Basic Optimisation

Machine Learning

Optimisation[1]

[1] Material derived from: R. Bronson and G. Naadimuthu, *Operations Research*; D. Rosenberg, "Extreme Abdridgement of Boyd and Vandenberghe's *Convex Optimisation*"; P.S. Sastry's course on Pattern Recognition

Single Variable Optimisation I

optimise: $f(x)$	OR	optimise: $f(x)$ subject to: $a \leq x \leq b$
(a)		(b)

- Here *optimise* means *maximise* or *minimise*
- The optimisation problem in (a) is called *unconstrained* optimisation, and in (b) is called *constrained* optimsation
- If a constrained optimisation problem has no solution, then constraining the value of x may give a solution
  - For example, $f(x) = x$ has no finite maximum (or minimum). But if $a \leq x \leq b$ then the maximum (and minimum) are well-defined

Single Variable Optimisation II

- Values of x satisfying the constraints are called *feasible* solutions. The constrained optimisation problem is to find the the optimal value of $f(x)$ amongst feasible solutions
- If for some feasible x, $f(x) \leq f(x')$ for values of x for which $f(x)$ is defined then x is called a *global* minimum (similarly for a global maximum). If $f(x) \leq f(x')$ for $x' \in Nbd(x)$ then x is said to be a *local* minimum (similarly for a local maximum)
    - A global optimum is a local optimum, but not *vice versa*.
    - Do not confuse constrained optimisation with finding a local optimum. The constrained optimisation problem requires us to find the optimal value in some interval $[a, b]$ which need not be a small

Single Variable Optimisation III

- The following results from the calculus are known:
  1. A function that is continuous in $[a, b]$ has a global minimum and global maximum in $[a, b]$
  2. if f has a local optimum at x_0 and $f'(x)$ is defined at x_0, then $f'(x_0) = 0$
  3. if f has a local optimum at x_0 and $f'(x)$ and $f''(x)$ are defined at x_0, then: (a) if $f'(x_0) = 0$ and $f''(x_0) < 0$ then x_0 is a local maximum for $f(x)$; and (b) if $f'(x_0) = 0$ and $f''(x_0) > 0$ then x_0 is a local minimum for $f(x)$
- So, for constrained optimisation problems, solutions are either: (a) at points where $f'(x)$ is not defined; or (b) at points where $f'(x) = 0$; or (c) at the end-points a or b.

Convex amd Concave functions I

- A set C is convex if for any $x_{1,2} \in C$ and any $\alpha \in [0,1]$:

$$x = \alpha x_1 + (1-\alpha) x_2 \in C$$

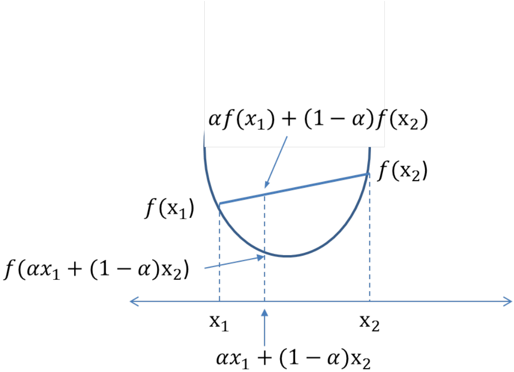
If $C = \Re^2$, then $x_{1,2}$ are points in 2-d space, and x is a point on the line segment joining x_1 and x_2

- A function f that satisfies

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2) \qquad (0 \leq \alpha \leq 1)$$

Geometrically:

Convex amd Concave functions II



So, for convex functions, the line joining a pair of points lies "above" the function

Convex amd Concave functions III

- For $x_{1,2} \in (a, b)$ is said to be *convex* on (a, b). If $(a, b) = (-\infty, +\infty)$ then f is simply said to be a convex function. If the inequality is reversed, then the function is said to be concave in (a, b) (that is, a function f is concave if the negative of the function is convex)
- Examples of convex functions are:
  - Linear functions of the form $ax + b$ (for all a, b)
  - Power functions of the form $|x|^p$ for $pfgeq1$
  - Exponential functionsof the form e^{ax} (for all a)
  - Norms on $\Re^n$ (like $\mid x \mid$ or $\mid x \mid_2$
  - $\max(x_1, x_2, \ldots, x_n)$ is convex
- A function is *strictly convex* if the line segment is strictly above the function (a linear function is not strictly convex)
- The following results are known:
  - For a convex function, any local minimum is also a global minimum

Convex amd Concave functions IV

- For a strictly convex function, if there is a local minimum then it is a unique global minimum

Multivariate Unconstrained Optimisation I

- We extend the univariate optimisation problem to a multivariate one. Now we want to optimise the value of $u = f(\mathrm{x})$.
  - the response y is a scalar field f that at each point $x_1, x_2, \ldots, x_k$ gives the response $f(x_1, x_2, \ldots, x_k)$
- The results from the calculus require counterparts to the first- and second-differentials
- The counterpart to the first-derivative with x is the *gradient*
  - From standard vector calculus, the gradient of f at the point gives the direction in which y will change most quickly (that is, the direction of steepest ascent).

Multivariate Unconstrained Optimisation II

- This gradient, usually denoted ∇f, is the vector:

$$\left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_k}\right)$$

  This is also denoted in matrix notation as:

$$\left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_k}\right]^T$$

- The *Hessian* matrix H_f associated the function $f(x)$ is the matrix $H|_f$

$$\left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right] \qquad (i, j = 1 \dots n)$$

  We are usually interested in the value of the Hessian matrix at some value x_0. This is denoted by $H|_{f}, x_0$

Multivariate Unconstrained Optimisation III

1. Let $f(x_1, x_2, x_3) = 3x_1^2 x_2 - x_2^2 x_3^3$. What is ∇f at $\mathrm{x}_0 = [1, 2, 3]^T$?

   Answer.

$$\nabla f = \begin{bmatrix} 6x_1 x_2 \\ 3x_1^2 - 2x_2 x_3^3 \\ -3x_2^2 x_3^2 \end{bmatrix}$$

$$\nabla f|_{\mathrm{x}_0} = \begin{bmatrix} 12 \\ -105 \\ -108 \end{bmatrix}$$

   This is the direction of greatest increase of f at the point x_0

2. What is the Hessian for f at x_0 as above?

Multivariate Unconstrained Optimisation IV

Answer.

$$H|_f = \begin{bmatrix} 6x_2 & 6x_1 & 0 \\ 6x_1 & -2x_3^3 & -6x_2x_3^2 \\ 0 & -6x_2x_3^2 & -6x_2^2x_3 \end{bmatrix}$$

Substituting the values for x_0 we get:

$$H|_{f}, x_0 = \begin{bmatrix} 12 & 6 & 0 \\ 6 & -54 & -108 \\ 0 & -108 & -72 \end{bmatrix}$$

Multivariate Unconstrained Optimisation V

- We will also need the following:
  - A matrix A is *symmetric* if $A = A^T$
  - A s ymmetric matrix is *negative definite* if $X^T AX$ is negative for every non-zero n-dimensional vector X
  - A symmetric matrix is *negative semi-definite* if $X^T AX$ is negative or 0 for every n-dimensional vector X
- Let:

$$A_1 = \begin{vmatrix} a_{11} \end{vmatrix} \; A_2 = (-1) \times \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \; A_3 = (-1)^2 \times \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \cdots$$

Or in general:

$$A_n = (-1)^{n-1} det(A)$$

Multivariate Unconstrained Optimisation VI

- A is negative definite iff $A_1, A_2, \ldots, A_n$ are all negative and negative semi-definite iff there exists some $r < n$ s.t. the A_i for $i \leq r$ are negative, and are 0 for $i > r$.
- If the second partial derivatives of a function f are continuous at x_0 then the Hessian $\mathrm{H}|_{f,\mathrm{x}_0}$ will be symmetric

3. Is the Hessian obtained earlier negative, or semi-negative, or neither at x_0?

   Answer. Since $A_1 = 12$ for the Hessian, it is not negative or semi-negative at x_0.

- The results from the calculus are slightly more restricted in the multivariate case:
  1. If $f(\mathrm{x})$ is continuous on a closed region then $f(\mathrm{x})$ has a global maximum and minimum in the region

Multivariate Unconstrained Optimisation VII

2. If $f(\mathrm{x})$ has a local maximum or minimum at some point x^* and ∇f is defined in some ϵ-neighbourhood around x^* then $\nabla f|_{\mathrm{x}}^* = 0$
3. If $f(\mathrm{x})$ has both ∇f and second partial derivates defined in some ϵ-neighbourhood around x^* and $\nabla f|_{\mathrm{x}}^* = 0$ and $\mathrm{H}|_f, \mathrm{x}^*$ is negative-definite then $f(\mathrm{x})$ has a local maximum at x^*

Numerical Optimisation I

- In general, analytical expressions for optimal values of a multivariate function $f(\mathrm{x})$ are hard to obtain
- The most commonly used numerical procedure uses the *graadient* to perform either *gradient ascent*, or *gradient descent*.
- Gradient ascent:
  1. Start with some guess x_0
  2. Determine subsequent vectors $\mathrm{x}_1, \mathrm{x}_2, \ldots$ using the update formula:

     $$\mathrm{x}_{k+1} = \mathrm{x}_k + \eta^* \nabla f|_{\mathrm{x}_k}$$

     where η^* is the value of a scalar η that results in the maximum value for $f(\mathrm{x}_k + \eta \nabla f|_{\mathrm{x}_k})$ (often, η^* is just taken to be a small constant)
  3. Stop when $\mathrm{x}_k \approx \mathrm{x}_{k+1}$

Numerical Optimisation II

- This is a greedy search in the direction of maximal increase. Replacing the $+$ sign by $-$ in the update formula will result in a search in the direction of maximal decrease. The resulting procedure is gradient *descent*
- The choice of the initial value x_0 can affect the quality of the solution found. There is no general way of determining a good starting point: some of the problem is alleviated by multiple random restarts, and selecting the best of the local optima obtained
- There are other numerical procedures (Newton-Raphson is a prominent example) that may converge faster. But all numerical methods only converge to a local maximum or minimum

Numerical Optimisation III

- The exceptions are if the functions are concave or convex, in which case any local optimum found is the global optimum. The definition of a convex function in the multivariate case is a generalisation of the univariate definition:

  $$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2)$$

  A function is concave if its negative is convex
- The following results are known:
  1. If a function $f(x)$ has second partial derivates defined on $\Re$ then $f(x)$ on $\Re$ iff its Hessian is negative semi-definite for all $x \in \Re$
  2. If a function $f(x)$ is concave in $\Re$ then any local maximum is a global maximum

Numerical Optimisation IV

That is, if the Hessian is negative semi-definite everywhere in $\Re$ then any local maximum is a global maximum. Alternatively, if the Hessian of $-f(\mathrm{x})$ is negative and semi-definite everywhere in $\Re$ then any local minimum is a global minimum

4. Show that at every interation, gradient ascent at a point x_k moves in the direction of greatest increase of $f(\mathrm{x}_k)$

Answer. The rate of change of $f(\mathrm{x})$ at x_k in the direction of any unit vector U is:

$$\nabla f|_{\mathrm{x}_k} \cdot \mathrm{U} = |\nabla f||\mathrm{U}| cos\theta$$

This is a maximum when $cos\theta = 1$ or $\theta = 0$. That is, U is in the same direction as $\nabla f|_{\mathrm{x}_k}$. Any scalar multiple $\eta^* \nabla f|_{\mathrm{x}_k}$ is in this direction.

5. For $f(x_1, x_2, x_3) = x_1(x_2 - 1) + x_3^3 - 3x_3$, find where $\nabla f = 0$, and the values of f at these points. Are any of these global maxima or minima?

   Answer.

$$\nabla f = \begin{bmatrix} x_2 - 1 \\ x_1 \\ 3x_3^2 - 3 \end{bmatrix}$$

Solving, we get $\nabla f = 0$ at $\mathrm{x}_1 = [0, 1, 1]^T$ and $\mathrm{x}_2 = [0, 1, -1]^T$. $F(\mathrm{x}_1) = 0(1 - 1) + 1 - 3 = -2$ and $f(\mathrm{x}_2) = 0(1 - 1) - 1 + 3 = 2$.
None of these are global optima, since f can increase or decrease without limit. So, it does not have a finite global maximum or minimum. as x_2

6. Maximise $z = f(x_1, x_2) = -(x_1 - \sqrt{5})^2 - (x_2 - \pi)^2 - 10$.

Answer. Note first that f is continuous on $\Re$ and that as x_1, x_2 become very large or very small, f behaves as $-x_1^2 - x_2^2 - 10$, which becomes arbitrarily small. Now $\nabla f = [-2(x_1 - \sqrt{5}), -2(x_2 - \pi)]^T$, which is 0 at $x_1 = \sqrt{5}$ and $x_2 = \pi$. The value f at this point is -10, which a maximum for f

7. Find the maximum for the function f above, using gradient ascent.

Answer. The steps are as follows:

Gradient. The gradient of f is:

$$\nabla f = \begin{bmatrix} -2(x_1 - \sqrt{5}) \\ -2(x_2 - \pi) \end{bmatrix}$$

Numerical Optimisation VII

Sample points for obtaining x_0. Here is a sample of possible start points:

x_1	−8.537	−0.9198	9.201	9.250	6.597	8.411	8.202	−9.173	−9.337	−5.794
x_2	−1.099	−8.005	−2.524	7.546	5.891	−9.945	−5.709	−6.914	8.163	−0.0210
z	−144.0	−144.2	−90.61	−78.59	−36.58	−219.4	−123.9	−241.3	−169.2	−84.48

(An obvious start point from this sample is $x_0 = [6.597, 5.891]^T$)

Iteration Step 1. For the first iteration:

$$x_1 = x_0 + \eta \nabla f|_{x_0}$$

from which we get
$f(x_1) = 106.3\eta^2 + 106.3\eta - 36.58.$
Solving, we find $f(x)_1)$ is maximised for $\eta = \eta^* 0.5$, So, $x_1 = x_0 + \eta^* \nabla f|_{x_0} = [2.236, 3.142]^T$

Numerical Optimisation VIII

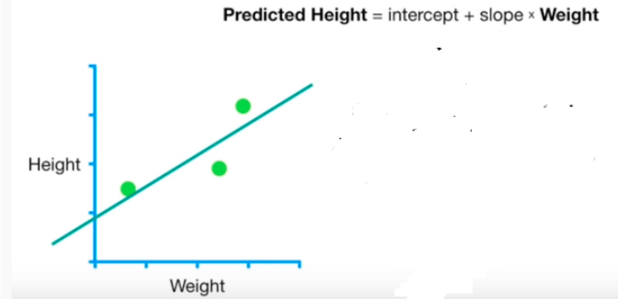
Stop? Checking, we find the value of $f(x_1) = -10$ is substantially different to $f(x_0) = -36.58$. So we continue.

Iteration Step 2. This is left as an exercise.

- In machine learning, we will often be minimising a *loss functin* (for example, MSE). In this case, we will be using *gradient descent* to find numerical values of parameters of a model that minimises the loss function

Example: Parameter Estimation using Gradient Descent I

- Let us look at estimating values of one parameter for a simple linear model:[2]



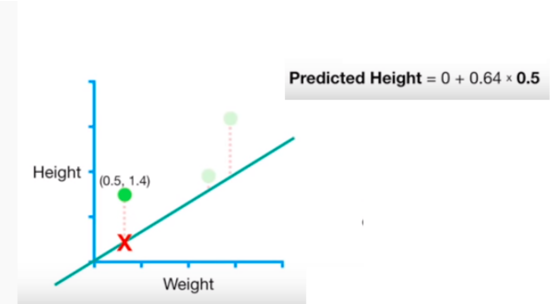
- For the moment, we will simply minimise the sum of squared residuals (SSR)
- The parameters of this model are the intercept and the slope. So, ideally, we want to find optimal values for these two quantities.

Example: Parameter Estimation using Gradient Descent II

- Using the usual method of partial differentiation w.r.t the slope and and the intercept gives optimal values of $Intercept = 0.95$ and $Slope = 0.64$
- BUT: analytical optimisation is difficult to automate, and may not even be possible in some cases
  - Instead, we will look at a general-purpose greedy search (gradient descent)
- Let us first look at how gradient descent works with one parameter. We will assume we have already found the value of the slope to be 0.64
- For any predicted line, let us take the sum of squared residuals as the *loss function*. This is sometimes called the *squared loss* function

Example: Parameter Estimation using Gradient Descent III

- We can now evaluate the sum of the squares of residuals for any predicted line



Example: Parameter Estimation using Gradient Descent IV

- We can write the sum of squares of residuals as the following:

$$SSR = [1.4-(I+0.64\times0.5)]^2+[1.9-(I+0.64\times2.3)]^2+[3.2-(I+0.64\times2.9)]^2$$

  This can be visualised as function between I (X-axis) and SSR (Y-axis). As I gets the optimal value, the slope of this function will be close to 0

- The derivative of the loss function w.r.t. I is then:

$$\frac{d(SSR)}{dI} = -2[1.4-(I+0.64\times0.5)]-2[1.9-(I+0.64\times2.3)]-2[3.2-(I+0.64\times2.9)]$$

- Let is start with a random value of *Intercept*, say $I = 0$.
- The value of this derivative at $I = 0$ (our first guess) is -5.7. This is the slope of the I vs SSR curve at $I = 0$

Example: Parameter Estimation using Gradient Descent V

- Gradient takes steps along I using the the slope:

$$I_{k+1} \;=\; I_k - \eta \times \frac{d(SSR)}{dI_k}$$

  Here, if $\eta = 0.1$, then the step is $-5.7 \times 0.1 = -0.57$ and:

$$I_2 \;=\; 0 - (-0.57) \;=\; 0.57$$

- Repeat the process of calculating the derivative with $I = 0.57$, which gives a slope of -2.3 at $I = 0.57$. The new step size is therefore -0.23 and:

$$I_3 \;=\; 0.57 + 0.23 \;+\; 0.8$$

- Repeating will result in intercepts $I = 0.89, 0.92, 0.94, 0.95$

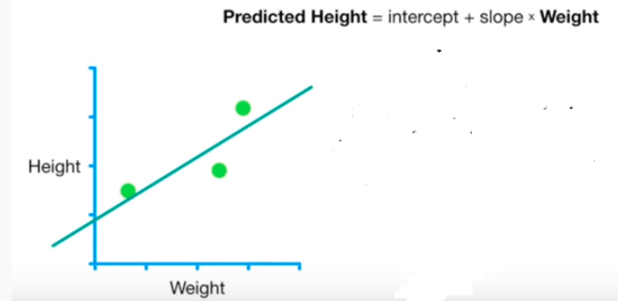
Example: Parameter Estimation using Gradient Descent VI

- So, gradient descent gets to the optimal value (0.95). But how does it know to stop? Stops when the step size is very close to 0. Since $Step = \eta Slope$, this must mean that *Slope* is very close to 0.
- In this case, gradient descent will stop with $I = 0.95$

[2] The following example from Statquest's online presentation on Gradient Descent

Example: Multi-parameter Gradient Descent I

- Let us look at finding values of both parameters for the simple linear model:[3]



- We will now look at estimating both *Slope* and *Intercept* using gradient descent. As before, we start with the loss function:

Example: Multi-parameter Gradient Descent II

- We can write the sum of squares of residuals as the following:

$$SSR = [1.4-(I+S\times0.5)]^2 + [1.9-(I+S\times2.3)]^2+[3.2-(I+S\times2.9)]^2$$

SSR is now a function of both I and S, and we want to estimate the optimal value of both I and S.

- We will need to calculate slopes w.r.t I and S separately. These are the following:

$$\frac{\partial(SSR)}{\partial I} = -2[1.4-(I+S\times0.5)] - 2[1.9-(I+S\times2.3)] - 2[3.2-(I+S\times2.9)]$$

$$\frac{\partial(SSR)}{\partial S} = -2\times0.5[1.4-(I+S\times0.5)] - 2\times2.3[1.9-(I+S\times2.3)] - 2\times2.9[3.2-(I+S\times$$

- Like before, we will start with random choices, say: $I = 0$, $S = 1$. This gives us 2 slopes:

$$\frac{\partial(SSR)}{\partial I} = -1.6 \qquad \frac{\partial(SSR)}{\partial S} = -0.8$$

Example: Multi-parameter Gradient Descent III

- With a learning rate of $\eta = 0.01$, we get step-sizes of 0.01×-1.6 for I and 0.01×-0.8 for S. So, the new values of I and S are

$$I_2 \ + \ I_1 - \eta \times \frac{\partial(SSR)}{\partial I_1} \ = \ 0 + 0.016 \ = \ 0.016$$

and

$$S_2 \ + \ S_1 - \eta \times \frac{\partial(SSR)}{\partial S_1} \ = \ 1 + 0.008- \ = \ 1.008$$

- Repeat with the new values of I and S, until step sizes are very small. Here, gradient descent terminates with $I = 0.95$ and $S = 0.64$

[3]The following example from Statquest's online presentation on Gradient Descent

Multivariate Optimisation with Constraints I

- For $\mathrm{x} \in \Re^n$:

$$
\begin{array}{lcl}
\text{optimise: } f(\mathrm{x}) & & \text{optimise: } f(\mathrm{x}) \\
\text{subject to:} & & \text{subject to:} \\
g_1(\mathrm{x}) = 0 & & g_1(\mathrm{x}) \leq 0 \\
g_2(\mathrm{x}) = 0 & \text{OR} & g_2(\mathrm{x}) \leq 0 \\
\vdots & & \vdots \\
g_m(\mathrm{x}) = 0 & & g_m(\mathrm{x}) \leq 0 \\
(\mathrm{a}) & & (\mathrm{b})
\end{array}
$$

  with $m < n$

- A variation requires $\mathrm{x} \geq 0$. This can be translated into one of the forms (a) or (b):
  - The translation to (b) is easy: we introduce n additional constraints $-x_1 \leq 0$, $x_2 \leq 0$, $\ldots x_n \leq 0$

Multivariate Optimisation with Constraints II

- The translation to (a) requires the introduction of *slack* variables $s_1, s_2, \ldots s_n$. additional constraints of the form $-x_1 + s_1^2 = 0$, $-x_2 + s_2^2 = 0$, $\ldots -x_n + s_n^2 = 0$ (the use of squared slack variables ensures that the added term is positive)
- The technique of using slack variables can also be used to transform any constrained optimisation problem of type (b) into one of type type (a)
- One technique for solving problems of type (a) is by penalising values of x that violate one or more constraints

Penalty-Based Optimisation I

- An alternative to the use of Lagrange multipliers is to use *penalty* functions. For example, the maximisation problem is transformed into:

  $$\text{maximise } z = f(\mathrm{x}) - \sum_{i=1}^{m} p_i g_i(\mathrm{x})$$

  Here the p_i are positive *penalties*, and the r.h.s. is called the *penalty function*.
- Forcing the p_i to large values will prefer x values for which the corresponding g_i closer to 0 (which is the value required by the constraint)
- This suggests the following iterative procedure:
  1. Transform the problem to the standard form (maximisation)
  2. Form the penalty function, by selecting random positive penalties p_i for each constraint g_i

Penalty-Based Optimisation II

3. Find the maximimum of the penalty function (usually only possible numerically)
4. For the solution x^*, find the constraints g_i s.t. $g_i(x^*$ are not close to 0. If no such constraint g_i exists, then stop. Otherwise increase the corresponding penalty p_i
5. Repeat the maximisation step using the new penalty function

- The intuitive idea of using a penalty function is developed more rigorously with the use of *Lagrange multipliers*

Lagrange Multipliers I

$$\begin{array}{lcl} \text{minimise: } f(\mathrm{x}) & & \text{maximise: } f(\mathrm{x}) \\ \text{subject to:} & & \text{subject to:} \\ g_1(\mathrm{x}) \leq 0 & & g_1(\mathrm{x}) \leq 0 \\ g_2(\mathrm{x}) \leq 0 & \text{OR} & g_2(\mathrm{x}) \leq 0 \\ \vdots & & \vdots \\ g_m(\mathrm{x}) \leq 0 & & g_m(\mathrm{x}) \leq 0 \\ \text{(a)} & & \text{(b)} \end{array}$$

- Provided some conditions on the partial derivatives of f and g are satisfied, then it can be shown that if for some x^*:

$$-\nabla f|_{\mathrm{x}}^{*} = \lambda_i \nabla g_i(\mathrm{x}^*)$$

then x^* is a solution the optimisation problem (a)

Lagrange Multipliers II

- Similarly if:

$$\nabla f|_{\mathrm{x}}^{*} = \lambda_i \nabla g_i(\mathrm{x}^*)$$

  then x^* is a solution to the optimisation problem (b)
- We define the *Lagrangian* for (a) as the function

$$L(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_m) \; + \; f(\mathrm{x}) - \sum_{i=1}^{m} \lambda_i g_i(\mathrm{x})$$

  Then:

$$\nabla L = \nabla f(\mathrm{x}) - \sum_i \lambda_i \nabla g_i(\mathrm{x})$$

- It is clear that for all points x^* s.t. $\nabla L|_{\mathrm{x}}^{*} = 0$ $\nabla f|_{mathbfx}^{*} + \sum \lambda_i g_i(\mathrm{x}^*) = 0$ and x^* is a solution to (a)

Lagrange Multipliers III

- Similarly, the Lagrangian for (b) is:

$$L(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_m) \; - \; f(\mathrm{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathrm{x})$$

  and a similar result follows
- $\nabla L = 0$ is a system of $n + m$ equations in $n + m$ unknowns:

$$\begin{aligned} \frac{\partial L}{\partial x_i} &= 0 \qquad\qquad & (i = 1, 2, \ldots n) \\ \frac{\partial L}{\partial \lambda_j} &= 0 \qquad\qquad & (j = 1, 2, \ldots, m) \end{aligned}$$

Lagrange Multipliers IV

- Often, it may not be possible to find an analytical solution. In that case, some technique of numerical optimisation to find a local maximum for L will have to suffice

8. Maximise $f(x_1, x_2, x_3) = -(x_1 + x_2 + x_3)$ subject to the constraints:

$$\begin{aligned} x_1^2 + x_2 &= 3 \\ x_1 + 3x_2 + 2x_3 &= 7 \end{aligned}$$

Answer. We first bring this into the standard form for the constraints:

maximise: $z = f(x_1, x_2, x_3) = -(x_1 + x_2 + x_3)$
subject to:
$x_1^2 + x_2 - 3 = 0$
$x_1 + 3x_2 + 2x_3 - 7 = 0$

Lagrange Multipliers V

The Lagrangian is the function:

$$L(x_1, x_2, x_3, \lambda_1, \lambda_2) = -(x_1+x_2+x_3)-\lambda_1(x_1^2+x_2-3)-\lambda_2(x_1+$$

The solution to the constrained maximisation problem is amongst the solutions to the equations in $\nabla L = 0$. That is:

Lagrange Multipliers VI

$$\frac{\partial L}{\partial x_1} = -1 - 2x_1\lambda_1 - \lambda_2 = 0$$
$$\frac{\partial L}{\partial x_2} = -1 - \lambda_1 - 3\lambda_2 = 0$$
$$\frac{\partial L}{\partial x_3} = -1 - 2\lambda_2 = 0$$
$$\frac{\partial L}{\partial \lambda_1} = -(x_1^2 + x_2 - 3) = 0$$
$$\frac{\partial L}{\partial \lambda_2} = -(x_1 + 3x_2 + 2x_3 - 7) = 0$$

Solving, we get $\lambda_1 = 0.5$, $\lambda_2 = -0.5$, $x_1 = -0.5$, $x_2 = 2.75$, and $x_3 = -0.375$. This gives $z = -1.875$

Lagrange Multipliers VII

as the maximum, and 1.875 as the minimum for $f(x_1, x_2, x_3)$

The KKT Conditions and Duality I

- We now consider the more general form of constraints that contain inequalities. For example:

$$\begin{aligned} &\text{minimise: } f(\mathrm{x}) \\ &\text{subject to:} \\ &g_1(\mathrm{x}) \leq 0 \\ &g_2(\mathrm{x}) \leq 0 \\ &\qquad \vdots \\ &g_m(\mathrm{x}) \leq 0 \end{aligned}$$

- Sometimes, additional equality constraints of the form $h_j(\mathrm{x}) = 0$. We will treat these as a pair of inequalities $h_j(\mathrm{x}) \leq 0$ and $h_j(\mathrm{x}) \geq 0$ (replacing the latter with $-h_j(\mathrm{x}) \leq 0$ to get it into

The KKT Conditions and Duality II

- The method of Lagrange multipliers is generalised by the *Karush-Kuhn-Tucker* (KKT) conditions to account for inequalities. The conditions state that the solution to the optimisation problem are to be found in the solution to the problem of minimising the function:

$$L(\mathrm{x}, \lambda) = f(\mathrm{x}) + \sum_i \lambda_i g_i(\mathrm{x})$$

with the additional constraint that the $\lambda_i \geq 0$. The λ_i are called the KKT multipliers. Conventionally, we still call L the Lagrangian, and the KKT multipliers are still call Lagrange multipliers

The KKT Conditions and Duality III

- If the optimisation problem is a maximisation one, then the corresponding function is:

$$f(\mathrm{x}) - \sum_i \lambda_i g_i(\mathrm{x})$$

with $\lambda_i \geq 0$

9. Show for a specific value $\mathrm{x} = \mathrm{x}_0$:

$$\begin{aligned} \max_{\lambda_i \geq 0} L(\mathrm{x}, \lambda) &= f(\mathrm{x}_0) \qquad g_i(\mathrm{x}_0) \leq 0 \quad (i = 1, \ldots, m) \\ &= \infty \qquad\quad g_i(\mathrm{x}_0) > 0 \quad (i = 1, \ldots, m) \end{aligned}$$

(Correctly, *max* should be *sup*.)

The KKT Conditions and Duality IV

Answer. This follows from the fact that the maximum value of L can be made arbitrarily large by choosing a large λ_i for any $g_i > 0$. On the other hand, for all g_i's ≤ 0, the value of $\lambda_i = 0$ ensures L is maximised. If all g_i's are ≤ 0, then all λ_i's are $= 0$, and $L = f$.

- So, the original optimisation problem can be reformulated as:

$$\text{Find: } \mathrm{p}^* = \min_{\mathrm{x}} \max_{\lambda_i \geq 0} L(\mathrm{x}, \lambda)$$

which is the same as:

$$\text{Find:}\mathrm{p}^* = \underset{\mathrm{x}}{min}\ L(x, \lambda^*)$$

where λ^* denotes an optimal value for λ

The KKT Conditions and Duality V

- This is called the *primal form* of the original minimisation problem There is a *dual form*:

$$\text{Find:d}^* = \max_{\lambda_i \geq 0} \min_{x} L(x, \lambda)$$

  which is the same as:

$$\text{Find:d}^* = \max_{\lambda_i \geq 0} L(x^*, \lambda_i)$$

  where x^* denotes an optimal value for x

- So, the dual problem is:

  maximise: $g(\lambda)$
  subject to:
  $\lambda \geq 0$

The KKT Conditions and Duality VI

Here $g(\lambda) = \min_{\mathrm{x}} L(\mathrm{x}, \lambda)$ and the constraint is short for m inequalities of the form $\lambda_i \geq 0$ $(i = 1, \ldots, m)$

- The dual problem may be easier to solve, since the constraints are simpler
- In general, $\mathrm{p}^* \neq \mathrm{d}^*$. But for any optimisation problem, it can be shown that $\mathrm{p}^* \geq \mathrm{d}^*$, and if f is convex, then usually $\mathrm{p}^* = \mathrm{d}^*$

10. Determine the KKT conditions for the problem of minimising $x = x_1^2 + 5x_2^2 + 10x_3^2 - 4x_1x_2 - 12x_1x_3 - 2x - 1 + 10x_2 + 5x_3$ subject to: $x_1 + 2x_2 + x_3 \geq 4$ and all the $x_i \geq 0$

    Answer. The steps of the solution are:

    1. Transform it to standard form by maximising $-z$

The KKT Conditions and Duality VII

2. Transform inequality constraints into equalities using 4 new KKT variables (1 for the constraint, and 3 for constraints of non-negative x_i).
3. Form the KKT function L, with 4 multipliers $\lambda_1, \ldots \lambda_4$, one for each of the equality constraints
4. The KKT conditions follow from the system of equations resulting from the $\nabla L = 0$

- Given convex functions $f, g_1, \ldots, g_m$:

$$
\begin{aligned}
&\text{minimise: } f(\mathrm{x}) \\
&\text{subject to:} \\
&g_1(\mathrm{x}) \leq 0 \\
&g_2(\mathrm{x}) \leq 0 \\
&\qquad \vdots \\
&g_m(\mathrm{x}) \leq 0
\end{aligned}
$$

The KKT Conditions and Duality VIII

- If the constraint functions are linear, then as long as there is at least one point in the domain that satisfies the constraints, then the primal and dual solutions will be identical ("strong duality")

11. Let x^* be the optimal solution to the primal form for a convex optimisation problem, and let λ^* be the optimal solution to the dual form of the problem. Then, assuming strong duality, show:

$$\lambda_i{}^* g_i(x^*) = 0 \quad (i = 1, \dots, m)$$

This is called the *complementary slackness* condition: it holds if strong duality holds. It says that for all $\lambda_i > 0$, $g_i(x^*) = 0$

The KKT Conditions and Duality IX

Answer. Assuming strong duality:

$$f(\mathrm{x}^*) \;=\; g(\lambda^*)$$

Now, we know:

$$f(\mathrm{x}^*) \;=\; \min_{\mathrm{x}} L(\mathrm{x}, \lambda^*) \leq L(\mathrm{x}^*, \lambda^*)$$

Since:

$$L(\mathrm{x}^*, \lambda^*) = f(\mathrm{x}^*) + \sum_i \lambda_i{}^* g_i(\mathrm{x}^*)$$

and every term in the sum is ≤ 0, then we have:

$$f(\mathrm{x}^*) \leq L(\mathrm{x}^*, \lambda^*) \leq f(\mathrm{x}^*)$$

That is:

$$\lambda_i g_i(\mathrm{x}^*) = 0 \qquad (i = 1, \ldots, m)$$

The Dual for Convex Optimisation I

- For convex optimisation, any x^* is a global minimum if and only if x^* satisfies the constraints (i.e. x^* is *feasible*) and there exists $\lambda^* in \Re^m$ s.t.
  1. $\nabla L(\mathrm{x}^*, \lambda^*) = 0$
  2. $\lambda_j{}^* \geq 0$ $(j = 1, \ldots, m)$
  3. $\lambda_j{}^* g_j(\mathrm{x}^* = 0$
- The first two are just the KKT conditions, and the third is the complemnentary slackness condition
- We will focus on minimisation problems where f is convex and the constraints g_i are linear. The Lagrangian is

$$L(\mathrm{x}, \lambda) = f(\mathrm{x}) + \sum_i \lambda_i (\mathrm{a}_i^T + \mathrm{b}_i)$$

The Dual for Convex Optimisation II

- As before, let:

$$g(\lambda) = \min_{\mathrm{x}} L(\mathrm{x}, \lambda)$$

Correctly, *min* should be *inf*, and the function can be $-\infty$ for some x

- Then the dual optimisation problem is:

$$\begin{array}{ll} \text{maximise:} & g(\lambda) \\ \text{subject to:} & \lambda_j \geq 0 \ (j = 1, \ldots, m) \end{array}$$

This problem requires optimisation over elements of $\Re^m$

- Since strong duality holds for convex optimisation, if the primal problem has a solution x^* then the dual problem has a solution λ^*, and $f(\mathrm{x}^*) = g(\lambda^*)$