

CSE Ph.D. Qualifying Exam, Fall 2008

You should choose two areas to work on. Each area consists of four problems, and you can choose three of them. Show all your work and write in a readable way.

1 Data Analysis

1. We are given an item space X (the exact nature of the item is irrelevant for this problem). For an item $x \in X$ is also associated a numerical grade y , i.e., a real number. For simplicity, we assume $y \in Y$ where Y is a finite subset of the real numbers. Let $P(x, y)$ be a probability distribution over $X \times Y$. For a fixed n , and a set of items $X_n = \{x_1, \dots, x_n\}$ with the corresponding numerical grades y_1, \dots, y_n , a ranking or an ordering of X_n is simply a permutation $\sigma = (\sigma(1), \dots, \sigma(n))$ of $(1, \dots, n)$. For a given set of real numbers $c_1 > \dots > c_n$, we use the following score to measure the quality of the ranking σ ,

$$s(\sigma; \{(x_i, y_i)\}) = c_1 y_{\sigma(1)} + \dots + c_n y_{\sigma(n)}.$$

- (a) [25 %] Show a ranking σ maximizes $s(\sigma)$ if

$$y_{\sigma(1)} > \dots > y_{\sigma(n)},$$

i.e., we should order the items in X_n by the decreasing order of the y_i 's.

- (b) [25 %] Now assume $(x_i, y_i), i = 1, \dots, n$ are iid from $P(x, y)$, show that

$$p(y_i | x_1, \dots, x_n) = p(y_i | x_i),$$

i.e., the conditional probability of y_i given x_1, \dots, x_n is the same as the conditional probability of y_i given x_i .

- (c) [50 %] A ranking function R is a mapping from X to the set of real numbers. We denote the ranking induced by R as σ_R , where σ_R is obtained by the decreasing order of $R(x_i), i = 1, \dots, n$. For a ranking function R , define the expected score as

$$\mathcal{E}s(\sigma_R; \{(x_i, y_i)\})$$

where the expectation \mathcal{E} is with respect to the product probability, i.e., $(x_i, y_i), i = 1, \dots, n$ are iid from $P(x, y)$. Show that the following ranking function

$$R^*(x) = \sum_y y P(y|x)$$

maximizes the expected score.

2. From the above Problem, we know R^* maximizes the expected score. We want to investigate whether the ranking (ordering) will change if we instead use

$$R_f^*(x) = \sum_y f(y)P(y|x)$$

as the ranking function, where f is a strictly monotonically increasing function.

- (a) [25 %] Show if f is linear, using $R^*(x)$ and $R_f^*(x)$ as the ranking functions produces the same ranking (ordering).
- (b) [50 %] Show if y can take exactly two distinct values, for an arbitrary f which is strictly monotonically increasing, using $R^*(x)$ and $R_f^*(x)$ as the ranking functions produces the same ranking (ordering).
- (c) [25 %] Show (b) is not true if y can take more than two distinct values.
3. Suppose you only know how to make density estimation methods. How can you phrase classification in terms of density estimation? How can you phrase regression in terms of density estimation? How can you phrase clustering in terms of density estimation? How can you phrase dimension reduction in terms of density estimation?
4. Let $X = (X_1, \dots, X_m), X_i \in \{0, 1\}$ be a binary random vector with the following probability function

$$p_\theta(X) = \exp \left(\sum_{i < j} X_i X_j \theta_{ij} - \log \psi(\theta) \right)$$

where $\psi(\theta)$ ensures normalization i.e. $\sum_X p_\theta(X) = 1$. Given several iid samples of the vector $X: X^{(1)}, \dots, X^{(N)}$ (where $X^{(j)}$ is a vector drawn iid from the distribution above) we can maximize the loglikelihood $\ell(\theta) = \sum_{j=1}^N \log p_\theta(X^{(j)})$ to obtain an estimate for θ . Unfortunately, the mle does not have a closed form and iterative optimization methods such as gradient descent or Newton's method are typically used.

- (a) [25 %] Express the gradient $\nabla \ell(\theta)$ in a simple way using expectations over $p_\theta(X)$ and over the empirical distribution (relative frequency of X in data)

$$\tilde{p}(X) = \frac{1}{N} \sum_{i=1}^N 1_{\{X=X_i\}}.$$

- (b) [35 %] What is the complexity of computing $\ell(\theta)$ and $\nabla \ell(\theta)$? Express your answer in terms of the dimensionality m and the number of samples N .
- (c) [40 %] An alternative estimator is the maximum pseudo likelihood which maximizes the pseudo loglikelihood function

$$p\ell(\theta) = \sum_{k=1}^N \sum_{i=1}^m p_\theta(X_i^{(N)} | \{X_j^{(N)} : j \neq i\}).$$

What is the complexity of computing $p\ell(\theta)$ and its gradient $\nabla p\ell(\theta)$? What are the situations in which computing $p\ell(\theta)$ is faster than computing $\ell(\theta)$?

2 High Performance Computing

1. **I/O complexity–lower bounds:** Consider the computation of the *outer product* operation, $C \leftarrow x \cdot y^T$, where x and y are column vectors of length n . Suppose you wish to implement this operation on a sequential machine with a simple memory hierarchy consisting of (a) an infinite-capacity but slow main memory, and (b) a small and fast fully-associative cache of size Z words.
 - (a) [60 %] Derive an asymptotic lower bound on the number of transfers between slow and fast memory for the outer product, as a function of n and Z .
 - (b) [40 %] Give an algorithm that attains this lower bound, and show that it does so.
2. **Designing a bus-based shared memory machine:** Dell is asking for your advice on the maximum number p of single-core processors they should offer for their new bus-based shared memory server. In particular, they want to choose p so that running p independent copies of the STREAM Triad benchmark just saturates the bus when running 1 independent copy on each processor.

Dell plans to use a single-core processor, configured as follows.

- The clock speed is 3 GHz.
- The processor can commit a maximum of 4 instructions per cycle, where up to 2 of the instructions are floating-point operations and up to 2 can be memory operations.
- The processor does *not* have a fused multiply-add instruction, so multiplies and adds are separate instructions.
- There is only 1 level of cache, with a line size of 128 bytes. The data and instruction caches are separate.

The server architecture is shown in Figure 1. The maximum bandwidth of the bus is 20 GB/s.

The pseudocode for the STREAM Triad benchmark is as follows:

```
do i = 1, n
  C[i] = q*A[i] + B[i]
enddo
```

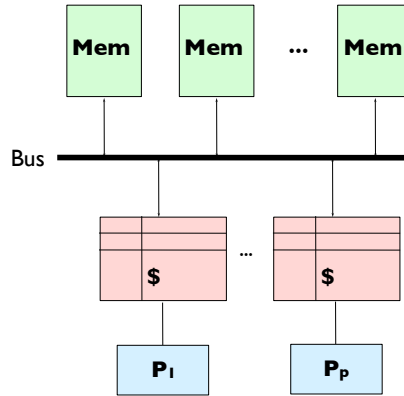


Figure 1: The architecture of Dell’s proposed bus-based shared memory server.

where A , B , and C are arrays of length n double-precision values, and q is a double-precision scalar variable.

Please answer the following.

- (a) [30 %] Estimate the data cache miss rate for STREAM Triad, assuming n is large and there is no hardware or software prefetching.
 - (b) [70 %] Suppose that the benchmark code is pre-loaded into all instruction caches but that the data caches are empty. We then run p independent copies of STREAM Triad, one on each of the p processors. What value of p saturates the bus?
3. **Smart disks for the N-body problem:** A new hardware vendor is planning to build a new type of “smart disk processor,” consisting of a conventional disk drive with a reconfigurable processor directly attached. The idea is that a user could perform a custom computation on the reconfigurable processor right as the data is coming off of the disk, before passing results along to the host processor. The company’s salesperson tells you that the latency to read data from the disk is 1 ms, and the read bandwidth is 250 MB/s.

Furthermore, the salesperson tells you that this system would be great for your N-body calculation, and only costs 50% more than a comparable workstation with the same disk but with a general purpose CPU.

Your data, which resides on disk, consists of a set of points, each represented by 4 single-precision floating-point values: 1 for the point’s mass and 3 more for its (x, y, z) coordinates. You wish to perform a direct N-body computation among subsets of your points, which we’ll suppose takes $\kappa \cdot n^2$ flops for n points. The κ is some known constant that depends on the force. (For this problem, ignore the symmetry of the force computation among points.)

- (a) [30 %] Suppose we decide that as we stream points off of the disk, we'll use the custom processor to perform the direct computation on the points in flight. How many such points will there be?
- (b) [30 %] To sustain the same 250 MB/s from the custom processor to the host processor, what does the flop rate of the custom processor need to be (in terms of κ)?
- (c) [40 %] Let's say κ for your force is 50 (flops per pairwise force computation). Your boss wants to know whether it makes sense to buy this system. What advice would you give and why?
4. **Parallel matrix transpose:** In this problem, you will propose and analyze parallel algorithms for the matrix transpose operation, $B \leftarrow A^T$, *i.e.*, $b_{ij} \leftarrow a_{ji}$ for all entries a_{ji} of A .
- (a) [35 %] Suggest 3 schemes for partitioning the two matrices among processes, and discuss the trade-offs among these approaches. Does it matter whether you consider a shared address space or message-passing machine? Why or why not?
- (b) [35 %] Write pseudocode for parallel matrix transpose on a shared address space machine, and then again for a message passing machine. Besides inherent communication and load balance, what other major performance issues do you consider in each case and how do you address them?
- (c) [40 %] Is there any benefit to blocking the parallel matrix transpose? Under what conditions? How would you block it (just describe, no pseudocode needed)? What, if anything, is different between blocking for matrix transpose vs. blocking for dense matrix multiply or dense Gaussian elimination?

3 Numerical Analysis

1. Let

$$A = \begin{pmatrix} a^T \\ \hat{A} \end{pmatrix} \in R^{m \times n}, \quad m > n.$$

Then we have

$$A^T A = aa^T + \hat{A}^T \hat{A}.$$

In rank-one downdating problem of Cholesky decomposition, we are to find the Cholesky factor \hat{R} for $\hat{A}^T \hat{A}$ that satisfies the relationship $\hat{A}^T \hat{A} = \hat{R}^T \hat{R}$ given the Cholesky factor R for $A^T A$ and the vector a . This can be done by using the relation

$$\hat{R}^T \hat{R} = \begin{pmatrix} R^T & ia \end{pmatrix} \begin{pmatrix} R \\ ia^T \end{pmatrix} = \begin{pmatrix} R^T & ia \end{pmatrix} J^T J \begin{pmatrix} R \\ ia^T \end{pmatrix}$$

where $i = \sqrt{-1}$, for any orthogonal transformation J .

- (a) [60 %] Show how J can be computed to make

$$J \begin{pmatrix} R \\ ia^T \end{pmatrix} = \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}$$

which gives the Cholesky factor \hat{R} for $\hat{A}^T \hat{A}$. Show that J can be chosen so that all the computations involved can stay in real field (e.g., no complex numbers are involved in the computation).

- (b) [40 %] Define a 2×2 hyperbolic transformation as $\begin{pmatrix} \cosh(\theta) & \sinh(\theta) \\ \sinh(\theta) & \cosh(\theta) \end{pmatrix}$. Note that

$$\begin{pmatrix} \cosh(\theta) & \sinh(\theta) \\ \sinh(\theta) & \cosh(\theta) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \cosh(\theta) & \sinh(\theta) \\ \sinh(\theta) & \cosh(\theta) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Present a Cholesky decomposition downdating algorithm based on hyperbolic transformations and describe how J is related to the hyperbolic transformations in the downdating.

2. Consider the linear system $Ax = b$ where

$$A = \begin{pmatrix} 6 & -3 & 2 \\ 3 & -2 & 1 \\ 2 & -1 & 1.2 \end{pmatrix} \quad b = \begin{pmatrix} 5 \\ 2 \\ 2.0 \end{pmatrix}$$

The condition number of A associated with the 1-norm is $\kappa_1(A) = 33$.

- (a) [35 %] Consider the vector $\tilde{x} = (1, 1, 1)^T$ as an approximate solution to the system. Calculate the residual norm $\|b - A\tilde{x}\|_1$.
Use the condition number and this residual norm to give an upper bound for the norm $\|x - \tilde{x}\|_1 / \|x\|_1$, where x is the exact solution of the system.
- (b) [25 %] Show that if the term $a_{33} = 1.2$ is replaced by $a_{33} = 0.6666\dots = 2/3$ the matrix A becomes singular. [Hint: there is a vector of the form $z = [-1, 0, \xi]^T$, where ξ is a certain scalar to be determined, such that $Az = 0$]
- (c) [40 %] Use the result of the previous question to find a lower bound for $\kappa_1(A)$. Compare with the condition number given above and verify that you do indeed obtain a lower bound.

3. Consider the following quadratic program:

$$\min_x \frac{1}{2} x^T A x - b^T x \quad \text{subject to} \quad Bx = 0, \quad (1)$$

where $x, b \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$, and $B \in \mathbb{R}^{m \times n}$, with $m < n$; A, B, b are assumed to be known.

- (a) [20 %] After introducing Lagrange multipliers p , state the first-order optimality conditions for Equation (1).
- (b) [20 %] State the second-order sufficient conditions for Equation (1).
- (c) [30 %] In order to solve for x , one has to “invert”¹

$$K = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix}.$$

Show that K is an indefinite matrix.

- (d) [15 %] Assuming A is invertible, derive an expression for the Schur-complement matrix of p in K .
 - (e) [15 %] Let V be an orthonormal basis for the null space of B . Using V , transform (1) to an unconstrained optimization problem and derive its first-order optimality condition.
4. Consider a two-dimensional dynamical system given by

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} -3 & 5 \\ 5 & -8 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}, \quad 0 < t \leq 10 \quad \text{and} \quad \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad (2)$$

where $\dot{z}(t) = \frac{dz(t)}{dt}$ for any given function $z(t)$.

- (a) [10 %] Derive an expression for the exact solution of (2).
- (b) [45 %] Implement the forward and backward Euler schemes (you can choose any programming language/mathematical package you wish) for (2). Perform a numerical simulation by discretizing $0 < t < 10$ using time step $dt = 1/2, 1/32, 1/128$. Tabulate the error of the solution at the Euler discretization points as a function of the time step index and report the relative mean square error for $t = 10$.
- (c) [45 %] Calculate the largest dt for both the forward and backward Euler methods for which the methods are numerically stable when applied to (2).

4 Discrete Algorithms

1. In computational biology, DNA can be represented as a sequence of characters drawn from an alphabet of four letters, A, C, T, and G, representing the four nucleotides. Given two sequences S_1 and S_2 of n and m characters, respectively, describe what is meant by a local alignment. Given a similarity score of +2, a mismatch penalty of -1, and a gap score of 0, give an efficient sequential algorithm to compute the score of the best local alignment between S_1 and S_2 . What is the asymptotic complexity of your algorithm? What are the space requirements? Suppose now that you are given a multi-core processor with p cores (with $1 < p < \min(n, m)$), design and analyze a multicore

¹By “inverting” K , we mean solving $Kv = g$ for v .

algorithm for sequence similarity problem using local alignments that scales with the number of cores. Describe in detail whether or not your algorithm is cache-friendly.

2. Given an undirected, connected, sparse graph $G = (V, E)$ with $n = |V|$, $m = |E|$ and an average vertex degree (m/n) of $O(1)$, give an algorithm to find a spanning tree of G starting from vertex $s \in V$. A spanning tree is a subset of $m = n - 1$ edges that form a tree of all of the n vertices in the graph such that no cycles (or loops) are formed.
 - (a) What data structures are selected and why?
 - (b) What is the complexity of this algorithm?
 - (c) Describe the performance one would expect from an implementation of this algorithm on a 32-bit uniprocessor computer (e.g. your PC), assuming $n < 100,000$.
 - (d) How much memory (as a function of n and m) is required?
 - (e) What do you expect dominates the running time of the implementation?
 - (f) Assume now that you wish to find a spanning tree on a graph with a billion vertices. Please identify strategies you could employ to solve this large problem.
3. It is often useful to partition graphs for various computational science and engineering applications. For example, given a graph $G = (V, E)$, we wish to partition the vertices into k sets such that each set contains about n/k vertices, and the total number of edges cut is minimized. Please describe two heuristics for partitioning graphs when the vertices have nodal coordinates. Describe how these approaches work for graph bisection ($k = 2$) versus multilevel partitioning (e.g., for $k = 16$). Give an example of a graph topology that works well for each of the heuristics, and an example of a topology that does not work well. Explain what is meant by coarsening a graph in multilevel partitioning, and give an example of an algorithm that could coarsen a graph without nodal coordinates.
4. In the RAM model, a balanced binary tree is often held in an array data structure of n elements where node i 's two children are held in locations $2i$ and $2i + 1$. Compare the time complexity of searching for a leaf in this tree using the RAM model and using the cache-oblivious model with block size B and memory size M , and $n \gg M$. If there is a more effective cache-oblivious data structure for this problem, describe in detail the layout and the new cost for performing a search.