Mathematical Appendix

Course director: Zoltán Rácz
5329: Inequality, Household Behavior, and the Macroeconomy

This file is updated as the course advances.

1 Differentiation of multivariate functions

You should know the following:

- What are partial derivatives, how to compute them for a given function?
- The vector containing the partial derivatives of f at point x is denoted $\nabla f(x)$ and is called the gradient.
- At a local maximum/minimum, partial derivatives are 0 (First-order conditions). A point where partial derivatives are 0 is called a critical point.
- It can happen, however, that a critical point is not a local extremum. If the function is strictly concave (for maximum) or concave (for minimum) at the point in question, then this anomaly cannot happen.
- If the function of interest is globally concave/convex, then any critical point is a global maximum/minimum.

2 Constrained optimization

In economics, we solve constrained maximization problems with constraints all the time. Constraints can be either equations or can be given by inequalities. We want to characterize the solutions of the following problem:

$$\begin{aligned} \max_x \ & f(x) \qquad (1) \\ \text{such that } g_i(x) &= 0 \quad \forall\, i \in \{1, \dots, m\} \\ h_j(x) &\le 0 \quad \forall\, j \in \{1, \dots, p\} \end{aligned}$$

Here x is an n-dimensional vector and f, the g_is and the h_js are all differentiable functions mapping $\mathbb{R}^n$ to $\mathbb{R}$. We have m equality constraints (encoded by the g functions) and p inequality constraints (encoded by the h functions).

Then if (a) x^* is a local solution of the program (1), and (b) at x^* the gradients of all binding constraint functions form a linearly independent collection of vectors[1], then there exist scalars $\lambda_1, \dots, \lambda_m, \mu_1, \dots, \mu_p$ such that the following set of equations is satisfied (besides the constraints in (1), obviously):

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{j=1}^{p} \mu_j \nabla h_j(x^*) = 0 \tag{2}$$

$$\mu_j h_j(x^*) = 0 \quad \forall\, j \in \{1, \dots, p\} \tag{3}$$

The first equation is a vector equation (remember, gradients have length n), so on the RHS there is an n-long 0 vector. Condition (3) says that if constraint j does not bind (so h_j is not 0), then μ_j has to be 0. This implies that the corresponding term drops out from condition (2). Therefore, non-binding constraints are irrelevant.

Some notes:

- You should think of this as an analog of first-order conditions in the unconstrained case (then this whole system simplifies to $\nabla f(x^*) = 0$ as it should).
- There exist sufficient conditions as well, based on concavity of f and quasi-concavity of constraint functions, but we will never spend time checking them (they always hold in our examples).
- In our examples in this course, condition (b) will always hold (unless you include the same constraint several times accidentally), you can feel free to forget about it.

3 Taylor approximation

Derivatives are useful to provide a local approximation of a differentiable function. A Taylor approximation uses information about the derivatives of a function at a particular point (therefore the approximation works better close to this point). The idea is that if you know what is the slope of a function (aka: derivative) and how the slope changes locally (this info is in higher order derivatives), then you know how the function behaves around the point where you take the derivatives.

[1] This is called the 'Constraint Qualification Condition'

3.1 First-order

A first-order Taylor approximation around a looks like this:

$$f(x) \approx f(a) + f'(x)\Big(x - a\Big)$$

This more or less directly follows from the definition of the derivative. This approximation would be exact for a linear function, but more generally it is an ok approximation only if x is close to a.

3.2 Second-order

A second-order Taylor approximation around a looks like this:

$$f(x) \approx f(a) + f'(x)\Big(x - a\Big) + \frac{f''(x)}{2}\Big(x - a\Big)^2$$

This approximation would be exact for a quadratic function, but more generally it is an ok approximation only if x is close to a.

3.3 Note

One can take higher-order approximations as well, but we won't need them. Also, this idea works for multivariate functions as well (with more complicated-looking notation), but again for this course we don't need that.

4 Geometric Series

In mathematics, a geometric series is the sum of an infinite number of terms that have a constant ratio between successive terms. For example, the series

$$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots$$

is geometric, because each successive term can be obtained by multiplying the previous term by $1/2$. In general, a geometric series is written as $a+ar+ar^2+ar^3+\ldots$, where a is the coefficient of each term and r is the common ratio between adjacent terms. Geometric series in their generator form can be written as

$$\sum_{k=0}^{\infty} ar^k$$

Convergence The convergence of the geometric series depends on the value of the common ratio r. We distinguish two cases:

- If $|r| < 1$ the terms of the series approach zero in the limit and the series converges to $\frac{a}{1-r}$.
- If $|r| \geq 1$, the series diverges.

Sum of the first n+1 terms of a geometric series. For $r \neq 1$, the sum of the first $n+1$ terms of a geometric series, up to and including the r^{n} term, is

$$a + ar + ar^2 + ar^3 + \cdots + ar^n = \sum_{k=0}^{n} ar^k = a\left(\frac{1-r^{n+1}}{1-r}\right)$$

where r is the common ratio. When $r = 1$ we simply have $\sum_{k=0}^{n} ar^k = a(n+1)$

5 Law of iterated expectations

The law of iterated expectations states that for any random variable z and two information sets J, I with $J \subset I$ (I contains more information than J), the following holds:

$$E[E(z|I)|J] = E(z|J). \tag{4}$$

5.1 Example

Imagine you want to compute the expectation of income y_{t+2} given your information in t, which can be summarized by a productivity shock z_t. The productivity shock is stochastic and evolves as a Markov chain. By the law of iterated expectations, we have that:

$$E_t(y_{t+2}) = E_t[E_{t+1}(y_{t+2})]$$

note that $E_t[y_{t+2}]$ can be written as $E_t[y_{t+2}|z_t]$) and that $= E_t[E_{t+1}[y_{t+2}]]$ can be written as $E[E[y_{t+2}|z_{t+1}]|z_t]$.

The reason this works is that you know more at time $t+1$ than at time t.

5.2 Intuition

So what does (4) mean? Let's rewrite (4):

$$0 = E[E(z|I)|J] - E(z|J) = E[E(z|I) - z|J] \tag{5}$$

Conditional expectation is a predictor. $E(z|I)$ is a predictor of z based on information set I. $E(z|I) - z$ is the error of this predictor. (5) says that the expected error of this predictor (conditioned on any coarser information than I) is 0. So first you only have little information J, but you can update your prediction when you get more information I. Today, however, you don't know neither what exactly your prediciton will be when you learn more, nor what will be the final realization of z. What you know now is that the expected error of the new conditional expectation prediction when you learn I is 0. This is what (4) and (5) tell you.

5.3 Implication: Forecast errors are orthogonal to variables known at time of forecast

At least, when we forecast using conditional expectation.

Let x be any variable in the information set I. This means that when you update your prediction knowing I, x will be known (so, will be a constant). However, x might not be part of the information set J. It turns out, x has to have 0 covariance with the forecast error, conditioned on info in J:

$$E[(E(z|I)-z)x|J] = E[(E(z|I)x-zx)|J] = E[(E(zx|I)-zx)|J] = E[zx|J]-E[zx|J] = 0$$