Examlet 1 Study Guide

Tuesday, January 31, 2017 11:48 AM

Here is a non-exhaustive list of questions you should be able to answer as you prepare for the examlet.

Introduction

- What does $\mathrm{f(n)} = \mathrm{O(g(n))}$ mean?

  $T(n) = C \cdot g(n)$

- If $\mathrm{T(n)}$ represents timing for a problem size n and you know $T(n) = O(n^2)$ as well as the timing for $n = 10$, what will the timing for $n = 20$ be?

  $$T(n = 20) = \left(\frac{20}{10}\right)^2 \cdot T(n = 10) = 4 \cdot T(n = 10)$$

- What is the (asymptotic, i.e. big-O) cost of matrix-matrix multiplication?

  $O(n^3)$

- What types of quantities can be estimated using Big-O notation?

  Runtime, space, etc.

Linear Algebra Recap

- What is a vector space? What conditions do they satisfy?
  a. There exists an zero vector $\vec{0}$
  b. Any linear combination of its vector is in the vector space
  c. Any scaled of them is in the vector space
  d. Distributive, commutative
- What is a linear function? What conditions do they satisfy?

  $f(a + b) = f(a) + f(b)$

- What does 'linearly independent' mean?

  They cannot be made by performing linear combination

- What is a basis? What conditions does it satisfy?

  A set of vectors is a basis of a field if and only if

  a. They are linearly independent
  b. They span the field

- Given a basis, how can a given vector be represented in coordinates?

- $B = [\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3]$
  - Given v, we can represent it as $a_1\boldsymbol{b}_1 + a_2\boldsymbol{b}_2 + a_3\boldsymbol{b}_3$
- How do matrices represent linear functions?
  - In a similar way as above
- What does it mean for a matrix to be invertible?

  A is invertible, i.e. **A** has an inverse, is nonsingular, or is nondegenerate.

  A is row-equivalent to the *n*-by-*n* identity matrix $\mathbf{I}_n$.

  A is column-equivalent to the *n*-by-*n* identity matrix $\mathbf{I}_n$.

  A has *n* pivot positions.

  det **A** ≠ 0. In general, a square matrix over a commutative ring is invertible if and only if its determinant is a unit in that ring.

  A has full rank; that is, rank **A** = *n*.

  The equation **Ax** = **0** has only the trivial solution **x** = **0**

  Null **A** = {0}

  The equation **Ax** = **b** has exactly one solution for each **b** in K^n.

  The columns of **A** are linearly independent.

  - The columns of **A** span K^n

  Col **A** = K^n

  The columns of **A** form a basis of K^n.

  The linear transformation mapping **x** to **Ax** is a bijection from K^n to K^n.

  There is an *n*-by-*n* matrix **B** such that **AB** = $\mathbf{I}_n$ = **BA**.

  The transpose $\mathbf{A}^\mathrm{T}$ is an invertible matrix (hence rows of **A** are linearly independent, span K^n, and form a basis of K^n).

  The number 0 is not an eigenvalue of **A**.

  The matrix **A** can be expressed as a finite product of elementary matrices.

  The matrix **A** has a left inverse (i.e. there exists a **B** such that **BA** = **I**) *or* a right inverse (i.e. there exists a **C** such that **AC** = **I**), in which case both left and right inverses exist and **B** = **C** = $\mathbf{A}^{-1}$.
- How does matrix-matrix (and matrix-vector) multiplication work, numerically?
  - If $C = AB$
  - Then $C_{ij} = \sum_{k=1}^{n} A_{ik}B_{kj}$
- What is a permutation matrix?

    - It is a matrix that is formed by switching rows of an identity matrix

Python

Note: You will have access to a set of documentation for Python and its numerical libraries.

- How do you express the following in Python: integer, real number, string, list, tuple?
    - (integer): 1, 2, 10, -2
    - (real number (float)): 1.2, 3., -.5
    - (string): "a", 'b'
    - (list): [1, 2, 3]
    - (tuple): (4, 5, 6), (2,)
- How do you assign a value to a variable in Python?
    - <variable> = <value>
- How do you write a for loop in Python?
    - for <blah> in <foo>:
- How do you write an if conditional statement in Python?
    - if <condition>:
- How do you write an while loop in Python?
    - Simply use "while <condition>:"
- What happens when a Python value (e.g. a list) is modified in-place?
    - All variables associated with this object will have the modified values
- How do you avoid in-place modification?

    Call copy()
- How do you create a numpy array? from given data? filled with zeros? with equally spaced values?
    - np.array([1, 2, 3, 4, 5])
    - np.zeros(50)
    - np.arange(10)
- What is the shape of a numpy array?
    - The dimension of the matrix
- How do you extract the nth row/column of a numpy array?
    - Row, column = a.shape
- How many entries are there in a numpy array of shape (10, 20)?
    - 200
- For a numpy array a of shape (10, 20, 30), what is the shape of a[:,3:5]?
    - (10, 2, 30)

- (10, 2, 30)

Taylor Approximation

- Given a function f, find its Taylor expansion about an expansion center c of a given order.

$$T_n(x) = f(c) + f^{(1)}(c)\cdot(x-c) + \frac{f^{(2)}(c)\cdot(x-c)^2}{2!} + \cdots + \frac{f^{(n)}(c)\cdot(x-c)^n}{n!}$$

- Provide an estimate (in Big-O notation) of the truncation error of a Taylor expansion.

For a degree n Taylor expansion, the truncation error is $|f(x) - \tilde{f}(x)| = O(h^{n+1})$

- Use the truncation error estimate to estimate the error $E(h)$ for one distance h_2 given the error for another distance h_1.

$$\mathrm{Error(h)} = C \cdot h^{n+1} = O(h^{n+1})$$

Thus $\mathrm{Error}(h_2) = C \cdot h_2^{n+1} = C \cdot h_1^{n+1} \cdot \left(\frac{h_2}{h_1}\right)^{n+1} = \mathrm{Error}(h_1)\cdot\left(\frac{h_2}{h_1}\right)^{n+1}$

- Have a heuristic understanding of when Taylor expansions will not converge, as demonstrated in class.

When $n+1$ term is worse than n term (the ratio of them should be less than 1 in order to converge)

Examlet 2 Study Guide

Sunday, February 19, 2017 1:47 PM

Here is a non-exhaustive list of questions you should be able to answer as you prepare for the examlet.

Past chapters

See the study guide for examlet 1. Recall from the course policies that our examlets are cumulative. However, the focus will be on new material.

Interpolation

- What is interpolation? What are interpolation nodes?
    - Interpolation is the process of recovering polynomials from nodes
    - Interpolation nodes are the points $\left(x_i, f(x_i)\right)$ where f is the true function that we want to recover
- What is a Vandermonde matrix?
    - It is a matrix where the rows are different points and the columns are different transformation of the points
- What is the monomial basis?
    - Monomial: polynomial with only one term. E.g. $x, x^2, x^3, \ldots$
- How does one determine the coefficients of a polynomial interpolant?
    - $coeff = V^{-1}f$
- What is a generalized Vandermonde matrix?
    - Instead of using monomial, in generalized Vandermonde matrix, the transformations are some general set of functions

    Is this technique limited to the monomials $\{1, x, x^2, x^3, \ldots\}$?

    No, not at all. Works for any set of functions $\{\varphi_1, \ldots, \varphi_n\}$ for which the generalized Vandermonde matrix

    - $$\begin{pmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \cdots & \varphi_n(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \cdots & \varphi_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(x_n) & \varphi_2(x_n) & \cdots & \varphi_n(x_n) \end{pmatrix}$$

is invertible.

- How can interpolation be used to predict interpolant integrals or derivatives based on function values at a set of nodes?
    - $$\begin{aligned} \int_s^t f(x)dx &\approx \int_s^t a_0 + a_1x + a_2x^2 + a_3x^3 dx \\ &= a_0 \int_s^t 1dx + a_1 \int_s^t x \cdot dx + a_2 \int_s^t x^2 dx + a_3 \int_s^t x^3 dx \end{aligned}$$
    - The same apply for generalized set of functions
- What is the asymptotic (big-O) behavior of the error in interpolation? What does h represent in the error term?
    - For an degree n polynomial, the interpolation generally result in $O(h^{n+1})$ error (using $n + 1$ points)
    - h is the interval between points
- For what types of functions do you expect interpolation to work well? For which is that not the case?
    - When we are estimating a polynomials where the degree of it is less than the number of nodes we are using

Monte Carlo

- What is a random variable?
    - A random variable X is a function that depends on "the (random) state of the world".
- What is a distribution function? What requirements does it satisfy?
    - Discrete of continuous. Describe how likely each individual value of X is.
    - Need $p_i \geq 0$ for discrete distribution, $p(x) \geq 0$ for continuous distribution to make sense
- What is a sample?

  Sample: A sample $s_1, \ldots, s_N$ of a random variable X, are instances of the random variables
- What is a sample mean?
    - The arithmetic means of the samples, or equivalently, the expected value of the random variable
- What is an expected value? What is variance?

Define the 'expected value' of a random variable.

For a discrete random variable X:

For a discrete random variable X:

- $$E[f(X)] = \sum_{i=1}^{n} p_i f(x_i)$$

  For a continuous random variable:

  $$E[f(X)] = \int_{\mathbb{R}} f(x) \cdot p(x) dx$$

- Define variance of a random variable.

  $$\sigma^2[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2.$$

- How does taking a function of a set of samples (equivalently, a random variable) change the probability density function of their distribution?
  - From HW3 Q1:
    - $$q(a) = \lim_{h \to 0} \frac{f^{-1}(a+h) - f^{-1}(a)}{h} = (f^{-1}(a))'$$
  - (not sure whether this what the question is asking for, though)
- What assumptions are needed on a set of samples to guarantee convergence of their average to the mean of a random variable?
  - The Law of Large Numbers: As the number of samples $N \to \infty$, the average of samples converges to the expected value with probability 1.

  What can samples tell us about the distribution?

  $$P\left[\lim_{N\to\infty} \frac{1}{N}\left(\sum_{i=1}^{N} s_i\right) = E[X]\right] = 1.$$

  Or for an expected value,

  $$E[X] \approx \frac{1}{N}\left(\sum_{i=1}^{N} s_i\right)$$

  - This will converge if the samples are from the given distribution
- How do you approximate an expected value of one random variable based on a sample of another?

- Use Monte Carlo:
  - Monte Carlo methods are algorithms that compute approximations of desired quantities or phenomena based on randomized sampling.
- If sampling from a distribution $p(x)$ is hard, then we can use another distribution $\tilde{p}(x)$ that we can sample from (e.g. uniform distribution), then
  - $$E[X] = \int_{\mathbb{R}} x \cdot p(x)dx = \int_{\mathbb{R}} x\frac{p(x)}{\tilde{p}(x)} \cdot \tilde{p}(x)dx$$
    $$= \int_{\mathbb{R}} \tilde{x}\frac{p(\tilde{x})}{\tilde{p}(\tilde{x})} \cdot \tilde{p}(\tilde{x})dx = E\left[\tilde{X} \cdot \frac{p(\tilde{X})}{\tilde{p}(\tilde{X})}\right]$$
- Then we can approximate $E[X]$ by sampling $\tilde{s}_i$ from $\tilde{X}$:
  - $$E[X] \approx \frac{1}{N}\sum_{i=1}^{N} \tilde{s}_i \cdot \frac{p(\tilde{s}_i)}{\tilde{p}(\tilde{s}_i)}$$

    (due to the Law of Large Number)

- How does one use Monte Carlo for integration?
  - $$\int_{\Omega} g(x)dx = \int_{\Omega} \frac{g(x)}{\tilde{p}(x)}\tilde{p}(x)dx$$
    $$= E\left[\frac{g(\tilde{X})}{\tilde{p}(\tilde{X})}\right] \approx \frac{1}{N}\sum_{i=1}^{N} \frac{g(\tilde{s}_i)}{\tilde{p}(\tilde{s}_i)}$$
  - For 2D integral:

    $$G = \int\int_{\Omega} f(x,y)dxdy = \int_0^L \int_0^L f(x,y)\mathbf{1}_{\Omega}(x,y)dxdy$$

    Using a uniform random variable with distribution $\tilde{p}(x,y) = 1/L^2$

    $$G = |\Omega|E\left[f(X)\frac{p(X)}{\frac{1}{L^2}}\right] = |\Omega|L^2E[f(X)p(X)]$$

    $$\approx \frac{|\Omega|L^2}{N}\sum_{i=1}^{N} f(x_i,y_i)p(x_i,y_i)$$

$$= \frac{L^2}{N} \sum_{i=1}^{N} f(x_i, y_i) \mathbf{1}_\Omega(x_i, y_i).$$

- What is the appropriate scaling factor if Monte Carlo integration is done based on samples from a larger region than the integration domain?
    - Not sure what the question is asking for.. Probably for uniform distribution, L for 1d and L^2 for 2d
- What is the asymptotic (big-O) behavior of the error in sampling?
    - The Central Limit Theorem states that with

      $$S_N := X_1 + X_2 + \cdots + X_n$$

      for the (X_i) independent and identically distributed according to random variable X with variance σ^2, we have that

      $$\frac{S_N - NE[X]}{\sqrt{\sigma^2 N}} \to \mathcal{N}(0,1),$$

      i.e. that term approaches the normal distribution. As we increase N, σ^2 stays fixed, so the asymptotic behavior of the error is

      $$\left|\frac{1}{N} S_N - E[X]\right| = O\left(\frac{1}{\sqrt{N}}\right).$$
- In relative terms, for what types of problems is Monte Carlo effective and for what types of problems is it not?
    - Monte Carlo converge rather slowly, so it might not be a good idea to use Monte Carlo if there is an analytical solution that is much easier to generated

Errors

- What are absolute and relative errors?
    - Absolute error: $|x_0 - \tilde{x}|$
    - Relative error: $\frac{|x_0 - \tilde{x}|}{|x_0|}$
- What does it mean for a result to have n accurate digits?
    - It means that there are n leading (most siginificants) non-zero digits that are accurates

are accurates

- E.g. "5 accurate digits":
  - 3.1415777777
- '$\tilde{x}$ has n accurate digits' is roughly equivalent to having a relative error of 10^{-n}.
$$\frac{|\tilde{x} - x_0|}{|x_0|} < 10^{-n}.$$

- What are common sources of error in numerical methods?
  - Truncation error:
    (E.g. Taylor series truncation, finite-size models, finite polynomial degrees)
  - Rounding error
    (Numbers only represented with up to ~15 accurate digits.)
- How does the number of accurate digits relate to rounding?
  - Rounding to n digits leaves n accurate digits–a relative error of 10^{-n}
- What is a condition number?
  - The smallest κ such that
    $\text{Relative error in output} \le \kappa \cdot \text{Relative error in input}$
- What can you say about a condition number given data points with relative errors on inputs and outputs?
  - $$\kappa = \max_x \frac{\text{rel error in output } f(x)}{\text{rel error in input } x} = \max_x \frac{\frac{|f(x) - f(x+\Delta x)|}{|f(x)|}}{\frac{|\Delta x|}{|x|}}$$

Floating Point Basics

- What is fixed point arithmetic? How are numbers represented in fixed point?
  - A fixed number of bits with exponents ≥ 0 and a fixed number of bits with exponents < 0
- What is the significand? the exponent? of a floating point number?
  - E.g.: for $(1.101)_2 \cdot 2^3$
    - 1.101 is significand
    - 3 is exponent
- What is floating point arithmetic? How do the significand and exponent

define a floating point number?

  - A fixed number of significand and a fixed number of exponent (signed integer)

- What relative error does a floating point number representation of a real number have?
  - Around the machine epsilon
- What numbers can be more accurately represented in fixed point than floating point? What about the other way around?
  - Fix point can represent numbers that are not multiples of 2 more accurately
  - Floating point can represent numbers that are extremely large (in magnitude) or extremely close to zero more accurately

Examlet 3 Study Guide

Tuesday, March 14, 2017 12:14 AM

Here is a non-exhaustive list of questions you should be able to answer as you prepare for the examlet.

Past chapters

See the
Study guide for examlet 1
Study guide for examlet 2
(Recall from the course policies that our examlets are cumulative.)

Floating Point

- What is fixed point arithmetic? How are numbers represented in fixed point?
  - Fix number of bits with exponent ≥ 0, fix number of bits with exponent < 0
- What is floating point arithmetic? How are numbers represented in floating point?
  - Fix number of significand, fix number of exponent
- What is the significand? the exponent? of a floating point number?
  - Significand: the base, Exponent: the power
- What is machine epsilon?
  - The smallest number such that $1 + \epsilon \neq 1$
  - Also the maximum relative error in any floating point operation
- How can you quantify the least possible amount of rounding error that floating point arithmetic introduces with every operation?
  - Machine epsilon?
  - Use condition number: $\kappa = \max_x \frac{\text{rel error in output } f(x)}{\text{rel error in input } x} = \max_x \frac{\frac{|f(x) - f(x+\Delta x)|}{|f(x)|}}{\frac{|\Delta x|}{|x|}}$
- How are floating point numbers stored? What is the 'implicit one' in the significand?
  - Significand + exponent.
- How is zero represented in floating point?
  - Subnormal number: Turn off the leading one

- What are subnormal numbers? What is (gradual and non-gradual) underflow? overflow?
    - Smaller than the smallest normal number.
- What can we say about error in the subnormal representation of numbers?
    - The error will be larger, because subnormal numbers don't have as many accurate digits as normal numbers.
- How is floating point addition performed?
    1. Bring both numbers onto a common exponent
    2. Do grade-school addition from the front, until you run out of digits in your system.
    3. Round result.
- What is catastrophic cancellation? How can you estimate the relative error (/number of digits) in the result of a calculation that incurs catastrophic cancellation?
    - Subtract two numbers of very similar magnitude that result in a significant increase of relative error

Computational Linear Algebra

- How are images represented as vectors? What does addition/scalar multiplication mean for them?
    - Numbers… larger number means more bright…?
- How are sound clips represented as vectors? What does addition/scalar multiplication mean for them?
    - Numbers as well, as sin curves. Increase/decrease in frequency & amplitude
- How are shapes represented as vectors? What does addition/scalar multiplication mean for them?
    - Vertices. Transformation..?
- How do matrices operate on bases? How can linear operations on a basis be expressed using a basis?

    It represents a *linear function* between two vector spaces $f : U \to V$ in terms of bases $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ of U and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ of V. Let

$$\boldsymbol{u} = \alpha_1 \boldsymbol{u}_1 + \cdots + \alpha_n \boldsymbol{u}_n$$

and

$$\boldsymbol{v} = \beta_1 \boldsymbol{v}_1 + \cdots + \beta_m \boldsymbol{v}_m.$$

- Then f can *always* be represented as a matrix that obtains the βs from the αs:

$$\boxed{\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_m \end{pmatrix}.}$$

Graphs and Sparsity

- What is an adjacency matrix?
  - i is connected to $j \rightarrow A_{i,j} = A_{j,i} = 1$
- What is a Laplacian matrix?
  - Let D = diagonal matrix where the diagonals are the degree of node
  - $L = D - A$
- How do these representations change for directed or weighted graphs?
  - Directed: from i to $j \rightarrow A_{j,i} = 1$
  - Weighed: i is connected to $j \rightarrow A_{i,j} = A_{j,i} = \text{weight}(i, j)$
- What does matrix-vector multiplication with an adjacency matrix mean?
  - Give the array of nodes that can be visited
- What is a Markov chain? What is the Markov property?
  - Weight(i, j)=probability of visiting node j from node I
  - Markov property: memory less (next state depend only on the current one)
- What is a transition matrix/graph? What is a steady state?
  - E.g. Markov chain
  - $Mv_s = v_s$
- What is a sparse matrix?
  - A matrix with lots of zeros
- How does CSR format for the representation of sparse matrices work?
  - Store row-starts, columns, and values.
  - RowStart: zero based value indicate the indices associate with the start of a row
  - Columns: column where each value is in
  - Values: non zero values
- What would matrix-vector multiplication with CSR matrices look like?

- What would matrix-vector multiplication with CSR matrices look like?
  - ```python
    import numpy as np
    Ax = np.zeros(4)

    for i in range(len(IA) - 1):
        row = V[IA[i]:IA[i + 1]]
        foo = 0
        for j in JA[IA[i]:IA[i + 1]]:
            Ax[i] += x[j] * row[foo]
            foo += 1
    ```
- What is the computational cost of sparse-matrix vector multiplication?
  - $O(\mathrm{nnz}(A) + n)$, where $\mathrm{nnz}(A)$ is the number of non-zero values in A

Vector and Matrix Norms

- What criteria does a vector norm have to satisfy?
  - $\|x\| > 0$ <=> $x \neq 0$
  - $\|\gamma x\| = |\gamma| \|x\|$ for all scalar γ
  - Obeys triangle inequality $\|x + y\| \leq \|x\| + \|y\|$
- What is the triangle inequality?
  - $\|x + y\| \leq \|x\| + \|y\|$
- What are the p-norms?
  - $$\left\|\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}\right\|_p = \sqrt[p]{|x_1|^p + \cdots + |x_n|^p} \quad (p \geqslant 1)$$
- What is the "unit ball" of a norm?
  - All x such that $\|x\|_2 = 1$
- What is an induced matrix norm?
  - $$\|A\| := \max_{\|\boldsymbol{x}\|=1} \|A\boldsymbol{x}\|$$
- What is the Frobenius matrix norm?
  - $$\|A\|_F := \sqrt{\sum_{i,j} a_{ij}^2}$$
- What does an induced matrix norm imply about the amplification of a vector norm during matrix-vector multiplication?
    ```

vector norm during matrix-vector multiplication?

Matrix norms inherit the vector norm properties:

1. $\|A\| > 0 \Leftrightarrow A \neq \mathbf{0}$.
2. $\|\gamma A\| = |\gamma| \, \|A\|$ for all scalars γ.
3. Obeys triangle inequality $\|A + B\| \leqslant \|A\| + \|B\|$

- But also some more properties that stem from our definition:

1. $\|A\boldsymbol{x}\| \leqslant \|A\| \, \|\boldsymbol{x}\|$
2. $\|AB\| \leqslant \|A\| \, \|B\|$ (easy consequence)

Both of these are called submultiplicativity of the matrix norm.

- How can the matrix norm of a diagonal matrix be computed?
  - The norm of any diagonal matrix (or for that matter, any normal matrix) is the maximum of the absolute values of its eigenvalues
- How can an induced matrix norm be estimated by sampling?
  - Sampling a bunch of vectors, bring the norm to 1, do the dot product, calculate the norm of the product, and find one with maximum norm

Examlet 4 Study Guide

Monday, April 10, 2017 1:59 PM

Here is a *non-exhaustive* list of questions you should be able to answer as you prepare for the examlet.

Past chapters

See the

- Study guide for examlet 1
- Study guide for examlet 2
- Study guide for examlet 3

(Recall from the course policies that our examlets are cumulative.)

Norms and Conditionining

- What criteria does a vector norm have to satisfy?
  - $\|\boldsymbol{x}\| > 0 \Leftrightarrow \boldsymbol{x} \neq \boldsymbol{0}$
  - $\|\gamma\boldsymbol{x}\| = |\gamma|\|\boldsymbol{x}\|$ for all scalar γ
  - Obeys triangle inequality $\|\boldsymbol{x}\| + \|\boldsymbol{y}\| \leq \|\boldsymbol{x} + \boldsymbol{y}\|$
- What is the triangle inequality?
  - $\|\boldsymbol{x}\| + \|\boldsymbol{y}\| \leq \|\boldsymbol{x} + \boldsymbol{y}\|$
- What are the p-norms?
  - $\left\| \begin{pmatrix} x_1 \\ \\ x_n \end{pmatrix} \right\|_p = \sqrt[p]{|x_1|^p + \cdots + |x_n|^p} \quad (p \geqslant 1)$
- What is the "unit ball" of a norm?
  - All vectors whose norm is 1
- What is a matrix norm? submultiplicativity?
  - Matrix norm: $\|A\| \coloneqq \max_{\|\boldsymbol{x}\|=1} \|A\boldsymbol{x}\|$
  - Submultiplicativity:
    - $\|A\boldsymbol{x}\| \leq \|A\|\|\boldsymbol{x}\|$
    - $\|AB\| \leq \|A\|\|B\|$
- How can the matrix norm of a diagonal matrix be computed?
  - p norm of diagonal matrix is the maximum absolute eigenvalue (maximum diagonal entry)

- What is special about matrix norms of orthogonal matrices?
    - 2 norm of orthogonal matrix is 1
- What is the condition number of solving a linear system? matrix-vector multiplication?
    - $$\begin{aligned} \frac{\text{rel err. in output}}{\text{rel err. in input}} &= \frac{\|\Delta \boldsymbol{x}\| \,/\, \|\boldsymbol{x}\|}{\|\Delta \boldsymbol{b}\| \,/\, \|\boldsymbol{b}\|} = \frac{\|\Delta \boldsymbol{x}\| \, \|\boldsymbol{b}\|}{\|\Delta \boldsymbol{b}\| \, \|\boldsymbol{x}\|} \\ &= \frac{\|A^{-1}\Delta \boldsymbol{b}\| \, \|A\boldsymbol{x}\|}{\|\Delta \boldsymbol{b}\| \, \|\boldsymbol{x}\|} \\ &\leqslant \|A^{-1}\| \, \|A\| \frac{\|\Delta \boldsymbol{b}\| \, \|\boldsymbol{x}\|}{\|\Delta \boldsymbol{b}\| \, \|\boldsymbol{x}\|} \\ &= \|A^{-1}\| \, \|A\| \,. \end{aligned}$$
- What is the condition number of a matrix?
    - $\mathrm{cond}(A) = \|A\| \|A^{-1}\|$
- How can the condition number of a diagonal matrix be calculated?
    - 2 norm: largest absolute eigenvalue times inverse of smallest absolute eigenvalue
- How does a condition number affect the number of accurate digits in a result?
    - Larger condition number: fewer accurate digits
- How can the norm/condition number of a matrix A be found from the plot of $A\boldsymbol{x}$ for $\|\boldsymbol{x}\| = 1$?
    - Sample a lot of x, and find one that gives maximum $\|A\boldsymbol{x}\|$

LU

- What do forward/backward substitution accomplish? How? At what computational cost?
    - Forward/backward substitution solve linear system
    - Computational cost: $O(n^2)$
- What is LU factorization? How does it work? What is its computational cost?

$$\boxed{A = \begin{bmatrix} a_{11} & \mathbf{a_{12}} \\ \mathbf{a_{21}} & A_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \mathbf{l_{21}} & L_{22} \end{bmatrix} \cdot \begin{bmatrix} u_{11} & \mathbf{u_{12}} \\ 0 & U_{22} \end{bmatrix}}$$

First, we can observe

$$\begin{bmatrix} a_{11} & \mathbf{a_{12}} \end{bmatrix} = 1 \cdot \begin{bmatrix} u_{11} & \mathbf{u_{12}} \end{bmatrix},$$

so the first row of U is just the first row of A.

- Second, we notice $\mathbf{a_{21}} = \mathbf{l_{21}} \cdot u_{11}$, so $\mathbf{l_{21}} = \mathbf{a_{21}}/u_{11}$.

  To get L_{22} and U_{22}, we use the equation,

  $$A_{22} = \mathbf{l_{21}} \cdot \mathbf{u_{12}} + L_{22} \cdot U_{22}.$$

  To solve, perform the Schur complement update and 'recurse',

  $$[L_{22}, U_{22}] = \text{LU-decomposition}(A_{22} - \underbrace{\mathbf{l_{21}} \cdot \mathbf{u_{12}}}_{\text{Schur complement}})$$

- Computation cost: $O(n^3)$

- How does arithmetic with block matrices work? What is the Schur complement update?
  - Same as above
- What is the asymptotic cost of LU factorization? How do you apply it to estimate factorization time?

  > What is the computational cost of carrying out LU factorization on an $n \times n$ matrix?

  - $O(n)$ cost to form $\mathbf{l_{21}}$

    $O(n^2)$ to perform Schur complement update $\mathbf{l_{21}}\mathbf{u_{12}}$

    Overall $O(n^3)$ since we continue for n steps

    More precisely, we have n outer products of decreasing size,

    $$\sum_{i=1}^{n} 2i^2 \approx 2n^3/3.$$

Pivoting

- Does the LU factorization always exist? Why is pivoting needed?
  - No. LU factorization does not exist if the matrix has zero diagonal.
  - Pivoting switch the rows so that the diagnals are non-zero.
- What is partial pivoting? What is its purpose? How does it work?
  - Partial pivoting is done by multiply a permutation matrix on each iteration with the original matrix, thereby move the largest leading entry to the top

- What is a permutation matrix? How does it help realize partial pivoting?
  - Permutation matrix is obtained by switching the rows of an identity matrix
- What is the form of an LU factorization with pivoting? (can be $PA = LU$ or $A = P^{-1}LU$ - note that $P^{-1} = P^{T}$)
  - $PA = LU$
  - Since $Ax = b$, we have $P^{-1}LUx = b \rightarrow x = U^{-1}L^{-1}Pb$
- What is the cost of LU factorization with pivoting?
  - Still $O(n^3)$ … ?

LU: Applications

- How is LU factorization used to solve a linear system of equations $A\boldsymbol{x} = b$?
  - $A = LU, A\boldsymbol{x} = \boldsymbol{b} \rightarrow \boldsymbol{x} = U^{-1}L^{-1}\boldsymbol{b}$
  - Runtime: $O(n^3) + O(n^2)$
- How is LU factorization used to solve many linear systems of equations $A\boldsymbol{x}_i = \boldsymbol{b}_i$ with many different right-hand sides?
  - Do LU for once and use forward/backward substitution on each of the pairs
  - Runtime: $O(n^3) + O(n^3)$
- How is LU factorization used to solve a matrix equation $AX = B$?
  - Same as above
- Be able to (write down algorithms to) solve more complicated matrix equations involving triangular/orthogonal/other matrices.
  - Sure

Eigenvectors and Eigenvalues

- What is an eigenvector? an eigenvalue of a matrix? (i.e. know the definition)
  - $A\boldsymbol{x} = \lambda\boldsymbol{x}$
  - $$\begin{aligned} & A\boldsymbol{x} = \lambda\boldsymbol{x} \\ \Leftrightarrow \quad & (A - \lambda I)\boldsymbol{x} = 0 \\ \Leftrightarrow \quad & A - \lambda I \text{ is singular} \\ \Leftrightarrow \quad & \det(A - \lambda I) = 0 \end{aligned}$$

$\iff \det(A - \lambda I) = 0$

- What are eigenvectors linearly independent?
  - When they have distinct absolute eigenvalues
- What is power iteration?
  - $\lim_{k \to \infty} \|(1/\lambda_1^k) A^k (\alpha_1 \boldsymbol{x}_1 + \cdots + \alpha_n \boldsymbol{x}_n)\| = \alpha_1$
- What can be obtained using power iteration?
  - An approximation of the closest eigenvector
- What is normalized power iteration? What problem does it address?
  - Overflow
- Given an approximate eigenvector, how can you estimate eigenvalues? What is the Rayleigh Quotient?
  - $\frac{\boldsymbol{x}^T A \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \lambda$ is the Rayleigh Quotient. It gives the eigenvalue of x if x is an eigenvector

Examlet 5 Study guide

Friday, April 28, 2017 1:40 PM

Here is a *non-exhaustive* list of questions you should be able to answer as you prepare for the examlet.

Past chapters

See the

- Study guide for examlet 1
- Study guide for examlet 2
- Study guide for examlet 3
- Study guide for examlet 4

(Recall from the course policies that our examlets are cumulative.)

Eigenvectors and Eigenvalues

- All topics for examlet 4 study guide
- How do eigenvectors/eigenvalues change under Shift? Inversion? Taking the nth power? Taking the inverse?
    - Shift: $A \to A - \sigma I$
        - $(A - \sigma I)\boldsymbol{x} = (\lambda - \sigma)\boldsymbol{x}$
    - Inversion: $A \to A^{-1}$
        - $A^{-1}\boldsymbol{x} = \lambda^{-1}\boldsymbol{x}$
    - Power: $A \to A^n$
        - $A^n\boldsymbol{x} = \lambda^n\boldsymbol{x}$
    - Inverse iteration: $A \to (A - \sigma I)^{-1}$
        - $(A - \sigma I)^{-1}\boldsymbol{x} = (\lambda - \sigma)^{-1}\boldsymbol{x}$
- When is a matrix diagonalizable? Are all matrices diagonalizable?
    - A matrix is diagonalizable if we have n eigenvectors with different eigenvalues.
    - No, definitely no.
- How does the error in power iteration behave?
    - For usual power iteration:
        - $e_{k+1} \approx \frac{|\lambda_2|}{|\lambda_1|} e_k$

- For inverse iteration:
  - $e_{k+1} \approx \dfrac{|\lambda_{\text{closest}} - \sigma|}{|\lambda_{\text{second-closest}} - \sigma|} e_k$
- Inverse iteration will converge to the eigenvector whose $|\lambda|$ is closest to σ

- Under what circumstances will power iteration converge? When can we not guarantee that it will?
  - It will not converge if the largest eigenvalue is as large as the second largest eigenvalue. i.e. $|\lambda_1| = |\lambda_2|$
- What is inverse iteration?
  - $\boldsymbol{x}_{k+1} = (A - \sigma I)^{-1}\boldsymbol{x}_k$
- What is Rayleigh quotient iteration?
  - $\boldsymbol{x}_{k+1} = (A - \sigma_k I)^{-1}\boldsymbol{x}_k$ where $\sigma_k = \boldsymbol{x}_k^T A \boldsymbol{x}_k / \boldsymbol{x}_k^T \boldsymbol{x}_k$
- How can the power method be applied to find the equilibrium distribution of a Markov chain?
  - The steady state is equivalent to a vector with eigenvalue of 1: $A\boldsymbol{p} = \lambda \boldsymbol{p}$ where $\lambda = 1$

SVD

- What is the singular value decomposition?

  The SVD is a factorization of an $m \times n$ matrix into

  $$A = U\Sigma V^T, \quad \text{where}$$

  - ▸ U is an $m \times m$ orthogonal matrix (Its columns are called 'left singular vectors'.)
  - ▸ Σ is an $m \times n$ diagonal matrix with the singular values on the diagonal

    $$\Sigma = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & & 0 \end{pmatrix} \qquad \text{Convention: } \sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n \geqslant 0.$$

  - ▸ V^T is an $n \times n$ orthogonal matrix (V's columns are called 'right singular vectors'.)

- What are left/right singular vectors with respect to $A^T A$ and AA^T? singular values?
  - The left singular vector U is the eigenvectors of AA^T
  - The right singular vector V is the eigenvectors of $A^T A$
- What properties do the singular vectors and singular values satisfy?
  - The singular obey the rule as indicated above
  - The singular values are square root of the eigenvalues of A
- How can the SVD be computed?

  1. Compute the eigenvalues and eigenvectors of $A^T A$.

  $$A^T A \boldsymbol{v}_1 = \lambda_1 \boldsymbol{v}_1 \quad \cdots \quad A^T A \boldsymbol{v}_n = \lambda_n \boldsymbol{v}_n$$

  2. Make a matrix V from the vectors $\boldsymbol{v}_i$:

  $$V = \begin{pmatrix} | & & | \\ \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_n \\ | & & | \end{pmatrix}.$$

  ($A^T A$ symmetric: V orthogonal if columns have norm 1.)

  3. Make a diagonal matrix Σ from the square roots of the eigenvalues:

  $$\boxed{\Sigma = \begin{pmatrix} \sqrt{\lambda_1} & & & \\ & \ddots & & \\ & & \sqrt{\lambda_n} & 0 \end{pmatrix}}$$

  4. Find U from

  - The left singular vector U is the eigenvectors of AA^T
  - The right singular vector V is the eigenvectors of A^T A

  $$A = U\Sigma V^T \quad \Leftrightarrow \quad U\Sigma = AV.$$

  (While being careful about non-squareness and zero singular values)
  In the simplest case:

  $$U = AV\Sigma^{-1}.$$

- Given a non-square matrix, what shape do the component matrices of the SVD have? In the 'full' case and the 'reduced' case?

the SVD have? In the "full" case and the "reduced" case?

- Assume A is an $m \times n$ matrix
- In "full" case:
  - U is a $m \times m$ matrix
  - Σ is a $m \times n$ matrix
  - V is a $n \times n$ matrix
- In "reduced" case:
  - U is a $m \times m$ matrix
  - Σ is a $m \times k$ matrix
  - V is a $k \times n$ matrix
  - Where $\mathrm{k} = \min(m, n)$

- How can the SVD be used for low-rank approximation?
  - $A_k = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \cdots + \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^T$
  - Then A_k is the best ran-k approximation to A
- What is the pseudoinverse? What properties does it satisfy?

  Define a 'pseudo-inverse' Σ^+ of a diagonal matrix Σ as

  - $$\Sigma_i^+ = \begin{cases} \sigma_i^{-1} & \text{if } \sigma_i \neq 0, \\ 0 & \text{if } \sigma_i = 0. \end{cases}$$
  - Then the pseudo-inverse of A is defined as $V\Sigma^+ U^T$

Least Squares

- How can you solve a (square) linear system using the SVD?
  - Solve $\Sigma V^T \boldsymbol{x} = U^T \boldsymbol{b}$
  - Cost: $O(n^2)$ - but more operations than using forward/backward substitution. Even worse when including comparison of LU vs. SVD.
- Why is the SVD helpful for (tall-and-skinny) least-squares system using the SVD? What is the residual in such a problem?
  - We can use SVD for lower rank approximation
  - The residual is

    $$\min_{\text{rank } B \leqslant k} \|A - B\|_F = \|A - A_k\|_F = \sqrt{\sigma_{k+1}^2 + \cdots \sigma_n^2}.$$
  - The residual is orthogonal to columns of A
- How can you solve a least-squares problem using the SVD?
  - Use pseudo-inverse.
  - $A\boldsymbol{x} \cong \boldsymbol{b}$ is solved by $A^+ \boldsymbol{b}$

,

- Given an SVD of the matrix and a right-hand side, how would you find the 2-norm of the residual of a least-squares problem?
  - The two norms are the square root of sum of the terms where the right hand side is nonzero whereas the singular value is zero
- How would you use code to solve a (short-and-fat/tall-and-skinny matrix) least-squares problem?
  - If short and fat -> infinitely many solutions
  - Tall and skinny -> using pseudo inverse

Interpolation

- What are the drawbacks of equal-spaced nodes in interpolation? How are those addressed by edge-clustered nodes?
  - Notes toward the end tend to be bad-behave
  - See Runge's phenomenon (oscillation of nodes at the end when using equal-space polynomial interpolation)
- What are the drawbacks of monomials as an interpolation basis? How are those addressed by orthogonal polynomials?
  - Monomials get closer and more similar as n increases (so becoming closer to linearly dependent)
  - Orthogonal polynomials make them linearly independent
- What does it mean for two functions to be orthogonal?
  - The dot product of the two functions is zero
  - $$\boldsymbol{f} \cdot \boldsymbol{g} = \sum_{i=1}^{n} f_i g_i = \langle \boldsymbol{f}, \boldsymbol{g} \rangle$$
    $$\langle f, g \rangle = \int_{-1}^{1} f(x) g(x) dx$$
    = 0
- What does it mean for two polynomials to be orthogonal?
  - Same. The inner product of the two polynomial should be zero.
  - E.g. Chebyshev
- How are the Legendre polynomials defined? and the Chebyshev polynomials?
  - Legendre polynomial: $q = ($ ... $q)$ for all prev q, then $q =$

- Legendre polynomial: $q = (prevq - proj_{\mathrm{prevq}} q)$ for all prev-q, then $q = \frac{q}{\|q\|}$ (so norm = 1)
  - Three equivalent definitions:
    - Result of Gram-Schmidt with weight $1/\sqrt{1-x^2}$

      What is that weight?

      1/ (Half circle), i.e. $x^2 + y^2 = 1$, with $y = \sqrt{1-x^2}$
    - $T_k(x) = \cos(k \cos^{-1}(x))$
    - $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$
- What are the Chebyshev interpolation nodes?
  - $$x_i = \cos\left(\frac{2i-1}{2k}\pi\right) \quad (i = \ldots, k).$$
  - Vandermonde of these can be applies in $O(n \log n)$ time
  - It is edge-clustered
  - (I think k is the n here)

Numerical Differentiation

- How can a first/second/third derivative be computed using interpolation? How would that be expressed using Vandermonde matrices?
  - First derivative: $\tilde{\boldsymbol{f}}' = V'V^{-1}\boldsymbol{f}$
  - Second derivative $\tilde{\boldsymbol{f}}' = V'V^{-1}V'V^{-1}\boldsymbol{f}$
  - Third derivative $\tilde{\boldsymbol{f}}' = V'V^{-1}V'V^{-1}V'V^{-1}\boldsymbol{f}$
- Given point values of a function, how can you use interpolation to compute an approximation of the derivative of that function at the same or different points?
  - Same point: just repeats above..?
  - Different point: use $V^{-1}V'V^{-1}$ to get an weight $\boldsymbol{\alpha}'$, then use it on top of new points
- What are finite difference formulas?
  - $$\frac{f(x+h) - f(x-h)}{2h}$$
- If you shorten the distance between points from, say, h to $h/2$, how will the finite difference formula change?

the finite difference formula change?

  - It will be more accurate...?
- If you shift a finite difference formula from, say $3+h$ to, say, $4+h$, how does the formula change?
  - Shift -> perhaps no change.
- What is the order of accuracy of this process? (I.e. how does the error depend on h?)
  - $$\max_{x\in[a,b]}\left|f'(x)-\tilde{f}'(x)\right| \leqslant C\cdot h^n$$
  - Where n is the degree (1 less than number of nodes)

Numerical Integration

- Given point values of a function, how can you use interpolation to compute an approximation of the definite integral (over some interval) of that function?
  Can call $\boldsymbol{w} := V^{-T}\boldsymbol{d}$ the quadrature weights and compute
  - $$\int_a^b \tilde{f}(x)dx = \boldsymbol{w}^T\boldsymbol{y} = \boldsymbol{w}\cdot\boldsymbol{y}.$$
  - $d_i = \int_a^b \varphi_i(x)dx$ can be computed ahead of time,
- How do quadrature rules make this process more efficient?
  - We can calculate the integral in the range $[0,1]$, and shift and scale them to calculate the integral in the range $[a,b]$
  - $$\int_a^b f(x)dx \underbrace{=}_{x=(b-a)\bar{x}+a} (b-a)\int_0^1 f((b-a)\bar{x}+a)d\bar{x}.$$
  - $$\int_a^b f(x)dx \approx (b-a)\mathbf{w}^T\mathbf{y}.$$
- If you shorten the distance between points from, say, h to $h/2$, how will the quadrature rule change?
  - See above. Halve of the original value.

- If you shift a quadrature rule from, say $3 + h$ to, say, $4 + h$, how does the formula change?
  - See above. No change in weight.
- What is the order of accuracy of this process? (I.e. how does the error depend on h?)
  - $$\left| \int_a^b f(x)dx - \int_a^b \tilde{f}(x)dx \right| \leqslant C \cdot h^{n+2}$$
  - Where $h = b - a$, n is the degree (1 less than the number of nodes)