  Solve factorization $\mathbf{Q}^{(k)}\mathbf{R}^{(k)} = \mathbf{A}^{(k-1)}$
  $\mathbf{A}^{(k)} \leftarrow \mathbf{R}^{(k)}\mathbf{Q}^{(k)}$
Result: $\overline{\mathbf{Q}}^{(k)}$

Corollary 3.15.5. For the QR iteration define $\overline{\mathbf{Q}}^{(k)} = \mathbf{Q}^{(1)}\ldots\mathbf{Q}^{(k)}$ and for both algorithms define $\overline{\mathbf{R}}^{(k)} = \mathbf{R}^{(k)}\ldots\mathbf{R}^{(k)}$. Then both algorithms generate the same $\overline{\mathbf{Q}}^{(k)}, \overline{\mathbf{R}}^{(k)}$ and $\mathbf{A}^{(k)}$ and $\mathbf{A}^k = \overline{\mathbf{Q}}^{(k)}\overline{\mathbf{R}}^{(k)}$.

Proof. For simultaneous iteration $\mathbf{A}^k = \mathbf{A}\overline{\mathbf{Q}}^{(k-1)}\overline{\mathbf{R}}^{(k-1)} = \overline{\mathbf{Q}}^{(k)}\mathbf{R}^{(k)}\overline{\mathbf{R}}^{(k-1)} = \overline{\mathbf{Q}}^{(k)}\overline{\mathbf{R}}^{(k)}$. For the QR iteration we have $\mathbf{A}^k = \mathbf{A}\overline{\mathbf{Q}}^{(k-1)}\overline{\mathbf{R}}^{(k-1)} = \overline{\mathbf{Q}}^{(k-1)} = \overline{\mathbf{Q}}^{(k-1)}\mathbf{A}^{(k-1)}\overline{\mathbf{R}}^{(k-1)} = \overline{\mathbf{Q}}^{(k)}\overline{\mathbf{R}}^{(k)}$ ■

Remark. We have $\mathbf{A}^k = \overline{\mathbf{Q}}^{(k)}\overline{\mathbf{Q}}^{(k)}$ so we have $\mathbf{A}^{-k} = (\overline{\mathbf{R}}^{(k)})^{-1}\overline{\mathbf{Q}}^{(k)^\top} = \overline{\mathbf{Q}}^{(k)}(\overline{\mathbf{R}}^{(k)})^{-\top}$. So then $\mathbf{A}^{-k}\mathbf{P} = (\overline{\mathbf{Q}}^{(k)}\mathbf{P})(\mathbf{P}\overline{\mathbf{R}}^{(k)^{-\top}}\mathbf{P})$

Algorithm 3.8: Shifted QR Simultaneous Iteration

Data: $\mathbf{A}$
$\mathbf{A}^{(0)} \leftarrow \mathbf{A}$
for $k = 1, 2, \ldots$ **do**
  Solve factorization $\mathbf{Q}^{(k)}\mathbf{R}^{(k)} = \mathbf{A}^{(k-1)} - \mu^{(k)}\mathbf{I}$
  $\mathbf{A}^{(k)} \leftarrow \mathbf{R}^{(k)}\mathbf{Q}^{(k)} + \mu^{(k)}\mathbf{I}$
Result: $\mathbf{Q}^{(k)}$

Remark. So first we apply the tridiagonalization. We then pick a shift $\mu^{(k)} = \mathbf{A}_{mm}$.

Lecture 17

22 Oct. 16:30

3.15.3 Real Schur Form

Remark. Working with complex eigenvalues is significantly more expensive than working with real eigenvalues.

Theorem 3.15.1 (Real Schur Form)**.** If $\mathbf{A} \in \mathbb{R}^{m \times n}$ then there exists real orthogonal $\mathbf{Q}$ such that $\mathbf{A} = \mathbf{QTQ}^\star$ where $\mathbf{T}$ is Schur with 2×2 blocks in the sub diagonal.

Remark. The 2×2 on the diagonal are the decoupled complex eigenvalues which we can then solve to retrieve the complex eigenvalues.

Chapter 4

Gradient Descent

4.1 Sparse Matrix

Remark. A sparse matrix is represented as $[x_i, y_i, v_i]$. We store only the non-zero elements.

Remark. Matrix-vector operations with this are of the degree of the sparsity and matrix-matrix operations in the product of the degrees.

Remark. Most matrices come out of graphs, and more graphs are sparse.

4.2 Quadratic form

Definition 4.2.1 (Quadratic form). A quadratic form is a scaler, quadratic function of a vector with the form $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x} - \mathbf{b}^\top \mathbf{x} + c$.

Corollary 4.2.1. We have the derivative of the quadratic form as $f'(\mathbf{x}) = \frac{1}{2}\mathbf{A}^\top \mathbf{x} + \frac{1}{2}\mathbf{A}\mathbf{x} - \mathbf{b}$ but it is $f'(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$ if $\mathbf{A}$ is symmetric.

4.3 Steepest Descent

Corollary 4.3.1. The direction of the steepest descent is $r_{(i)} - f'(\mathbf{x}_{(i)}) = \mathbf{b} - \mathbf{A}\mathbf{x}_{(i)}$. This gives us the iteration $\mathbf{x}_{(i+1)} = \mathbf{x}_{(0)} + \alpha r_{(i)}$.

We now need to decide α i.e. "how big a step to take".

Corollary 4.3.2. The error $\mathbf{e}_{(i)} = \mathbf{x}_{(i)} - \mathbf{x}$. So we have $r_{(i)} = -\mathbf{A}\mathbf{e}_{(i)}$.

Remark. We see that α minimizes f when the directional derivative is equal to 0. So we have $\frac{\mathrm{d}}{\mathrm{d}\alpha} f(\mathbf{x}_{(i)}) = f'(\mathbf{x}_{(i)})^\top \frac{\mathrm{d}}{\mathrm{d}\alpha}\mathbf{x}_{(i)} = f'(\mathbf{x}_{(i)})^\top r_{(i-1)}$. We can then set this to 0 then find the size of α. Ultimately we get $\alpha = \frac{r_{(i-1)}^\top r_{(i-1)}}{r_{(i-1)}^\top \mathbf{A} r_{(i-1)}}$ which is the inverse raleigh quotient.

Remark. We see that if we are on an eigenvector then we directly reach the minima in one step.

Corollary 4.3.3. We have $r_{(i+1)} = r_{(i)} - \alpha_{(i)}\mathbf{A}r_{(i)}$

Lecture 18

23 Oct. 11:50

Remark. We can write any matrix as $\mathbf{A} = \mathbf{D} + \mathbf{E}$ where $\mathbf{D}$ is a diagonal matrix. We then have a fixed point iteration for $\mathbf{Ax} = \mathbf{b}$, called the Jacobi iteration.

Remark. Consider the case where $r_{(i)} = -\mathbf{Ae}_{(i)} = -\lambda \mathbf{e}_{(i)}$ then we have $\mathbf{e}_{(i+1)} = 0$. So if $r_{(i)}$ is an eigenvector then we see that we converge directly.

4.4 Conjugate Directions

Remark. For steepest descent we often takes steps in the same direction. A suggestion to improve is to look at orthogonal directions, and each time adjust α such that we take the largest possible step in that direction and line up with $\mathbf{x}$. So we take only one step in each direction.

Remark. We take the conjugacy $\mathbf{d}_i^\top \mathbf{A} \mathbf{d}_i$, i.e. looking for vectors that are "A-orthogonal". Then taking the derivative we get $\mathbf{d}_{(i)}^\top \mathbf{A} \mathbf{e}_{(i+1)} = 0 \Rightarrow \alpha_{(i)} = \frac{\mathbf{d}_{(i)}^\top r_{(i)}}{\mathbf{d}_{(i)} \mathbf{A} \mathbf{d}_{(i)}}$.

Lecture 19

Lecture 20

11 Nov. 11:50

Missed above two classes. Conjugate gradient descent, Newton and Levenberg-Marquardt to be studied from PDFs. 13 Nov. 11:50

Chapter 5

Linear Programming

Lecture 21: Online

We will being by trying to gain a geometric understanding of linear programming. .

see PDFs

5.1 Cones

Definition 5.1.1 (Cones). Given a set of of vectors $\{\mathbf{g}_i\}$ the set $\{\sum_i a_i \mathbf{g}_i | a_i \geq 0\}$ of all their non-negative linear combinations is a cone.

Definition 5.1.2 (Dual of a Cone). The dual of a cone C is written as $C^\perp$ and is the set of all vectors that have a non-positive dot product with everyone vector in C. If C is a cone then $(C^\perp)^\perp = C$.

Definition 5.1.3 (Flat Cone). If a cone contains both a non-zero vector and its negative it is a flat cone.

Corollary 5.1.1.

Lecture 22: Online

Online class, repeating the geometric interpretation from previous days.

20 Nov. 11:50

Lecture 23

25 Nov. 11:50

5.2 Linear Programming Algebra

Definition 5.2.1 (Linear Program). A linear program is an intance of the problem $\mathbf{Ax} \leq \mathbf{b}$

Definition 5.2.2 (Feasible Solution). This is the set $\{\mathbf{x} : \mathbf{Ax} \leq \mathbf{b}\}$. A feasible solution obtaining the maximum is optimal.

Definition 5.2.3 (Polyhedra). A polyhedra in $\mathbb{R}^n$ is a set of the type $P = \{\mathbf{x} : \mathbf{Ax} \leq \mathbf{b}\}$ for some matrix $\mathbf{A} \in \mathbb{R}^{\{}m \times n\}$. A bounded polyhedra is also a polytope.

Definition 5.2.4 (Dimension). If X is a non-empty set then $\dim(X) = n - \text{rank}(A)$ where A is a $n \times n$ matrix where $\mathbf{Ax} = \mathbf{Ay}$ for all $x, y \in X$.

Definition 5.2.5 (Supporting Hyperplane)**.** Let $P = \{\mathbf{x} : \mathbf{Ax} \leq \mathbf{b}\} \neq \phi$. If $\mathbf{c}$ is a non-zero vector for which $\delta = \max\{\mathbf{c}^\top \mathbf{x} | \mathbf{x} \in P\}$ then $\{\mathbf{x} : \mathbf{c}^\top \mathbf{x} = \delta\}$ is a supporting hyperplane. A point $\mathbf{x}$ for which $\{\mathbf{x}\}$ is a face is called a vertex of P and is also a basic solution of the system $\mathbf{Ax} \leq \mathbf{b}$.

Theorem 5.2.1 (Faces)**.** Let P be a polyhedron and $F \subseteq P$. Then the following are equivalent:

1. F is a face of P
2. $\exists C$ such that $\delta = \max\{\mathbf{c}^\top \mathbf{x} : x \in P\}$ is finite and $F = \{x \in P : \mathbf{c}^\top \mathbf{x} = \delta\}$
3. $F = \{x \in P : \mathbf{Ax}' = \mathbf{b}'\}$ for some subsytem of $\mathbf{Ax} \leq \mathbf{b}$.

This implies that if the max is bounded then the maximum is attained at a face of P. Let P be a polyhedron and F be a face of P. Then F is again a polyhedron. Further a set $F' \subseteq F$ is a face of P iff it is a face of F.

Definition 5.2.6 (Facet)**.** Let P be a polyhedron. A facet of P is amaximum face distinct from P.

Theorem 5.2.2 (Facets)**.** Let P be a feasible region with dimension $n - \text{rank}(\mathbf{A})$. Let $\mathbf{Ax}' = \mathbf{b}'$ be a minimal inequality system such that $P = \{\mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{Ax}' \leq \mathbf{b}'\}$. Then each inequality $\mathbf{Ax}' \leq \mathbf{b}'$ is facet defining and each facet of P is defined by an inequality.

Theorem 5.2.3 (Hoffman and Kruskal)**.** Let $P = \{\mathbf{Ax} \leq \mathbf{b}\}$ be a polyhedron. A non-empty subset $F \subseteq P$ is a minimal fae of P if and only if $F = \{\mathbf{x}' : \mathbf{Ax}' = \mathbf{b}'\}$ for some subsystem of $\mathbf{A}'\mathbf{x} \leq \mathbf{b}'$ of $\mathbf{Ax} \leq \mathbf{b}$.

Corollary 5.2.1. The minimal faces of polytopes are vertices.