Old, New, Borrowed, and Blue in the Marriage of Statistics and Optimization

Margaret H. Wright

Computer Science Department

Courant Institute of Mathematical Sciences

New York University

SAMSI Workshop on Statistics and Optimization

February 8, 2017

I'm delighted and honored to have been invited to give this lecture.

Thank you very much for the invitation!

Nothing will be said about "big data".

Many apologies in advance for my omission of important references!

Title of the talk? In North Carolina today, *assuming one can find the right judge*, statistics and optimization can celebrate their marriage with a Victorian English tradition.

Each participant would bring

- something old to honor past contributions;
- something new to signal openness to novel ideas;
- something borrowed to welcome other people and their ideas.

And they would share

- one BLUE, the best linear unbiased estimator.

The content of this talk, conceived in advance: a close look at revealing instances in the joint history of statistics and optimization.

The actual content, produced less than 24 hours ago: a superficial look at a very small number of these instances, related in some way to optimization, numerical analysis, and software development.

To set the stage: let's pay homage to Gauss, who fits in every category, and a great 1995 paper by G. W. Stewart, "Gauss, statistics, and Gaussian elimination" in the *Journal of Computational and Graphical Statistics*.

From that paper:

> *Gauss [in the early 1800s] introduced Gaussian elimination as a mathematical tool to get at the precision of least-squares estimates. . . . Gaussian elimination was not conceived as a general numerical algorithm with applications in statistics and least squares. Rather it was a procedure that sprang from the interface of statistics and computation.*

In this spirit, let's think about connected aspects of statistics and numerical analysis [aka scientific computing], especially linear algebra and optimization.

One instance that comes from both statistics and optimization: regularization.

It entered optimization via a 1943 paper by Richard Courant, "Variational methods for the solution of problems in equilibrium and vibrations". (The word "penalty" does not appear in his paper.)

His idea is that, rather than trying to satisfy a "hard" boundary condition, one poses instead a sequence of problems with "mild" constraints, each of which is a relaxation of the hard conditions, with eventual satisfaction of the desired constraint.

Attention is paid throughout the Courant paper to quadratic functionals, and he emphasizes that

> . . . *rigid constraints are only idealized limiting cases of very large restoring forces*.

Translating Courant's ideas into optimization, consider an equality-constrained problem:

$$\text{minimize} \quad f(x) \ \text{ subject to } \ c_i(x) = 0, i = 1 : m.$$

If we think this problem is too difficult, one strategy is to define a nonnegative *penalty function* whose magnitude reflects the size of the constraint functions and that vanishes when the constraints are satisfied. In particular, we know from Courant that a quadratic function will work:

$$P_Q(x) = \|c(x)\|_2^2.$$

Given a sequence of positive penalty parameters $\{\rho_k\}$, $k = 0,\ldots$, and assuming that unconstrained optimization is "easy", we can then solve a sequence of unconstrained problems:

$$\text{minimize} \quad f(x) + \rho_k P_Q(x), \ \text{ with } \ \rho_k \to \infty.$$

Under relatively mild conditions, the minimizers of the penalty function converge to a minimizer of the constrained problem, just as Courant had observed about rigid constraints being limiting cases.

It is straightforward to see that the Hessian of the penalty function is

$$\nabla^2 P_Q(x, \rho) = W(x, \lambda) + \rho J(x)^T J(x),$$

where W is an approximation to the Hessian of the Lagrangian function, λ is a Lagrange multiplier estimate, and J is the Jacobian of c. Hence the structure of the penalty Hessian is an "innocent" matrix W plus, as $\rho \to \infty$, an increasingly large multiple of a matrix whose form is $J^T J$.

But let's get back to Courant—what happened next?

The famous 1968 book by Fiacco and McCormick, *Nonlinear programming: sequential unconstrained minimization techniques*, surveyed the then-state of the art in what they called "interior-point" methods (revived in the 1980s) and "exterior-point" methods.

Perhaps surprisingly, no papers in optimization appeared to pursue Courant's ideas rigorously for more than a decade. However, various versions of the quadratic penalty function were proposed during the 1950s, and the software package `SUMT` (sequential unconstrained minimization technique) was widely used in the late 1960s.

Because of concerns about the quadratic penalty function (the infinite penalty parameter and ill-conditioning), other penalty functions were proposed.

Zangwill suggested the ℓ_1 penalty function $\|c(x)\|_1$ in 1967, and Pietrzykowski (1970) showed that there is a finite value of ρ such that the minimizer of $f(x) + \rho\|c(x)\|_1$ is a minimizer of the original problem. However, using the ℓ_1 penalty function is more complicated because of difficulties with software that assumes smoothness.

And what about trust region methods?

Their history is interesting because the paper usually cited as the origin of trust-region methods is by Levenberg (1944), who proposed adding a positive multiple of the identity to the Hessian of a nonlinear least-squares problem to improve stability.

The next citation about trust region methods is to a 1963 paper by Marquardt, SIAM *Journal on Applied Mathematics*, "An algorithm for least-squares estimation of nonlinear parameters", which does not mention Levenberg. Marquardt describes his result as developing a "maximum neighborhood" that interpolates between a Taylor-series model and the steepest descent method. Marquardt seems to reveal himself as a statistician in a 1970 paper called "Generalized inverses, ridge regression, biased linear estimation, and nonlinear estimation", but does not use the word "regularization".

In *unconstrained* minimization of the nonlinear function $f(x)$, it is typical to choose the search direction to minimize a certain model of f, most often quadratic (as in Newton's method). But what should be done if the quadratic model is flawed, e.g., does not represent f well or has an indefinite Hessian?

The idea is that, at the current iterate x_k, the local model is trustworthy only nearby, and the search direction p is chosen by minimizing the model subject to a size constraint:

$$\text{minimize} \quad Q(p) \ \text{ subject to } \ \|p\| \leq \delta,$$

where Q is a quadratic model of f with Hessian H and gradient g.

The bound on $\|p\|$ defines the trust region. With the two-norm, p is a solution of a linear system with the following form for some nonnegative λ:

$$(H + \lambda I)p = -g$$

and g is the gradient of f, so that the model Hessian is perturbed by a positive multiple of the identity. The idea of a trust region method is to produce good steps to the minimizer. If a really good model were known, in some sense there would be no need for a trust region and λ could be zero.

The two most common choices of norm are the two-norm and the infinity norm, where the latter imposes bounds on the size of each component of p.

How about regularization from the statistics side? The ordinary least-squares estimate is the minimizer of $\|y - X\beta\|_2^2$, but this linear regression has shortcomings.

The so-called ridge regression coefficients β are the solution of

$$\min \|y - X\beta\|_2 + \lambda\|\beta\|_2^2,$$

where $\lambda \geq 0$ is often called a tuning parameter. The value of λ reflects a balancing between variance and bias; we may want to decrease the variance, by paying a penalty in increasing the bias.

Many ideas labeled as regularization have been suggested with the idea of adding a penalty to the least-squares problem (or a variation):

$$\text{minimize} \quad \|y -_x \beta\| + \lambda P(\beta),$$

where $\lambda > 0$ and $P(\beta)$ might be $\|\beta\|_2^2$ or $\|\beta\|_1$ or some other combination.

The name most commonly associated with the introduction of regularization is Tikhonov, whose famous paper on the subject appeared in Russian in 1943, and is titled (in English) "On the stability of inverse problems".

Tikhonov introduced the concept of regularization in the context of solving integral equations. Rather than solving the operator equation $Af = g$, which is ill-posed or very ill-conditioned, one instead minimizes $\|Af - g\|_2^2 + \gamma W(f)$, where $\gamma > 0$ is a regularization parameter and $W(f)$ is a smoothness penalty such as $\int [f'(x)]^2 dx$.

FYI: Wikipedia says that Tikhonov regularization is known in the statistical literature as ridge regression.

How to compute the best λ? Generalized cross validation (Golub, Heath, Wahba; implemented in R and Matlab).

And then there's the lasso, which uses an ℓ_1 penalty.

Thousands of references and (varying) explanations.

> *Loosely, regularization is the class of methods needed to modify maximum likelihood to give reasonable answers in unstable situations.*
>
> Bickel and Li (2006)

Why have I gone on and on about these topics? Because they are closely related but also very different, and thus illustrate the growing (and occasionally confusing) connections between statistics and optimization.

The connections are very clearly illustrated in a recent paper by Higham, Strabić, and Šego, "Restoring definiteness via shrinking, with an application to correlation matrices with a fixed block", *SIAM Review* (2016).

The paper addresses the situation when a matrix should, in theory, be positive semidefinite, but is not—for example, because of missing data or inaccuracies in computing the matrix. In the examples of interest, it is not feasible to use an indefinite matrix, and a "repair" is made by developing a "replacement" matrix.

The strategy is based on the idea of "shrinking", by forming a convex linear combination of two matrices. One of the matrices is indefinite, which is untypical in statistics but often true in optimization. In addition, the authors make no statistical assumptions about either of the matrices, so that their technique may fit well into applications in optimization.

A treat (for a numerical analyst) is that three proposed methods are given, based on ingredients we know and love: bisection, Newton's method, and solving a symmetric definite generalized eigenvalue problem. The test for positive definiteness using bisection uses the Cholesky factorization.

And the icing on the cake: software implementing these methods is available in the 2015 NAG library, and Matlab and Python versions are available from `github`.

A different topic: Derivative-free optimization: a thought sink for me for several years.

The context is minimizing $f(x)$, where no derivatives of f are available. Furthermore,

1. calculation of f is very expensive or time-consuming (e.g., involves collection of data in real time);
2. f is unpredictably non-nice (may have discontinuous derivatives or discontinuities or singularities);
3. f may include low-accuracy measured data;
4. calculating f may include running a complicated simulation of "black box" calculation whose details are not available to the optimization software.

Because f is very expensive to compute, the methods of greatest interest should not calculate f very many times.

This property effectively rules out classes of methods that typically require many function evaluations, in particular nature-based methods such as genetic methods, swarm methods, simulated annealing, etc.

Derivative-free methods have an interesting history:

1. In the 1950s and 1960s, papers about optimization methods were published in good journals with no proofs at all, usually with reports on performance on a few problems;

2. Once "mathematization" started, such papers were not highly regarded—in fact, they were seen by some as embarrassing. Theory was essential;

3. Matters have loosened up somewhat, but work in optimization without theory is, for many (probably most) optimization researchers, unacceptable. (But some engineering journals still welcome this kind of work.)

Here is an instance of the first point: The Nelder–Mead "simplex" method (1965) [not to be confused with the simplex method for linear programming] was developed by John Nelder and Roger Mead, two UK statisticians who worked then at the National Vegetable Research Station Wellesbourne, Warwick.

The NM method was inspired by a 1962 method of Spendley, Hext, and Himsworth (SHH), "Sequential application of simplex designs in optimisation and evolutionary operation", published in *Technometrics*, which proposed forming and moving a regular simplex in n dimensions. The SHH method was intended to be used in empirical process optimization.

[An aside: the SHH paper is fascinating.]

Like SHH, the NM method develops a simplex that moves around, but it does not retain the same shape. The NM paper states that their simplex "adapts itself to the local landscape, elongating down long inclined planes, changing direction on encountering a valley at an angle, and contracting in the neighborhood of a minimum".

The NM method was published in the *Computer Journal*, the same journal that published the 1963 Fletcher–Powell paper on quasi-Newton methods. It contains no proofs, several numerical examples, and very informal commentary on how the method performs.

NM became, almost immediately, wildly popular with practitioners who wanted to optimize but did not want to write code to evaluate derivatives of their functions. It was by far the most popular optimization routine in the Numerical Algorithms Group software library in the 1970s and 1980s.

Today (February 2016) the original NM paper has been cited 24,228 times, this number is still growing—and the method is being used!

To confirm this, consider the 5,138 (and growing) citations for a 1998 paper that derives theory for the NM method in low dimensions, and which is cited by the MathWorks documentation for routine `fminsearch`.

Although NM is extremely simple to grasp, it is very difficult to analyze, and there is a 2-d counterexample in which it fails to converge to the minimizer. For this reason alone, some respected optimization researchers have explicitly advised against using it.

In addition, Kelley (1999) noted that "Unlike . . . [direct search algorithms with provable convergence], the Nelder–Mead algorithm can stagnate and converge to a nonoptimal point". He accordingly proposes strategies for detecting and remediating this stagnation.

So why would anyone use NM, especially when convergence proofs are known for a large number of derivative-free methods and there is good software available for those methods?

Convergence proofs are reassuring, but they are (of course) mathematical. Even with provable convergence, certain direct search algorithms can "stagnate in practice" in the sense of making very little progress for a long time!

When a method is in a *resolution ridge* when maximizing (a resolution valley when minimizing), extremely slow progress may result. This property was described in the 1960s by Wilde (1964) and Brent (1973).

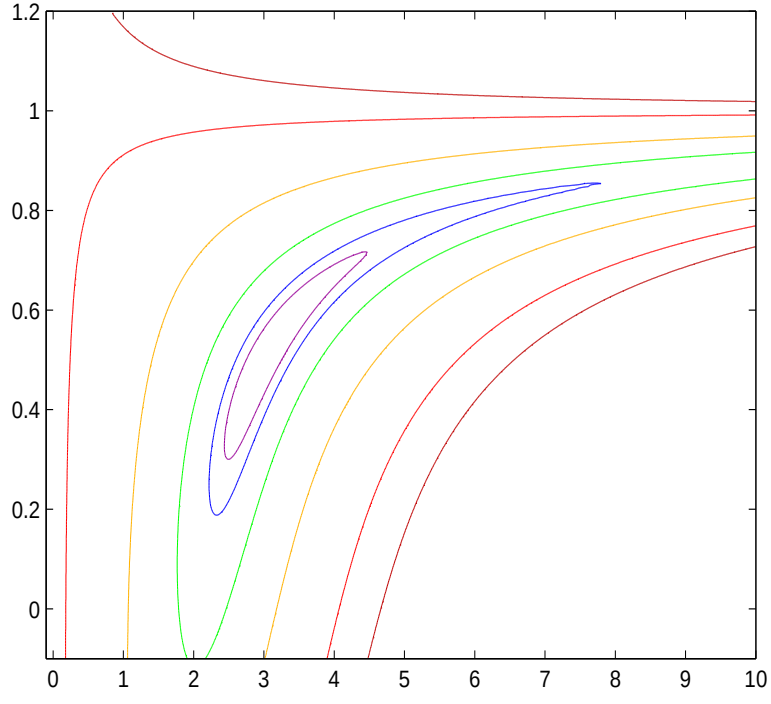
Let δ_i be a "resolution" in variable i. Then x is *in a resolution valley* if

$$f(x) < f(x - \delta_i e_i) \quad \text{and} \quad f(x) < f(x + \delta_i e_i),$$

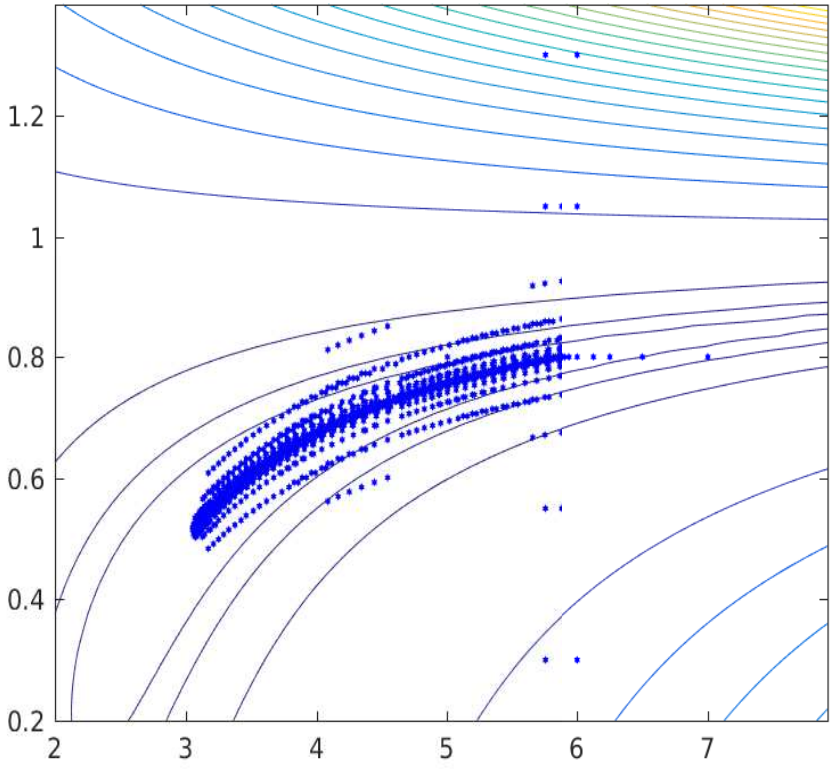
where e_i is the ith coordinate vector.

The implication of this property is that the optimization method "thinks that" x has the locally smallest value of f even though x is far from the real minimizer. Of course, there is no guaranteed way to check whether this property holds in general.

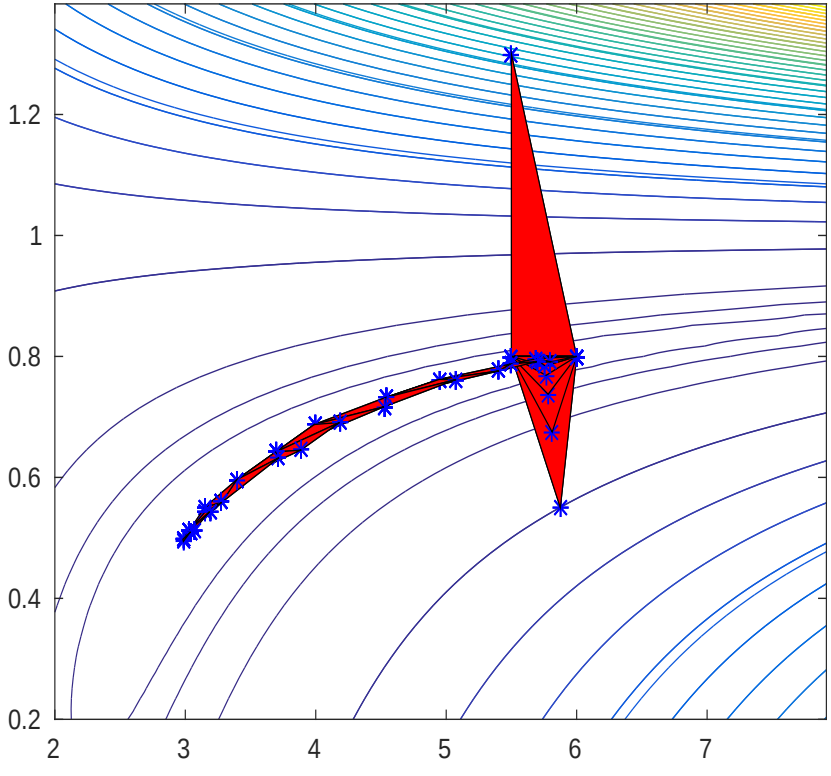
Here is a function containing a curving resolution valley.



Here are the function evaluations in a direct search method applied to that function:



For comparison, are the NM simplices (each vertex represents a function evaluation), with the same starting point.



It might be argued that the only interesting optimization problems today involve "big data", and certainly these pose lively, challenging, and absorbing challenges for optimization.

But, in the view of this speaker, many interesting and challenging problems do not have those properties.

A remarkable number of papers inspired by medicine turn out to use a selection of MathWorks software for derivative-free optimization.

A tiny sample:

[personal anecdote] D. Schreck and R. Fishberg (2013), Diagnostic accuracy of a new cardiac electrical biomarker for detection of electrocardiagram changes suggestive of myocardial ischemic injury, *Annals of Noninvasive Electrocardiology* 19, 129–144. (508 ECGs.)

D. Calzolari, S. Bruschi, L. Coquin et al. (2008), Search algorithms as a framework for the optimization of drug combinations, *Public Library of Science, Computational Biology* 4, 1–14.

A. Zimmer, I. Katzir, et al. (2016), Prediction of multidimensional drug dose responses based on measurements of drug pairs, *Proceedings of the National Academy of Sciences*.

These problems are, to put it mildly, fascinating, and some optimization researchers want to help solve them.

Help is needed from statistics in both modeling and in software development!

"Statistics" here does not mean performance and data profiles about the relative performance of methods on mostly academic test problems.

The effectiveness of derivative-free methods depends to a large extent on specific properties of the relevant functions that can be extracted and analyzed using statistical methods.

Learning about these properties will use function evaluations, but pay off in the long run.

Many ideas and resources are already available; here is a tiny selection.

- Adaptively rescaling the mesh for improved efficiency. Audet, LeDigabel, and Tribes, "Dynamic scaling in the mesh adaptive direct search algorithm for blackbox optimization". *Optimization and Engineering* (2016).
- Using regularization to estimate intervals for numerical differentiation. Cullum, "Numerical differentiation and regularization", SIAM *Journal on Numerical Analysis* (1971)
- Knowles and Renka, "Methods for numerical differentiation of noisy data", *Electronic journal of differential equations* (2014).

- Fitting response surfaces with stochastic processes. Jones, Schonlau, and Welch, "Efficient global optimization of expensive black-box functions", *Journal of Global Optimization* (1998).
- Sacks, Welch, Mitchell, and Wynn, "Design and analysis of computer experiments", *Statistical Science* (1989). Software available in R and Matlab.
- Adjengue, Audet, Ben Yahia, "A variance-based method to rank input variables of the Mesh Adaptive Direct Search algorithm", *Optimization Letters* (2014).
- Wahba, Link to `stat.wisc.edu` software, University of Wisconsin.

- `cnoise`, Matlab scripts that estimate the accuracy of a user-provided function as well as computational noise; Moré and Wild (2011).
- Analysis and characterization of sensitivities; sIPOPT, Pirnay, López-Negrete, Biegler (2012).
- `Convex`, a convex optimization modeling framework in Julia; Udell et al. (2014).
- `Jump`, a language for mathematical optimization; Dunning et al. (2015).

What's the big message about connections between statistics and optimization? To optimize impact and sample size, produce great theory and available code.

Donald Knuth (2002):

> *My main conclusion after spending ten years of my life working on the TeX project is that software is hard. It's harder than anything else I've ever had to do.*
>
> *The creation of good software demands a significantly higher standard of accuracy than [proving theorems or writing books], and it requires a longer attention span than other intellectual tasks.*

Knuth is right, and we are very lucky that dedicated people have produced and maintained and expanded R and Matlab just for us.


...and they lived
happily ever after.
choosinglove.net