University of Illinois
Fall 2016

Department of Economics
Roger Koenker

Economics 536
Lecture 15

Optimization

Iterative methods of numerical optimization play an important role throughout economics, but are especially crucial in modern econometrics. In this lecture I would like to sketch some basic principles and try to convey some notion what is happening "inside the chip" while you are waiting for the computer to produce a result.

Let's begin by considering a univariate optimization problem:

$$\min_{x \in \Re} f(x)$$

for some nice function $f : \Re \Rightarrow \Re$. We will (gradually) be more specific about what is meant by nice. If we imagine starting at an arbitrary point x_0 and asking: how should move to find a minimum? We are naturally led to consider the derivative. If

$$f'(x_0) > 0$$

the function is increasing so we need to go to the left, if

$$f'(x_0) < 0$$

then the function is decreasing and we need to move to the right.

So far everything is quite obvious, but the next question is: how far should we move in the direction we choose? Imagine for the moment that it is cheap to evaluate $f(\cdot)$, then we could continue in this direction until $f(x)$ stopped decreasing, this would give us a local min, but it would be useful to have some better guidance about step lengths.

This suggests using second derivative information about the curvature of the function f. Suppose we take as a working model for f, the quadratic approximation

$$f_0(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{1}{2}(x - x_0)^2 f''(x_0)$$

then given any starting point x_0 we could pretend that $f_0(x)$ was a reasonable approximation of f and we could find x_1 to minimize $f_0(x)$ by solving the first order conditions,

$$x_1 = x_0 - (f''(x_0))^{-1} f'(x_0).$$

This is called Newton's method. Obviously, if f *is* quadratic, then this method yields the exact solution in one step. Otherwise it may, or may not, perform well.

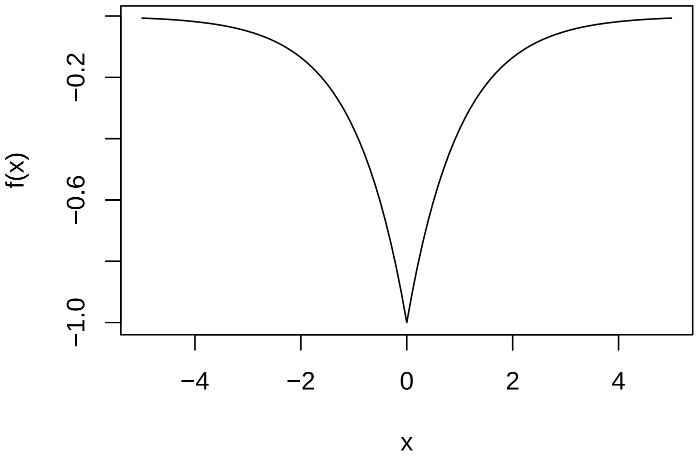

Figure 1: Newton's First Fiasco: Note that second order conditions are violated at the minimum.

Example 1: Suppose $f(x) = -e^{-|x|}$, then

$$f'(x) = e^{-x} \text{ sgn } (x)$$

so it is clear that $f'(x)$ is increasing for $x > 0$ and decreasing for $x < 0$ and thus that $f(x)$ has a unique minimum at $x = 0$. But Newton's method is a disaster. Why? See Figure 1. □

Example 2: Suppose $f(x) = x^6 - x^4 - x^3 - 2x^2 + 4$. Now

$$f'(x) = 6x^5 - 4x^3 - 3x^2 - 4x$$

and now it is no longer obvious what the minimizer is, since there are several roots to the equation $f'(x)$. If we take another derivative

$$f''(x) = 30x^4 - 12x^2 - 6x - 4$$

then we can compute the Newton steps from any starting point, x_0. We have a function that looks like Figure 1.

If we start near one of the local optima, we converge to it, but this may or not be a global minimum. For example, starting at $x = .5$ yields the sequence

$$x_1 = .123$$
$$x_2 = .012$$
$$x_3 = .0001$$

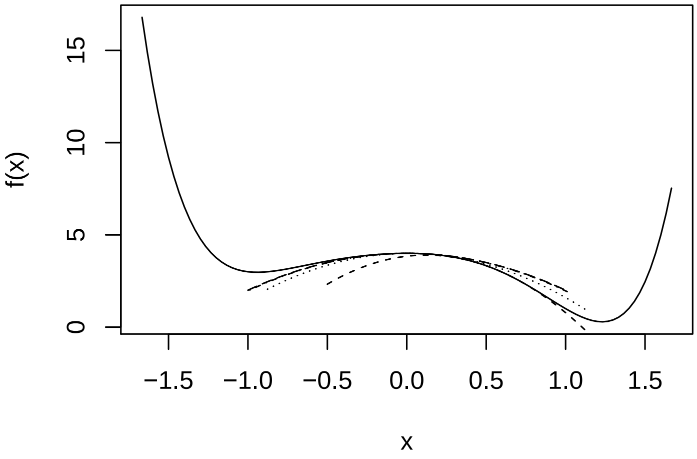

Figure 2: Newton's Second Fiasco: The dashed line is the quadratic approximation of the function at the starting point $x = 0.5$, and the two quadratic approximations at the next two iterates are almost indistinguisable. Note that second order conditions for finding a minimum are violated at the point of convergence– obviously a local maximum.

converging rapidly to zero, but note zero isn't even a local min it is local max! □

In higher dimensions the situation is more challenging, but the principles are quite familiar from the univariate case. We have two ancient strategies, we discuss both.

Cauchy's Method (Steepest Descent) Oddly, the simpler of the two approaches was formalized only in the mid 19th century long after Newton was dead. The idea is based on computing the gradient, or directional derivative of f and moving in the direction of steepest descent. Recall that, by the chain rule we can differentiate the expression

$$\varphi_0(t, u) = f(x_0 + tu)$$

with respect to the scalar t in the direction u, to obtain

$$\varphi_0'(t, u) = \nabla f(x_0 + tu)^\top u$$

so that at the point x_0 we have

$$\varphi_0'(0, u) = \nabla f(x_0)^\top u.$$

This is usually called the directional derivative of f at x_0 in the direction u. Now we look at all possible choices of directions u such that $\| u \|= 1$ and try to find the one that makes the directional derivative smallest. Note that if we are to find a direction of descent, a direction that makes f smaller than it is at x_0 we need to find a direction for which $\varphi_0'(0, u) < 0$, if $\varphi_0'(0, u) \geq 0$ for all u, then we are already at a minimum. As we have seen, this doesn't insure that we have found a global minimum, but at least locally we are at a minimum we can't improve upon x_0 by a *small* move.

Now by the Cauchy-Schwarz inequality[1]

$$- \| \nabla f(x_0) \| \ \| u \| \leq \nabla f(x_0)^\top u \leq \| \nabla f(x_0) \| \ \| u \|$$

Recall that $\| u \|= 1$, so choosing u to make $\varphi_0'(0, u)$ as small as possible means choosing it to hit the lower bound and this occurs when $u = -\nabla f(x_0)/ \| \nabla f(x_0) \|$. To see this, note that the only way that we can get equality is to choose u so that we get $||\nabla f||$ in the middle expression is by letting $u = -\nabla f$ and then normalizing it to have length one. This is called the *direction of steepest descent*

Remark. In the simplex method of linear programming, one follows edges of a polyhedral convex constraint set and at each vertex one chooses the next edge to travel along as the one that is steepest. This works well in certain Portuguese hill towns: at each intersection choose the path of steepest descent. I don't recommend it for hiking in the Himalayas, or even in the Alps.

A curious and not necessarily obvious property of the Cauchy method is that it results in a sequence of steps that are sequentially orthogonal. This is illustrated in the following figure. If we follow the gradient direction we eventually come to a tangency with some level curve and at that point we are no longer going down, then at that point we want to find a new direction and this new direction is necessarily orthogonal to the prior one because of this tangency property. So we end up taking a series of steps each one orthogonal to the previous one.

[1] In statistical jargon this is sometimes called the correlation inequality since it can be expressed as

$$\frac{\text{Cov}(X, Y)^2}{V(X)V(Y)} \leq 1.$$

This is a convenient memory device if nothing else.

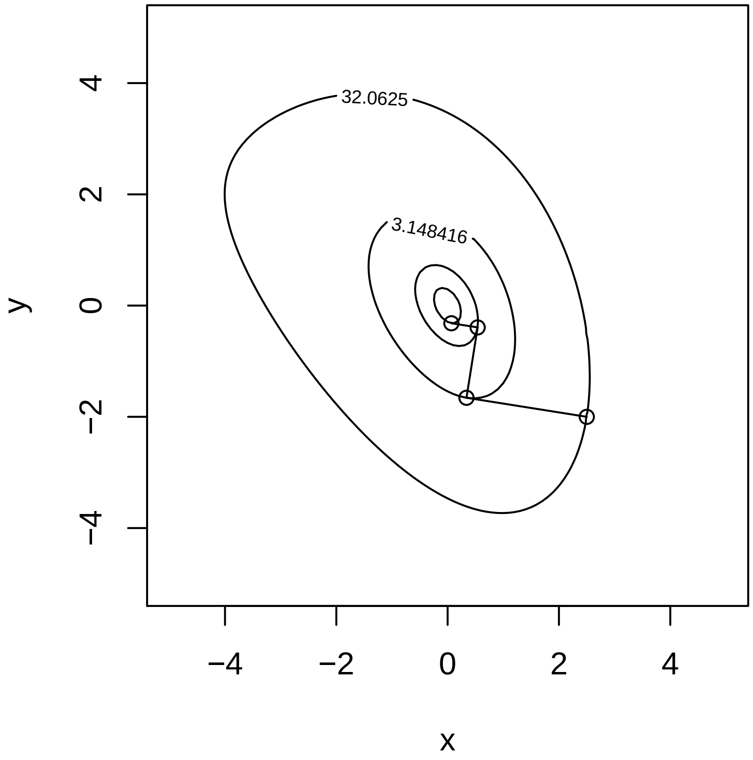

Figure 3: Cauchy's Method: Travel in the direction of steepest descent, or ascent, from the initial point of each iteration. Travel until the step fails to continue to improve things. This puts you (in smooth problems at least) on a new point of tangency from which you move normal to the tangent so successive steps are orthogonal to one another.

Cauchy's method has the virtue that it is always pointing us in a direction that decreases the function. However, it may not always be well-advised to travel as far as the strict version of the method we have described.

Shor's Method. An interesting rather recent development that provides a bridge between steepest decent and Newton's method is Shor's r-algorithm. The basic idea is to make a transformation of coordinates at each iteration that attempts to use some Hessian information generated by the successive gradient directions. I won't try to explain the details, which are elaborated in Kappel and Kuntsevich (2000), but I've illustrated an example that can be compared with the previous Cauchy method plot in the next figure. Shor's algorithm has been found to be particularly successful in problems with non-differentiable objective functions whose gradients exist except on a set of measure zero. Shor's method has a tuning parameter that when set to zero yields Cauchy's method and when set to one yields a version of the conjugate gradient method. In the example we illustrate this parameter has been set to 0.5.

Newton's Method. The foregoing discussion and our experience with the univariate case suggests that we might profit from the curvature information of the function. Again we consider a two term Taylor's series expansion of the function,

$$f_0(x) = f(x_0) + (x - x_0)^\top \nabla f(x_0) + \frac{1}{2}(x - x_0)^\top \nabla^2 f(x_0)(x - x_0).$$

Again, we hope this is a good approximation and consider the step that would minimize $f_0(x)$ as if it were the real function. This leads to the iteration sequence

$$x_1 = x_0 - (\nabla^2 f(x_0))^{-1} \nabla f(x_0).$$

This is really just a matrix analogue of our prior formula. Instead of taking a step in direction $-\nabla f(x_0)$ as suggested by Cauchy, we are modifying the steepest descent by premultiplying by the inverse of the Hessian. And rather than having a step of indeterminant length we are provided with a quite explicit step length.

When the function f is really quadratic, or nearly quadratic, this works quite brilliantly, giving exactly the global minimizer in the strictly quadratic case. But it can also be badly fooled in situations where the quadratic approximation is poor. Good algorithms often begin with gradient steps and gradually adopt Newton steps as confidence in the quadratic model increases. This is sometimes called the "region of trust" method.

Some Special Methods for Statistical Problems

Statistical optimization problems sometimes exhibit special features that can be exploited by more specialized methods. I'll discuss very briefly two of these.

Method of Scoring

Consider a typically maximum likelihood problem

$$\max_{\theta \in \Re^p} \ell(\theta)$$

where $\ell(\cdot)$ denotes the log likelihood. Newton's method yields

$$\theta_1 = \theta_0 - (\nabla^2 \ell(\theta_0))^{-1} \nabla \ell(\theta_0).$$

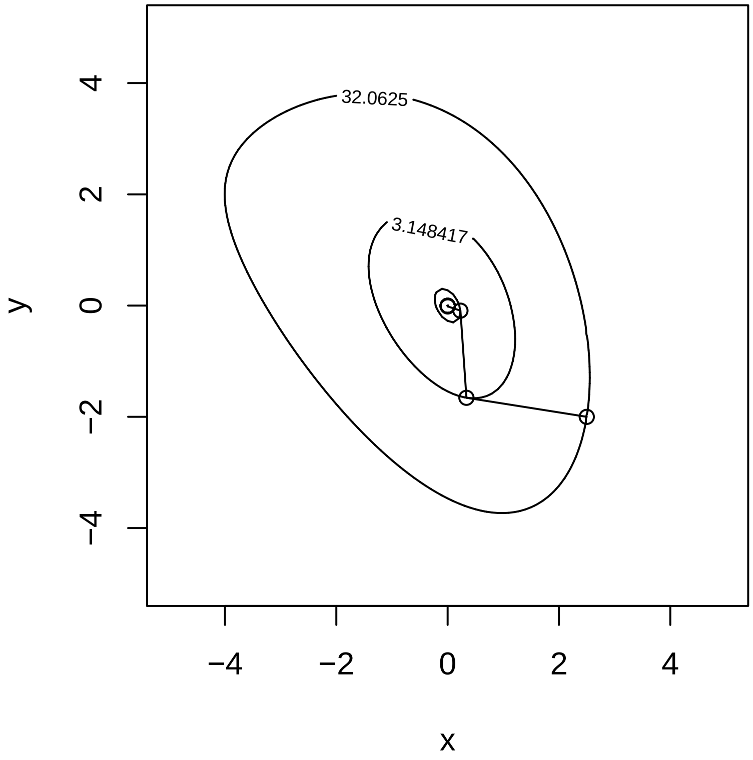

Figure 4: Shor's Method: This is a modified version of Cauchy's method, the first step is pure Cauchy, but subsequent steps use some Hessian like information provided by the successive differences in the gradients. Note that this means that the steps are no longer orthogonal, and the method has somewhat quicker convergence behavior.

Typically, we can write $\ell(\theta)$ as a sum of contributions from n observations

$$\ell(\theta) = \sum_{i=1}^{n} \ell_i(\theta)$$

and consequently we have

$$\nabla \ell(\theta) = \sum_{i=1}^{n} \nabla \ell_i(\theta).$$

When evaluated at $\theta = \theta_0$, and when the model is correctly specified,

$$E\nabla \ell(\theta_0) = 0$$

and furthermore

$$-E\nabla^2 \ell(\theta_0) = E\nabla \ell(\theta_0)\nabla \ell(\theta_0)^\top$$

and this leads to attractive simplifications of the Newton method in which we substitute either this outer product of the gradient, or the expectation of the Hessian in place of the naïve "observed Hessian," an approach called Fisher's method of scoring.

Gauss-Newton Method: for models with Gaussian likelihoods, but which are nonlinear in parameters, for example

$$y_i = g(x_i, \theta) + u_i$$

with $u_i \sim \mathcal{N}(0, \sigma^2)$ we also have considerable special structure. The crucial component of the log likelihood looks like

$$f(\theta) = \sum_{i=1}^{n} (y_i - g(x_i, \theta))^2$$

so

$$\nabla f(\theta) = \nabla g^\top (y - g)$$

where $\nabla g = (\nabla g(x_i, \theta))$ is a $n \times p$ matrix and $y - g$ is the usual residual vector. Another derivative yields

$$\nabla^2 f(\theta) = \nabla^2 g(y - g) - \nabla g^\top \nabla g.$$

If we now do something a bit strange, in effect neglecting the first term of the Hessian, we obtain the Newton step:

$$\theta_1 = \theta_0 + (\nabla g^\top \nabla g)^{-1} \nabla g^\top (y - g)$$

so what is happening? If we interpret the step as a regression, then the matrix ∇g is playing the role of the X matrix, and $y - g$ the role of y. So we are just doing a sequence of least squares regressions of current residuals on a linear approximation of g.

Linear Programming Many interesting optimization problems can be formulated as linear programs, that is as problems of maximizing a linear function subject to linear equality and inequality constraints, for example,

$$\max\{c^\top x \mid Ax = b, x \geq 0\}.$$

The classical approach to solving such problems is a variant of Cauchy's steepest descent (ascent in this case) method. This was developed more or less simultaneously by Dantzig in the U.S. and Kantorovich in the U.S.S.R. But the basic idea of their solution method was nicely described by Edgeworth (1888):

> The method may be illustrated thus:–Let $C - R$ (where C is a constant, [and R denotes the objective function]) represent the height of a surface, which will resemble the roof of an irregularly built slated house. Get on this roof somewhere near the top, and moving continually upwards along some one of the edges, or *arrétes*, climb up to the top. The highest position will in general consist of a solitary pinnacle. But occasionally there will be, instead of a single point, a horizontal ridge, or even a flat surface.

If Edgeworth had been able to translate this vivid geometric description into an explicit algorithm, we would have had linear programming 60 years before Dantzig and Kantorovich. Edgeworth's description is quite an accurate representation of the simplex method. The problem consists in finding a point within a convex polyhedral set that maximizes the linear function $c^\top x$. The point will generally be a vertex of the constraint set $\{x \mid Ax = b, x \geq 0\}$, but as Edgeworth notes it may be an edge or even a higher dimensional surface.

The theory of the simplex algorithm is quite arcane, sufficiently complex that it has spawned many careers. It is remarkably efficient even on very large problems. Indeed, this efficiency was itself an important research problem in the 1970's involving many prominent participants including Gerard Debreu and Steven Smale. But in the early 1980's a new approach began to emerge; the earliest work was independently done in the U.S.S.R., by Khachiyan, and in the U.S. somewhat later by Karmarker. Ironically, again it was eventually discovered that the basic ideas underlying the new approach had been suggested much earlier by Ragnar Frisch. Frisch's basic idea was to replace the inequality constraints of the linear program by a penalty term, somewhat in the spirit of Lagrange. In the simple case with $x \geq 0$, he suggested the penalty $\sum \log x_i$. Starting from any point in the positive orthant we can solve:

$$\max\{c^\top x \mid \lambda^\top Ax + \mu \sum \log x_i\}.$$

Note that as elements of x approach zero the last term imposes a serious cost. But it is clear that optimization of the original problem requires that many elements of x should be driven to zero. Recall the famous Stigler Diet problem.[2] To reconcile this conflict we need to consider algorithms that allow $\mu \to 0$. For fixed values of μ solutions to the Frisch barrier problem lie on what is called the "central path" and as $\mu \to 0$ the central path leads to the optimal vertex solution, or to a point in the convex set-valued solution.

The beauty of this approach is that optimizing the barrier problem for fixed μ is a relatively simple convex optimization problem, with the barrier term ensuring that we have well-defined Newton steps. Combined with a sensible strategy for adjusting μ toward zero, this yields very efficient algorithms even for very large problems. Figures 4 and 5 illustrate the process for a trivial problem in which we start in the middle of a polytope and iterate toward the NE vertex.

Accelerated Gradient Descent Irony abounds in the episodic development of methods of convex optimization. In the last decade or two it has become evident that Newton type methods including the interior point methods described above are impractical in large problems where factorization of

[2] In 1945, well before the Dantzig work became public, George Stigler published a paper on minimal expenditure diets. Using 9 nutrient requirements and 77 foods he constructed a diet of only five foods: wheat flour, evaportated milk, cabbage, spinach and beans satisfying the requirements and costing a mere $39.93 *per year.* With improved, but certainly more arduous methods Dantzig and coworkers improved this by a whopping 24 cents somewhat later. In 2005 dollars the cost of Stigler's diet is roughly $500.

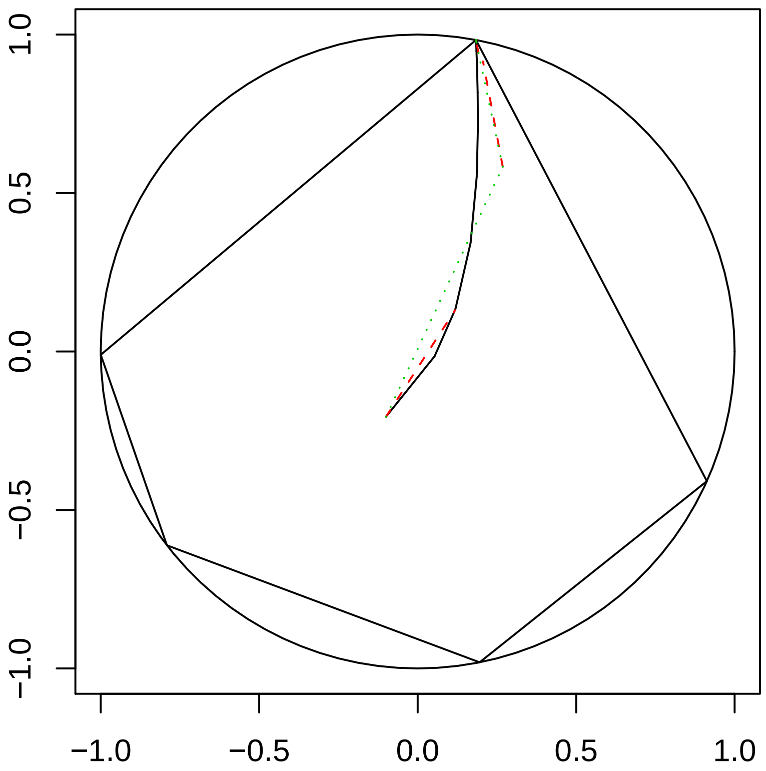

Figure 5: A Simple Example of Interior Point Methods for Linear Programming: The figure illustrates a random pentagon of which we would like to find the most northeast vertex. The central path beginning with an equal weighting of the 5 extreme points of the polygon is shown as the solid curved line. The dotted line emanating from the this center is the first affine scaling step. The dashed line is the modified Newton direction computed according to the proposal of Mehrotra. Subsequent iterations are unfortunately obscured by the scale of the figure.

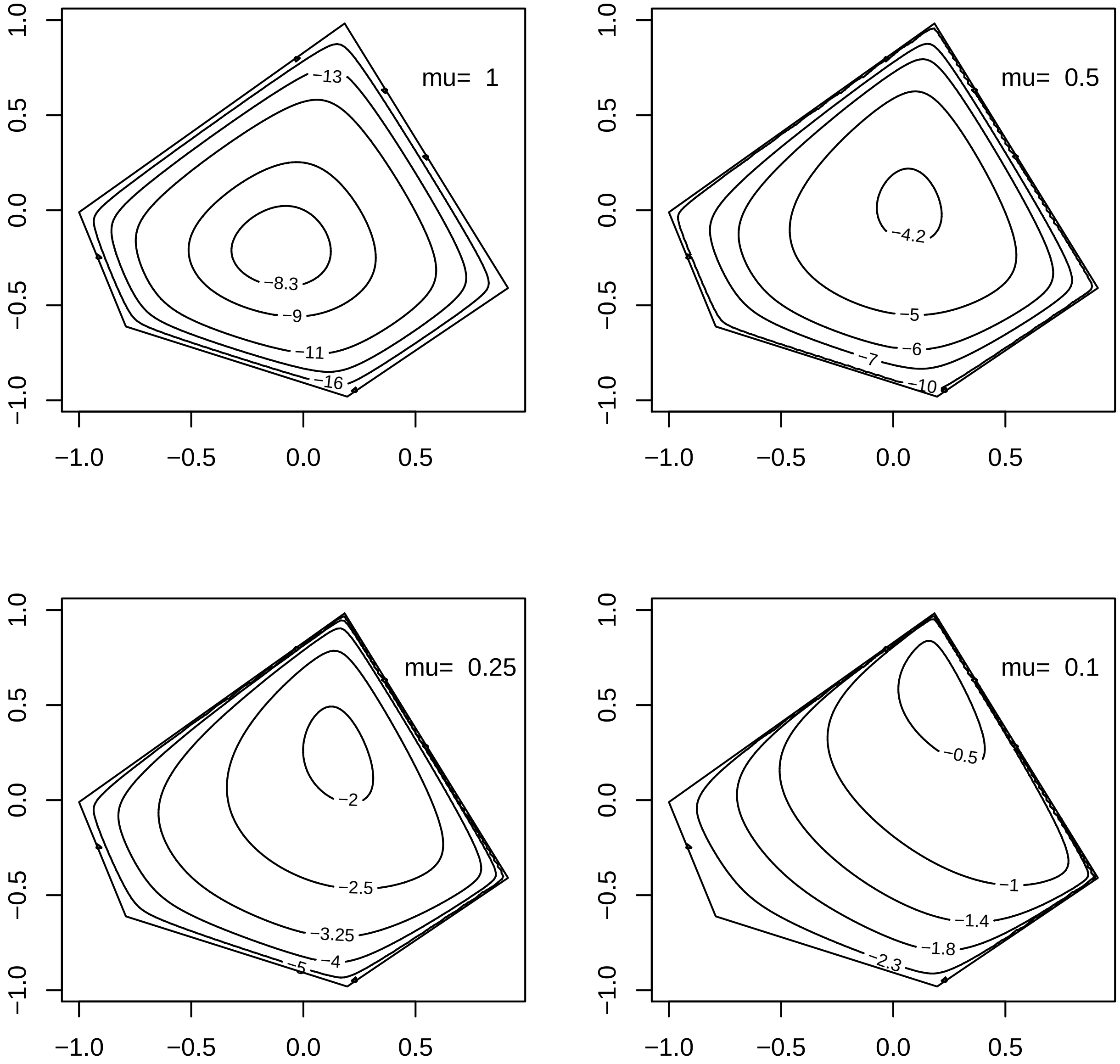

Figure 6: Contours of the Log Barrier Objective Function for the Simple Polygonal Linear Program: The figure illustrates four different contour plots of the log barrier objective function corresponding to four different choices of μ. In the first panel, $\mu = 1$ and the contours are centered in the polygon. As μ is reduced the penalized objective function is less influenced by the penalty term and more strongly influenced by the linear component of the original LP formulation of the problem. Thus, for $\mu = .1$ we find that the unconstrained maximum of the log barrier function occurs quite close to the optimal vertex of the original LP. The locus of solutions to the log barrier problems for various μ's is called the central path, and is illustrated in Figure by the solid curved line.

Hessians is slow at best and impossible at worst when memory constraints are binding. So researchers have returned to gradient descent methods where only linear operations are needed and can be easily parallelized. Again, drammatic innovations were hidden in plain sight in the Soviet literature. Nesterov[3] (1983) provided a crucial insight that has led to an outpouring of new work on accelerated gradient descent (AGD) that is now ubiquitous in machine learning and related fields.

Recall that the fundemental flaw in the classical gradient descent method is its tendency to take orthogonal steps. Nesterov proposed a modified gradient step that substantially speeds things up. Classical gradient descent has a convergence rate of $\mathcal{O}(1/k)$ after k steps, Nesterov was able to show that his AGD achieved a rate of $\mathcal{O}(1/\sqrt{k})$. Experts seem to agree that the intuition behind this is rather murky even after more than 30 years. This was evident in Michael Jordan's talk recent talk at UIUC in September, 2016, which embedded AGD into a continuous time, differential equations framework.

I won't attempt to describe the algorithm any further, although code for the following figure will be available on the class website. Instead I'll contrast performance in an extremely simple problem where gradient descent in its original form already encounters difficulties. In Figure 7 we see contours of a quadratic function that has a minimum at the origin. Classical gradient descent takes the jagged, grey path from its initial point $(-0.75, 0.15)$ and after 100 iterations is about 2/3 of the way to its final objective. In contrast after 42 iterations, AGD has achieved the optimum with almost no perceptible oscillation. Of course one may say that we could have done this in one step with Mr. Newton's method: to which one can only respond, yes, but try that in dimension 10^8. One might also object that the earliest versions of AGD only apply to strongly convex functions, and thus not to the typical nonsmooth problems of modern machine learning, but this too has been extensively considered and accelerated proximal gradient methods now power many important applications.

Stochastic Approximation For nice functions Newton's method has good convergence properties, however we sometimes face more challenging problems in which objective is non-smooth. An important class of practical problems arises when the objective function involves a stochastic component. A simple example of this general class is the monopoly problem of psM. The monopolist faces an unknown, stochastic demand function, he would like to maximize expected profits by chosing a sequence of prices, but observed profits is a highly non-smooth function of price. Given a stationary demand environment, we may posit a smooth expected profit function,

$$\mu_0(p) = \mathbb{E}\pi(p) \equiv \mathbb{E}[pX(p) - C(X(p))].$$

Kiefer and Wolfowitz (1952), extending the stochastic approximation methods introduced by Robbins and Monro (1951), proposed finding a maximizer $p^* = \text{argmax}\mu_0(p)$, by iterations of the form,

$$p_{t+1} = p_t + a_t(\pi(p_t + c_t) - \pi(p_t - c_t))/c_t,$$

where $\{a_t\}$ and $\{c_t\}$ denote deterministic sequences satisfying the conditions, $c_t \to 0$, $a_t \to 0$, $\sum a_t \to \infty$ and $\sum a_t^2/c_t^2 < \infty$. In the R package `monopoly` accompanying the problem set, they are set by default to $a_t = a_0 n^{-1}$ and $c_t = c_0 n^{-1/6}$. As noted by Ruppert (1991), these choices achieve a certain optimality and yield a limiting normal theory for the sequence $\{p_t\}$:

$$n^{1/3}(p_t - p^*) \rightsquigarrow \mathcal{N}(\beta(p^*), \sigma^2(p^*))$$

[3] Nesterov also made enormous contributions to the early theory behind interior point methods.

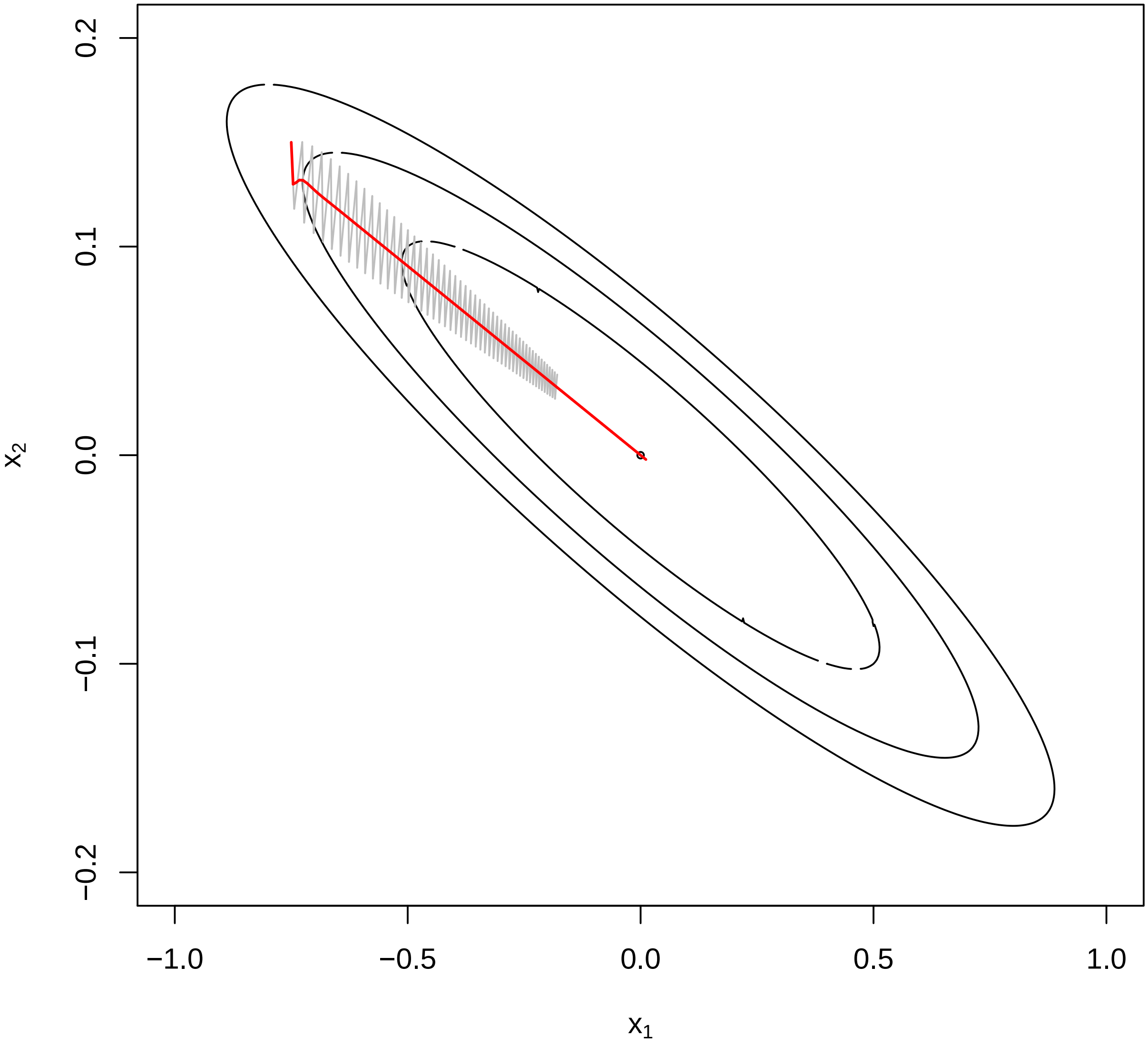

Figure 7: Elliptical contours depict a somewhat ill-conditioned quadratic with minimum at the origin. The jagged, grey path represents the classical gradient descent after 100 iterations. The heavier (red) solid line is the path of the AGD iterations after 42 iterations, almost no oscillation and much more rapid progress toward the objective. Full disclosure: the oscillation of the classical gradient decent path has been exaggerated somewhat as revealed in the code for the figure on the class website.

where

$$\beta(p) = \frac{-a_0 c_0^2 \mu_0^{'''}(p)}{3(2\mu_0^{''}(p) - 2/3)}$$

and

$$\sigma(p) = \frac{a_0^2 \sigma^2(p)}{2c_0^2(2a_0\mu_0^{''}(p) - 2/3)}.$$

denote the asymptotic bias and variance of the sequence respectively. More recent developments have extended the approach to multivariate settings and, under more stringent smoothness conditions improved the rate of convergence.

A curious feature of all this in the monopoly setting is that even with the original $\mathcal{O}(n^{-1/3})$ convergence rate to the profit maximizing value, the convergence is too rapid to enable the monopolist to consistently estimate the parameters of a simple quadratic demand curve. It should be noted that the usual approach to such problems in economics is to formulate them as dynamic programming problems with Bayesian updating of the demand parameters in each period, but this is computationally much more complicated.

References

Nazareth, J.L. (1994) *The Newton-Cauchy Framework: A Unified Approach to Unconstrained Nonlinear Minimization*, Springer.

Kappel, F. and A. V. Kuntsevich, (2000) An Implementation of Shors r-Algorithm, *Computational Optimization and Applications*, 15, 193205.

Kiefer, J., and J. Wolfowitz (1952): "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, pp. 462–466.

Nesterov, Y.E. (1983) "A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/\sqrt{k})$, Dokl. Akad. Nauk SSR, 269, 543-47 (in Russian).

Robbins, H., and S. Monro (1951): "A stochastic approximation method," *The Annals of Mathematical Statistics*, pp. 400–407.

Stigler, G. (1945): "The Cost of Subsistence," *J. of Farm Economics*, 27, 303-14.

Ruppert, D. (1991): "Stochastic approximation," in *Handbook in Sequential Analysis*, ed. by B. Ghosh, and P. Sen, pp. 503–529. Dekker: New York.