620-361 Operations Research Techniques and Algorithms

Practice Class 5

1. How can you tell if a matrix is positive definite, positive semi-definite, negative definite or negative semi-definite?
   Solution:

   (a) Positive definite - we say a matrix, M, is positive definite if $x^T Mx > 0$ for all nonzero vectors x. Quite simply, if all the eigenvalues are strictly greater than 0, then the matrix is positive definite. Alternatively, if all the leading principal minors are positive, then the matrix is positive definite.

   (b) Positive semi-definite - we say a matrix, M, is positive semi-definite if $x^T Mx \geq 0$ for all nonzero vectors x. A matrix is positive-semi definite if there are some zero eigenvalues and some positive eigenvalues.

   (c) Negative definite - we say a matrix, M, is negative definite if $x^T Mx < 0$ for all nonzero vectors x. We look for a matrix with strictly negative eigenvalues.

   (d) Negative semi-definite - we say a matrix, M, is negative semi-definite if $x^T Mx \leq 0$ for all nonzero vectors x. We look for a matrix with some zero eigenvalues while the rest are negative.

2. (a) Name two ways to calculate, from a given matrix, whether it is positive definite.
   Solution:

   i. Determine the eigenvalues of a matrix as follows:

   $$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

   We take λ from the diagonal values:

   $$M = \begin{bmatrix} a - \lambda & b \\ c & d - \lambda \end{bmatrix}$$

   Now find the determinant of this matrix:

   $$(a - \lambda)(d - \lambda) - bc$$

   and solve it $= 0$ for λ. These are your eigenvalues. If they are all strictly positive then you have a positive definite matrix.

ii. Determine the leading principal minors of the matrix as follows:

$$M = \begin{bmatrix} \mathbf{a} & b \\ c & d \end{bmatrix}$$

a is the first leading principal minor. We want this to be positive.

$$M = \begin{bmatrix} \mathbf{a} & \mathbf{b} \\ \mathbf{c} & \mathbf{d} \end{bmatrix}$$

The entire matrix is the next leading principal minor. Find the determinant. We want this to be positive also. If both principal minors are positive then we have a positive-definite matrix.

For larger matrices, you can extend this by finding all the eigenvalues for the first way, and finding the determinants of all the leading principal minors for the second way.

(b) Which way is quickest for a 2x2 and 3x3 matrix?
Solution: For a 2x2 find the leading principal minors, as this only requires a simple determinant calculation. For a 3x3, calculating the determinant is far more tedious, and it is sometimes faster to calculate the eigenvalues of this matrix.

3. In the context of what we have been studying, why is it sometimes important to have a positive definite matrix?
Solution: For unconstrained optimisation we would like to have a convex function. We use the fact that the Hessian of the objective function is positive definite to indicate that the function is also convex. For Newton's method, we show that the direction chosen is actually a descent direction by demonstrating that the inverse of the Hessian is positive definite. Similarly for quasi-Newton methods (e.g. BFGS), we require the Hessian to be positive definite for the same reason.

4. What do we mean by 'full rank'?
Solution:
We mean that the columns of the matrix are linearly independent.

5. What is an affine function? Give an example.
Solution:
An affine function is a linear function which may contain a constant. For example, $f(x_1, x_2) = A_1x_1 + A_2x_2 + b$.

6. What do we mean when we say a function is C^1 or C^2?
Solution:
When we say C^1 we mean that the function belongs to the space of continuously differentiable functions (which are differentiable once and have continuous derivative). Similarly with C^2, we mean it is differentiable twice and has continuous second derivative.

7. What is the Jacobian matrix?
Solution:
Given a set $\mathbf{y} = \mathbf{f}(\mathbf{x})$ of m equations in n variables $x_1, \ldots, x_n$ written as:

$$\mathbf{y} = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix}$$

The Jacobian is defined by:

$$J(x_1, \ldots, x_n) = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_1}{\partial x_n} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

Note that sometimes the determinant of the Jacobian matrix is called the Jacobian.

8. What is the Hessian matrix?
Solution:
Given a function $f(x_1, x_2, \ldots, x_n)$, the Hessian is defined by:

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

9. Why is unimodality a concern?
Solution:
The methods we have learned about are designed to search a line in a descending fashion until a good approximation for the minimum is found. They are not designed to climb out of local minima in search of global minima. Unimodality guarantees that only one minima exists, and hence we can be confident that the method will find the right one.

10. When would you use Fibonacci search?
Solution:
When we have a 1-D unconstrained optimisation problem and we want an efficient method to find the minimum. We know exactly when we want to stop the algorithm *a priori*.

11. When would you use Golden Section search?
Solution:
When we have a 1-D unconstrained optimisation problem and we don't know *a priori* when we want to stop the algorithm.

12. When would you use the Method of False Position?
Solution:
When we have a 1-D unconstrained optimisation problem and we can evaluate the derivative at each iteration.

13. When would you use Newton's method for 1D problems?
Solution:
When we have a 1-D unconstrained optimisation problem where we can calculate the derivative and the second derivative of the function and evaluate these at each iteration.

14. What do we mean by 'step size'? What is this used for?
Solution:
At each iteration of a direct search method, a step is taken in some search direction that will improve the objective function. We want to know how far we should travel in that direction. This is the step size.

15. How can you tell if a function is convex?
Solution:
Visually, we know if a function is convex if we can draw a line between any two points in the function and find that the line we have drawn is above the function. If the Hessian of the function is positive definite, then the function is convex.

16. Why is it sometimes important for a function to be convex?
Solution:
For minimisation problems, convexity guarantees unimodality. Basically all the methods we have learned so far are only guaranteed to work if we have unimodality.

17. What do we mean when we say $||x^k - x^*||$?
Solution:
We want to find the error in our estimate x^k - the distance between our estimate and the actual optimal solution.

18. What do we mean when we say $||\nabla f(x^k)||$?
Solution:
The gradient is the vector of first derivatives. Its norm is the magnitude of the gradient vector. Because we often want $\nabla f(x^k) = 0$, the norm should be close to 0 if we are close to the optimal solution.

19. What do we mean by 'rate of convergence'?
Solution:
We refer to the rate at which the error in our estimate shrinks. We compare the error in our current estimate to the error in our previous estimate to obtain this rate.

20. When performing a direct search, what are some suitable stopping criteria?
Solution:

(a) Gradient norm - We expect this number to be small near the optimum. So when $||\nabla f(x^k)|| < \epsilon$, we can stop.

(b) Function difference - Near the optimum, the difference between successive values of the objective function will become small. When $|f(x^k) - f(x^{k-1})| < \epsilon$, stop.

(c) Step size - Near the optimum, one expects this step to become smaller and smaller. When $t < \epsilon$, stop.

(d) Number of iterations - When the program has reached some maximum level of iterations, stop.

21. When would you use Newton's method for n-D problems?
Solution:
We use Newton's method when we have an unconstrained n-D problem where the objective function is C^2.

22. When would you use BFGS for n-D problems?
Solution:
This method is designed for problems where the Hessian is difficult to evaluate or calculate.

23. When would you use the Lagrange method?
Solution:
When you have an optimisation model with nonlinearity in either the objective function or constraints (or both), and all the constraints are expressed as equality constraints.

24. When would you use the Karush-Kuhn-Tucker conditions?
Solution:
When you have an optimisation model with nonlinearity in either the objective function or constraints (or both), and some of the constraints are expressed as inequality constraints.

25. How do we solve the KKT conditions?
Solution:
If the KKT conditions are easy to solve by inspection, we do that. If not, we see what happens when each set of cosntraints are set to be active or inactive. If we are unable to solve the equations algebraically, we can try penalty methods.

26. When would you use a penalty method?
Solution:
When you cannot solve the KKT equations due to nonlinearity and/or a large number of variables (or just plain tedium!).