Announcements

- Remaining things to hand in
  - Take home M2 to be posted soon (Tuesday).
  - We will also have a take home final F1
- Today's class will end a few minutes early
  - (SISTA seminar is in Marley)

Markov chain Monte Carlo methods

- Samples are conditioned on the previous one (this is the Markov chain).
- We have given up the very natural preference for independent samples.
- Basic intuition why this might be a good idea
  - If you have **finally** found a region of high probability, stick around for a bit, enjoy yourself, grab some more samples.
- MCMC is generally a good hammer for complex, high dimensional, problems.

Ergodic chains

- Different starting probabilities will give different chains
- We want our chains to converge (in the limit) to the same stationary state, regardless of starting distribution.
- Such chains are called ergodic, and the common stationary state is called the equilibrium state.
- Ergodic chains have a unique equilibrium.

When do our chains converge?

- Important theorem tells us that (finite*) our chains converge to equilibrium under two relatively weak conditions.
- (1) Irreducible
  - We can get from any state to any other state
- (2) Aperiodic
  - The chain does not get trapped in cycles
- These are true for detailed balance which is sufficient, but not necessary for convergence.

*Infinite or uncountable state spaces introduces additional complexities.

Recall terminology and chain evolution

Chains (think ensemble) evolve according to:

$$p(z) = \sum_{z'} p(z')\, T(z', z)$$

Matrix vector representation:

$$\mathbf{p} = \mathbf{T}\mathbf{p}'$$

And, after *n* iterations after a starting point:

$$\mathbf{p}^{(n)} = \mathbf{T}^{N}\mathbf{p}^{(0)}$$

Aside on stochastic Matrices

- T is a left (column) stochastic matrix.
  - If you are right handed, take the transpose
- The column vector, **p**, also has non-negative elements, that sum to one (sometimes this is called a stochastic vector).
- Fun facts that we did on the board
  - The product of a stochastic matrix and vector is a stochastic vector.
  - The product of two stochastic matrices is a stochastic matrix.

Aside on (stochastic) Matrix powers

Consider the eigenvalue decomposition of T, $T = E\Lambda E^{-1}$

So, $\mathrm{T}^{N} = E\Lambda^{N}E^{-1}$

Since T^{N} cannot grow without bound, the eigenvalues are inside $[-1,1]$.

In fact, for our situation, the second biggest absolute value of the eigenvalues is less than one (not so easy to prove).

Aside on (stochastic) Matrix powers

We have $\mathrm{T}^{N} = E\Lambda^{N}E^{-1}$

$$\Lambda = \begin{pmatrix} 1 & & & \\ & \lambda_2 & & \\ & & \cdots & \\ & & & \lambda_K \end{pmatrix} \text{ and } \Lambda^{\infty} = \begin{pmatrix} 1 & & & \\ & 0 & & \\ & & \cdots & \\ & & & 0 \end{pmatrix}$$

$$\Lambda^{\infty}E^{-1} = \begin{pmatrix} \mathbf{e}_1^T \\ 0 \\ \ldots \\ 0 \end{pmatrix} \quad \text{and } \mathrm{E}\Lambda^{\infty}E^{-1}\mathbf{p} \parallel \mathbf{e}_1 \parallel \mathbf{p}^{*}$$

(End of last lecture)

Aside on (stochastic) Matrix powers

We have $T^N = E\Lambda^N E^{-1}$

$$\Lambda = \begin{pmatrix} 1 & & & \\ & \lambda_2 & & \\ & & \cdots & \\ & & & \lambda_K \end{pmatrix} \text{ and } \Lambda^\infty = \begin{pmatrix} 1 & & & \\ & 0 & & \\ & & \cdots & \\ & & & 0 \end{pmatrix}$$

So, $\Lambda^\infty E^{-1} = \begin{pmatrix} \mathbf{e}_1^T \\ 0 \\ \ldots \\ 0 \end{pmatrix}$

Aside on (stochastic) Matrix powers

We have $\Lambda^\infty E^{-1} = \begin{pmatrix} \mathbf{e}_1^T \\ 0 \\ \ldots \\ 0 \end{pmatrix}$

So, $\Lambda^\infty E^{-1}\mathbf{p} = \begin{pmatrix} \mathbf{e}_1^T \cdot \mathbf{p} \\ 0 \\ \ldots \\ 0 \end{pmatrix}$

And, $E\Lambda^\infty E^{-1}\mathbf{p} = ?$

Aside on (stochastic) Matrix powers

We have $\Lambda^\infty E^{-1}\mathbf{p} = \begin{pmatrix} \mathbf{e}_1^T \cdot \mathbf{p} \\ 0 \\ \ldots \\ 0 \end{pmatrix}$

So, $E\Lambda^\infty E^{-1}\mathbf{p} = \mathbf{e}_1\left(\mathbf{e}_1^T \cdot \mathbf{p}\right) \parallel \mathbf{e}_1 \parallel \mathbf{p}^*$

In summary, $\mathbf{p}^* \parallel \mathbf{e}_1$ and $\mathbf{p}^*$ stochastic means that $E\Lambda^\infty E^{-1}\mathbf{p} = \mathbf{p}^*$

This is true, no mater what the initial point $\mathbf{p}$ is.

So, glossing over details, we have convergence to equilibrium.

Demo

- According to the previous, if T is a stochastic matrix, then:

$\mathbf{p}^* \cong \mathbf{T}^\mathbf{N}\mathbf{p}$
(No matter what p! They all will give the same answer).

Also, $\mathbf{p}^* \parallel \mathbf{e}^{(\mathbf{1})}$

Justification relies on Perron Frobenious theorem

Let $A = (a_{ij})$ be an $n \times n$ positive matrix: $a_{ij} > 0$ for $1 \le i, j \le n$. Then the following statements hold.

1. There is a positive real number r, called the **Perron root** or the **Perron–Frobenius eigenvalue**, such that r is an eigenvalue of A and any other eigenvalue λ (possibly, complex) is strictly smaller than r in absolute value, $|\lambda| < r$. Thus, the spectral radius $\rho(A)$ is equal to r.
2. The Perron–Frobenius eigenvalue is simple: r is a simple root of the characteristic polynomial of A. Consequently, the eigenspace associated to r is one-dimensional. (The same is true for the left eigenspace, i.e., the eigenspace for A^T.)
3. There exists an eigenvector $v = (v_1,\dots,v_n)$ of A with eigenvalue r such that all components of v are positive: $A v = r v$, $v_i > 0$ for $1 \le i \le n$. (Respectively, there exists a positive left eigenvector w : $w^T A = r w^T$, $w_i > 0$.)
4. There are no other positive (moreover non-negative) eigenvectors except v (respectively, left eigenvectors except w), i.e. all other eigenvectors must have at least one negative or non-real component.
5. $\lim_{k\to\infty} A^k/r^k = vw^T$, where the left and right eigenvectors for A are normalized so that $w^T v = 1$. Moreover, the matrix $v w^T$ is the projection onto the eigenspace corresponding to r. This projection is called the **Perron projection**.
6. **Collatz–Wielandt formula**: for all non-negative non-zero vectors x, let $f(x)$ be the minimum value of $[Ax]_i / x_i$ taken over all those i such that $x_i \neq 0$. Then f is a real valued function whose maximum is the Perron–Frobenius eigenvalue.
7. A "Min-max" Collatz–Wielandt formula takes a form similar to the one above: for all strictly positive vectors x, let $g(x)$ be the maximum value of $[Ax]_i / x_i$ taken over i. Then g is a real valued function whose minimum is the Perron–Frobenius eigenvalue.
8. The Perron–Frobenius eigenvalue satisfies the inequalities

$$\min_i \sum_j a_{ij} \le r \le \max_i \sum_j a_{ij}.$$

From Wikipedia

Main points about P-F

- The maximal eigenvalue is strictly maximal (item 1).
- The corresponding eigenvector is "simple" (item 2)
- It has all positive (or negative) components (item 3).
- There is no other eigenvector that can be made non-negative.
- The maximal eigenvalue of a stochastic matrix has absolute value 1 (item 8 applied to stochastic matrix).

Aside on (stochastic) Matrix powers

Summary

$\mathbf{p}^* = \mathbf{T}\mathbf{P}^*$ is an eigenvector with eigenvalue one.

We have written it as $\mathbf{p}^* \parallel \mathbf{e}^1$ because $\mathbf{e}^1$ is the eigenvector normalized to norm 1 (standard form).

Intuitively (perhaps), T will reduce any component of p orthogonal to p^*, and T^N will kill off such components as $\mathrm{N} \to \infty$.

Algebraic proof

Neal '93 provides an algebraic proof which does not rely on spectral theory.

(Likely homework assignment will be study this further).

Summary so far

- Under reasonable (easily checked and/or arranged) conditions, our chains converge to an equilibrium state.
- Easiest way to prove (or check) that this is the case is to show detailed balance.
- To use MCMC for sampling a distribution, we simply ensure that our target distribution is the equilibrium state.
- Variations on MCMC are mostly about improving the speed of convergence for particular situations.

Summary so far

- The time it takes to get reasonably close to equilibrium (where samples come from the target distribution) is called "burn in" time.
  - I.E., how long does it take to forget the starting state.
  - There is no general way to know when this has occurred.
- The average time it takes to visit a state is called "hit time".
- What if we really want independent samples?
  - We can take every N^{th} sample (some theories about how long to wait exist, but it depends on the algorithm and distribution)

Are we learning anything useful?

In putting together this issue of *Computing in Science & Engineering*, we knew three things: it would be difficult to list just 10 algorithms; it would be fun to assemble the authors and read their papers; and, whatever we came up with in the end, it would be controversial. We tried to assemble the 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century. Following is our list (here, the list is in chronological order; however, the articles appear in no particular order):

- Metropolis Algorithm for Monte Carlo
- Simplex Method for Linear Programming
- Krylov Subspace Iteration Methods
- The Decompositional Approach to Matrix Computations
- The Fortran Optimizing Compiler
- QR Algorithm for Computing Eigenvalues
- Quicksort Algorithm for Sorting
- Fast Fourier Transform
- Integer Relation Detection
- Fast Multipole Method

With each of these algorithms or approaches, there is a person or group receiving credit for inventing or discovering the method. Of course, the reality is that there is generally a culmination of ideas that leads to a method. In some cases, we chose authors who had a hand in developing the algorithm, and in other cases, the author is a leading authority.

1521-9615/00/$10.00 © 2000 IEEE

Jack Dongarra
University of Tennessee and Oak Ridge National Laboratory
Francis Sullivan
IDA Center for Computing Sciences

In this issue

Monte Carlo methods are powerful tools for evaluating the properties of complex, many-body systems, as well as nondeterministic processes. Isabel Beichl and Francis Sullivan describe the Metropolis Algorithm. We are often confronted with problems that have an enormous number of dimensions or a process that involves a path with many possible branch points, each of which is governed by some fundamental probability of occurence. The solutions are not exact in a rigorous way, because we randomly sample the problem. However, it is possible to achieve nearly exact results using a relatively small number of samples compared to the problem's dimensions. Indeed, Monte Carlo methods are the only practical choice for evaluating problems of high dimensions.

John Nash describes the Simplex method for solving linear programming problems. (The use of the word *programming* here really refers to scheduling or planning—and not in the way that we tell a computer what must be done.) The Simplex method relies on noticing that the objective function's maximum must occur on a corner of the space bounded by the constraints of the "feasible region."

Large-scale problems in engineering and science often require solution of sparse linear algebra problems, such as systems of equations. The importance of iterative algorithms in linear algebra stems from the simple fact that a direct approach will require $O(N^3)$ work. The Krylov subspace iteration methods have led to a major change in how users deal with large, sparse, nonsymmetric matrix problems. In this article, Henk van der Vorst describes the state of the art in terms of

- Metropolis Algorithm for Monte Carlo
- Simplex Method for Linear Programming
- Krylov Subspace Iteration Methods
- The Decompositional Approach to Matrix Computations
- The Fortran Optimizing Compiler
- QR Algorithm for Computing Eigenvalues
- Quicksort Algorithm for Sorting
- Fast Fourier Transform
- Integer Relation Detection
- Fast Multipole Method

Metropolis-Hastings MCMC method

- Like Metropolis, but now q() is not symmetric.

Metropolis-Hastings MCMC method

While not_bored

{

Sample $q\left(z \mid z^{(prev)}\right)$

Accept with probability $A\left(z, z^{(prev)}\right) = \min\left(1, \frac{\tilde{p}(z)\, q\left(z^{(prev)} \mid z\right)}{\tilde{p}\left(z^{(prev)}\right) q\left(z \mid z^{(prev)}\right)}\right)$

If accept, emit z, otherwise, emit $z^{(prev)}$.

}

Does Metropolis-Hastings have detailed balance?

$$
\begin{aligned}
p(z')\,q(z \mid z')\,A(z, z') &= \min\left(p(z')\,q(z \mid z'),\, p(z)\,q(z' \mid z)\right) \\
&= p(z)\,q(z' \mid z)\min\left(\frac{q(z \mid z')}{q(z' \mid z)}\frac{p(z')}{p(z)}, 1\right) \\
&= p(z)\,q(z' \mid z)\min\left(1, \frac{p(z')}{p(z)}\frac{q(z \mid z')}{q(z' \mid z)}\right) \\
&= p(z)\,q(z' \mid z)\,A(z', z)
\end{aligned}
$$

Metropolis-Hastings comments

- Again it does not matter if we use unnormalized probabilities.
- It should be clear that the previous version, where q() is symmetric, is a special case.