Editor: Dianne P. O'Leary, oleary@cs.umd.edu

BLIND DECONVOLUTION: ERRORS, ERRORS EVERYWHERE

By Dianne P. O'Leary

THIS HOMEWORK ASSIGNMENT FOCUSES ON A CLASS OF METHODS THAT ACCOUNTS FOR UNCERTAINTY IN THE MODEL AS WELL AS THE DATA. OUR EXAMPLE CONCERNS *spectroscopy*—the attempt to reconstruct a true spectrum from an observed one. The problem we're considering is sometimes called *blind deconvolution*, because we're trying to unravel not only the spectrum, but the function that caused the blurring. These problems also arise in image deblurring.

Spectroscopy

Consider the data in Figure 1, which represents counts measured by a spectrometer. Suppose we have particles whose energy ranges from e_{low} to e_{high} and define some intermediate energy levels $e_{low} = e_0 < e_1 < \ldots < e_{n_b-1} < e_{n_b} = e_{high}$. This creates n_b bins, where the count for the jth bin is the number of particles determined to have energies between e_{j-1} and e_j. Our spectrometer records n_b counts, one for each bin; in the figure, we've passed a curve through these counts.

Ideally, the count in bin j is exactly the number of particles with energies in the range $[e_{j-1}, e_j]$. But some blurring occurs due to the measurement process, and a particle in that energy range might instead be included in the count for a different nearby bin. The probability that a particle with energy e is assigned to bin j is often modeled as a normal distribution with mean $(e_j + e_{j-1})/2$ and variance s_j^2.

One way to model this system is to try to determine the correct counts f_j and the correct blurring given the measured counts $g_j, j = 1, \ldots, n_b$ and estimates of the values s_j.

This gives us a matrix equation $(\boldsymbol{K} + \boldsymbol{E})\,\mathbf{f} \approx \mathbf{g} + \mathbf{r}$, where $\boldsymbol{E}$ accounts for errors in modeling the spectrometer's blur, and $\mathbf{r}$ accounts for errors in counts. The matrix entry $k_{j\ell}$ is computed as the probability that a particle whose energy is in the interval $[e_{\ell-1}, e_\ell]$ is assigned to bin j ($j, \ell = 1, \ldots, n_b$).

There are several sources of differences between the true spectrum and the recorded spectrum:

- We effectively assign energy $(e_j + e_{j-1})/2$ to all particles in bin j, which isn't correct.
- A count's value depends on the number of particles with the energies that it represents, but there is some smearing, so it also depends on the number of particles with nearby energies.
- The values of the counts often have some error because they're finite-precision representations rather than the infinite-precision "real ones." In our data, the counter is incremented by 0.1 for every 100 particles detected.

Therefore, if we model our problem by $\boldsymbol{K}\mathbf{f} \approx \mathbf{g}$, the solution to this problem doesn't give us the true spectrum; we have uncertainty in $\boldsymbol{K}$ as well as in $\mathbf{g}$! In this homework, we'll investigate the effects of such error and some ways to cope with it.

Models

We could assume that the main error is in our estimates of the s_j values and try to estimate both the correct values of the variances and the correct counts. We'll take a different approach that is appropriate even when the probabilities aren't exactly normal and need a correction applied to them. Our approach is useful whenever both $\boldsymbol{K}$ and $\mathbf{g}$ have errors whose relative variance is known or can be estimated. We'll experiment with two models:

- *Model 1, least squares*. Most of the error is in $\mathbf{g}$ (so $\boldsymbol{E}$ is assumed to be zero).
- *Model 2, total least squares*. There is significant error in both $\mathbf{g}$ and $\boldsymbol{K}$.

We've used Model 1 in a previous homework (see the "Tools" sidebar). As in that problem, the matrix $\boldsymbol{K}$ $(m \times n)$ can be quite ill-conditioned, so small changes in the measured counts $\mathbf{g}$ make large changes in the resulting $\mathbf{f}$. Therefore, as in that homework, we'll add regularization to make the problem better conditioned, so we focus on the truncated singular value decomposition (SVD) as a tool for doing this.

Our major algorithmic task is to figure out how to solve

these problems without much computational effort. We don't want to have to start over again if we change our mind about how many singular values to drop. In fact, we want to compute an SVD only once.

Let's consider each case in turn.

Model 1: Least Squares ($E = 0$).

Define $\boldsymbol{K} = \boldsymbol{U}\Sigma\boldsymbol{V}^T$, where

- $\boldsymbol{U}$ has dimension $m \times m$ and $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$, the identity matrix.
- Σ has dimension $m \times n$, the only nonzeros are on the main diagonal, and they are nonnegative real numbers $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$.
- $\boldsymbol{V}$ has dimension $n \times n$ and $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$.

In Model 1, we determine $\mathbf{f}$ by solving the least squares problem

$$\min_{\mathbf{f}} \left\| \boldsymbol{K}_n \mathbf{f} - \mathbf{g} \right\|_2,$$

where

$$\boldsymbol{K}_n = \boldsymbol{U} \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{V}^T,$$

the dimensions of the zero columns are $m \times (n - \tilde{n})$, and $\tilde{n} \leq n$ is the number of singular values retained, so $\Sigma_{\tilde{n}}$ is a diagonal matrix with entries $\sigma_1, \ldots, \sigma_{\tilde{n}}$. The solution to the problem isn't unique if $\tilde{n} < n$, but the solution of minimal norm is found by taking

$$\mathbf{f}_n = \boldsymbol{V} \begin{bmatrix} \Sigma_n^{-1} & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{U}^T \mathbf{g}.$$

(Here, the dimensions of the zero columns are $n \times (m - \tilde{n})$.) Thus, we can compute different estimates of the solution, for various values of $\tilde{n}$, without recomputing the SVD.

Which value of $\tilde{n}$ should we pick? One rule of thumb (called the *discrepancy principle*) is to choose the value of $\tilde{n}$ that makes the residual norm $\| \mathbf{g} - \boldsymbol{K}\mathbf{f}_{\tilde{n}} \|$ close to its expected value. If the errors in the data values $\mathbf{g}$ are independent and normally distributed with mean zero and variance δ^2, then this expected value is $\delta\sqrt{m}$. To estimate the value δ^2, we use other rules of thumb. For many ill-posed problems, for example, the left singular vectors corresponding to very small singular values are highly oscillatory, so they capture white noise in the measured data. Therefore, if we assume that these components are entirely due to noise, and if we believe that the noise has mean zero, we might estimate the variance by computing the variance of the last few components of $\hat{\mathbf{g}} = \boldsymbol{U}^T\mathbf{g}$ using the formula

$$\delta^2 \approx \frac{1}{m - \hat{m} + 1} \sum_{i=\hat{m}}^{m} \hat{\mathbf{g}}_i^2$$

for some value of $\hat{m}$ close to m.

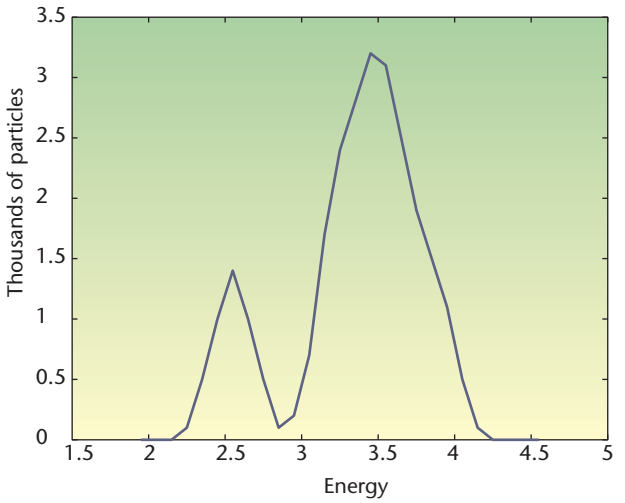

Figure 1. (Simulated) data from a spectrometer. Given that these are particles with at most five different energy levels, determine these energies and the relative abundance of the particles.

Problem 1.

Program this algorithm and try it on the data of Figure 1 for various values of $\tilde{n}$. The matrix $\boldsymbol{K}$ is 27×22, and we assume that the true counts for the first two and the last three bins are zero. Note how ill-conditioned the original matrix $\boldsymbol{K}$ is (by recording σ_1/σ_n).

Model 2: Total Least Squares (TLS)

If we allow both $\boldsymbol{E}$ and $\mathbf{r}$ to be nonzero, how can we solve the problem?

First, we need a way to measure the size of these quantities. One reasonable way is to use the Frobenius norm of the errors:

$$\left\| \begin{bmatrix} \boldsymbol{E} & \mathbf{r} \end{bmatrix} \right\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} e_{ij}^2 + \sum_{i=1}^{m} r_i^2.$$

(If we expect the errors e_{ij}^2 to be much different in size than the errors r_i^2, we might want to use weights for each term in this expression, but for this homework, we'll just leave them equally weighted.)

Let's rewrite $\mathbf{K}\boldsymbol{f} \approx \mathbf{g}$ as

Tools

This homework is related to an earlier homework problem,[1,2] but in that case, the matrix $\boldsymbol{K}$ was assumed to be known exactly.

The current homework relies on your ability to manipulate the singular value decomposition (SVD).[3,4]

Instead of using truncated SVD for regularization, we might use Tikhonov. This method is quite well studied for least squares problems;[5] for total least squares (TLS), additional analysis appears in a paper by Gene Golub, Per Christian Hansen, and Dianne P. O'Leary.[6]

The standard reference for TLS is the book by Sabine Van Huffel and Joos Vandewalle.[7] Ricardo Fierro and colleagues[8] give further information on the truncated TLS algorithm used in Problem 3.

References

1. J.G. Nagy and D.P. O'Leary, "Image Deblurring: I Can See Clearly Now," *Computing in Science & Eng.*, vol. 5, no. 3, 2003, pp. 82–85.
2. J.G. Nagy and D.P. O'Leary, "Image Deblurring: I Can See Clearly Now (Solution)," *Computing in Science & Eng.*, vol. 5, no. 4, 2003, pp. 72–74.
3. G.H. Golub and C.F. van Loan, *Matrix Computations*, Johns Hopkins Press, 1989.
4. C.F. van Loan, *Introduction to Scientific Computing*, Prentice Hall, 2000.
5. P.C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM Press, 1998.
6. G.H. Golub, P.C. Hansen, and D.P. O'Leary, "Tikhonov Regularization and Total Least Squares," *SIAM J. Matrix Analysis and Applications*, vol. 21, 1999, pp. 185–194.
7. S. Van Huffel and J. Vanderwalle, *The Total Least Squares Problem*, SIAM Press, 1991.
8. R. Fierro et al., "Regularization by Truncated Total Least Squares," *SIAM J. Scientific Computing*, vol. 18, 1997, pp. 1223–1241.

$$[\,\boldsymbol{K}\;\mathbf{g}\,]\begin{bmatrix}\mathbf{f}\\ -1\end{bmatrix}\approx 0.$$

Notice these facts:

- If this equation were exactly satisfied, matrix $[\boldsymbol{K}, \mathbf{g}]$ would have a singular value of zero, and $[\mathbf{f}^T, -1]^T$ would be a corresponding right singular vector.
- We need a matrix $[\boldsymbol{E}, \mathbf{r}]$ to add to $[\boldsymbol{K}, \mathbf{g}]$ to make
$$[\,\boldsymbol{K}+\boldsymbol{E}\quad \mathbf{g}+\mathbf{r}\,]\begin{bmatrix}\mathbf{f}\\ -1\end{bmatrix}=0.$$
- Among all such matrices $[\boldsymbol{E}\,\mathbf{r}]$, we need the one with smallest Frobenius norm.

Finding this matrix is a well-studied problem.

PROBLEM 2.

Suppose we have the SVD of $[\mathbf{K}, \mathbf{g}] = \tilde{\boldsymbol{U}}\tilde{\Sigma}\tilde{\boldsymbol{V}}^T$. Assume that $\boldsymbol{K}$ has rank n and that $\tilde{v}_{nn} > \tilde{v}_{n+1,\,n+1} \neq 0$. Show that the solution to

$$\min_{\boldsymbol{E},\mathbf{r}} \left\| \begin{bmatrix}\boldsymbol{E} & \mathbf{r}\end{bmatrix} \right\|_F,$$

subject to the constraint

$$[\,\boldsymbol{K}+\boldsymbol{E}\quad \mathbf{g}+\mathbf{r}\,]\begin{bmatrix}\mathbf{f}\\ -1\end{bmatrix}=0$$

is

$$[\,\boldsymbol{E}\;\mathbf{r}\,] = -\tilde{\sigma}_{n+1}\tilde{\mathbf{u}}_{n+1}\tilde{\mathbf{v}}_{n+1}^T, \text{ with}$$

$$\begin{bmatrix}\mathbf{f}\\ -1\end{bmatrix} = -\frac{1}{\tilde{v}_{n+1,n+1}}\,\tilde{\boldsymbol{v}}_{n+1},$$

where $\tilde{\mathbf{u}}_{n+1}$ is the $(n+1)$st column of $\tilde{\boldsymbol{U}}$ and $\tilde{\mathbf{v}}_{n+1}$ is the $(n+1)$st column of $\tilde{\boldsymbol{V}}$.

Hint:

a. First show that this solution satisfies the constraint and that of the resulting $\|[\boldsymbol{E}\,\mathbf{r}]\|_F = \tilde{\sigma}_{n+1}$.

b. Show that

$$\|\tilde{\boldsymbol{U}}^T\boldsymbol{A}\tilde{\boldsymbol{V}}\|_F = \|\boldsymbol{A}\|_F$$

for any matrix $\boldsymbol{A}$ of size $m \times (n+1)$.

c. Then transform the problem to minimizing $\|[\tilde{\boldsymbol{E}}, \tilde{\boldsymbol{r}}]\|_F$ subject to $(\tilde{\Sigma} + \tilde{\boldsymbol{E}})\tilde{\mathbf{f}} = \mathbf{0}$ for some vectors $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{r}}$ and matrix $\tilde{\boldsymbol{E}}$, solve the problem in this coordinate system, and show that no solution gives a value of the minimization function smaller than $\tilde{\sigma}_{n+1}$.

If we want to truncate our model at $\tilde{n}$, the solution becomes

$$\mathbf{f} = -\frac{1}{\|\tilde{\boldsymbol{V}}_{22}\|^2}\,\tilde{\boldsymbol{V}}_{12}\tilde{\boldsymbol{V}}_{22}^T$$

where $\tilde{\boldsymbol{V}}_{12}$ consists of rows 1 through n and columns $\tilde{n}+1$ through $n+1$ of $\tilde{\boldsymbol{V}}$, and $\tilde{\boldsymbol{V}}_{22}$ contains the last row of these columns of $\tilde{\boldsymbol{V}}$.

Problem 3.

Write a Matlab function to solve Model 2 using this truncated technique for various values of $\tilde{n}$. The input values should be $\boldsymbol{K}$, $\mathbf{g}$, and a range of $\tilde{n}$ values. Include appropriate documentation, and use your function to solve our problem.

Problem 4.

Write a brief summary of the results you obtained using Model 1 and Model 2 to solve the problem of Figure 1. Give your best estimate of the number of different peaks (energy levels) in the original data $\mathbf{f}_{true}$, the relative heights of the peaks, and the centers of the peaks. Make a convincing argument to justify your estimate and your choice of parameters (δ and $\tilde{n}$) for each method.

Finally, we're ready to answer our original question.

How to Reach *CiSE*

Writers

For detailed information on submitting articles, write to cise@computer.org or visit www.computer.org/cise/author.htm.

Letters to the Editors

Send letters to Jenny Ferrero, Contact Editor, jferrero@computer.org. Please provide an email address or daytime phone number with your letter.

On the Web

Access www.computer.org/cise/ or http://cise.aip.org for information about CiSE.

Subscribe

Visit https://www.aip.org/forms/journal_catalog/order_form_fs.html or www.computer.org/subscribe/.

Subscription Change of Address (IEEE/CS)

Send change-of-address requests for magazine subscriptions to address.change@ieee.org. Be sure to specify CiSE.

Subscription Change of Address (AIP)

Send general subscription and refund inquiries to subs@aip.org.

Missing or Damaged Copies

If you are missing an issue or you received a damaged copy (IEEE/CS), contact membership@computer.org. For AIP subscribers, contact kgentili@aip.org.

Reprints of Articles

For price information or to order reprints, send email to cise@computer.org or fax +1 714 821 4010.

Reprint Permission

To obtain permission to reprint an article, contact William Hagen, IEEE Copyrights and Trademarks Manager, atcopyrights@ieee.org.

www.computer.org/cise/

PARTIAL SOLUTION TO
LAST ISSUE'S HOMEWORK ASSIGNMENT

MULTIDIMENSIONAL INTEGRATION: PARTITION AND CONQUER

By Dianne P. O'Leary

IN THE LAST ISSUE, WE INVESTIGATED VARIOUS MONTE CARLO INTEGRATION METHODS ON A SAMPLE PROBLEM—FINDING THE AREA OF A QUARTER CIRCLE—AND ON A PARTITION FUNCTION.

PROBLEM 1.

Suppose we want to estimate the area of a quarter circle with radius r:

$$I = \int_0^r \sqrt{r^2 - x^2}\, dx = \int_0^r f(x) dx .$$

Let $r = 0.5$ and use two methods:

- Method 1, testing whether randomly generated points in the rectangle $[0, r] \times [0, r]$ are inside or outside the quarter circle Ω and multiplying the fraction inside by r^2, the area of the rectangle.
- Method 2, computing the average value of $f(x)$ and multiplying by r, the length of $\Omega = [0, r]$.

Compare the quality of the two estimates for 10, 100, 1,000, 10,000, and 100,000 points by measuring the error and the convergence rate.

How many function evaluations does your favorite integration routine use to get an estimate of comparable quality?

Answer:

Sample code is given on the *CiSE* Web site (www.computer.org/cise/homework). Method 2 gives somewhat better results because it averages the function values themselves rather than just using them to decide whether a point is inside or outside the region. Three-digit accuracy is achieved for 100,000 points in Method 1 and for 1,000 and 100,000 points for Method 2. The convergence rate for Method 1 is consistent with $1/\sqrt{n}$, because the product of the error with $\sqrt{n}$ is approximately constant for large n, but for Method 2, the results are somewhat more variable. Matlab's function `quad` uses 13 function evaluations to get three-digit accuracy.

Clearly, for low-dimensional integration of smooth functions, Monte Carlo methods are not the methods of choice. Their value becomes apparent only when the dimension d is large so that methods like `quad` would be forced to use a lot of function evaluations.

PROBLEM 2.

Compute new estimates of the integral of Problem 1 using Method 3, importance sampling, taking 10 samples of $f(x)$ to determine the function $p(x)$. Compare with your previous results.

Answer:

Sample code is available on the *CiSE* Web site. Importance sampling produces better estimates at lower cost: see the answer to Problem 4 for detailed results.

PROBLEM 3.

Generate 500 pseudorandom points $\mathbf{v}_p$ in $\Re^2$ and 500 quasirandom points $\mathbf{w}_p$ in $\Re^6$ ($p = 1, \ldots, 500$). Plot the pseudorandom points. Then for the quasirandom points, make a plot of the first two coordinates, a separate plot of the third and fourth coordinates, and a final plot of the fifth and sixth coordinates. Discuss the desirability of using each of these four choices for "random" points.

Answer:

Figure A shows the results. The pseudorandom points from Matlab's `rand` are designed to have good statistical properties, but they leave large gaps in space. The quasirandom points are both more predictable and more evenly distributed: they tend to lie on diagonal lines, with longer strings as the coordinate number increases. Other algorithms for generating quasirandom points avoid this defect.

PROBLEM 4.

Compute new estimates of the integral of Problem 1 using quasirandom numbers in Method 2 instead of pseudorandom numbers. Compare the results.

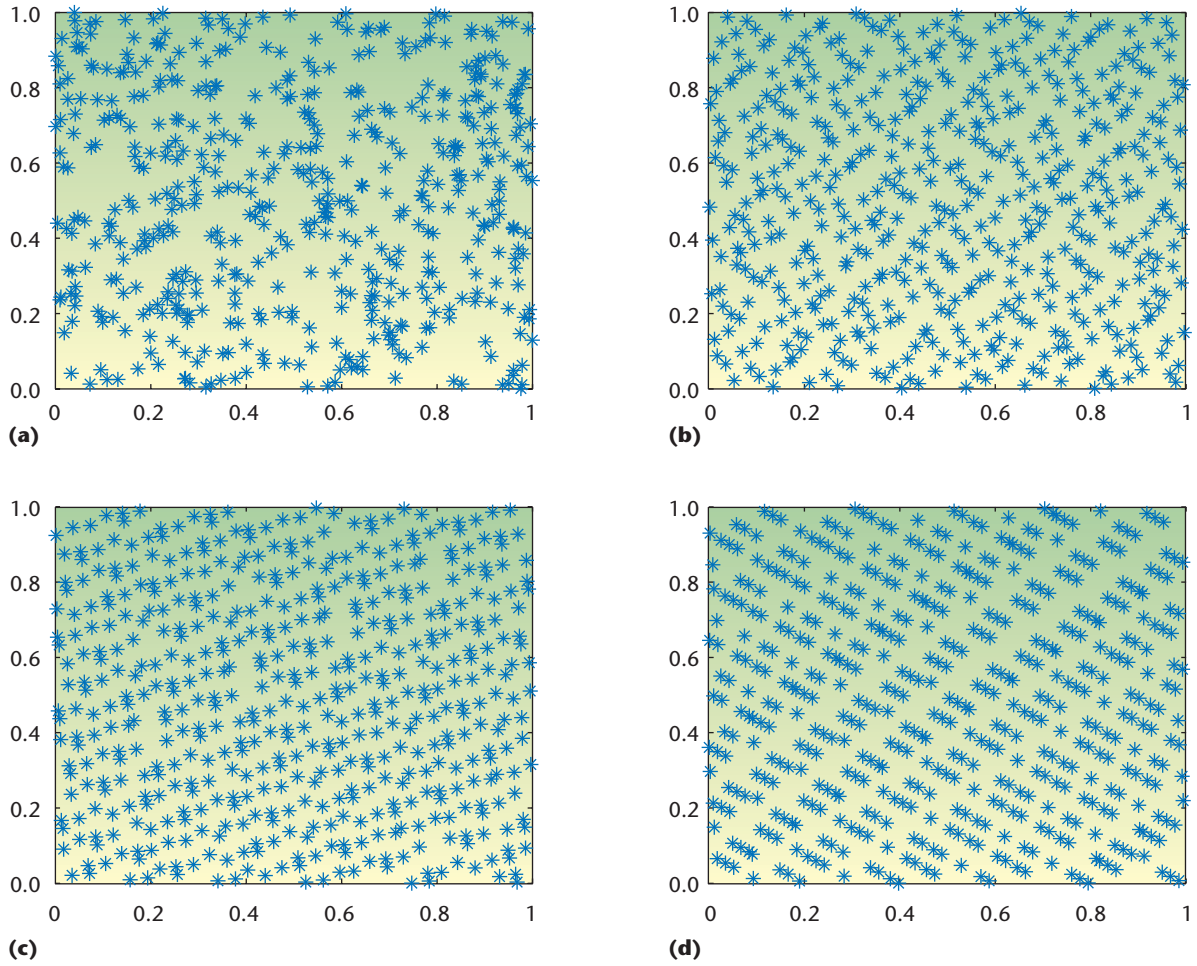

Figure A. The answer to Problem 3. (a) 500 pseudorandom points, and (b) the first two quasirandom coordinates, (c) the second two, and (d) the third two.

Answer:

Table 1 gives the absolute value of the errors in the estimates from each of the four methods. We can determine the convergence rates from the slope of a straight line fit to the logs of each set of errors. We get the best results with Method 4, using quasirandom numbers in Method 2. Method 3, importance sampling, was also quite good.

Consider a chain of particles in which each particle interacts with its one or two neighbors on the chain. For this set of particles, the partition function is

$$Z_d(L) = \int_{-\infty}^{\infty} \rho(a, a, L)da, \quad (1)$$

where $L = 1/(\alpha T)$, T is the temperature, α is the Boltzmann constant, d is the number of particles, and ρ is

$$\rho_d(a, b, L) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} g(a, x_1, L)g(x_1, x_2, L) \dots g(x_d, b, L)dx_1dx_2 \dots dx_d,$$

where

$$g(x, y, L) = \frac{1}{\sqrt{2\pi\delta}} \exp\left(-\frac{1}{2\delta}(x-y)^2 - \frac{1}{2}\delta\left(V(x)+V(y)\right)\right),$$

with $\delta = L/(d+1)$.

PROBLEM 5.

a. Let $L = 1$ and $k = 1.38$ Angstroms^2 g/sec^2 K and consider the harmonic oscillator potential $V(x)$. Determine finite integration limits for $a, x_1, \dots, x_d$ so that the partition function $Z_d(L)$ in Equation 1 can be approximated to three-digit accuracy. (Do this by bounding the neglected part of the integral.)

b. Use your favorite one-dimensional integration routine to estimate the partition function $Z_d(L)$. When you need a

Table 1. Absolute value of estimated errors.

n	Method 1	Method 2	Method 3	Method 4
10	3.65e-03	1.11e-02	4.67e-03	1.50e-02
100	1.35e-03	3.38e-03	1.02e-03	2.49e-03
1,000	2.85e-03	2.38e-04	1.22e-05	3.00e-04
10,000	1.57e-03	1.14e-03	1.75e-04	4.10e-05
100,000	4.97e-04	1.72e-04	1.44e-05	5.14e-06

function value $\rho(a, a, L)$, use Monte Carlo integration to obtain it. Try n = 100, 1,000, 10,000, 100,000, 1,000,000 (if possible), and d = 1, 2, 4, 8, 16.

c. Repeat the experiment using quasirandom points in the Monte Carlo integration. What can you say about the accuracy and convergence rate of your estimates?

Answer:

There are several ways to get bounds. The one used here is a variation on Mei Huang's ideas.

For (a), let $x_0 = x_{d+1} = a$ and let $\mathbf{x} = [x_0, \ldots, x_d]^T$ and $\boldsymbol{P}\mathbf{x} = [x_1, \ldots, x_{d+1}]^T$. Note that

$$g(a, x_1, L)g(x_1, x_2, L) \ldots g(x_d, a, L)$$

$$= \left(\frac{1}{\sqrt{2\pi\delta}}\right)^{d+1} \exp\left(-\frac{1}{2\delta}\sum_{i=1}^{d+1}(x_i - x_{i-1})^2 - \frac{\alpha\delta}{4}\left(\sum_{i=1}^{d+1}x_i^2 + \sum_{i=0}^{d}x_i^2\right)\right)$$

$$= \left(\frac{1}{\sqrt{2\pi\delta}}\right)^{d+1} \exp\left(-\frac{1}{2\delta}\left(2\|\mathbf{x}\|^2 - 2\mathbf{x}^T\boldsymbol{P}\mathbf{x}\right) - \frac{\alpha\delta}{2}\|\mathbf{x}\|^2\right)$$

$$= \left(\frac{1}{\sqrt{2\pi\delta}}\right)^{d+1} \exp\left(-\left(\frac{1}{\delta} + \frac{\alpha\delta}{2}\right)\|\mathbf{x}\|^2 + \frac{1}{\delta}\mathbf{x}^T\boldsymbol{P}\mathbf{x}\right).$$

Now, because $\|\mathbf{x}\| = \|\boldsymbol{P}\mathbf{x}\|$ and because $\mathbf{x}^T\mathbf{y} = \|\mathbf{x}\|\,\|\mathbf{y}\|\cos(\theta)$, where θ is the angle between $\mathbf{x}$ and $\mathbf{y}$, we see that $\mathbf{x}^T\boldsymbol{P}\mathbf{x} \le \mathbf{x}^T\mathbf{x}$, so

$$g(a, x_1, L)g(x_1, x_2, L) \ldots g(x_d, a, L)$$

$$\le \left(\frac{1}{\sqrt{2\pi\delta}}\right)^{d+1} \exp\left(-\left(\frac{1}{\delta} + \frac{\alpha\delta}{2}\right)\|\mathbf{x}\|^2 + \frac{1}{\delta}\mathbf{x}^T\mathbf{x}\right)$$

$$= \left(\frac{1}{\sqrt{2\pi\delta}}\right)^{d+1} \exp\left(-\frac{\alpha\delta}{2}\|\mathbf{x}\|^2\right).$$

Now if we integrate the function $g(a, x_1, L)\, g(x_1, x_2, L) \ldots g(x_d, a, L)$ over the region $\Omega = [-b, b]^{d+1} \subset \mathcal{R}^{d+1}$ instead of $\mathcal{R}^{d+1}$, the error is bounded by

$$\int_{\mathcal{R}^{d+1}-\Omega} g(a, x_1, L)g(x_1, x_2, L) \ldots g(x_d, a, L)d\mathbf{x}$$

$$\le \int_{\mathcal{R}^{d+1}-\Omega}\left(\frac{1}{\sqrt{2\pi\delta}}\right)^{d+1} \exp\left(-\frac{\alpha\delta}{2}\|\mathbf{x}\|^2\right)d\mathbf{x}$$

$$= \int_{\mathcal{R}^{d+1}}\left(\frac{1}{\sqrt{2\pi\delta}}\right)^{d+1} \exp\left(-\frac{\alpha\delta}{2}\|\mathbf{x}\|^2\right)d\mathbf{x} - \int_{\Omega}\left(\frac{1}{\sqrt{2\pi\delta}}\right)^{d+1} \exp\left(-\frac{\alpha\delta}{2}\|\mathbf{x}\|^2\right)d\mathbf{x}$$

$$= \left(\frac{1}{\sqrt{2\pi\delta}}\int_{-\infty}^{\infty}\exp\left(-\frac{\alpha\delta}{2}x^2\right)dx\right)^{d+1} - \left(\frac{1}{\sqrt{2\pi\delta}}\int_{-b}^{b}\exp\left(-\frac{\alpha\delta}{2}x^2\right)dx\right)^{d+1}.$$

We can now use the fact that these integrals are closely related to the integral for the normal distribution:

$$F(x) = \frac{1}{\sqrt{2\pi\sigma}}\int_{-\infty}^{x}\exp\left(-t^2/(2\sigma^2)\right)dt,$$

with $\sigma^2 = 1/(\alpha\delta)$, so we see that

$$\int_{\mathcal{R}^{d+1}-\Omega} g(a, x_1, L)g(x_1, x_2, L) \ldots g(x_d, a, L)\, d\mathbf{x} \le (\delta\sqrt{\alpha})^{d+1}(1 - (F(b)) - F(-b))^{d+1}) \equiv e(\delta, \alpha, d, b).$$

Using tables or computer functions of the normal distribution function, we can compute this bound.

For (b) and (c), visit the *CiSE* Web site for a sample program:

- Getting a good solution to this problem via Monte Carlo integration is difficult. It's very easy to underestimate the answer if you spend a lot of function evaluations on regions that have function values close to zero.
- For efficiency and to avoid intermediate underflow, the program should use an expression like `exp(a+b)` instead of `exp(a)*exp(b)`.

Importance sampling, using a normal distribution, would be a better approach to this problem, but the best approach uses the partition-and-conquer method discussed in the last issue.

Acknowledgments

I'm grateful to Mei Huang for her work on Problem 5.

Dianne P. O'Leary is a professor of computer science and a faculty member in the Institute for Advanced Computer Studies and the Applied Mathematics Program at the University of Maryland. She received a BS in mathematics from Purdue University and a PhD in computer science from Stanford. She is a member of SIAM, ACM, and AWM. Contact her at oleary@cs.umd.edu; www.cs.umd.edu/users/oleary/.