Homework 1. Due Wednesday, Sept. 6.

1. (**4 pts**) (Problem 7 from Section 2.10 in D. Bindel's and J. Goodman's book "Principles of Scientific Computing") Starting with the declarations

```
      float x, y, z, w;
const float oneThird = 1/ (float) 3;
const float oneHalf  = 1/ (float) 2;
                  // const means these never are reassigned
```

   we do lots of arithmetic on the variables `x, y, z, w`. In each case below, determine whether the two arithmetic expressions result in the same floating point number (down to the last bit) as long as no `NaN` or `inf` values or denormalized numbers are produced.

   (a)

```
( x * y ) + ( z - w )
( z - w ) + ( y * x )
```

   (b)

```
( x +   y ) + z
  x + ( y   + z )
```

   (c)

```
  x * oneHalf + y * oneHalf
( x + y ) * oneHalf
```

   (d)

```
  x * oneThird + y * oneThird
( x + y ) * oneThird
```

2. **(10 pts)** The *tent map* of the interval $[0, 1]$ onto itself is defined as

$$f(x) = \begin{cases} 2x, & x \in [0, 1/2), \\ 2 - 2x, & x \in [1/2, 1]. \end{cases} \qquad (1)$$

   Consider the iteration $x_{n+1} = f(x_n)$, $n = 0, 1, 2, \ldots$.

   (a) What are the fixed points of this iteration, i.e. the points x^* such that $x^* = f(x^*)$? Show that these fixed points are unstable, i.e., if you start iteration at $x^* + \delta$ for any δ small enough then the next iterate will be farther away from x^* then $x^* + \delta$.

   (b) Prove that if x_0 is rational, then the sequence of iterates generated starting from x_0 is periodic.

(c) Show that for any period length p, one can find a rational number x_0 such that the sequence of iterates generated starting from x_0 is periodic of period p.

(d) Generate several long enough sequences of iterates on a computer using any suitable language (Matlab, Python, C, etc.) starting from a pseudorandom x_0 uniformly distributed on $[0, 1)$ to observe a pattern. I checked in Python and C that 100 iterates are enough. Report what you observe. If possible, experiment with single precision and double precision.

(e) Explain the observed behavior of the generated sequences of iterates.

Homework 2. Due Wednesday, Sept. 13.

1. (**6 pts**) Consider the polynomial space $\mathcal{P}_n(x)$, $x \in [-1,1]$. Let T_k, $k = 0, 1, \ldots, n$, be the Chebyshev basis in it. The Chebyshev polynomials are defined via

$$T_k = \cos(k \arccos x).$$

   (a) Use the trigonometric formula

$$\cos(a) + \cos(b) = 2\cos\left(\frac{a+b}{2}\right)\cos\left(\frac{a-b}{2}\right)$$

   to derive the three-term recurrence relationship for the Chebyshev polynomials

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x), \quad k = 1, 2, \ldots. \tag{1}$$

   (b) Consider the differentiation map

$$\frac{d}{dx} : \mathcal{P}_n \to \mathcal{P}_{n-1}.$$

   Write the matrix of the differentiation map with respect to the Chebyshev bases in $\mathcal{P}_n$ and $\mathcal{P}_{n-1}$ for $n = 7$. *Hint: you might find helpful properties of Chebyshev polynomials presented in Section 3.3.1 of Gil, Segure, Temme, "Numerical Methods For Special Functions". Chapter 3 of this book is added to Files/Refs on ELMS.*

2. (**6 pts**) Let $A = (a_{ij})$ be an $m \times n$ matrix.

   (a) Prove that the l_1-norm of A is

$$\|A\|_1 = \max_j \sum_i |a_{ij}|,$$

   i.e., the maximal column sum of absolute values. Find the maximizing vector.

   (b) Prove that the max-norm or l_∞-norm of A

$$\|A\|_{\max} = \max_i \sum_j |a_{ij}|,$$

   i.e., the maximal row sum of absolute values. Find the maximizing vector.

3. **(6 pts)** Consider the matrix

$$A = \begin{bmatrix} 1 & 10 \\ 0 & 1 \end{bmatrix}. \tag{2}$$

   (a) Find the Jordan form of A.

   (b) Find the 2-norm of A.

Homework 3. Due Wednesday, Sept. 20.

1. (**4 pts**) Let A be an $n \times n$ matrix. The Rayleigh quotient $Q(x)$ is the following function defined on all $x \in \mathbb{R}^n$:

$$Q(x) := \frac{x^\top Ax}{x^\top x}.$$

   (a) Let A be symmetric. Prove that $\nabla Q(x) = 0$ if and only if x is an eigenvector of A.

   (b) Let A be asymmetric. What are the vectors x at which $\nabla Q = 0$?

2. (**8 pts**) The goal of this exercise is to understand how one can compute a QR decomposition using *Householder reflections*.

   (a) Let u be a unit vector in $\mathbb{R}^n$, i.e., $\|u\|_2 = 1$. Let $P = I - 2uu^\top$. This matrix performs reflection with respect to the hyperplane orthogonal to the vector u. Show that $P = P^\top$ and $P^2 = I$.

   (b) Let $x \in \mathbb{R}^n$ be any vector, $x = [x_1, \dots, x_n]^\top$. Let u be defined as follows:

$$\tilde{u} := \begin{bmatrix} x_1 + \mathsf{sign}(x_1)\|x\|_2 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \equiv x + \mathsf{sign}(x_1)\|x\|_2 e_1, \quad u = \frac{\tilde{u}}{\|\tilde{u}\|_2}, \qquad (1)$$

   where $e_1 = [1, 0, \dots, 0]^\top$. The matrix with the vector u constructed according to (1) will be denoted $\mathsf{House}(x)$:

$$P = I - 2uu^\top \equiv I - 2\frac{\tilde{u}\tilde{u}^\top}{\tilde{u}^\top \tilde{u}} \equiv \mathsf{House}(x).$$

   Calculate Px.

   (c) Let A be an $m \times n$ matrix, $m \geq n$, with columns a_j, $j = 1 \dots, n$. Let $A_0 = A$. Let $P_1 = \mathsf{House}(a_1)$. Then $A_1 := P_1 A_0$ has the first column with the first entry nonzero and the other entries being zero. Next, we define P_2 as

$$P_2 = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{P}_2 \end{bmatrix}.$$

   where the matrix $\tilde{P}_2 = \mathsf{House}\left(A_1(2 : m, 2)\right)$. The notation $A_1(2 : m, 2)$ is Matlab's syntax indicating this is the vector formed by entries 2 through m of the 2nd column on A_1. Then we set $A_2 = P_2 A_1$. And so on.

This algorithm can be described as follows. Let $A_0 = A$. Then for $j = 1, 2, \ldots, n$ we set

$$P_j = \begin{bmatrix} I_{(j-1)\times(j-1)} & 0 \\ 0 & \tilde{P}_j \end{bmatrix}; \quad \tilde{P}_j = \mathsf{House}\left(A_{j-1}(j:m, j)\right), \quad A_j = P_j A_{j-1}.$$

Check that the resulting matrix A_n is upper triangular, its entries $(A_n)_{ij}$ are all zeros for $i > j$. Propose an `if`-statement in this algorithm that will guarantee that A_n has positive entries $(A_n)_{jj}$, $1 \le j \le n$.

(d) Extract the QR decomposition of A given the matrices P_j, $1 \le j \le n$, and A_n.

3. **(6 pts)** Prove items (1)–(6) of Theorem 3 on page 14 of `LinearAlgebra.pdf`.

4. **(4 pts)** Let A be an $m \times n$ matrix where $m < n$ and rows of A are linearly independent. Then the system of linear equations $Ax = b$ is underdetermined, i.e., infinitely many solutions. Among them, we want to find the one that has the minimum 2-norm. Check that the minimum 2-norm solution is given by

$$x^* = A^\top (AA^\top)^{-1} b.$$

Hint. One way to solve this problem is the following. Check that x^* is a solution to $Ax = b$. Show that is $x^* + y$ is also a solution of $Ax = b$ then $Ay = 0$. Then check that the 2-norm of $x^* + y$ is minimal if $y = 0$.

5. **(3 pts)** Let A be a 3×3 matrix, and let T be its Schur form, i.e., there is a unitary matrix Q (i.e., $Q^*Q = QQ^* = I$ where Q^* denotes the transpose and complex conjugate of Q) such that

$$A = QTQ^*, \quad \text{where} \quad T = \begin{bmatrix} \lambda_1 & t_{12} & t_{13} \\ 0 & \lambda_2 & t_{23} \\ 0 & 0 & \lambda_3 \end{bmatrix}.$$

Assume that λ_j, $j = 1, 2, 3$ are all distinct.

(a) Show that if v is an eigenvector of T then Qv is the eigenvector of A corresponding to the same eigenvalue.

(b) Find eigenvectors of T. *Hint: Check that $v_1 = [1, 0, 0]^\top$. Look for v_2 of the form $v_2 = [a, 1, 0]^\top$, and then for v_3 of the form $v_3 = [b, c, 1]^\top$, where a, b, c are to be expressed via the entries of the matrix T.*

(c) Write out eigenvectors of A in terms of the found eigenvectors of T and the columns of Q: $Q = [q_1, q_2, q_3]$.

Homework 4. Due Wednesday, Sept. 27.

1. (**5 pts**) Read Sections 4.1–4.3 on Ky-Fan norms and low-rank approximations based on SVD in `LinearAlgebra.pdf`. Prove the Eckart-Young-Mirsky theorem for any Ky-Fan norm.

   Theorem 1. *Let $A = U\Sigma V^\top$ be an SVD of A and M be any matrix of the size of A such that $\mathsf{rank}(M) \le k$. Then*

   $$\|A - M\| \ge \|A - U_k\Sigma_k V_k^\top\| \quad \textit{for any Ky-Fan norm} \quad \|\cdot\|,$$

   where U_k and V_k consist of the first k columns of U and V, respectively, and $\Sigma_k = \mathsf{diag}\{\sigma_1, \ldots, \sigma_k\}$.

   You can use Lemma 1 in Section 4.3 in `LinearAlgebra.pdf`.

2. (**5 pts**) Find an upper bound for the condition number for eigenvector r_j of a non-symmetric matrix A assuming that all its eigenvalues are distinct. In what case will this condition number be large?

3. Consider the Rayleigh Quotient Iteration, a very efficient algorithm for finding an eigenpair of a given matrix

```
Input:   $x_0 \neq 0$ is the initial guess for an eigenvector
$v = x_0/\|x_0\|$
for k = 0, 1, 2, ...
    $\mu_k = v^T A v$
    Solve $(A - \mu_k I)w = v$ for  $w$
    $v = w/\|w\|$.
end for
```

   Here is Matlab program implementing the Rayleigh Quotient Iteration for finding an eigenpair of a random $n \times n$ symmetric matrix starting from a random initial guess:

```
function RayleighQuotient()
n = 100;
A = rand(n);
A = A' + A;
v = rand(n,1);
v = v/norm(v);
k = 1;
mu(k) = v'*A*v;
tol = 1e-12;
I = eye(n);
```

```
res = abs(norm(A*v - mu(k)*v)/mu(k));
fprintf('k = %d: lam = %d\tres = %d\n',k,mu(k),res);
while res > tol
    w = (A - mu(k)*I)\v;
    k = k + 1;
    v = w/norm(w);
    mu(k) = v'*A*v;
    res = abs(norm(A*v - mu(k)*v)/mu(k));
    fprintf('k = %d: lam = %d\tres = %d\n',k,mu(k),res);
end
end
```

(a) **(2 pts)**Let A be a symmetric matrix with all distinct eigenvalues. Let μ be not an eigenvalue of A. Show that if (λ, v) is an eigenpair of A then $((\lambda-\mu)^{-1}, v)$ is an eigenpair of $(A-\mu I)^{-1}$.

(b) **(4 pts)**The Rayleigh Quotient iteration involves solving the system $(A-\mu_k I)w = v$ for w. The matrix $(A-\mu_k I)$ is closed to singular. Nevertheless, this problem is well-conditioned (in exact arithmetic). Explain this phenomenon. Proceed as follows. Without the loss of generality assume that v is an approximation for the eigenvector v_1 of A, and μ is an approximation to the corresponding eigenvalue λ_1. Let $\|v\| = 1$. Write v as

$$v = \left(1 - \sum_{i=2}^{n} \delta_i^2\right)^{1/2} v_1 + \sum_{i=2}^{n} \delta_i v_i,$$

where δ_i, $i = 2, \ldots, n$, are small. Show that the condition number $\kappa((A-\mu I)^{-1}, v)$ (see page 88 in [1]) is approximately $\left(1 - \sum_{i=2}^{n} \delta_i^2\right)^{-1/2}$ which is close to 1 provided that δ_i are small.

(c) **(4 pts)** It is known that the Rayleigh Quotient iteration converges cubically, which means that the error $e_k := |\lambda - \mu_k|$ decays with k so that the limit

$$\lim_{k\to\infty} \frac{e_{k+1}}{e_k^3} = C \in (0, \infty).$$

This means, that the number of correct digits in μ_k triples with each iteration. Try to check this fact experimentally and report your findings. Proceed as follows. Run the program. Treat the final μ_k as the exact eigenvalue. Define $e_j := |\mu_j - \mu_k|$ for $j = 1, \ldots, k-1$. Etc. Pick several values of n and make several runs for each n. Note that you might not observe the cubic rate of convergence due to too few iterations and floating point arithmetic.

References

[1] Bindel and Goodman, Principles of scientific computing

Homework 5. Due Wednesday, Oct. 4.

1. **(5 pts)**

   (a) Consider the set $\mathcal{L}$ of all $n \times n$ lower-triangular matrices with positive diagonal entries.

   i. Prove that the product of any two matrices in $\mathcal{L}$ is also in $\mathcal{L}$.

   ii. Prove that the inverse of any matrix in $\mathcal{L}$ is also in $\mathcal{L}$.

   This means that the set of all $n \times n$ lower-triangular matrices with positive diagonal entries forms a group with respect to matrix multiplication.

   (b) Prove that the Cholesky decomposition for any $n \times n$ symmetric positive definite matrix is unique. *Hint. Proceed from converse. Assume that there are two Cholesky decompositions $A = LL^\top$ and $A = MM^\top$. Show that then $M^{-1}LL^\top M^{-\top} = I$. Conclude that $M^{-1}L$ must be orthogonal. Then use item (a) of this problem to complete the argument.*

2. **(5 pts)** The Cholesky algorithm is the cheapest way to check if a symmetric matrix is positive definite.

   (a) Program the Cholesky algorithm. If any L_{jj} turns out to be either complex or zero, make it terminate with a message: "The matrix is not positive definite".

   (b) Generate a symmetric 100×100 matrix as follows: generate a matrix $\tilde{A}$ with entries being random numbers uniformly distributed in $(0, 1)$ and define $A := \tilde{A} + \tilde{A}^\top$. Use the Cholesky algorithm to check if A is symmetric positive definite. Compute the eigenvalues of A using a standard command (e.g. `eig` in MATLAB), find minimal eigenvalue, and check if the conclusion of your Cholesky-based test for positive definiteness is correct. If A is positive definite, compute its Cholesky factor using a standard command (e.g. see this help page for MATLAB) and print the norm of the difference o the Cholesky factors computed by your routine and by the standard one.

   (c) Repeat item (b) with A defined by $A = \tilde{A}^\top \tilde{A}$. The point of this task is to check that your Cholesky routine works correctly.

3. **(4 pts)** An $n \times n$ matrix is called *tridiagonal* if it is of the form

$$A = \begin{bmatrix} b_1 & c_1 & 0 & \dots & 0 \\ a_2 & b_2 & c_2 & & 0 \\ 0 & a_3 & b_3 & c_3 & \\ & & \ddots & \ddots & \ddots \\ 0 & \dots & 0 & a_n & b_n \end{bmatrix}.$$

There is a fast algorithm for solving linear systems $Ay = f$ with invertible and strictly diagonally dominant (i.e., $|b_i| > |a_i| + |c_i| \ \forall i$) tridiagonal matrices A. Sometimes it is referred to as the *Thomas algorithm*:

```
function TridiagSolver(a,b,c,f)
n = length(f);
v = zeros(n,1);
y = v;
w = b(1);
y(1) = f(1)/w;
for i=2:n
    v(i-1) = c(i-1)/w;
    w = b(i) - a(i)*v(i-1);
    y(i) = ( f(i) - a(i)*y(i-1) )/w;
end
for j=n-1:-1:1
   y(j) = y(j) - v(j)*y(j+1);
end
end
```

Calculate the number of flops for the Thomas algorithm.

4. **(4 pts)** Calculate (approximately) the number of flops for the modified Gram-Schmidt algorithm for computing the QR factorization of an $n \times n$ matrix A. Here is a vectorized Matlab code implementing the modified Gram-Schmidt.

```
A = rand(n);
Q = zeros(n); R = zeros(n);
for i = 1 : n
    Q(:,i) = A(:,i);
    for j = 1 : i-1
        R(j,i) = Q(:,j)'*Q(:,i);
        Q(:,i) = Q(:,i) - R(j,i)*Q(:,j);
    end
    R(i,i) = norm(Q(:,i));
    Q(:,i) = Q(:,i)/R(i,i);
end
```

Hint: The command `Q(:,j)'*Q(:,i)` *means* $\sum_{k=1}^{n} Q_{kj}Q_{ki}$.

The command `Q(:,i) = Q(:,i) - R(j,i)*Q(:,j)` *means the for-loop*

```
for k = 1 : n
Q(k,i) = Q(k,i) - R(j,i)*Q(k,j);
end
```

5. **(6 pts)**

   (a) Prove the cyclic property of the trace:

   $$\mathsf{trace}(ABC) = \mathsf{trace}(BCA) = \mathsf{trace}(CAB) \tag{1}$$

   for all A, B , C such that their product is defined and is a square matrix.

   (b) Prove that

   $$\|A\|_F^2 = \sum_{i=1}^{d} \sigma_i^2. \tag{2}$$

   Hint: use the full SVD of A and *the cyclic property of trace.*

   (c) Prove that

   $$\|A + B\|_F^2 = \|A\|_F^2 + \|B\|_F^2 + 2\langle A, B\rangle_F, \tag{3}$$

   where $\langle A, B\rangle_F$ is the Frobenius inner product. The Frobenius inner product is defined as

   $$\langle A, B\rangle_F := \sum_{i,j} a_{ij}b_{ij} = \mathsf{trace}(A^\top B) = \mathsf{trace}(B^\top A). \tag{4}$$

Homework 6. Due Wednesday, Oct. 11.

Dataset: An incomplete spreadsheet of movie ratings. Data file: `MovieRankingData.csv`. If you haven't done it, please feel free to manually add your own row there. Format: CSV (can be opened and edited e.g. using Numbers (Mac OS), Excel (Windows)).

Programming: Pick any language you wish. High-level language, e.g. Matlab or Python, is preferable. All requested algorithms should be programmed from scratch. Please use standard functions for SVD.

1. **(10 pts)** Do **matrix completion** in two ways:

   (a) Use the low-rank factorization model $A \approx XY^\top$ and the objective function of the form

   $$F(X,Y) = \frac{1}{2}\|P_\Omega(A - XY^\top)\|_F^2 + \frac{\lambda}{2}\left(\|X\|_F^2 + \|Y\|_F^2\right).$$

   Try values of λ 0.1, 1, and 10, and $\mathsf{rank}(X) = \mathsf{rank}(Y) = k$, $k = 1, 2, \ldots, 7$. Find X and Y using alternating iteration

   $$X^{m+1} = \arg\min_X F(X, Y^m), \tag{1}$$
   $$Y^{m+1} = \arg\min_Y F(X^{m+1}, Y). \tag{2}$$

   Each of these steps can be further decomposed into a collection of small linear least squares problems. For example, at each substep of (1), we solve the linear least squares problem to compute the row i of X:

   $$\mathbf{x}_i^\top = \arg\min_{\mathbf{x}} \frac{1}{2}\left\|\mathbf{x}^\top Y_{\Omega_i}^\top - a_{\Omega_i}\right\|_2^2 + \frac{\lambda}{2}\|\mathbf{x}\|^2, \tag{3}$$

   where $\Omega_i := \{j \mid (i,j) \in \Omega\}$, $Y_{\Omega_i}^\top$ is the set of columns of $Y^\top$ with indices in Ω_i, and a_{Ω_i} is the set of known entries of A in its row i. A similar problem can be set up for each column of Y. Work out solutions to these problems in a manner similar to the one in Sections 5.3.1 and 5.3.2 of `LinearAlgebra.pdf` except that there will be no constraint requiring the entries to be positive. Implement the resulting algorithm. Comment on how the value of λ and the choice of rank affects the result. Which values of the rank and λ seem the most reasonable to you? You can judge by your own row.

   (b) Use the approach of penalizing the nuclear norm in Section 6.3 of `LinearAlgebra.pdf` and the iteration

   $$M^{j+1} = S_\lambda(M^j + P_\Omega(A - M^j))$$

   Experiment with different values of λ.

Compare these two approaches for matrix completion. Which one gives more sensible results? Which one is easier to use? Which one do you find more efficient?

2. **(10 pts)** Extract a complete submatrix from the dataset `MovieRankingData.csv` by visual inspection with as many columns as you can find. This reduced dataset is needed for practicing algorithms for nonnegative matrix factorization (NMF).

   Compute the NMF $A \approx WH$ where W has k columns using

   (a) Projected gradient descend.

   (b) Lee-Seung scheme.

   Plot the Frobenius norm squared vs. iteration number for each solver. Which one do you find to be the most efficient?

Homework 7. Due Wednesday, Oct. 18.

Reference: [NW] J. Nocedal and S. Wright, "Numerical Optimization", Second Edition, Springer, 2006 (available online e.g. via UMD library).

1. **(5 pts)** Prove that for the sequence of iterates of the conjugate gradient algorithm, the preliminary version (Algorithm 5.1, page 108 in [NW]), the residuals are orthogonal, i.e.,
$$r_k^\top r_i = 0, \quad i = 0, 1, \ldots, k-1,$$
You can use facts proven in class:
$$\mathsf{span}\{p_0, \ldots, p_k\} = \mathsf{span}\{r_0, \ldots, r_k\} = \mathsf{span}\{r_0, Ar_0, \ldots, A^k r_0\} = \mathcal{K}(r_0; k),$$
$$p_k^\top A p_i = 0, \quad i = 0, 1, \ldots, k-1,$$
and Theorem 5.2 from [NW].

2. **(5 pts)** Prove that the conjugate gradient algorithm, the preliminary version (Algorithm 5.1, page 108 in [NW]), is equivalent to Algorithm 5.2 (CG) (page 112 in [NW]), i.e., that
$$\alpha_k = \frac{r_k^\top r_k}{p_k^\top A p_k}$$
and
$$\beta_{k+1} = \frac{r_{k+1}^\top r_{k+1}}{r_k^\top r_k}.$$

3. **(5 pts)** Let A be an $n \times n$ matrix. A subspace spanned by the columns of an $n \times k$ matrix B is an invariant subspace of A if A maps it into itself, i.e., if $AB \subset \mathsf{span}(B)$. This means that there is a $k \times k$ matrix C such that $AB = BC$.

   Prove that if a vector $r \in \mathbb{R}^n$ lies in the k-dimensional subspace spanned by the columns of B, i.e., if $r = By$ for some $y \in \mathbb{R}^k$ (r is a linear combination of columns of B with coefficients y_1, ..., y_k) then the Krylov subspaces generated by r spot expanding at degree $k-1$, i.e,
$$\mathsf{span}\{r, Ar, \ldots, A^p r\} = \mathsf{span}\{r, Ar, \ldots, A^{k-1} r\} \quad \forall p \geq k.$$

4. **(5 pts)** Prove Theorem 5.5 from [NW], page 115. Here are the steps that you need to work out.

   (a) Construct a polynomial $Q(\lambda)$ of degree $k+1$ with roots λ_n, λ_{n-1}, ..., λ_{n-k+1}, and $\frac{1}{2}(\lambda_1 + \lambda_{n-k})$ such that $Q(0) = 1$.

(b) Argue that $P(\lambda)$ defined as

$$P(\lambda) = \frac{Q(\lambda) - 1}{\lambda}$$

is a polynomial, not a rational function, by referring to the theorem about factoring polynomials. Cite that theorem.

(c) Use the ansatz

$$\|x_{k+1} - x^*\|_A^2 \le \min_{P \in \mathcal{P}_k} \max_{1 \le i \le n} [1 + \lambda_i P_k(\lambda_i)]^2 \|x_0 - x^*\|_A^2$$

Argue that

$$\|x_{k+1} - x^*\|_A^2 \le \max_{1 \le i \le n} Q^2(\lambda_i) \|x_0 - x^*\|_A^2.$$

(d) Show that

$$\max_{\lambda \in [\lambda_1, \lambda_{n-k}]} [Q(\lambda)]^2 \le \max_{\lambda \in [\lambda_1, \lambda_{n-k}]} \left| \frac{\lambda - \frac{1}{2}(\lambda_1 + \lambda_{n-k})}{\frac{1}{2}(\lambda_1 + \lambda_{n-k})} \right|^2.$$

(e) Find the maximum of the function in the right-hand side of the last equation in the interval $[\lambda_1, \lambda_{n-k}]$.

(f) Finish the proof of the theorem.

5. **(5 pts)** The goal of this problem is to practice the *conjugate gradient algorithm (CG) with and without preconditioning* on a meaningful problem. An explanation for its setup is a bit long, but it is a typical case that a lot of effort is spent on problem setup. I have done the setup for you. Your job will be just to code the CG algorithms.

Consider a maze with two exits, A, and B, shown in Fig. 1 taken from this paper by W. E and E. Vanden-Eijnden. This maze consists of a 20×20 array of cells, $N = 400$ cells in total. We will number the cells from 1 to 400 column-wise, i.e., the first column contains cells with indices from 1 to 20, the second one – from 21 to 40, and so on. Exit A is at cell 1 while Exit B is at cell 400.

Let A be the adjacency matrix for this maze. A is 400×400, $A_{ij} = 1$ if cells i and j are adjacent and there is no wall between them, and $A_{ij} = 0$ otherwise. A random walker makes a step from a cell to any adjacent cell not separated by a wall with equal probability. The stochastic matrix for this random walk can be found as $P = R^{-1}A$ where R is a diagonal matrix with row sums of A along its diagonal. Row sums of P are all equal to 1, and P_{ij} is the probability that the random walker located at cell i will next move to cell j.

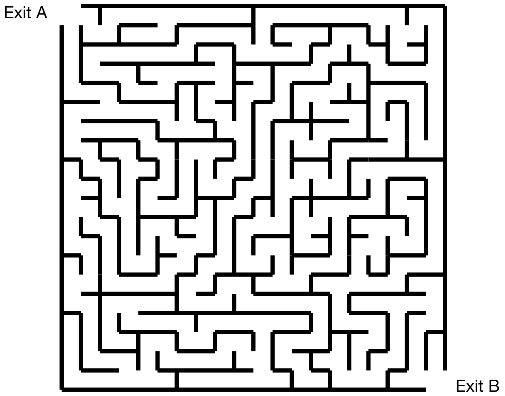


Figure 1: A maze with two exits.

Our goal is to compute the *committor* $x \in \mathbb{R}^N$ for this random walk, i.e., the vector of probabilities whose component x_i is the probability that the walker located at cell i will first arrive at exit B rather than A. This vector of probabilities satisfies:

$$x_1 = 0, \quad x_N = 1, \quad x_i = \sum_{j=1}^{N} P_{ij} x_j, \quad 2 \le i \le N-1. \tag{1}$$

Equation (1) is obtained from the following reasoning. The probability of exiting via B rather than A from cell i is equal to the sum of products of probabilities to move to a cell j from i and exit from j via B. The sum is over all other cells j.

Equation (1) can be written in the matrix form as follows. We denote $P - I$ by L. Then

$$L_{2:(N-1),2:(N-1)} x_{2:(N-1)} = b_{2:(N-1)}, \tag{2}$$

where b is obtained by $b = -Le_N$, where e_N is the vector with entries 1, ..., $N-1$ equal to zero and entry N equal to 1. You can check this, or just believe me.

Recall that $L = R^{-1}A - I$. It is not a symmetric matrix, but it can be symmetrized as follows. We will mark with tilde all submatrices $(2:(N-1), 2:(N-1))$ and all subvectors with indices in $2:(N-1)$.

$$\tilde{L}\tilde{x} = \tilde{b} \tag{3}$$

$$(\tilde{R}^{-1}\tilde{A} - \tilde{I})\tilde{x} = \tilde{b} \tag{4}$$

$$\tilde{R}^{1/2}(\tilde{R}^{-1}\tilde{A} - \tilde{I})\tilde{R}^{-1/2}\tilde{R}^{1/2}\tilde{x} = \tilde{R}^{1/2}\tilde{b} \tag{5}$$

$$(\tilde{R}^{-1/2}\tilde{A}\tilde{R}^{-1/2} - \tilde{I})\tilde{R}^{1/2}\tilde{x} = \tilde{R}^{1/2}\tilde{b} \tag{6}$$

The matrix $\tilde{R}^{-1/2}\tilde{A}\tilde{R}^{-1/2} - \tilde{I} =: L_{\mathsf{symm}}$ is symmetric. $\tilde{R}^{1/2}\tilde{x} =: y$, a new vector of unknowns. $\tilde{R}^{1/2}\tilde{b} = b_{\mathsf{symm}}$ is the new right-hand side. It is possible to check (just believe me), that the matrix $-L_{\mathsf{symm}}$ is symmetric positive definite.

Thus, here is the linear system with a symmetric positive definite matrix:

$$\boxed{-L_{\mathsf{symm}}y = -b_{\mathsf{symm}}, \quad x_{2:(N-1)} = \tilde{R}^{-1/2}y.} \tag{7}$$

The Matlab code `random_walk_in_maze.m` visualizes the maze, sets up this linear system, solves it using the built-in solver "\", visualizes the solution, and plots the eigenvalues of $-L_{\mathsf{symm}}$.

Task. Modify the code to solve Eq. (7) using the conjugate gradient algorithm without and with preconditioning (Algorithms 5.2 and 5.3) in [NW]. Use the incomplete Cholesky preconditioner. The corresponding Matlab command is

```
ichol_fac = ichol(sparse(A));
M = ichol_fac*ichol_fac';
```

Stop iterations when the residual will have a norm less than 10^{-12}. Plot the norm of the residuals after each iteration for the CG algorithm with and without preconditioning in the same figure. Use the logarithmic scale along the y-axis. Visualize the computed solution. Link your code to the pdf file with the homework.

Homework 7. Due Wednesday, Oct. 18.

Reference: [NW] J. Nocedal and S. Wright, "Numerical Optimization", Second Edition, Springer, 2006 (available online e.g. via UMD library).

1. **(5 pts)** Prove that for the sequence of iterates of the conjugate gradient algorithm, the preliminary version (Algorithm 5.1, page 108 in [NW]), the residuals are orthogonal, i.e.,

   $$r_k^\top r_i = 0, \quad i = 0, 1, \ldots, k-1,$$

   You can use facts proven in class:

   $$\mathsf{span}\{p_0, \ldots, p_k\} = \mathsf{span}\{r_0, \ldots, r_k\} = \mathsf{span}\{r_0, Ar_0, \ldots, A^k r_0\} = \mathcal{K}(r_0; k),$$

   $$p_k^\top A p_i = 0, \quad i = 0, 1, \ldots, k-1,$$

   and Theorem 5.2 from [NW].

2. **(5 pts)** Prove that the conjugate gradient algorithm, the preliminary version (Algorithm 5.1, page 108 in [NW]), is equivalent to Algorithm 5.2 (CG) (page 112 in [NW]), i.e., that

   $$\alpha_k = \frac{r_k^\top r_k}{p_k^\top A p_k}$$

   and

   $$\beta_{k+1} = \frac{r_{k+1}^\top r_{k+1}}{r_k^\top r_k}.$$

3. **(5 pts)** Let A be an $n \times n$ matrix. A subspace spanned by the columns of an $n \times k$ matrix B is an invariant subspace of A if A maps it into itself, i.e., if $AB \subset \mathsf{span}(B)$. This means that there is a $k \times k$ matrix C such that $AB = BC$.

   Prove that if a vector $r \in \mathbb{R}^n$ lies in the k-dimensional subspace spanned by the columns of B, i.e., if $r = By$ for some $y \in \mathbb{R}^k$ (r is a linear combination of columns of B with coefficients y_1, ..., y_k) then the Krylov subspaces generated by r spot expanding at degree $k-1$, i.e,

   $$\mathsf{span}\{r, Ar, \ldots, A^p r\} = \mathsf{span}\{r, Ar, \ldots, A^{k-1} r\} \quad \forall p \geq k.$$

4. **(5 pts)** Prove Theorem 5.5 from [NW], page 115. Here are the steps that you need to work out.

   (a) Construct a polynomial $Q(\lambda)$ of degree $k+1$ with roots λ_n, λ_{n-1}, ..., λ_{n-k+1}, and $\frac{1}{2}(\lambda_1 + \lambda_{n-k})$ such that $Q(0) = 1$.

Homework 8. Due Wednesday, Oct. 25.

The goal of this homework is to understand the Nested Dissection algorithm (George, 1973).

Reference: G. Martinsson, 10 lectures on fast direct solvers (2014). Lecture 6.

1. **(5 pts)** Suppose an invertible matrix A has a block form

$$A = \left[\begin{array}{c|c|c} A_{11} & & A_{13} \\ \hline & A_{22} & A_{23} \\ \hline A_{31} & A_{32} & A_{33} \end{array}\right]. \tag{1}$$

Assume that LU decompositions for A_{11} and A_{22} are available: $A_{11} = L_{11}U_{11}$, $A_{22} = L_{22}U_{22}$.

(a) Show that A can be factored as

$$A = \left[\begin{array}{c|c|c} L_{11} & & \\ \hline & L_{22} & \\ \hline A_{31}U_{11}^{-1} & A_{32}U_{22}^{-1} & I \end{array}\right] \left[\begin{array}{c|c|c} I & & \\ \hline & I & \\ \hline & & S_{33} \end{array}\right] \left[\begin{array}{c|c|c} U_{11} & & L_{11}^{-1}A_{13} \\ \hline & U_{22} & L_{22}^{-1}A_{23} \\ \hline & & I \end{array}\right], \tag{2}$$

where the matrix S_{33} is called the *Schur compliment*. Derive the formula for S_{33}.

(b) Suppose that the LU decomposition of S_{33} is found: $S_{33} = L_{33}U_{33}$. Write out the LU decomposition of A.

2. **(5 pts)** Modify the provided Matlab or Python code implementing the nested dissection algorithm to replace the LU factorizations with Cholesky factorizations. This modification will be specifically designed for symmetric positive definite matrices A. You can use a built-in function that computes Cholesky factorization.

Test it on the linear system from the problem with the maze from the previous homework. Save the symmetric positive definite linear matrix, the corresponding right-hand side, and the solution to it to a file and read this file in your new modified code. Paste your code to the pdf file with your homework. Report the norm of the difference between the solution computed by your code and the solution computed by a standard built-in linear solver.

3. **(5 pts)** Let the input matrix A be $n \times n$, symmetric positive definite. Estimate the number of flops in the resulting nested dissection with Cholesky factorizations. Do not count multiplications by permutation matrices as, if they were implemented in e.g. C, they would do only reindexing but involve no flops. Your answer should contain the exact coefficient next to the highest power of N. Terms with smaller powers of N can be incorporated in $O(\cdot)$.

Homework 9. Due Friday, Nov. 3.

Reference: [NW] J. Nocedal and S. Wright, "Numerical Optimization", Second Edition.

1. **(5 pts)** Suppose that a smooth function $f(x)$ is approximated by a quadratic model in the neighborhood of a current iterate x:

$$m(p) = f(x) + \nabla f(x)^\top p + \frac{1}{2} p^\top B p,$$

where B is a symmetric positive definite matrix. Show that then the direction p found by setting the gradient of $m(p)$ to zero is a descent direction for $f(x)$, i.e.,

$$\cos\theta := -\frac{\nabla f(x)^\top p}{\|\nabla f(x)\|\|p\|} > 0.$$

Also, bound $\cos\theta$ away from zero in terms of the condition number of B, i.e., $\kappa(B) = \|B\|\|B^{-1}\|$.

2. **(5 pts)** Let $f(x)$, $x \in \mathbb{R}^n$, be a smooth arbitrary function. The BFGS method is a quasi-Newton method with the Hessian approximate built recursively by

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \quad \text{where } s_k := x_{k+1} - x_k\,, \;\; y_k := \nabla f_{k+1} - \nabla f_k\,.$$

Let x_0 be the starting point and let the initial approximation for the Hessian is the identity matrix.

(a) Let p_k be a descent direction. Show that Wolfe's condition 2,

$$\nabla f_{k+1}^\top p_k \geq c_2 \nabla f_k^\top p_k, \quad c2 \in (0,1)$$

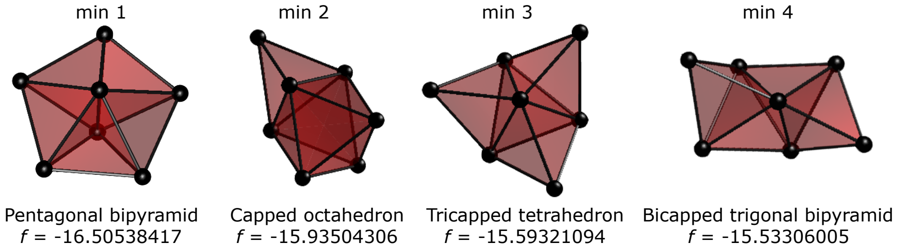
implies that $y_k^\top s_k > 0$.

(b) Let B_k be symmetric positive definite (SPD). Prove that then B_{k+1} is also SPD, i.e., for any $z \in \mathbb{R}^n \backslash \{0\}$, $z^\top B_{k+1} z > 0$. You can use the previous item of this problem and the Cauchy-Schwarz inequality for the B_k-inner product $(u,v)_{B_k} := v^\top B_k u$.

3. **(5 pts)** The goal of this problem is to code, test, and compare various optimization techniques on the problem of finding local minima of the potential energy function of the cluster of 7 atoms interacting according to the Lennard-Jones pair potential (for brevity, this cluster is denoted by LJ$_7$):

$$f = 4\sum_{i=2}^{7}\sum_{j=1}^{i}\left(r_{ij}^{-12} - r_{ij}^{-6}\right), \quad r_{ij} := \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}. \quad (1)$$

It is known that LJ_7 has four local energy minima:



min 1 min 2 min 3 min 4

Pentagonal bipyramid
f = -16.50538417

Capped octahedron
f = -15.93504306

Tricapped tetrahedron
f = -15.59321094

Bicapped trigonal bipyramid
f = -15.53306005

Add the following search directions to the provided Matlab code `LJ_line_search` (or to the Python code that I will provide soon):

- BFGS ([NW], page 24),
- (FRCG) Fletcher-Reeves nonlinear CG ([NW], page 121),
- (PRCG) Polak-Ribiere nonlinear CG ([NW], page 122, eq. (5.45)).

Note that it is recommended to reset the matrix B_k in the BFGS method to identity every m steps. Try $m = 5$ and $m = 20$

Compare the performance of the five algorithms, the three algorithms above, steepest descent, and Newton's (already encoded) in terms of the number of iterations required to achieve convergence and by plotting the graph of f and $\|\nabla f\|$ agaist the iteration number for each test case. Do it for each of the four initial conditions approximating the four local minima and ten random initial conditions.

4. **(5 pts)** (Approx. Problem 3.1 from [NW])

(a) Compute the gradient and the Hessian of the Rosenbrock function

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2. \tag{2}$$

Show that $(1, 1)$ is the only local minimizer, and that the Hessian is positive definite at it.

(b) Program the steepest descent, FRCG, PRCG, Newton's, and BFGS algorithms using the backtracking line search. Use them to minimize the Rosenbrock function (2). First start with the initial guess $(1.2, 1.2)$ and then with the more difficult one $(-1.2, 1)$. Set the initial step length $\alpha_0 = 1$ and plot the step length α_k versus k for each of the methods.

Plot the level sets of the Rosenbrock function using the command `contour` and plot the iterations for each method over it.

Plot $\|(x_k, y_k) - (x^*, y^*)\|$ versus k in the logarithmic scale along the y-axis for each method. Do you observe a superlinear convergence? Compare the performance of the methods.

Homework 10. Due Friday, Nov. 10.

Reference: [NW] J. Nocedal and S. Wright, "Numerical Optimization", Second Edition.

1. **(5 pts)**

   (a) Let $\langle \cdot, \cdot \rangle$ be an inner product defined on a vector space V. Prove the Cauchy-Schwarz inequality

   $$|\langle u, v\rangle|^2 \le \langle u, u\rangle\langle v, v\rangle \quad \forall u, v \in V. \tag{1}$$

   Hint: Consider the quadratic polynomial

   $$p(t) = \langle u + tv, u + tv\rangle, \quad t \in \mathbb{R} \text{ (or } \mathbb{C}). \tag{2}$$

   Can this polynomial take negative values? Use your answer to conclude what should be the sign of the discriminant if you set $p = 0$.

   (b) Let B be a real symmetric positive definite $n \times n$ matrix. Use the Cauchy-Schwarz inequality to prove that

   $$(g^\top Bg)(g^\top B^{-1}g) \ge (g^\top g)^2 \quad \forall g \in \mathbb{R}^n. \tag{3}$$

2. **(5 pts)** Consider Newton's algorithm for solving the trust-region subproblem ([NW], Algorithm 4.3, page 87). Prove that Eq. (4.43) is equivalent to Eq. (4.44) in [NW], i.e., that for

   $$\phi(\lambda) = \frac{1}{\Delta} - \left[\sum_{j=1}^{n} \frac{(q_j g)^2}{(\lambda_j + \lambda)^2}\right]^{-1/2},$$

   where (q_j, λ_j) are the eigenpairs of B, the Newton iteration

   $$\lambda^{(l+1)} = \lambda^{(l)} - \frac{\phi(\lambda^{(l)})}{\phi'(\lambda^{(l)})}$$

   is given by

   $$\lambda^{(l+1)} = \lambda^{(l)} + \left(\frac{\|p_l\|}{\|z_l\|}\right)^2 \frac{\|p_l\| - \Delta}{\Delta},$$

   where $z_l = L^{-1}p_l$, $p_l = -(B + \lambda^{(l)}I)^{-1}g$, and L is the Cholesky factor of $B + \lambda^{(l)}I$, i.e., $B + \lambda^{(l)} = LL^\top$. Note: $R = L^\top$ in Algorithm 4.3.

   Hint: You will need to compute the derivative of ϕ and express it in terms of $\|p_l\|$ and $\|(B + \lambda^{(l)}I)^{-1}g\|^2$. Also, you will need to use the fact that the Cholesky factor of any SPD matrix M is related to $M^{1/2}$ via an orthogonal transformation.

3. **(5 pts)** Consider the problem of finding local energy minima of the LJ_7 as in Problem 3 of HW9. Consider the same set of initial conditions: four initial conditions close to its four local minima, and ten random initial conditions.

   Implement the BFGS trust-region method with the dogleg subproblem solver. Compare its performance with the trust-region Newton with the exact subproblem solver implemented in the provided code by creating a table with the number of iterations required to achieve convergence and plotting the graph of f and $\|\nabla f\|$ against the iteration number for each test case (the four initial conditions close to the minima and one representative random configuration initial condition). Do it for each of the four initial conditions approximating the four local minima and ten random initial conditions. The set of figures to include is the same as for Problem 3 in HW9.

   Comment on the performance of trust-region methods compared to the performance of line-search methods.

4. **(5 pts)** (Approx. Problem 3.1 from [NW]) Write a code that applies the two algorithms from the previous problem (the trust-region BFGS with the dogleg solver and the trust-region Newton with the exact subspace solver) to the Rosenbrock function as in Problem 4 of HW9:

   $$f(x,y) = 100(y - x^2)^2 + (1 - x)^2. \tag{4}$$

   Experiment with the same two initial conditions: $(1.2, 1.2)$ and $(-1.2, 1)$.

   Plot the level sets of the Rosenbrock function using the command `contour` and plot the iterations for each method over it. Plot $\|(x_k, y_k) - (x^*, y^*)\|$ versus k in the logarithmic scale along the y-axis for each method. Compare the performance of the methods.

AMSC660/CMSC660 Maria Cameron

Homework 11. Due Monday, Nov. 20

(20 pts)
The goals of this project are

- to acquire practice in working with real data;
- to explore various optimization methods for solving classification problems and understand how their performance is affected by their settings.

What to submit. Please submit a report with figures and comments. Link your codes to the report pdf. These can be e.g. Dropbox links or GitHub links, etc. All optimizers should be coded from scratch.

Programming language You can use any suitable programming language. Matlab or Python are preferable. I provide an auxiliary package in Matlab, but it is easy to rewrite whatever you need from it in Python. The content of this package is described throughout the rest of this problem description.

- If you are going to use Matlab, you can use `mnist.mat` as input. Note that 4-pixel paddings are removed in `minst.mat`.
- If you are using Python, you can read the binary data files (the link is below) directly in Python. In this case, you can but do not have to remove the 4-pixel padding. Or read `mnist.mat` in Python. Or, even simpler, save the data postprocessed with SVD in Matab and then read them in Python.

1 MNIST dataset

You will experiment with the MNIST dataset of handwritten digits 0, 1, ..., 9 available at http://yann.lecun.com/exdb/mnist/. The training set has 60000 28 x 28 grayscale images of handwritten digits (10 classes) and a testing set has 10000 images.

The data files are in binary format. The code `readMNIST.m` written by Siddharth Hegde and slightly modified by me reads these binary files and strips 4-pixel paddings from the images. The code `saveMNIST2mat.m` saves the resulting data to a mat file `mnist.mat`. The file `mnist.mat` and all codes I am mentioning are packaged to `MNISTaux.zip`.

2 Classification problem

The task is to select all images with digits 1 and all images with digits 7 from the training set, find a dividing surface that separates them, and test this dividing surface on the 1's and 7's from the test set. A sample of 1's and 7's from the training set is shown in Fig. 1.

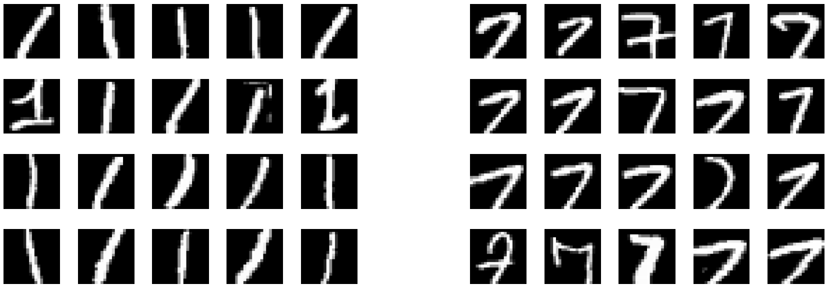


Figure 1: Samples of 20-by-20 images of 1's (left) and 7's (right) from MNIST.

Each image is a point in $\mathbb{R}^{400}$ (the images with stripped paddings are 20-by-20). It is convenient to reduce dimensionality of data by using SVD and mapping the set to $\mathbb{R}^d$ where $d \ll 400$, e.g. $d = 3$, $d = 10$, $d = 20$ – see `mnist_2categories_hyperplane.m`. We label all images with "1" by 1 and all images with "7" by -1. The training data set `Xtrain` (or `X` for brevity) is `Ntrain`-by-`d` matrix. The vector of labels `y` is `Ntrain`-by-`1`.

We pose three kinds of unconstrained optimization problems.

2.1 A smooth loss function for the optimal hyperplane with Tikhonov regularization

In the simplest setting, we aim at finding a dividing hyperplane $w^\top x + b = 0$ with that $w^\top x_j + b > 0$ for all (almost all) x_j corresponding to 1 (labelled with $y_j = 1$) and $w^\top x_j + b < 0$ for all (almost all) x_j corresponding to 7 (labelled with $y_j = -1$). Hence, x_j is classified correctly if

$$\mathsf{sign}(y_j(w^\top x_j + b)) = 1.$$

Instead of the discontinuous sign function, we use a smooth sigmoid-type function (we call it *residual*)

$$r_j \equiv r(x_j; \{w, b\}) := \log\left(1 + e^{-y_j(w^\top x_j + b)}\right) \tag{1}$$

that is close to zero if $y_j(w^\top x_j + b) > 0$ and grows linearly in the negative range of the aggregate $y_j(w^\top x_j + b)$. For brevity, we will denote the d+1-dimensional vector of parameters $\{w, b\}$ by $\mathbf{w}$. We form the loss function by averaging up the residuals and adding a Tikhonov regularization term:

$$f(\mathbf{w}) = \frac{1}{n}\sum_{j=1}^{n} \log\left(1 + e^{-y_j(w^\top x_j + b)}\right) + \frac{\lambda}{2}\|\mathbf{w}\|^2. \tag{2}$$

Here n is the number of data points and λ is a parameter for the Tikhonov regularization. This loss function and its derivatives are encoded in functions `fun0` and `gfun0` at the bottom of the code `mnist_2categories_hyperplane.m`. The optimization problem in `mnist_2categories_hyperplane.m` is solved using the stochastic inexact Newton method. The approximations for Newton's directions are found by the conjugate gradient method. A line search algorithm is used along each proposed direction. If you set `nPCA = 3` in line 5, i.e., $d = 3$, then the dividing hyperplane is visualized.

2.2 A smooth loss function for the optimal quadratic hypersurface with Tikhonov regularization

As you will see, a quadratic dividing hypersurface may lead to much fewer misclassified digits. We are seeking a quadratic hypersurface of the form:

$$x^\top W x + v^\top x + b.$$

Hence, the quadratic test function is

$$q(x_j; \mathbf{w}) := y_j \left(x^\top W x + v^\top x + b \right). \tag{3}$$

The loss function is defined in a similar manner:

$$f(\mathbf{w}) = \frac{1}{n} \sum_{j=1}^{n} \log\left(1 + e^{-q(x_j;\mathbf{w})}\right) + \frac{\lambda}{2}\|\mathbf{w}\|^2. \tag{4}$$

Here $\mathbf{w}$ denotes the $d^2 + d + 1$-dimensional vector of coefficients of $\{W, v, b\}$. This loss function and its gradient are available in the file `qloss.m`.

2.3 A nonlinear least squares problem for the optimal quadratic hypersurface

Finally, we can design the loss function to fit the framework of the nonlinear least squares problem:

$$f(\mathbf{w}) = \frac{1}{2} \sum_{j=1}^{n} [r_j(\mathbf{w})]^2, \quad r_j(\mathbf{w}) = \log\left(1 + e^{-q(x_j;\mathbf{w})}\right). \tag{5}$$

The vector of the residuals and the Jacobian matrix are available in the file `Res_and_Jac.m`.

3 The research tasks

3.1 Levenberg-Marquardt

Set the number of PCAs $d = 20$. Find the optimal quadratic dividing surface. With this number of PCAs and the quadratic surface, you should be able to achieve good accuracy

(the ratio of correctly classified test data to the total number of test data is about 99%). Implement the Levenberg-Marquardt algorithm. A driver for it is
`mnist_2categories_quadratic_NLLS.m`.
It calls (line 94)

```
[w,f,gnorm] = LevenbergMarquardt(r_and_J,w,kmax,tol);
```

You need to code the function `LevenbergMarquardt` yourself. To avoid problems with inverting the matrix $J^\top J$, regularize it by changing it to

$$J^\top J + I \cdot 10^{-6}.$$

3.2 Stochastic optimizers

Implement the following stochastic optimizers.

1. Stochastic gradient descent (experiment with various batch sizes and stepsize decreasing strategies.

2. Stochastic Nesterov (experiment with various batch sizes). Its deterministic version is given by Eqs. (61)–(62) in `Optimization.pdf`.

3. Stochastic Adam (experiment with various batch sizes). Its deterministic version is proposed in a paper by D. P. Kingma and J. L. Ba "Adam: A Method for Stochastic Optimization" where ADAM is introduced: https://arxiv.org/pdf/1412.6980.pdf.

Use these optimizers to minimize the loss function (4) and find the optimal dividing quadratic hypersurface. For each solver, experiment with the appropriate settings. Compare the performance of these optimizers to each other.

Run the stochastic optimizers for the same number of epochs (if you have n data points and your batch size is m then round(n/m) timesteps is one epoch). Which stochastic optimizer do you find the most efficient?

Include a detailed discussion on the performance of these solvers in various settings in your report. Supplement your report with plots of the estimates for the loss function and the norm of its gradient. Include tables, if appropriate.

AMSC660/CMSC660 Maria Cameron

Homework 12. Due Friday, December 1

1. **(5 pts)**

   Consider the KKT system

$$\begin{bmatrix} G & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} -\mathbf{p} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix}. \qquad (1)$$

   where G is $d \times d$ symmetric positive definite and A is $m \times d$ and has linearly independent rows. Show that the matrix

$$K := \begin{bmatrix} G & A^\top \\ A & 0 \end{bmatrix}$$

   is of *saddle-point type*, i.e., it has d positive eigenvalues and m negative ones. *Hint: Find matrices X and S (S is called the* ***Schur compliment****) such that*

$$\begin{bmatrix} G & A^\top \\ A & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ X & I \end{bmatrix} \begin{bmatrix} G & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & X^\top \\ 0 & I \end{bmatrix}.$$

   Then use Sylvester's Law of Inertia (look it up!) to finish the proof.

2. **(5 pts)** Consider an equality-constrained quadratic program QP

$$\frac{1}{2}\mathbf{x}^\top G\mathbf{x} + \mathbf{c}^\top \mathbf{x} \;\rightarrow\; \min \quad \text{subject to} \qquad (2)$$

$$A\mathbf{x} = \mathbf{b}. \qquad (3)$$

   The matrix G is symmetric. Assume that A is full rank (i.e., its rows are linearly independent) and $Z^\top GZ$ is positive definite where Z is a basis for the null-space of A, i.e., $AZ = 0$.

   (a) Write the KKT system for this case in the matrix form.

   (b) Show that the matrix of this system K is invertible. *Hint: assume that there is a vector $\mathbf{z} := (\mathbf{x}, \mathbf{y})^\top$ such that $K\mathbf{z} = 0$. Consider the quadratic form $\mathbf{z}^\top K\mathbf{z}$, use logical reasoning and algebra, and arrive at the conclusion that then $\mathbf{z} = 0$.*

   (c) Conclude that there exists a unique vector $(\mathbf{x}^*, \boldsymbol{\lambda}^*)^\top$ that solves the KKT system. Note that since we have only equality constraints, the positivity of $\boldsymbol{\lambda}$ is irrelevant.

3. **(5 pts)** Consider the following quadratic program with inequality constraints:

$$\begin{aligned} f(x,y) &= (x-1)^2 + (y-2.5)^2 \;\to\; \min \quad \text{subject to} && (4)\\ [1] \quad & x - 2y + 2 \ge 0 && (5)\\ [2] \quad & -x - 2y + 6 \ge 0 && (6)\\ [3] \quad & -x + 2y + 2 \ge 0 && (7)\\ [4] \quad & x \ge 0 && (8)\\ [5] \quad & y \ge 0 && (9) \end{aligned}$$

(a) Plot level sets of the objective function and the feasible set.

(b) What is the exact solution to (4)–(9)? Find it analytically with the help of your figure.

(c) Suppose the initial point is $(2,0)$. Initially, constraints 3 and 5 are active, hence start with $\mathcal{W} = \{3,5\}$. Work out all iterations of the active-set method analytically. The arising linear systems should be very easy to solve. For each iteration, you need to write out the set $\mathcal{W}$, the KKT system, its solution, i.e., (p_x, p_y), the vector of Lagrange multipliers, and the current iterate (x_k, y_k). Plot all iterates on your figure. There should be a total of 5 iterations.

AMSC660/CMSC660 Maria Cameron

Homework 13. Due Friday, December 8.

1. **(10 pts)** The invariant probability density for the system evolving in the double-well potential $V(x) = x^4 - 2x^2 + 1$ according to the overdamped Langevin dynamics[1] at temperature $\beta^{-1} = 1$ is given by the Gibbs pdf

$$f(x) = \frac{1}{Z} e^{-(x^4 - 2x^2 + 1)}, \quad \text{where} \quad Z = \int_{-\infty}^{\infty} e^{-(x^4 - 2x^2 + 1)} dx. \tag{1}$$

   (a) Use the composite trapezoidal rule to find the normalization constant Z. Pick an interval of integration $[-a, a]$ where a is large enough so that $e^{-(a^4 - 2a^2 + 1)} < 10^{-16}$.

   (b) Find the optimal value of σ in order to use the pdf of the form

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/(2\sigma^2)}$$

   for sampling RV with pdf $f(x)$ (Eq. (1)) by means of the acceptance-rejection method. The optimal σ minimizes the constant c.

   Hint: First find analytically

$$x^* = \arg\max_{x \in \mathbb{R}} \frac{f(x)}{g_\sigma(x)}$$

   as a function of σ. Then you can find the optimal σ using e.g. the function `fminbnd` *in MATLAB. If you use a programming language that does not have standard function to find a minimum of a function in 1D, plot a graph $c(\sigma)$ and pick σ close to the optimal one.*

   (c) Sample RV η with pdf $f(x)$ (Eq. (1)) using the acceptance-rejection method. Check that the ratio of the total number of samples and the number of accepted samples is close to C. Plot a properly scaled histogram for the obtained samples and compare it with the exact distribution (with Z found numerically). An example of generating such a histogram is given in the code in Section 3.3 in `MonteCarloAMSC660.pdf`.

   Hint: to generate samples of $\mathcal{N}(0, \sigma^2)$, generate samples from $\mathcal{N}(0, 1)$ and multiply them by σ.

   (d) Find $E[|x|]$ for the pdf $f(x)$ using the Monte Carlo integration.

Submit a single pdf document. *Link your codes to it, or print them to pdfs and append them to the main pdf.*

[1] The overdamped Langenin stochastic differential equation is $dX = -\nabla V(X)dt + \sqrt{2\beta^{-1}}dW$ where dW is the increment of the standard Brownian motion.

2. **(10 pts)**

The unit cube in $\mathbb{R}^d$ centered at the origin is the set

$$C^d = \left\{ \mathbf{x} \in \mathbb{R}^d \mid \max_{1 \le i \le d} |x_i| \le \tfrac{1}{2} \right\},$$

while the unit ball in $\mathbb{R}^d$ centered at the origin is the set

$$B^d = \left\{ \mathbf{x} \in \mathbb{R}^d \mid \sum_{i=1}^{d} x_i^2 \le 1 \right\}.$$

Obviously, all centers of the $(d-1)$-dimensional faces of C^d, i.e., the points with one coordinate $\pm\frac{1}{2}$ and the rest zeros, lie inside B^d. The most remote points of C^d from the origin are the corners with all coordinates $\pm\frac{1}{2}$. The distance of the corner of C^d from the origin is $\sqrt{d}/2$. For $d \ge 5$, the corners of C^d and some their neighborhoods lie outside B^d. The d-dimensional volume of C^d is 1, while the volume of the d-dimensional unit ball B^d tends to zero as $d \to \infty$:

$$\mathsf{Vol}(C^d) = 1, \quad \mathsf{Vol}(B^d) = \frac{\pi^{d/2}}{\frac{d}{2}\Gamma\left(\frac{d}{2}\right)} \to 0 \quad \text{as} \quad d \to \infty.$$

Therefore, the fraction of the unit cube C^d lying inside B^d also tends to zero as $d \to \infty$. You can read about this phenomenon in [1].

Task. Calculate $\mathsf{Vol}(B^d \cap C^d)$ in $d = 5, 10, 15, 20$ using Monte Carlo integration in two ways.

(a) Use a sequence of independent uniformly distributed random variables in the unit cube C^d.

(b) Use a sequence of independent uniformly distributed random variables in the unit ball B^d. (You need to think of a way to generate such a random variable.)

References

[1] High-Dimensional Space, notes by Venkatesan Guruswami (Professor, Computer Science Dept, Carnegie Mellon University)