Sinkhorn ratings, and new strongly polynomial time algorithms for Sinkhorn balancing, Perron eigenvectors, and Markov chains

Warren D. Smith*
WDSmith@fastmail.fm

July 7, 2005

Abstract — **This paper simultaneously makes new contributions to political science, probability and statistics, and linear algebra.**

We describe a new method, "Sinkhorn ratings," for pairwise-comparison based ranking of chessplayers, web pages, or football teams; it also may be used to rank the candidates in an election in which each vote is a partial ordering of the candidates.

We also describe the first "strongly polynomial time" algorithm for finding the Perron-Frobenius eigenvector of a matrix with non-negative entries, and the second strongly polynomial time algorithm (and the first practical one) for Sinkhorn balancing a matrix with non-negative entries. The former also may be regarded as the first strongly polynomial algorithm for finding the stationary distribution of an N-state Markov chain with known transition matrix. Along the way we also present a new formulation of the Perron-Frobenius eigenvector and Markov stationary distribution problems as concave-∪ minimization problems, and present a powerful new technique for proving monotonicity statements e.g. about Markov chains.

Keywords — Voting paradoxes, rating systems, Sinkhorn matrix balancing, Perron-Frobenius eigenvectors, Markov chains, strongly polynomial algorithms, linear algebra, minimum-entropy statistical estimators, concavity, monotonicity.

1 The voting and ranking problems

Suppose there are N chessplayers and we know the results of all the games they have played between themselves so far. These could be written in a table, e.g. U_{ab} could denote the number of victories of player a over player b (with each drawn game counting as half a victory and half a defeat).

If instead of "chessplayers," we have "football teams," a similar matrix U could be written, but this time U_{ab} could sum some monotonic function of the score by which a defeated b.

In a political science setting in which each voter provides a rank-ordering of all N candidates (or perhaps only some subset of the candidates), we could have U_{ab} be the number of voters who have expressed preference for candidate a over b.

In any of these scenarios, regard the $N \times N$ matrix U as the "input data." Our problem is to determine, as output data, a real number "rating" for each chessplayer, football team, or candidate, with the "best" ones receiving the highest ratings and the worst ones the lowest.

It is not clear what to do. Many ideas have been suggested, but we shall only discuss one (by J.P.Keener [26]) besides our own, because it is highly analogous to it. Both methods have a high degree of mathematical "elegance."

2 The Keener-eigenvector and Sinkhorn solutions

Keener's proposal is to find the Perron-Frobenius eigenvector $\vec{x}$ – that is, the unique eigenvector with positive real entries – of the matrix U. Then the rating of candidate (or chessplayer) c is x_c.

Our new "Sinkhorn rating" proposal is to find the unique diagonal matrices R and C of positive reals such that $S = RUC$ is doubly-stochastic, i.e. S has non-negative entries and all row-sums and column-sums of S are 1. Then the rating of chessplayer p is C_{pp}/R_{pp}.

As we shall see, both Keener and Sinkhorn ratings are obtainable via remarkably simple iterative processes. But Sinkhorn ratings enjoy some advantages that Keener ratings do not: They exhibit "reversal symmetry," they are interpretable as minimum entropy statistical estimators, and they are directly connected to gambling odds for the next chess game.

3 Deeper look at the mathematical underpinnings of Keener-eigenvector

Keener's idea [26] actually yields a family[1] of methods, not just one. It also was re-invented later and without mentioning Keener by Dwork et al. [16] for the purpose of rating "web pages."

It employs the Perron-Frobenius theorem [32].

*21 Shore Oaks Drive, Stony Brook NY 11790.

[1]Both Keener and Dwork et al also considered several other natural Markov chains and/or matrices and explained how they too could be used for voting and ranking.

We now state a fairly strong form of that theorem, compiled from the beginning of [46], theorem 3.1 of section 2.3 of [35], and [22] chapter 8. Skip to theorem 2 for a simpler version.

Theorem 1 (Perron-Frobenius theorem (1907-1912)). *Let U be an $N \times N$ matrix all of whose entries are nonnegative. Suppose for some k with $k \geq (N-1)^2+1$ that U^k has all entries positive. (Wielandt showed that if this is true for any such k, it is true for all of them.) (Equivalently: the directed graph whose adjacency structure is represented by the nonzero entries of U, is "strongly connected" and each of its vertices is in a directed ℓ-cycle for all sufficiently large integer sizes ℓ, i.e. the digraph is "aperiodic.") Then U has a unique (non-multiple) positive real eigenvalue r, such that $r > |\lambda|$ for all eigenvalues λ of U with $\lambda \neq r$. Indeed $r \geq |\lambda|$ for all eigenvalues λ of B where B is any $N \times N$ matrix with $0 \leq B_{ij} \leq U_{ij}$, with equality only occurring if $B = D^{-1}AD$ for some diagonal matrix D; and $|B\vec{y}| \leq r|\vec{y}|$ for all $\vec{y}$. And indeed for each $k \geq 1$*

$$\frac{|\lambda|}{r} \leq \left(\frac{1-\mu}{1+\mu}\right)^{1/k} \quad \text{where } \mu = \frac{\min_{ij}(U^k)_{ij}}{\max_{st}(U^k)_{st}}. \tag{1}$$

A simple bound on r is

$$\min_j \sum_h U_{jh}\frac{y_h}{y_j} \leq r \leq \max_s \sum_t U_{st}\frac{y_t}{y_s}\,; \tag{2}$$

for any given vector $\vec{y}$ of positive reals; many stronger bounds are available in [22][35]. This Perron eigenvalue r corresponds to a unique (up to scaling) eigenvector $\vec{x}$ of U (obeying $U\vec{x} = r\vec{x}$). This $\vec{x}$ is the unique eigenvector of U that consists of N positive real numbers. And indeed $\vec{x}$ obeys

$$\frac{\max_j \sum_h U_{jh}}{\min_s \sum_t U_{st}} \leq \frac{\max_k x_k}{\min_h x_h} \leq \max_j \frac{\max_{s\neq j} U_{sj}}{\min_{t\neq j} U_{tj}} \tag{3}$$

so that $\min_h x_h$ is not only positive, it furthermore cannot be too small.

The above theorem is complicated to state mainly because we have been unable to refrain from giving a strong form of it. The following simpler version suffices for most of our purposes!

Theorem 2 (Perron 1907). *Let U be a square matrix with all entries positive. Then U has a unique (i.e. non-multiple) positive real eigenvalue r, such that $r > |\lambda|$ for all eigenvalues λ of U with $\lambda \neq r$. This eigenvalue corresponds to the unique eigenvector of U with all entries positive real.*

Keener proposed finding the Perron eigenvector $\vec{x}$ of U and then using it as the ratings, i.e. x_5 would be the rating of chessplayer #5. (To make $\vec{x}$ unique we recommend agreeing to rescale it so that $\sum_k x_k = 1$.) Actually, this is not necessarily going to work because, e.g, there might not be any $k > 0$ with U^k consisting entirely of positive numbers. Therefore, instead of U_{ab} counting the number of victories of a over b, or voter preferences for a over b – which might be zero – it is better to define U_{ab} to be the number of victories *plus one*. That way all entries of U are positive so the simpler Perron theorem may be applied. This "add one" idea also would be the recommendation of "Laplace's rule" from Bayesian statistics (§5).

We shall discuss Keener's scheme more later. For now let us point out an obvious **reason to question it.** If the results of all chess or football games were reversed, or if all voter preference orderings were reversed, or equivalently if the input matrix U were replaced by its *transpose*, we would hope that the ordering of the output ratings would *reverse*. (Even more powerfully, we might want the rating vector $\vec{x}$ to get replaced by $-\vec{x}$, or some other involution.) Keener's system disobeys this kind of reversal symmetry. For example, the Perron eigenvector (with eigenvalue $r = 10.4773$) of

$$U = \begin{pmatrix} 1 & 9 & 5 \\ 1 & 1 & 7 \\ 5 & 3 & 1 \end{pmatrix} \tag{4}$$

is[2] (0.4204, 0.2717, 0.3078). Meanwhile for U^T it is (0.2462, 0.3588, 0.3949) which seems unrelated and does *not* order the three candidates in reverse order; indeed the first set of ratings ranks the 2nd candidate bottom, but the second set of ratings ranks him in the middle.

4 Deeper look at the mathematical underpinnings of Sinkhorning

Our new rating system, *Sinkhorn voting*, overcomes that criticism. Instead of the Perron-Frobenius theorem, it depends on the following, extracted from [6] [15] [48] [41] [47] [50] [1] [3] [31] [33] [19] [23][24] [40] [42] [7]. Again, we have been unable to refrain from stating a very strong form of the theorem, summarizing a large number of papers in the literature,[3] and which applies to nonnegative matrices; please skip to theorem 4 for a simpler version about matrices with all entries positive. (A list of definitions of terms in the theorem statement appears immediately after it.)

Theorem 3 (Sinkhorn balancing). *Let U be an $N \times N$ matrix all of whose entries are nonnegative.*

I. If U is fully indecomposable then there exist diagonal matrices R and C with positive entries so that $S = RUC$ is doubly-stochastic, i.e. U's rows and columns may be scaled so that they each simultaneously have sum=1; and further, these scalings are unique (i.e. R, C, and S are unique) except for renormalizations of the form $R \to \kappa R$, $C \to C/\kappa$.

[2] Both eigenvectors have been normalized so that the sum of their entries is 1 and rounded to 4 decimal places.

[3] Nevertheless, our statement does not summarize all that is known. Sinkhorn balancing has been generalized to allow specifiying the row and column sums, to allow rectangular matrices, and to allow multidimensional arrays instead of 2-dimensional ones. We also mention that Igor Rivin and I jointly proved that transition probabilities may be assigned to the arcs of a given directed graph, to cause it to turn into a Markov chain that stationarizes the uniform distribution, if and only if that directed graph is cycle coverable. However, we later realized that result is almost trivial after one knows the following. Perfect & Mirsky [41] showed that *nonzero* transition probabilities may be assigned to the arcs of a given directed graph (i.e. to the nonzero entries of a matrix) to cause it to turn into a Markov chain that stationarizes the uniform distribution (i.e. to turn it into a doubly stochastic matrix) if and only if that directed graph is *fully* cycle coverable.

[4] Parlett & Landis ([40] p.65) mentioned that Kahan had proved that for matrices whose nonzero patterns were not cycle coverable, the Sinkhorn iteration would ultimately cycle without converging.

II. If the directed graph associated with U's nonzero pattern is cycle coverable[4] *(equivalently if* $\operatorname{per} U > 0$*), then for each $\epsilon > 0$ there exist diagonal matrices R and C with positive entries so that $S = RUC$ is ϵ-nearly doubly-stochastic. Furthermore, the Sinkhorn iteration on U will converge to a doubly-stochastic limit matrix S (although limit R and C can fail to exist). The limit matrix S is the unique doubly-stochastic matrix which minimizes* $\mathrm{KLdist}(S, U)$.

III. The following 7 statements all are equivalent:

1. *The directed graph arising from U's nonzero pattern is fully cycle coverable*
2. *There exists a doubly-stochastic matrix with the same pattern of zeros*
3. *U's rows and columns may be permuted to cause U to be a direct sum of fully indecomposable matrices*
4. *Positive diagonal matrices R and C exist so $S = RUC$ is doubly stochastic (but these R and C can be non-unique; they are unique exactly in case I)*
5. *Convergence of S, R, and C all occur in the Sinkhorn iteration starting from U (and suitable R,C, and S are produced to satisfy the preceding claim)*
6. *The convergence of S ultimately is geometric in nature*
7. *The zero pattern of the Sinkhorn limit matrix S is the same as that of U.*

IV. If U consists entirely of integers, and B is the sum of the bit-lengths of those integers (expressed in binary), then the Sinkhorn iteration after t steps will yield[5] *an ϵ-nearly doubly-stochastic matrix with $\epsilon = O(B/t)$ The individual dependencies of this bound on t and B both are best possible because of these examples:*

$$\begin{pmatrix} 1 & 0 \\ \alpha & 1-\alpha \end{pmatrix}, \quad \begin{pmatrix} K & K & 0 \\ 1 & 1 & 2K-2 \\ 1 & 1 & 2K-2 \end{pmatrix}; \tag{5}$$

t Sinkhorn steps applied to the first matrix yield $\left(\begin{smallmatrix} 1 & 0 \\ \alpha(t) & 1-\alpha(t) \end{smallmatrix}\right)$ where $\alpha(t) = \alpha/(1 + 2\alpha t)$ for each $t \ge 0$; and when K is a large integer at least $\log_5(2K^2)$ Sinkhorn steps are required to cause the second matrix to become 0.1-nearly doubly stochastic.[6]

V. Each Sinkhorn step monotonically increases the permanent $\operatorname{per} U$ *(and also* $|\det U|$*) if nonzero, and monotonically decreases the difference between the maximum and minimum column sums. Franklin & Lorenz [19][6][29] showed that each Sinkhorn step was a contraction in the Hilbert projective metric within either the space of row scalings, column scalings, or matrices, and showed that the contraction constant γ, $0 < \gamma < 1$, is a function purely of the initial matrix U (i.e. does not change as Sinkhorn steps proceed*[7]*), and is upper bounded by $\gamma \le K^2$ where*

$$K = \frac{1-\mu}{1+\mu} \quad \textit{where} \quad \mu = \sqrt{\max_{h,j,k,\ell} \frac{U_{hk}U_{j\ell}}{U_{jk}U_{h\ell}}} \le \frac{\min_{hj} U_{hj}}{\max_{k\ell} U_{k\ell}}. \tag{6}$$

Consequently $O(V|\log\epsilon|)$ Sinkhorn steps suffice to reach ϵ-near double stochasticity if all the matrix entries are in $[1, V]$. Kalantari & Khachiyan [23] further showed that $O([\epsilon^{-1} + \log N]\sqrt{N} \log V)$ Sinkhorn steps suffice.

VI. If we solve[8] *the following $(2N-2)$-dimensional concave-$\cup$ minimization problem:*

$$\min_{\vec{r},\vec{c}} \sum_{j=1}^{N} \sum_{k=1}^{N} U_{jk} \exp(r_j + c_k) \tag{7}$$

subject to

$$\sum_{j=1}^{N} r_j = \sum_{k=1}^{N} c_k = 0 \tag{8}$$

then the row- and column-scaling factors $\exp(r_j)$ *and* $\exp(c_k)$ *will (up to two scalar multiples) make the $N \times N$ matrix U become doubly-stochastic. It is also possible to deduce the scaling factors from the solution of the following unconstrained N-dimensional concave-$\cup$ minimization problem:*

$$\min_{\vec{r}} \sum_{k=1}^{N} \ln \left[\sum_{j=1}^{N} U_{jk} \exp(r_j) \right] - \sum_{j=1}^{N} r_j. \tag{9}$$

Associated definitions:

1. The "permanent" $\operatorname{per} U$ of an $N \times N$ matrix U is the "determinant with all plus signs," i.e. $\operatorname{per} U = \sum_\pi \prod_{k=1}^N U_{k,\pi(k)}$ where the sum is over all $N!$ permutations π of the set $\{1, 2, \ldots, N\}$.
2. The "bipartite double" of directed graph with N vertices, is the undirected bipartite graph with N red and N blue vertices, where red vertex i is connected to blue vertex j if and only if there is a directed arc $i \to j$ in the directed graph.
3. A directed graph is "cycle coverable" if a set of disjoint directed cycles exist, which cover each vertex exactly once. Equivalently: if its bipartite double contains a perfect matching. Equivalently: the permanent of its $N \times N$ adjacency matrix is nonzero. Equivalently (by the Frobenius-Konig theorem 2.1 in section 4.2 of [35]) if its $N \times N$ adjacency matrix, no matter what permutation of its rows and columns is taken, has no $s \times t$ block of 0s with $s + t = N + 1$.
4. A directed graph is "*fully* cycle coverable" if a set of disjoint directed cycles exist, which not only cover each vertex exactly once, but *also* contain the directed arc a, and this statement is true (perhaps with different cycle covers) for *each* arc a. Equivalently: if its bipartite double contains a perfect matching containing edge e and this statement is true (perhaps with different matchings) for *each* edge e. Parlett & Landis ([40] corollary on p.64) showed that the Sinkhorn iteration on a non-negative matrix with cycle coverable but not fully cycle coverable nonzero pattern, would converge to a matrix S with zero entries $S_{ab} = 0$ on all arcs ab of the directed graph that were not members of a cycle covering.

[5] The $O(B/t)$ claim is taken from Linial et al [31], who attribute it to Franklin and Lorenz [19]. But in fact, Franklin and Lorenz did not explicitly say this.

[6] This second example matrix is fully indecomposable.

[7] Observe that the formulas (6) for μ and K are unaffected by any row- and column-scaling of U.

[8] Each Sinkhorn step may be thought of as performing a minimization over half of the coordinates with the other coordinates held fixed. That and the concave-$\cup$ nature of the objective function is the simplest proof of the global convergence of the Sinkhorn iteration, and also explains why it generically ultimately converges geometrically.

5. A directed graph is "strongly connected" if any vertex is reachable (via a directed path) from any other. The "figure 8" directed graph, consisting of two directed cycles sharing a common vertex, is an example of a graph which is strongly connected but not cycle coverable. The "$\circ \to \circ$" directed graph, consisting of two directed cycles linked by an arc, is an example of a graph which is cycle coverable but neither strongly connected nor fully cycle coverable.
6. A square matrix U is "fully indecomposable" if there do not exist permutation matrices P and Q so that $PUQ = \left(\begin{smallmatrix} X & Y \\ 0 & Z \end{smallmatrix}\right)$ where X, Y, Z are blocks with X and Y square in shape, and 0 denotes an all-zero block. Equivalently, the bipartite $N + N$-vertex graph associated with the nonzero pattern of U has a subset of s ($0 < s < N$) inputs which are connected to, and only to, a subset of s outputs. Full indecomposability implies, but is not implied by, the requirement that the directed graph arising from U's nonzero pattern be fully cycle coverable. It also implies, but is not implied by, the requirement that the directed graph arising from U's nonzero pattern be strongly connected.
7. A square matrix S is "doubly-stochastic" if all its entries are nonnegative and all its row sums and column sums are 1. Birkhoff [5] (theorem 3.1 sec 5.3 of [35]; theorem 8.7.1 of [22]) showed the doubly stochastic matrices are precisely the convex combinations of permutation matrices. It is "ϵ-nearly doubly-stochastic" if all its entries are nonnegative and all its row sums and column sums are within $\pm\epsilon$ of 1.
8. A "Sinkhorn step" consists of renormalizing the columns of U to make them sum to 1, then renormalizing the rows of U to make them sum to 1. The "Sinkhorn iteration" is to repeatedly do Sinkhorn steps.[9] (See §6 for a formal algorithm statement.) If desired, the product of all the row-renormalizing (respectively column-renormalizing) factors can be accumulated in R and C respectively.
9. The "Kullback-Leibler distance" between two probability distributions p and q is $\text{KLdist}(q, p) = \sum_j q_j \log(q_j/p_j)$. This is not actually a metrical distance since $\text{KLdist}(p, q)$ and $\text{KLdist}(q, p)$ need not be equal, but it does satisfy these distance-like properties: $\text{KLdist}(p, q) \geq 0$ with equality if and only if $p = q$, and $\text{KLdist}(p, q)$ is a concave-$\cup$ function. In the Theorem we regard S and U (after normalization to make their N^2 entries sum to 1) as probability distributions on event-pairs.
10. The "Hilbert projective metric" among real N-vectors with all entries positive, is
$$\text{HPdist}(\vec{x}, \vec{y}) = \log \max_{i,k} \frac{x_i y_k}{x_k y_i} \tag{10}$$
and this may also be regarded as a metric on $N \times N$ matrices A, B with all entries positive where A and B are equivalent under some row- and column-scaling vectors $\vec{r}$ and $\vec{c}$:
$$\text{HPdist}(A, B) = \text{HPdist}(\vec{r}, \vec{1}) + \text{HPdist}(\vec{c}, \vec{1}) \tag{11}$$
where $\vec{1} = (1, 1, 1, \ldots, 1)$. The "contraction constant" $K(A)$ of a matrix A with nonnegative entries in the Hilbert projective metric is
$$K(A) = K(A^T) = \sup_{\vec{x},\vec{y}>\vec{0}} \frac{\text{HPdist}(A\vec{x}, A\vec{y})}{\text{HPdist}(\vec{x}, \vec{y})} \tag{12}$$
where $\vec{x}$ and $\vec{y}$ are not allowed to be proportional.[10]

Theorem 4 (Simpler Sinkhorn balancing). *Let U be a square matrix all of whose entries are positive. There exist diagonal matrices R and C with positive entries so that $S = RUC$ is doubly-stochastic, i.e. U's rows and columns may be scaled so that they each simultaneously have sum=1; and further, these scalings are unique (i.e. R, C, and S are unique) except for renormalizations of the form $R \to \kappa R$, $C \to C/\kappa$; and further, the Sinkhorn iteration on U will converge to R, C, and S and this convergence will ultimately behave geometrically.*

Our new suggestion is to compute the Sinkhorn row and column scalings R and C for the vote-matrix U. Again the "add one" definition of U is recommended, allowing us to use the simpler Sinkhorn theorem concerning matrices with all entries positive. Then the Sinkhorn rating of chessplayer $\#p$ is C_{pp}/R_{pp}. See §6 for a formally-stated algorithm.

Like Keener-eigenvector ratings, Sinkhorn ratings then are positive numbers, and only are defined only up to a common rescaling factor but are otherwise unique. Both kinds of ratings may be defined completely uniquely by agreeing on some normalizing scheme, for example by scaling all N Sinkhorn ratings to make their geometric mean be 1.

We shall see that there are other parallels as well, relating to the fact that both rating schemes have interpretations in terms of Markov chains.

In contrast to Keener-eigenvector ratings (EQ 4), Sinkhorn ratings (obviously) obey

Lemma 5 (Sinkhorn reversal symmetry). *The Sinkhorn ratings arising from U^T are the elementwise reciprocals of those arising from the matrix U, i.e. reversing all vote-preference orderings (or all game results) causes each Sinkhorn rating x to become $1/x$. (Or, if we prefer [54] to use the logs of these ratings, then reversal negates them.)*

5 Laplace's rule in Bayesian statistics

P.S.Laplace was once asked[11]: given that in N independent coin-flipping experiments, H heads appear ($0 \leq H \leq N$) what should we deduce about the probability p that the next coin toss will yield a head?

[9] Actually, according to Bregman this iteration was used in Russia in the 1930s.

[10] Incidentally, the contraction behavior of the map $\vec{x} \to A\vec{x}$ in the Hilbert projective metric with contraction constant $K < 1$ if $\min_{j,k} A_{jk} > 0$, immediately proves the Perron theorem 2. The Hilbert metric thus can be used to say something useful about both the Perron and Sinkhorn problems.

[11] Actually, the story was that he was asked to give odds against the sun rising tomorrow, given that it rose on N previous days. Laplace suggested the odds $N + 1$:1.

The naive estimate would be $p = H/N$. However, if $H = 0$ this yields $p = 0$, and since in that circumstance all the statistical experimental error (if there is any) is necessarily *one-sided*, this estimate is probably biased, i.e. likely to be an underestimate. Can we suggest an unbiased estimator?

If we suppose ("uniform prior") that all p within $0 \le p \le 1$ are equally likely to arise from equal-measure subsets, then the expected value of p is

$$\overline{p} = \frac{\int_0^1 p^{H+1}(1-p)^{N-H}\mathrm{d}p}{\int_0^1 p^{H}(1-p)^{N-H}\mathrm{d}p} = \frac{H+1}{N+2}. \tag{13}$$

This is Laplace's unbiased estimator.[12]

In the voting or gameplaying context: if there are G games between Amy and Bob and Amy wins W of them (and loses $G-W$), then we estimate the probability of Amy winning their next game to be $(W+1)/(G+2)$. If among N random voters H prefer candidate Amy over Bob (and $T = N - H$ prefer Bob) then an unbiased estimate of the Amy:Bob preference ratio is $(H+1):(T+1)$. This provides some justification (besides the fact that it makes Keener and Sinkhorn work better) for our "add one" suggestion for each U_{ab}.

6 Formal algorithm statements

The first step in the algorithm is to get a matrix U that summarizes the votes or game results. Since there could be several ways to do that, it is best to regard the $N \times N$ matrix U (rather than the votes or game results themselves) as the "input" of our algorithms. But one concrete suggestion is to let U_{ab} be *one plus* the number of wins of a over b (or number of voters expressing the preference $a > b$). That is motivated both by Laplace's rule and by the fact that if all $U_{ab} > 0$ then the algorithms below will converge ultimately geometrically.

procedure Keener-rating

Start with $\vec{x} = (1, 1, 1, \dots, 1)^T$.
repeat
  $\vec{x} \leftarrow U\vec{x}$;
  $y \leftarrow \sum_h x_h$;
  $\vec{x} \leftarrow \vec{x}/y$;
until $\vec{x}$ and y have converged
Output $x_1, x_2, \dots, x_N$.

procedure Sinkhorn-rating

Start with $\vec{r} = \vec{c} = (1, 1, 1, \dots, 1)$ and $S = U$.
repeat
  for $k = 1, \dots, N$ **do**
    $r_k \leftarrow r_k / \sum_\ell S_{k\ell}$;
  end for
  for $k = 1, \dots, N$ **do**
    $c_k \leftarrow c_k / \sum_\ell S_{\ell k}$;
  end for
  for $k = 1, \dots, N$, $\ell = 1, \dots, N$ **do**
    $S_{k\ell} = r_k U_{k\ell} c_\ell$;
  end for
until $\vec{r}$ and $\vec{c}$ have converged
Output $c_1/r_1, c_2/r_2, \dots, c_N/r_N$, or some agreed normalization of this N-vector (for example we could agree to scale it so its geometric-mean entry is 1).

7 Interpretations in terms of Markov chains, probability, entropy

Keener's eigenvector voting system may also be thought of in the following way. (We assume $\vec{x}$ is the Perron eigenvector of U.) Consider this Markov Chain on the N candidates. You are sitting on candidate A. You select a random voter v (or a random chess game, assuming for simplicity we are discussing a draw-free variant of chess) and a random candidate B. If v prefers B to A then you jump to candidate B, otherwise you stay with A. Keep doing this process forever. You tend to sit on better candidates for a larger fraction of the time than poor ones. The fraction of the time you will be at candidate n is x_n.

But, this interpretation really only is true assuming

1. We have *not* adopted the "add one suggestion,"
2. All voters provide *full* preference orders as their votes (or, with chessplayers, every player plays each other player an equal number of times according to a "round robin" tournament).
3. We agree to make U_{aa} be whatever value is required to make all columns of U have equal sum E, namely $U_{aa} = E - \sum_{b \ne a} U_{ba}$.

But, in various ways, it would be possible to enforce these rules about the structure of U even if we do not have a round-robin tournament or not all voters provide full preference rankings. All that is necessary is to have some suitable agreed-on procedure for constructing the matrix U. That is one reason why we said that Keener's idea leads to a "family" of methods, not just one – there is considerable freedom of choice here. For example if a plays 20 chess games versus b but c plays only 3 games versus d, we could agree to multiply all the c-d game results by 20/3 and thus "simulate" a true 20-round-robin tournament. This would allow the Markov chain interpretation but would not necessarily be desirable because it would overemphasize the importance of the cd games, "distorting the data."

One of the joys of **Sinkhorn** voting is that no such unpleasant choice is forced upon us – there is a nice Markov Chain interpretation even with *un*distorted data, and there is no need to do anything peculiar with the diagonal entries of U.

Here is that interpretation: R_{pp} and C_{pp} are "amplification factors" associated with chessplayer (or political candidate) #p. Namely, R_{pp} is the factor by which we need to "amplify" p's win-count (row p of U), while C_{pp} is the factor by which we need to amplify p's loss-count (column p of U) in order to make U become the transition matrix of Markov chain with *uniform* stationary distribution. Thanks to the Sinkhorn theorem, p's Sinkhorn rating C_{pp}/R_{pp} exists and is unique (except for an arbitrary normalization factor that is independent of p). For the weakest chessplayers p, we expect that C_{pp}/R_{pp} will be small since we would expect $0 \ll R_{pp} < 1$ and $0 < C_{pp} \ll 1$, i.e. large amplification of their win-counts and deamplification of their loss-counts would be required to make weak players look the same as strong ones. For the strongest

[12] The integrals are of "Euler beta function" type, which is why they can be done in closed form. Thus $1/\int_0^1 p^H(1-p)^{N-H}\mathrm{d}p = (N+1)\binom{N}{H}$.

chessplayers p, the situation is the opposite: we expect that C_{pp}/R_{pp} will be large since we would expect $0 < R_{pp} \ll 1$ and $0 \ll C_{pp} < 1$.

Furthermore, if two players a and b with Sinkhorn ratings s_a, s_b (normalized so that $s_a + s_b = 1$) were to play a G-game chess match, we would expect (in the absence of other information about a and b) that the $a : b$ win-ratio probably would be approximately

$$(G+2)s_a - 1 \, : \, (G+2)s_b - 1 \tag{14}$$

i.e, in the $G \to \infty$ limit (which seems the most relevant if we want to estimate win probabilities from ratings – the Laplace rule presumably already was applied when those s_a were computed and it is senseless to use it a second time) just

$$s_a \, : s_b. \tag{15}$$

EQ 14 arises because Sinkhornizing the matrix

$$\begin{pmatrix} 1 & A+1 \\ B+1 & 1 \end{pmatrix} \tag{16}$$

(corresponding to a $G = A + B$-game match with A wins for a and B wins for b) leads to $R_{aa} = \sqrt{B+1}$, $R_{bb} = \sqrt{A+1}$, $C_{aa} = (A+1)\sqrt{B+1}/K$, $C_{bb} = (B+1)\sqrt{A+1}/K$, where $K = (B+1)(A+1)[1 + \sqrt{(B+1)(A+1)}]$, i.e. to Sinkhorn ratings

$$s_a = \frac{A+1}{G+2}, \quad s_b = \frac{B+1}{G+2} \tag{17}$$

precisely corresponding to the Laplace-rule estimates of win-probabilities.

Thus Sinkhorn ratings are not just meaningless numbers – EQ 15 allows them to be **used by gamblers** to determine the odds for their next bet.

The Kullback-Leibler distance result in theorem 3 II also allows us to interpret Sinkhorn ratings as **minimum entropy statistical estimators.** Specifically, KLdist(p, q) is the "relative entropy" [12] of p with respect to q. KL distance can also be interpreted as the needed extra message-length per datum for sending messages distributed as p, if the messages are encoded using a code optimal for distribution q. Equivalently, choosing p to minimize KLdistance causes p to represent that "theory" of the world that is "simplest" thus capturing "Occam's razor." I.e. p is the "theory" describable (for those whose already know the "data" q) with the fewest bits – and hence which seems "maximally likely" (assuming #bits is proportional to negated log likelihood).

Therefore, the Sinkhorned matrix S may be regarded as the Markov chain which has uniform stationary distribution and whose transition matrix is the closest in KLdistance to the original matrix U, i.e. which has the smallest relative entropy with respect to U, i.e. which is the "theory of the world"[13] which is the "simplest" given that we already know the "data" U.

8 Voting system properties of Keener-eigenvector and Sinkhorn

We shall now examine "voting paradoxes" in both voting systems.

An example of "favorite betrayal" in both systems: In the following 19-voter election

#voters	their vote
8	$B > C > A$
6	$C > A > B$
5	$A > B > C$

both Sinkhorn voting and Keener eigenvector voting (and most other voting systems) exhibit "favorite betrayal." In other words, these (and most) voting systems are flawed in the sense that "dishonest exaggeration pays."

There is a Condorcet cycle, and the winner is B under either Keener-eigenvector or Sinkhorn voting or most other voting systems, such as Plurality, Borda, and Condorcet Least Reversal.

But if the 6 $C > A > B$ voters insincerely switch to $A > C > B$ ("betraying their favorite" C) then A becomes the winner under all these voting systems (and is the Condorcet-winner), which from their point of view is a better outcome.

My favorite voting system,[14] "range voting," does *not* exhibit favorite betrayal in 3-candidate elections, in the sense that each voter's strategically best vote orders the 3 candidates (via the $\ge$ relation) in a manner that is not incompatible with that voter's honest ordering. However, range voting seems not useable for the purpose of rating chessplayers based on tournament results.

Let us now confirm the above statements. In the situation above, the U-matrix (without using the "add one" option) is the left-hand matrix below:

$$\begin{pmatrix} 0 & 11 & 5 \\ 8 & 0 & 13 \\ 14 & 6 & 0 \end{pmatrix} \qquad \begin{pmatrix} 0 & 11 & 11 \\ 8 & 0 & 13 \\ 8 & 6 & 0 \end{pmatrix} \tag{18}$$

and alteration of the 6 $C > A > B$ votes to $A > C > B$ changes it to the right-hand matrix (in which A is a Condorcet winner). The eigenvectors are respectively

$$(0.301, \underbrace{0.361}_{B \text{ wins}}, 0.338) \text{ and } (\underbrace{0.372}_{A \text{ wins}}, 0.353, 0.275) \tag{19}$$

with respective eigenvalues 18.841 and 18.551.

The Sinkhorn ratings are

$$(0.789, \underbrace{1.164}_{B \text{ wins}}, 1.089) \text{ and } (\underbrace{1.237}_{A \text{ wins}}, 1.164, 0.695). \tag{20}$$

Dark-horse+3 pathology: A different and perhaps even more damaging kind of strategic voting pathology, which both Keener, Sinkhorn, and a great number of other ranked voting systems exhibit, is the "dark horse 3" pathology. There are 4 candidates, A, B, C, D and all voters agree that the "dark

[13] Where "permissible theories" are Markov chains with uniform stationary distribution.

[14] Your vote in an N-candidate range election is an N-vector of numbers each in the range $[0, 1]$. The vectors are averaged and the candidate with the highest mean wins.

horse" D is inferior to A, B, and C. However, the voters disagree about the best choice among $\{A, B, C\}$. Therefore, to maximize the impact of their vote, each ranks his favorite candidate top, D artificially second, and the other two candidates artificially last (in random order). Assuming the A-, B-, and C-fans are roughly equinumerous, the result will be a U matrix proportional to

$$\begin{pmatrix} 0 & 3 & 3 & 2 \\ 3 & 0 & 3 & 2 \\ 3 & 3 & 0 & 2 \\ 4 & 4 & 4 & 0 \end{pmatrix}. \tag{21}$$

Then the Perron eigenvector is $(0.229, 0.229, 0.229, 0.314)$ and the Sinkhorn ratings are $(0.841, 0.841, 0.8411.682)$ so that, in either system, the "sure loser" dark horse D *wins.*

Example of failure to elect Condorcet-Winner: When a Condorcet-winner ("beats-all winner") exists, neither Keener nor Sinkhorn necessarily elects him. In the 4-candidate election among $\{A, B, C, D\}$ with the following U-matrix,

$$U = \begin{pmatrix} 0 & 11 & 11 & 11 \\ 9 & 0 & 20 & 20 \\ 9 & 0 & 0 & 20 \\ 9 & 0 & 0 & 0 \end{pmatrix} \tag{22}$$

the Condorcet-Winner is A but B is the Keener winner because the Perron eigenvector is $(0.309, 0.372, 0.205, 0.113)$ and B's entry 0.372 is maximum.

If we add 1 to all entries of the U matrix to get

$$U = \begin{pmatrix} 1 & 12 & 12 & 12 \\ 10 & 1 & 21 & 21 \\ 10 & 1 & 1 & 21 \\ 10 & 1 & 1 & 1 \end{pmatrix} \tag{23}$$

the Condorcet-Winner is A but B is the Sinkhorn winner because the Sinkhorn ratings are $(1.147, 4.276, 0.955, 0.213)$. With Keener, B also still uniquely wins since the Perron eigenvector is $(0.304, 0.357, 0.213, 0.127)$. For the original U matrix without 1s added, Sinkhorning converges very slowly and no row- and column-scalings, and hence no finite Sinkhorn ratings, exist; but the Sinkhorn rating of B appears to go to ∞ faster than any other Sinkhorn rating, in that scenario. ▲

However, we claim

Theorem 6 (Neither Keener nor Sinkhorn can elect Condorcet-loser). *In a k-fold round-robin chess tournament, or election in which each voter provides a full ranking as his vote, neither the Keener nor the Sinkhorn tournament/election winner can be a Condorcet loser, i.e. neither can be a player with a losing record against each other player.*

Proof: Keener: Let $\vec{x}$ be the Perron eigenvector. The jth row-sum of U where j corresponds to a Condorcet loser, must be below average, and indeed the jth *weighted* row sum with any fixed positive weights – in particular if we use $\vec{x}$ as the weights – must be below average, and hence cannot be maximal. But $U\vec{x}$ is proportional to $\vec{x}$! Thus the assumption that j wins the election, i.e. that $x_j = \max_k x_k$, leads to a contradiction.

Sinkhorn: Same proof, except use the column scaling factors as the "positive weights," and then the largest weighted row-sum tells us the Sinkhorn winner. It cannot correspond to a below-average weighted row-sum, such as must happen for a Condorcet loser. □

"Add-top" and "no-show" failures? If, in an election, the addition of some set of identical votes all ranking candidate A top, prevents A from winning, that is an "add-top failure." More generally, if adding some set of identical votes causes the election to yield a worse result (from the point of view of those votes), that is a "no-show paradox." (So called because those voters would have been better off "not showing up.")

On the left is the U-matrix (e.g. B beats A pairwise 17 votes to 1) of a hypothetical 4-candidate election.

$$\begin{pmatrix} 1 & 3 & 1 & 9 \\ 15 & 1 & 10 & 3 \\ 17 & 8 & 1 & 7 \\ 9 & 15 & 11 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 4 & 2 & 10 \\ 15 & 1 & 11 & 4 \\ 17 & 8 & 1 & 8 \\ 9 & 15 & 11 & 1 \end{pmatrix} \tag{24}$$

On the right is the U-matrix after adding one extra vote $A > B > C > D$, represented by the middle matrix. The Sinkhorn ratings arising from the left matrix are $(0.385, 1.181, 1.492, 1.473)$ representing the ranking $C > D > B > A$, and from the right matrix are $(0.467, 1.160, 1.356, 1.361)$ representing the ranking $D > C > B > A$. (Both vectors have been normalized to have unit geometric mean.) This situation is a Sinkhorn no-show failure because the extra votes ranking D bottom caused D to win the election. (However, Keener regards $D > C > B > A$ in both situations.) Due to our reversal-symmetry lemma 5, in the same scenario with all votes reversed, an extra vote ranking D top would cause the Sinkhorn election to rank D last.

I am unaware of any add-top failure situation for Sinkhorn or Keener. However, Keener can exhibit a milder kind of add-top "failure." On the left is a U-matrix (e.g. B beats A pairwise 19 votes to 1 vote):

$$\begin{pmatrix} 0 & 19 & 19 & 17 \\ 1 & 0 & 17 & 2 \\ 1 & 3 & 0 & 2 \\ 3 & 18 & 18 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 20 & 20 & 18 \\ 1 & 0 & 18 & 3 \\ 1 & 3 & 0 & 3 \\ 3 & 18 & 18 & 0 \end{pmatrix} \tag{25}$$

On the right is the U-matrix after adding one extra vote $A > B > C > D$. The Perron eigenvector of the left matrix is $(0.494, 0.134, 0.081, 0.292)$ and of the right matrix is $(0.485, 0.145, 0.088, 0.282)$ where both vectors have been normalized to have unit sum. In both cases A is the winner, but A's eigenvector entry has *diminished* in the second case, which is somewhat "paradoxical." Because A still wins, this is not a failure of "add-top" for Keener eigenvector voting, but it is a failure in the weaker sense that adding an A-top vote actually diminishes A's rating. In the chessplayer view, this is very odd. It means that adding an additional round of games in which A wins all of his games (i.e. one win versus every other player), actually causes A's rating to decrease.

This kind of pathology appears to be *extremely* rare with Sinkhorn ratings – but (a slightly milder form of) it still can happen:

$$\begin{pmatrix} 1 & 2 & 17 & 11 & 11 \\ 16 & 1 & 10 & 7 & 1 \\ 1 & 8 & 1 & 5 & 1 \\ 7 & 11 & 13 & 1 & 1 \\ 7 & 17 & 17 & 17 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 3 & 18 & 12 & 12 \\ 16 & 1 & 11 & 8 & 2 \\ 1 & 8 & 1 & 6 & 2 \\ 7 & 11 & 13 & 1 & 2 \\ 7 & 17 & 17 & 17 & 1 \end{pmatrix} \tag{26}$$

On the right is the U-matrix after adding one extra vote $A > B > C > D > E$. The Sinkhorn ratings (normalized to have unit geometric mean) for the left matrix are $(1.6537, 0.7169, 0.2909, 0.7184, 4.0368)$ and of the right matrix is $(1.6382, 0.8535, 0.3356, 0.7418, 2.8725)$ and A's rating has diminished.

Here are some additional examples of no-show paradoxes.

$$\begin{pmatrix} 1 & 9 & 8 & 2 \\ 1 & 1 & 1 & 4 \\ 2 & 9 & 1 & 6 \\ 8 & 6 & 4 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 9 & 1 \\ 9 & 1 & 7 & 6 \\ 1 & 3 & 1 & 9 \\ 9 & 4 & 1 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 2 & 6 \\ 9 & 1 & 1 & 2 \\ 8 & 9 & 1 & 4 \\ 4 & 8 & 6 & 1 \end{pmatrix} \tag{27}$$

In the first example, the Sinkhorn ratings are $(1.857, 0.319, 1.300, 1.298)$ corresponding to the ranking $A > C > D > B$. After adding one additional $A > B > C > D$ vote, they become $(2.020, 0.382, 1.136, 1.141)$ corresponding to the ranking $A > D > C > B$. Thus an extra D-bottom Sinkhorn vote caused D's election result to improve from 3rd to 2nd place.

In the second example, the Perron eigenvector is $(0.191, 0.339, 0.236, 0.235)$ corresponding to the ranking $B > C > D > A$. After adding one additional $A > B > C > D$ vote, it becomes $(0.215, 0.338, 0.223, 0.225)$ corresponding to the ranking $B > D > C > A$. Thus an extra D-bottom Keener vote caused D's election result to improve from 3rd to 2nd place.

In the third example, the Perron eigenvector is $(0.1899, 0.1895, 0.3190, 0.3016)$ corresponding to the ranking $C > D > A > B$. After adding one additional $A > B > C > D$ vote, it becomes $(0.2066, 0.2086, 0.3098, 0.2750)$ corresponding to the ranking $C > D > B > A$. In this case an extra A-top Keener vote caused the election to rank A last.

Monotonicity: A rating system is "monotonic" if switching an adjacent preference $A > B$ to $B > A$ in a vote (or in the chessplaying view, reversing the result of the A-B game so B now wins) cannot cause A's rating to increase or B's to decrease.

Are Keener-eigenvector and Sinkhorn voting monotonic? A computer search of millions of random U-matrices arising from round-robin tournaments failed to find any example of a monotonicity failure for either system. That suggested that such failures are either impossible, or else so rare that they are not worth worrying about in practice. But, frustratingly, for a long time I was unable to prove this, until finally a breakthrough allowed the proof. We shall first set the stage by examining the difficulties.

The following theorem goes in the direction of, but seems inadequate to prove, the desired result.

Theorem 7 (Markov chain monotonicity). *Let $p(X,Y)$ and $q(X,Y)$ be irreducible Markov chain transition matrices on the same finite state space. Suppose there exist three distinct states A, B, C such that*

$$p(A,B) < q(A,B), \quad p(A,C) > q(A,C) \tag{28}$$

and that $p(X,Y) = q(X,Y)$ for all other entries (X,Y). Let $\pi_p(\cdot)$ and $\pi_q(\cdot)$ be the stationary distributions. Then $\pi_q(B) > \pi_p(B)$.

Proof (by David Aldous): The key realization is that the expected time between returns to B is $1/\pi(B)$, because if it were anything else, then by the law of large numbers, after a very long time the number of returns would, with probability$\to 1$, tend to a different number than it should, forcing an incorrect occupancy probability $\pi(B)$.

So it suffices to prove that the expected return time to B is smaller in the q-chain. This is fairly obvious. There are two kinds of return paths to B, those that go through A and those that do not. The latter kind are not affected.

The former kind return faster to B after the alteration since it takes 1 step from A to reach B directly, but ≥ 2 steps to reach B through C, and the probability of going directly to B increases, whereas the probability of going to C decreases (and all else stays the same) so the net effect on expected return time to B starting at A is

$$(1-c)\,\mathrm{d}p \quad \text{for some} \quad c \geq 2 \tag{29}$$

which is negative. □

Unfortunately the power of this kind of reasoning is very limited. Consequently almost every other monotonicity question about Markov chains still seems open![15]

Here are two kinds of Markov chain *non*monotonicity:

Example #1: Let A and A' be connected in a 2-cycle. Let B and B' be connected in a 2-cycle. Let there be weak couplings (0.1 probability) of going between A and B in either direction. Finally, let there be a very long directed path from B to A (say length 10^4 arcs) with very small probability (say 10^{-7}) of transitioning from B onto the path. If $P(B,A)$ is decreased by ϵ while $P(B, \text{path})$ is increased by ϵ (in the limit $\epsilon \to 0+$) then the return time T_A should increase by a small amount, but T_B will increase by a larger amount. This is an example of a non-monotonic Markov chain in the sense that decreasing $P(B,A)$ while increasing $P(B,C)$ (both by ϵ) causes $\pi(A) - \pi(B)$ actually to *in*crease. This construction is an approach suggested by Aldous, here made more concrete.

[15] I consulted Markov chain experts D.Aldous and L.Snell, and neither knew of any, or knew how to establish any, other Markov chain monotonicity results besides theorem 7. There are none to be found in all the (many) books on Markov chains. And theorem 7 appears to be an isolated special case; its proof technique seems incapable of proving any related results.

Example #2:

$$\begin{bmatrix} 0 & 1 & 0.02 & 0 & 0 & 0 \\ 0.9 & 0 & 0 & 0 & 0.0001 & 0 \\ 0.1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0.899 & 0 & 0 & 0 \\ 0 & 0 & 0.081 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0.9999 & 0 \end{bmatrix} \quad (30)$$

$$\begin{bmatrix} 0 & 1 & 0.101 & 0 & 0 & 0 \\ 0.9 & 0 & 0 & 0 & 0.0001 & 0 \\ 0.1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0.899 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0.9999 & 0 \end{bmatrix} \quad (31)$$

In the 6-state Markov chain with the first transition matrix, increasing $P(C,A)$ from 0.02 to 0.101 while decreasing $P(C,E)$ from 0.081 to 0 (yielding the second matrix) actually causes $\pi(A)/\pi(C)$ to *de*crease from 1.0100041 to 1.01.

Fortunately, we now present a simple new techique of immense power for proving Markov chain monotonicity results. It s based on our new characterization (theorem 13) of Markov stationary distributions and Perron eigenvectors as the solution of a concave-∪ minimization problem, and the following

Lemma 8 (Perturbation of concave-∪ minimization problems). *Let $F(\vec{x})$ be a concave-∪ function with a (necessarily unique) minimum at $\vec{x} = \vec{m}$. If a perturbation $G(\vec{x}) = F(\vec{x}) + \delta F(\vec{x})$ is made, where δF is a smooth function which is (i) bounded and (ii) has gradient $\vec{g} \neq \vec{0}$ at $\vec{m}$, and (iii) is (non-strictly) concave-∪, then the minimum $\vec{n}$ of $G(\vec{x})$ (which due to iii is still necessarily unique) obeys $(\vec{n} - \vec{m}) \cdot \vec{g} < 0$.*

Proof: This is actually merely a special case of the much more general (and well-known [21]) fact that the minimum of a strictly concave-∪ function must lie strictly within the "downward" halfspace arising by creating a hyperplane through the current position that is orthogonal to the gradient vector there. □

Corollary 9 (Keener monotonicity). *Keener eigenvector voting is a monotonic voting system: perturbing the U matrix by adding ϵ to U_{ab} and subtracting ϵ from U_{ba} (in the limit $\epsilon \to 0+$ and assuming both U_{ab} and U_{ba} are positive) cannot cause $x_a - x_b$ to decrease where $\vec{x}$ is the Perron eigenvector of U.*

Many other monotonicity statements about Markov chains ought to be provable in a similar fashion. The same lemma 8 instead applied to the concave-∪ minimization problem in theorem 3(VI) leads to

Corollary 10 (Sinkhorn monotonicity). *Sinkhorn voting is a monotonic voting system: perturbing the U matrix by adding ϵ to U_{ab} and subtracting ϵ from U_{ba} (for any $\epsilon > 0$ and assuming both U_{ab} and U_{ba} are positive) cannot cause s_a/s_b to decrease where $s_p = C_{pp}/R_{pp}$ are the Sinkhorn ratings arising from U.*

Failure of "immunity to clones": Neither Keener nor Sinkhorn is "immune to clones" as the following example demonstrates. The U-matrix on the left

$$\begin{pmatrix} 0 & 2 & 1 \\ 1 & 0 & 2 \\ 2 & 1 & 0 \end{pmatrix} \qquad \begin{pmatrix} 0 & 2 & 1 & 1 & 1 \\ 1 & 0 & 2 & 2 & 2 \\ 2 & 1 & 0 & 2 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 2 & 1 & 2 & 1 & 0 \end{pmatrix} \quad (32)$$

represents a Condorcet cycle situation with the three candidates A, B, C tied. It changes to the matrix on the right upon splitting C into three cloned candidates C_1, C_2, C_3 themselves forming a tied Condorcet cycle. The eigenvectors are

$$(1,1,1)/3 \text{ and } (0.177,\, 0.229,\, 0.198, 0.198, 0.198) \quad (33)$$

respectively so that the cloning of C breaks the tie[16] and causes B to become the winner. The same thing happens under Sinkhorn; the Sinkhorn ratings are

$$(1,1,1) \text{ and } (0.763, 1.311, 1.000, 1.000, 1.000) \quad (34)$$

respectively.

However, somehow these clone problems do not seem very serious because in 1-on-1 contests, cloning the candidates does not seem to affect Keener or Sinkhorn.

9 Runtime bounds for these and a smorgasbord of other algorithms

Runtimes for the procedures of §6. Assuming our "add one" suggestion for obtaining U from the V votes is adopted so that $\min_{j,k} U_{jk} \geq 1$, then the Keener-rating procedure in §6 will only need to perform $O(V|\log \epsilon|)$ powering iterations to reduce the vector L^2 norm of the additive error in $\vec{x}$ below ϵ. (This also is true in the chess-game picture with V games.) This follows from the upper bound on $|\lambda|/r$ with $\lambda \neq r$ given in theorem 1. Keener-rating's total runtime is therefore $O(VN^2|\log \epsilon|)$ at most.

The exact same argument, but now based instead on the upper bound on the contraction constant in theorem 1(V), proves that the same upper bound holds for Sinkhorn-rating's total runtime.[17]

These theoretical runtime bounds are best possible for any voting algorithm, in the sense that simply computing the matrix U from the V ranked ballots takes order VN^2 steps,[18] which is the same order as the runtime bound for the subsequent iteration for any fixed $\epsilon > 0$.

[16] And of course, by slightly perturbing this example we can cause the original scenario to have no ties and to have anybody we want as winner.
[17] We have used the obvious fact that each Sinkhorn or powering step runs in $O(E+V)$ steps, i.e. $O(N^2)$ steps for a dense matrix.
[18] At least employing the naive method where each ranked-ballot is converted to an $N \times N$ matrix, which then is added on to the total matrix.

Experimentally, the above runtime bounds are often unduly pessimistic. For square $N \times N$ matrices with random elements independently sampled from the uniform distribution on $[0, 1]$, experiments indicate that ≤ 7.5 Sinkhorns are required (on average) to cause convergence to the point where $\det C > 0.999999999$. This is true regardless of N: the slowest convergence is with $N = 2$ when 7.5 Sinkhorns are needed (on average); but when $N \geq 100$ only 3 Sinkhorns are needed.

For the same kind of matrices, experiments indicate that ≤ 20 matrix-vector multiplication iterations are required (on average) to cause convergence of the Perron eigenvector $\vec{x}$ to the point where $|\vec{x} - \vec{x}_{\text{old}}|^2 \leq 10^{-19}$. Again this is true regardless of N. Indeed ≈ 20 iterations are required (on average) when $N = 2$ but for $N \geq 99$ at most 7.1 are needed.

The performance in these experiments in undoubtably aided by the fact that this way of producing random matrices tends to cause the Perron eigenvector to be near the initial guess $(1, 1, \dots, 1)$ and the matrix to be about $N^{-1/2}$-nearly doubly stochastic, when N is large.

We shall now discuss pretty much all the other notions in this paper, but now algorithmically.

The condition in theorem 1 that the $N \times N$ matrix U have all entries non-negative and that U^k have all entries positive when $k = (N-1)^2+1$, may be checked in $O(N^2)$ steps plus the amount of time required to square U successively $\lceil 2 \log_2 N \rceil$ times in all.[19]

Deciding whether an N-vertex digraph is **cycle coverable**, or equivalently whether its adjacency matrix has nonzero permanent, or equivalently whether its bipartite double graph has a perfect matching, may be accomplished in $O(N^{2.376})$ arithmetic operations by replacing the nonzero entries of the matrix by random numbers mod P (for some large random prime P, with, say $N^2 < P < 2N^2$) and computing its determinant in mod P arithmetic via the fastest known algorithm. If the result is nonzero, then so is the permanent. If the result is 0, then either the permanent is 0, or it is nonzero and we were unlucky, but such bad luck happens with probability $\leq N/P$ by the Schwartz-Zippel lemma [45] (and the error probability may be reduced to exponentially tiny values by repeatedly rerunning the algorithm with independent random numbers each time).

The fastest known *deterministic* algorithm for deciding if there is a perfect matching (and which will **find a maximum-cardinality matching**) takes $O(E\sqrt{V})$ steps for a V-vertex, E-edge bipartite graph [17], i.e. $O(N^{2.5})$ for us. However, on a large class of random graphs (selection of edges according to biased coin flips) this algorithm requires only $O(V^{1.5} + E \log V)$ expected steps, i.e. $(N^2 \log N)$ for us, essentially because the augmenting paths it uses are usually only logarithmically long [4].

To check whether this digraph is **fully cycle coverable**, or equivalently whether its bipartite double graph always has a perfect matching containing edge e, no matter what e you choose, can be accomplished in $O((V+E)E)$ steps, i.e. $O(N^4)$ for us. To do that we find a perfect matching and then for each edge $e = (a, b)$ not yet known to be in a matching we decide whether an "augmenting cycle" exists in the current matching that joins a to b. This decision process may be accomplished by actually finding that cycle (consisting of edges alternately in and not in the current matching) by an $O(V + E)$-time graph exploration reminiscent of breadth-first search. The exploration starts at a and explores suitable non-e edges until either running out of edges or arriving at b after an odd number of edge-traversals.

To decide whether an $N \times N$ matrix U is **fully indecomposable**, we may use the following lemma of Dustin Stewart ([55] theorem 4.1). Because we may find the strongly connected components of a directed graph in $O(V + E)$ steps [56], this leads to an $O((V + E)V)$-step decision-procedure, i.e. for us $O(N^3)$ steps.

Lemma 11 (Stewart's characterization of full indecomposability). *The adjacency matrix U of a "tournament"[20] directed graph D is fully indecomposable if and only if (1) D is strongly connected and (2) for all vertices v, $DG-v$ does not contain a single-vertex strong component.*

Unfortunately, the Stewart characterization, and hence our $O(N^3)$-time test, are only *applicable* if $U^T + U + I$ has all positive entries: In the chessplayer view, that means each pair of players have played at least one game, and in the voting view that means for each pair of candidates, at least one voter has expressed a preference for one over the other.

How can we test full indecomposability for fully general matrices U? We can employ the following known[21]

Lemma 12 (Characterization of full indecomposability). *A matrix U is fully indecomposable if and only if (1) its nonzero pattern is the same as the nonzero pattern of some doubly-stochastic matrix and (2) its directed N-node graph is strongly connected.*

We can test (2) in $O(E + V)$ steps [56], and can test (1) by linear programming (the conditions for double stochasticity with a given nonzero pattern form a linear program). Thus there is a polynomial-time algorithm for testing matrix full-indecomposability. Unfortunately the runtime upper bounds for this algorithm are extremely large and in fact the best test I presently know involves simply using the best available Sinkhorn balancing algorithm to find an ϵ-nearly doubly-stochastic matrix where (by bounds mentioned in §10.2) it suffices to take $\epsilon = N^{-3N}$ (where we assume wlog that the original matrix is boolean) to get a clear discrimination. That yields a $O(N^{5.01})$-step test by using either the Kalantari-Khachiyan algorithm [23], the Nemirovsky-Rothblum algorithm [37], or the present paper's new strong polynomial algorithm (§10.8). The fact that all three of these (rather poor) time bounds happen to agree suggests that perhaps the present paper's new strong polynomial algorithm is not easily improveable. *But* by using the *sped up* version of Kalantari-Khachiyan that employs fast matrix multiplication (mentioned in our §10.2) the runtime is reduced to $O(N^{4.376})$ steps.

The fastest algorithm I know of that **both**

[19] We use here the fact that if U has non-negative entries and U^k has all entries positive for some $k \geq 1$, then U^m has all entries positive for each $m \geq k$. E.g. see ex.3 p.522 of [22].

[20] A "tournament" is a directed graph that has a arc joining each pair of vertices in one or the other direction, i.e. is such that $U + U^T + I$ has all positive entries.

[21] It is deducible from theorem 3.

1. finds a perfect matching of maximum weight-sum in an $N+N$-vertex bipartite graph with edge weights, and *also*
2. solves the dual linear program by finding two N-vectors $\vec{r}$ and $\vec{c}$ such that (i) $r_a + c_b = W_{ab}$ if the edge ab (with weight W_{ab}) is in the matching and (ii) $r_a + c_b \geq W_{ab}$ and (iii) $\sum_a r_a + \sum_b c_b$ is minimum possible,

is the Kuhn-Munkres "Hungarian method for solving the assignment problem" (chapter 11.2 of [39]). It runs in $O(N^3)$ steps.

10 Other algorithms for Sinkhorn balancing and for finding the Perron-Frobenius eigenvector

The simple Sinkhorn and powering iterations of §6 converge ultimately geometrically if the matrix has all entries positive, and indeed our theorems 1 and 3 have provided upper bounds on the contraction constants. And for our applications in voting *if the "add-one" suggestion is adopted* our matrix *does* have all entries positive, and indeed in $[1, V+1]$, leading to good upper bounds ($\leq 1 - 1/(V+1)$) on the contraction constants, and hence to excellent runtime bounds.

However, for matrices with nonnegative entries, some of which may be *zero*, we have *not* given any upper bound below 1 on the contraction constant. Therefore, these algorithms might run slowly.

So in this section we are going to discuss more sophisticated algorithms, some new, designed to try to guarantee fast runtimes even for matrices with some very small or zero entries.

What do we mean by "fast"? We would like our algorithms to run in a number of bit-operations *polynomial* in the bitlength of the input matrix (assumed to consist entirely of nonnegative integers) and $|\log \epsilon|$, to get a final answer with accuracy parameter ϵ. The Sinkhorn and powering iterations of §6 fail to meet this criterion and both can require exponential time.

Another possible criterion – "*strong* polynomial time"[22] – would be that our algorithms run in a number of arithmetic operations polynomial in $|\log \epsilon|$ and in the *size* N of the $N \times N$ input matrix, *regardless* of its contents. Best of all would be to be in the intersection of both kinds of polynomial time.

10.1 The NP-hardness of Sinkhorning general matrices

A symmetric matrix U is *singly-scalable* if there exists a positive diagonal matrix X such that the row and column sums of XUX are all ones. A matrix A is *doubly-scalable* if there exist positive diagonal matrices X, Y so that the row and column sums of XAY are all ones. (Note: here the entries of U, A, XUX, and XAY can include *negative* entries.) Khachiyan [27] showed both that testing the single-scalability of an arbitrary symmetric matrix U is NP-hard, and that testing the double-scalability of an arbitrary matrix A is NP-hard. (This is true even if A and U are $N \times N$ and its entries are integers $\leq N$ in absolute value.) Double scalability of A is equivalent to single scalability of $\left(\begin{smallmatrix} 0 & A \\ A^T & 0 \end{smallmatrix}\right)$ so that the single-scalabilty NP-hardness statement is a corollary of the double-scalability one.[23]

Thus the demand that our matrix's entries be *nonnegative* is essential to have any hope of fast Sinkhorning. This is not at all surprising since it was also essential to make the minimization problem in theorem 3(VI) be concave-∪, i.e. easy.

10.2 Weakly-polynomial-time Sinkhorning algorithms from concave-∪ minimization

Recall from theorem 3 part VI [42] that Sinkhorn balancing may be phrased as a concave-∪ minimization problem in $(2N-2)$-dimensional space. Therefore it may be solved by general purpose optimization algorithms [57] in (weakly) polynomial time.

Indeed Kalantari & Khachiyan [24] show by using the ellipsoid method [21] that if the maximum row-sum of U is V and its minimum nonzero entry is 1, then $O(N^4 [\ln \frac{N}{\epsilon}] \ln V)$ arithmetic operations suffice to find row- and column-scaling factors that make U become ϵ-nearly doubly stochastic. If we use Vaidya's convex programming method [57] instead of the ellipsoid method, that speeds their time bound up to $O(N \mathrm{Matmul}(N) [\ln \frac{N}{\epsilon}] \ln V)$ arithmetic operations. For this purpose it suffices [24] to minimize the potential function to within an additive height accuracy $\pm(\epsilon/4)^2$.

Furthermore, all of these scaling factors will lie in a common real interval $[\ell, u]$ with $0 < \ell \leq u$ and $u/\ell \leq V^{2N+1}$; and this result is near-tight because they have an example matrix U for which $u/\ell \geq V^{N/4}$.

Nemirovsky & Rothblum [37] improved the Kalantari & Khachiyan time bound to then $O(N^4 \ln[N \ln(NV)/\epsilon])$ but by using a more complicated algorithm. This also may be sped up with Vaidya's method.

Further, the convergence can be made ultimately quadratic in nature by eventually switching, e.g, to a conjugate gradient based optimizer [18].

10.3 Weakly-Polynomial-time algorithm for computing Perron-Frobenius eigenvectors, based on matrix powering

If some power A^p of a matrix has all entries positive (even though A itself might not) then the powering iteration will converge ultimately geometrically. It is known (ex.3 p.522 of [22]) that if some power A^p of a matrix A with non-negative entries has all entries positive, then this is also the case for A^{p+1}.

So if we compute some high power A^p of our matrix in $O(\log p)$ matrix multiplications by binary powering, then the usual powering algorithm but with A^p rather than A will converge ultimately geometrically but now with an easily computed bound on the contraction constant. These thoughts lead to the following algorithm. Let $\mathrm{Matmul}(N)$ be the number of

[22] We follow historical convention by using this poor choice of name. A better name would be "polynomial time in the real-number-model."

[23] Khachiyan erroneously made the opposite statement.

arithmetic operations required to multiply two $N \times N$ matrices. It is known [11][9] that $\text{Matmul}(N) = O(N^{2.376})$ because the usual $O(N^3)$-time matrix multiplication algorithm is not optimal.

procedure Faster-Perron-Frobenius
$B \leftarrow U$;
repeat
  $B \leftarrow B^2$;
until $\min_{s,t} B_{st} > 0$ and $\max_{s,t} B_{st} < N^{-2}\text{Matmul}(N) \min_{jk} B_{jk}$
$\vec{x} \leftarrow (1, 1, 1, \dots, 1)^T$;
repeat
  $\vec{x} \leftarrow B\vec{x}$;
  $y \leftarrow \sum_h x_h$;
  $\vec{x} \leftarrow \vec{x}/y$;
until $\vec{x}$ and y have converged ($\max_{j,k} x_k x_j^{\text{old}}/(x_j x_k^{\text{old}}) < 1 + \epsilon$)
Output $x_1, x_2, \dots, x_N$.

Assume the matrix U consists of integers in $[0, V]$. Then in total, at most $O(\text{matmul}(N)[N + \log(VN)] + N^2|\log \epsilon|)$ arithmetic steps are performed. In practice to prevent floating point overflow we would recommend rescaling the matrix after each squaring (and keep track of the scaling factor).

This algorithm could be improved to make it have ultimately quadratic convergence (i.e. so its runtime depends on $\log|\log\epsilon|$ rather than $|\log\epsilon|$) by switching to a different kind of iteration once it starts to work better – which will happen automatically when the eigenvector's entries have small enough relative errors. Newton's or Broyden's method [14] for solving simultaneous nonlinear equations would do.

10.4 Weakly-Polynomial-time algorithm for computing Perron-Frobenius eigenvectors, based on new minimization formulation

If we change variables from $\vec{x}$ to $\vec{y}$ where $x_j = \exp(y_j)$ then we automatically have $x_j \geq 0$.

Theorem 13 (New concave-∪ minimization formulation of Perron eigenvector). *Let U be an $N \times N$ matrix with all entries non-negative and having an eigenvector with all entries positive. Then*

$$F_j(\vec{y}) = \log \sum_k U_{jk} \exp(y_k - y_j) = (y_j)^{-1} \log \sum_k U_{jk} \exp(y_k) \tag{35}$$

are concave-∪ functions of the real N-vector $\vec{y}$, and

$$F(\vec{y}) = \max_j F_j(\vec{y}) \tag{36}$$

is also. F's minimum value is the log of the Perron eigenvalue r and its location gives the Perron eigenvector $\vec{x}$ via $x_j = \exp(y_j)$. (Optionally, we can demand the normalization $\sum_j y_j = 0$ throughout.)

Remark. This also may be thought of as a new concave-∪ minimization formulation of the problem of finding the stationary distribution of an N-state Markov chain with known transition matrix.

Proof: Because its second derivative $K^2 e^{Kz}$ is non-negative, $\exp(Kz)$ is a concave-∪ function of real z for any real K. Since any weighted sum of concave-∪ functions with positive constant weights is concave-∪, we know $\sum_k U_{jk} \exp(y_k - y_j)$ is concave-∪ along any line in $\mathbf{R}^N$, i.e, is concave-∪. More impressively, the *log* of this (namely F_j) is *also* concave-∪, since we indeed claim the log of any positively-weighted sum of exponentials of z, is concave-∪:

$$\frac{\mathrm{d}^2}{\mathrm{d}z^2} \log \sum_j c_j \exp(K_j z) = \frac{\mathrm{d}}{\mathrm{d}z} \frac{\sum_j c_j K_j \exp(K_j z)}{\sum_j c_j \exp(K_j z)} \tag{37}$$

$$= \frac{\sum_j c_j K_j^2 \exp(K_j z)}{\sum_j c_j \exp(K_j z)} - \left(\frac{\sum_j c_j K_j \exp(K_j z)}{\sum_j c_j \exp(K_j z)} \right)^2 \geq 0 \tag{38}$$

where the final "≥ 0" is because we may multiply by $\left(\sum_j c_j \exp(K_j z)\right)^2$ to get that it is equivalent to

$$\left(\sum_j K_j^2 A_j \right) \left(\sum_j A_j \right) \geq \left(\sum_j K_j A_j \right)^2 \tag{39}$$

where $A_j = c_j \exp(K_j z)$. This in turn follows from squaring

$$(\sum_j A_j B_j^2)^{1/2} (\sum_j A_j)^{1/2} \geq (\sum_j A_j B_j) \tag{40}$$

where $A_j \geq 0$, which is just an instance of the Cauchy-Schwartz inequality for one vector whose entries are $A_j^{1/2}$ and another vector whose entries are $A_j^{1/2} B_j$.

Because the max of a set of concave-∪ functions is concave-∪, we have now proven that $F(\vec{y})$ is a concave-∪ function; and the the optional normalization $\sum_j y_j = 0$ is just a restriction to a subspace (which preserves concavity).

It remains to prove that $F(\vec{y})$ has a unique minimum at $\vec{y}$ such that $x_j = \exp(y_j)$ are the entries of the Perron eigenvector, and that the minimum value $\log r = \min F$ is the log of the Perron eigenvalue.

It is immediate from F's definition that *if* $\vec{y}$ is such that $x_j = \exp(y_j)$ are the entries of the Perron eigenvector, *then* $F(\vec{y}) = F_j(\vec{y}) = \log r$ for all j.

Finally, recall that a concave-∪ function has a unique min (it it has any min) and cannot have any maxes or saddlepoints. But we shall now argue that the Perron eigenvector is a stationarizing point. Hence it must be the global min.

That is because the Perron eigenvalue is the spectral radius, so the Perron eigenvector is a local max of $|U\vec{x}|^2/|\vec{x}|^2$. Considering the general inequality for positive numbers

$$\min_j \frac{p_j}{q_j} \leq \frac{\sum_j p_j}{\sum_j q_j} \leq \max_j \frac{p_j}{q_j}, \tag{41}$$

in the special case $p_j = (U\vec{x})_j^2$ and $q_j = x_j^2$ yields the conclusion that all small perturbations of $\vec{x}$, since they approximately preserve $|U\vec{x}|^2/|\vec{x}|^2$, must increase some ratio $(U\vec{x})_j/x_j$ above r (min) or leave it the same (stationarizer). □

It is surprising that theorem 13 has escaped previous notice; it is the first characterization of the Perron eigenvector as the solution of a concave-∪ minimization problem. It is now a standard matter to minimize $F(\vec{y})$ to find the Perron eigenvector and eigenvalue in weakly-polynomial time by the ellipsoid [21] or Vaidya's method [57]. The runtime is $O(LN\text{Matmul}(N)|\log\epsilon|)$ steps where L is the bitlength of the (integer) input matrix.

This constitutes a new and well behaved, weakly polynomial method of finding Perron eigenvectors to arbitrary relative accuracy. Its runtime bound is worse than the powering-based method. However, what is more important is that we shall be able to employ theorem 13 as an ingredient in a *strongly* polynomial time algorithm.

10.5 Strongly polynomial-time algorithm for Perron-Frobenius eigenvectors

This section presents the first strongly polynomial time algorithm for finding the Perron-Frobenius eigenvector of an $N \times N$ matrix U with non-negative entries. It consists of four stages:

1. Verify [56] that the nonzero pattern of U represents a strongly connected directed graph, and more generally verify the condition in theorem 1 as in §9. (If not, then refuse to continue – Perron-Frobenius eigenvector is non-unique, contains non-positive entries, or does not exist.)
2. Schneider-Preprocessing prescales U's rows and columns (scaling the jth row by a factor reciprocal to the factor employed for scaling the jth column) in such a way that U's maximum entry in the kth row is the same as U's maximum entry in the kth column, for each k.
3. We then apply a standard concave-∪ minimizer (as in theorem 13) to find the Perron eigenvector of the rescaled U, to relative error $\ll 1/N$.
4. A quadratically convergent process is employed to decrease the relative error further, to ϵ, and finally the Perron eigenvector $\vec{x}$ of the original matrix is got from the Perron eigenvector of our rescaled matrix by applying the rescaling.

Let us now discuss the three steps in more detail.

1. Runs in $O(V+E)$ steps, i.e. $O(N^2)$ for us [56]; the more general verification, as we showed in §9, takes $O(N^2 + \text{Matmul}(N)\log N)$ steps..

2. Schneider and Schneider [43] called a directed graph with real arc-weights "max-balanced" if for any nontrivial subset W of the vertices, the maximum weight of an arc entering W is the same as the maximum weight of an arc leaving W. Equivalently, a $N \times N$ *matrix* with real entries is max-balanced if any nontrivial subset W of $\{1, 2, \dots, N\}$, the maximum value of an entry in the rows with indices in W is the same as the maximum entry in the columns with indices in W. (But note: the locations of these two equal entries need not be the same.)

They showed that there exists a unique "potential function" P_v so that adding $P_a - P_b$ to the weight of arc $a \to b$ yields a max-balanced directed graph. If U is an $N \times N$ matrix with non-negative entries, and we regard $\log U_{ab}$ as the weight of arc $b \to a$, then the row and column rescaling $\exp(P_b - P_a)U_{ab}$ of U_{ab} will uniquely cause U to become a max-balanced matrix.

The following algorithm finds this scaling. Use Karp's dynamic programming algorithm [25] to find the directed-cycle with maximum mean arc weight in the directed graph. Now choose P_v for v on the cycle, to equilibrate all the weights of the arcs of that cycle (and to make all other, off-cycle, weights be smaller). Now shrink the cycle to a point (making the new arc weights be the maximum of the old arc weights on any set of parallel arcs that results) and do it again. Because Karp's method runs in $O(VE)$ steps if the digraph is strongly-connected, and the shrink reduces the number of vertices of the graph, the Schneider-Schneider rescaling procedure may be carried out in $O(V^2E)$ steps, i.e. $O(N^4)$ steps for an $N \times N$ matrix.[24]

3. After step 2 rescaling U, automatically $\exp F(\vec{y})$ at $\vec{y} = \vec{0}$ is at most a factor of N greater than its minimum value, i.e. $F(\vec{0})$ is at most an additive amount $\log N$ greater than $\min_{\vec{y}} F(\vec{y})$. That is because we know that (i) the maximum row sum of U is an upper bound on the Perron eigenvalue, (ii) the maximum row sum of U exceeds the maximum entry of U by at most a factor of N, (iii) the maximum entry $M = \max_{k\ell} U_{k\ell}$ of U is a lower bound on U's Perron eigenvalue r, because just consider the entries x_j of the Perron eigenvector $\vec{x}$ which happen to lie on the maximum-mean cycle, all of whose arcs ab have equal matrix entries $U_{ab} = M$. In view of $U_{k\ell} \ge 0$, at least one of these x_j must get multiplied by at least M when we replace $\vec{x}$ by $U\vec{x}$. (iv) The Perron eigenvalue is the minimum value of e^F.

So we know that in terms of the *value* of F we are already fairly close to minimal, just by starting from $\vec{y} = \vec{0}$ without even doing any minimization. It is also possible to see that $\vec{y} = \vec{0}$ is also fairly close to the *location* of the min. View $\vec{x}$ (after normalization) as the occupation probability in a Markov chain. Because of the max-balanced nature of the transition matrix, $\max_k x_k \le N^{2N} \min_j x_j$ because no vertex-subset can have more than a factor of N more probability than its complement subset (and consider a chain of at most $2N$ subset inclusions).

Hence we know automatically that the minimizing $\vec{y}$ must lie in a ball of radius $\log(1 + N^{2N}) = O(N \log N)$ centered at $\vec{0}$. It therefore follows immediately that by using Vaidya's minimization algorithm, in $O(N^3 \log \frac{N}{\delta} + N\text{Matmul}(N) \log \frac{N}{\delta})$ steps we can reduce the error in $\vec{y}$ below δ. We suggest choosing δ to be some negative power of N, e.g. N^{-9}.

4. Once the absolute error in the y_j, and hence the relative error in the $x_j = \exp(y_j)$, have been reduced sufficiently – and certainly $\delta = O(N^{-9})$ suffices – we than may switch to a quadratically convergent algorithm for finding the Perron eigenvector $\vec{x}$. That is because small relative changes in the x_j will lead to small relative changes in the $M^{-1}(U\vec{x})_j$, and in its normalized version with sum 1; the effects of these changes are roughly linear and the quadratic and higher terms are neglectible. (Note that it is important here that U be max-balanced and with non-negative entries.) That means that we are well within the local quadratic convergence regime of standard numerical procedures intended to solve N-variable systems of nonlinear equations. Newton's method is the simplest to consider. Each Newton step takes $O(\text{Matmul}(N))$ steps and will roughly square the error.

[24] I do not know if this time bound is improveable.

Finally, once we have found the Perron eigenvector $\vec{x}$ of the *rescaled* U, the Perron eigenvector of the original U may be recovered by dividing x_j by $\exp(P_j)$ for $j = 0, 1, \ldots, N$. (The eigenvalue is unchanged, since the Schneider-Schneider rescaling was a similarity transformation of U.) We conclude:

Theorem 14 (Strongly polynomial algorithm for Perron eigenvector). *Our algorithm will find the Perron eigenvector (and eigenvalue) of an $N \times N$ matrix U with $U_{k\ell} \ge 0$ such that the nonzero pattern of U represents a strongly-connected directed graph, with all $N+1$ outputs accurate to relative error ϵ for arbitrary $0 < \epsilon < N^{-N}$ and $N \ge 2$. The runtime is $O(N^4 + \mathrm{Matmul}(N) \log|\log \epsilon|)$ steps, each an arithmetic operation or* exp *or* log *evaluation on real numbers. This also may be thought of as a strongly polynomial algorithm for finding the stationary distribution of an N-state Markov chain with known transition matrix. The memory requirement is $O(N^2)$ real numbers.*

This runtime is of the the same order as the runtime of the best known weakly polynomial algorithm (in our §10.3), provided that conventional N^3-time matrix multiplication is used there. This suggests that the algorithm of theorem 14 cannot easily be improved.

10.6 Are there other strong polynomial algorithms in eigen-related linear algebra?

Inspired by these successes, one might now ask whether there is a strongly polynomial algorithm for the topmost eigenvector of a general $N \times N$ symmetric matrix (provided a unique topmost eigenvector exists) or for finding the topmost singular vector of a general $N \times N$ real matrix. Or, one might ask if these problems may be formulated as concave-∪ minimization problems. I suspect the answers all are "*no*."[25]

If so, then the Perron-Frobenius eigenvector and Sinkhorn problems are atypically nice cases, perhaps nearly unique in all of linear algebra, where strongly polynomial algorithms exist. Such books as [20] are somewhat misleading about this; they leave the careless reader with the impression that strongly polynomial algorithms exist for these problems, but in fact do not even prove their algorithms are weakly polynomial. Nevertheless, weakly polynomial algorithms do exist for the general-matrix eigenproblem for matrices with rational number entries. That fact may be seen by combining known bounds ([34] section 4.6.1, e.g. his theorem 4.6) on root-separation for polynomials with rational coefficients, with known strongly polynomial algorithms for polynomial rootfinding [38], and finish off with the "inverse iteration" [20] to deduce good approximate eigenvectors from well-isolated good approximate eigenvalues.

10.7 Some experiments

To appreciate the importance of the distinction between a polynomial and superpolynomial number of arithmetic operations, consider the following experimental results.

Experiment #1: We have already described the excellent typical behavior of the simple `Keener-rating` powering iteration on $N \times N$ matrices filled with i.i.d. uniform random numbers in $[0, 1]$: it converges in a small constant number of $\vec{x} \leftarrow M\vec{x}$ iterations, not depending on N and only depending on the precision ϵ requirement for the output.

Experiment #2: But now suppose that instead, our matrix entries are $\exp(Kr)$ where r is a standard normal random deviate (i.i.d. deviates, one per matrix entry) and K is a constant. The experimental results with $K = 5$ and $K = 9$ are in table 10.1. They are quite different from experiment #1! First of all, the expected runtimes are far larger now – sometimes over 100,000 times larger – and they evidently grow *exponentially* with K. Second, these runtimes also have enormous variances.

N	#iterations			
2	31745	1047	572	13752
3	90	81296	1085	1477
4	1319	4346	2023	1917
5	6770	40700	6007	366
6	68473	3839	851	119
2	14251	32719	2613000	52
3	222840	497090	3877	13667000
4	1086100	2640300	514683	21463
5	185250	4631000	511570	2089800

Figure 10.1. Experimental numbers of $\vec{x} \leftarrow M\vec{x}$ iterations needed to get convergence to the point where $|\vec{x} - \vec{x}_{\mathrm{old}}| < 10^{-5}$. Upper portion of table: with $K = 5$. Lower portion: $K = 9$. Each number is the average number of iterations employed during 25 independent runs, so each line of the table summarizes 100 runs. ▲

I have also conducted some experiments with some more sophisticated algorithms that attempted to get strongly polynomial time. These algorithms will not be described here since I abandoned them when I discovered the strongly polynomial algorithm above. Suffice it to say that, on the same matrices as in table 10.1, the new algorithms experimentally require on the order of 1000 times fewer matrix-vector multiplications when $K = 9$, and on the order of 10 times fewer when $K = 5$.

10.8 Strongly polynomial time algorithm for matrix Sinkhorn balancing

The first strongly polynomial time algorithm for Sinkhorning was found by Linial et al [31]. We shall present a much better one here. It consists of three stages:

1. Hungarian-Preprocessing prescales U's rows and columns to maximize $|\det U|$ and $\mathrm{per}\, U$ subject to the constraints $U_{jk} \le 1$. This takes $O(N^3)$ steps.
2. Repeated Sinkhorn steps are done[26] until the matrix is $N^{-1.01}$-nearly doubly stochastic. Thanks to the preprocessing, at most $O(N^{2.02} \ln N)$ Sinkhorn steps are required, each one taking $O(N^2)$ steps.

[25] Because the topmost and second-to-top eigenvalues could be arbitrarily close to one another. It would therefore seem that no strong-polynomial algorithm could isolate the top eigenvalue from the others well enough to prevent too-severe contamination of the top eigenvector by the second-to-top eigenvector (which is orthogonal to it).

[26] It is instead possible to design the algorithm around a minimizer for the concave-∪ minimization problem in theorem 3(VI). Conjecturally that might allow reducing the "5.01" in theorem 15 to 4.01 or 4.38.

3. A new quadratically-convergent iterative process is used to push this further down to ϵ-near double stochasticity. At most $\log|\log \epsilon|$ iterations are required, each one taking $O(\text{Matmul}(N))$ steps.

Let us now discuss the three steps in more detail.

1. A preprocessing method one may apply to a matrix U is the following.[27]

procedure Hungarian-Preprocessing
Apply the Kuhn-Munkres Hungarian method (mentioned at the end of §9) with weights $W_{ab} = \log U_{ab}$ to find a maximum-sum-weight matching and suitable $\vec{r}$ and $\vec{c}$.
for $k = 1, 2, \ldots, N$ **do**
  Scale the kth row of U by $\exp(-r_k)$;
end for
for $k = 1, 2, \ldots, N$ **do**
  Scale the kth column of U by $\exp(-c_k)$;
end for

This takes $O(N^3)$ steps and reduces U to a form in which the N entries of U_{ab} corresponding to the Hungarian matching are 1s, while the remaining entries obey $0 \le U_{ab} \le 1$. This new U has $1 \le \operatorname{per} U \le N!$.

2. One may now apply Sinkhorn iterations until the product of the column sums of U becomes $\ge e^{-\kappa}$. (This product is always ≤ 1 due to the arithmetic-geometric mean inequality, and equality is achieved if and only if U is doubly-stochastic.) The point is that this will take at most $O(\kappa^{-1} N \ln N)$ Sinkhorn steps because (as we shall argue next paragraph) each one increases the permanent by at least a factor of e^{κ} and a total increase factor of at most $\ln(N!)$ is possible. The total runtime (assuming each arithmetic operation takes one "step") is then

$$O(N^3 + \kappa^{-1} N^3 \ln N). \tag{42}$$

Let us analyse this in detail. We shall employ the following refinement ([8] p.98) of the arithmetic-geometric mean inequality:[28] if $\sum_{j=1}^{N} X_j = N$, $N \ge 2$, and $X_j \ge 0$, then

$$\frac{1}{2} + \frac{1}{4}\left(\frac{\max}{\min} + \frac{\min}{\max}\right) \le \frac{1}{P} \tag{43}$$

where

$$P = \prod_{j=1}^{N} X_j, \quad \max = \max_j X_j, \quad \min = \min_k X_k. \tag{44}$$

(The unrefined arithmetic-geometric mean inequality merely states that $P \le 1$ with equality if and only if all the X_j are equal.)

Before each Sinkhorn step, the column sums c_j have average value 1. Therefore, by the arithmetic-geometric mean inequality, their product P obeys $P < 1$. After these sums are used to renormalize the columns (increasing the |determinant| and permanent by a factor of $1/P$), the row sums have average value 1. After they are used to renormalize the rows, the |determinant| and permanent increase further. Therefore, each Sinkhorn step increases both the |determinant| and permanent by a factor of at least e^{κ}.

Now by our refined A-G inequality, once the increase factor $1/P$ goes below e^{κ}, we have

$$\max_j c_j - 1,\ 1 - \min_k c_k \ \le\ O\left(\sqrt{e^{\kappa} - 1}\right) \tag{45}$$

for all sufficiently small κ. That proves $\kappa = O(N^{-2})$ suffices to force all the row and column sums to be within $\pm N^{-1}$ of 1.

3. We now apply some quadratically convergent iteration. Most simply, we could apply a conjugate-gradient minimizer [18] to the objective function in theorem 3(VI). This has the advantage of being very easy to program, and since that objective function is concave-∪, global convergence is guaranteed. Each CG step, which involves computation of both the objective function and its gradient, runs in $O(N^2)$ arithmetic operations. A total of $O(N)$ CG steps are required to complete a "phase." Asymptotically (in arithmetic free of roundoff error), each phase roughly squares the error ("quadratic convergence"). The total runtime of a CG phase is $O(N^3)$ steps.[29]

A theoretically faster globally quadratically-convergent iterative algorithm would involve computing the function value, gradient, and Hessian matrix, all in $O(N^2)$ steps, and then performing a safeguarded Newton iteration (i.e. perform a line-minimization along the line segment to the Newton-approximation to the min). This could be done in only $O(\text{Matmul}(N))$ steps.

We can in fact perform the new iteration in parallel with the old kind from stage 2 and use whichever decreases the error the most. That would enable a seamless transition between stages 2 and 3.

The key question is: how small must $e^{-\kappa}$ be in order to assure us that the quadratic convergence behavior will happen starting *immediately*? For that we need that the function behaves "locally quadratically," i.e. we need the cubic terms to be neglectible.

For that it plainly suffices (considering the Hungarian-Preprocessing – we perform the scaling-factor minimization using that preprocessed U and starting from the initial guess that came from the Sinkhornings in stage 2) if the present relative error on each scaling factor is $\ll N^{-1}$, which as we have seen is assured if $\kappa \ll N^{-2}$.

Theorem 15 (Strongly polynomial algorithm for Sinkhorn balancing). *The total runtime of the 3-stage algorithm is at most*

$$O\left(N^3 + N^{5.01} + \text{Matmul}(N) \log|\log \epsilon|\right) \tag{46}$$

steps to get the Sinkhorn row and column scaling factors all equal to within relative error ϵ.

In practice I suspect it will often run faster because fewer Sinkhorns will be required in stage 2 than our worst case bound $O(N^{3.01})$. Even as is, though, our runtime bound is superior to the only previously known runtime bound for a strongly-polynomial time Sinkhorn balancing procedures [31]

[27] This is a substantial improvement of the first algorithmic idea in [31].
[28] Other refinements, less useful for our purposes, are by Kober [28] and Cartwright & Field [10].
[29] Each "step" is an arithmetic operation or $\exp(x)$ evaluation.

and is comparable to the known weak-polynomial algorithms [24][37] for[30] the case of a boolean $N \times N$ matrix if $\epsilon \approx N^{-N}$.

And our algorithm simultaneously is much simpler to program than all those previous procedures.

11 Conclusion

Sinkhorn ratings seem to have convincing (if small) theoretical and experimental advantages over rating systems like Keener's based on Perron-Frobenius eigenvectors:

1. Sinkhorn ratings have an interpretation as a minimum entropy statistical estimator; Keener ratings do not.
2. Sinkhorn ratings obey "reversal symmetry"; Keener ratings do not.
3. Both Sinkhorn and Keener ratings have an interpretation in terms of Markov chains but far less strain is necessary to make that interpretation in the Sinkhorn case.
4. Sinkhorn ratings can be simply used by gamblers to generate odds for betting on the next chess game; not true for Keener ratings.
5. Experimentally, it seems extremely rare that adding an extra round robin tournament in which A wins every one of his games, can lower A's Sinkhorn rating. But it seems much more common that this can lower A's Keener rating.

Both Sinkhorn and Keener ratings may be interpreted as arising from the unique solution of a concave-∪ minimization problem, both obey and disobey similar voting system properties, and both are computable with simple algorithms (§6) as well as strongly polynomial algorithms.

In my opinion Sinkhorn ratings are a good way to rate chess-players and football teams. But I prefer range voting [51] (for single-winner elections) or reweighted range voting [52] or asset voting [53] (for multiwinner elections) as voting systems, because they seem to behave better in the presence of strategic (i.e. dishonest) voters. For rating chess players, I recommend also reading my [54], whose discussion is beautifully complementary to the present paper, and whose ratings correspond to the logs of our Sinkhorn ratings.

References

[1] Eva Achilles: Implications of Convergence Rates in Sinkhorn Balancing, Linear Algebra & Applications 187 (1993) 109-112.

[2] Foreman S. Acton: Numerical Methods That Work, 2nd printing. Math. Assoc. Amer. 1990. (Original printing Harper & Row 1970.)

[3] A.Bachem & B.Korte: On the RAS algorithm, Computing 23 (1979) 189-198.

[4] H.Bast, K.Mehlhorn, G.Schäfer, H.Tamaki: Matching algorithms are fast in sparse random graphs, pp. 81-92 in STACS 2004, Springer (Lecture Notes in Comput. Sci. #2996) 2004.

[5] G.D. Birkhoff: Tres Observaciones Sobre el Algebra Lineal, Universidad Nacional de Tucuman Revista, Serie A, 5 (1946) 147-151.

[6] G. Birkhoff: Extensions of Jentzsch's theorem, Transactions of the AMS 85 (1957) 219-227.

[7] A. Borobia & R. Canto: Matrix Scaling: A Geometric Proof of Sinkhorn's Theorem, LAA 268 (1998) 1-8.

[8] P.S. Bullen, D.S. Mitrinovic, P.M. Vasic: Means and their inequalities, D.Reidel 1988.

[9] P. Bürgisser, M. Clausen, A. Shokrollahi: Algebraic Complexity Theory, Springer Verlag (GMW #315) 1996.

[10] D.I. Cartwright & M.I. Field: A refinement of the AG mean inequality, Proc. Amer. Math'l. Soc. 71,1 (1978) 36-38.

[11] D. Coppersmith & S. Winograd: Matrix multiplication via arithmetic progressions, J. Symbolic Comput. 9 91990) 251-280.

[12] T.M.Cover & J.A. Thomas: Elements of Information Theory, Wiley 1991.

[13] G.B. Dantzig: Linear Programming and Extensions, Princeton University Press, Princeton, NJ, 1963.

[14] D.E. Dennis, Jr. & R.B. Schnabel: Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, 1983.

[15] D.J. Djokovic: Note on nonnegative matrices, Proc. Amer. Math'l. Soc. 25,1 (1970) 80-82.

[16] Cynthia Dwork, Ravi Kumar, Moni Naor, D.Sivakumar: Rank Aggregation for the web, 10th World Wide Web conference, Hong Kong (2001) 613-622. www.wisdom.weizmann.ac.il/~naor/PAPERS/rank_www10.html

[17] S. Even & R.E. Tarjan: Network Flow and Testing Graph Connectivity, SIAM J. Comput. 4 (1975) 507-518.

[18] R. Fletcher: Practical methods for optimization, Wiley 1980.

[19] J. Franklin & J. Lorenz: On the scaling of multidimensional matrices, LAA 114/115 (1989) 717-735.

[20] G.H. Golub & C.F. Van Loan: Matrix computations, 3rd edition, Johns Hopkins University Press 1996.

[21] M.Grötschel, L.Lovasz, A.Schrijver: Geometric algorithms and combinatorial optimization, Springer 1993 (2nd corrected ed.).

[22] Roger A. Horn & Charles R. Johnson:, Matrix Analysis, Cambridge Univ. press, 1985.

[23] B. Kalantari & L. Khachiyan: On the rate of convergence of deterministic and randomized RAS matrix scaling algorithms, Oper. Res. Lett. 14,5 (1993) 237-244.

[24] B. Kalantari & L. Khachiyan: On the complexity of nonnegative-matrix scaling, Linear Algebra Appl. 240 (1996) 87-103.

[25] R.M. Karp: A characterization of the minimum cycle mean in a digraph, Discrete Maths. 23 (1978) 309-311.

[26] James P. Keener: The Perron-Frobenius Theorem and the Ranking of Football Teams, SIAM Review 35,1 (March 1993) 80-93.

[27] Leonid Khachiyan: Diagonal matrix scaling is NP-hard, Linear Algebra Appl. 234 (1996) 173-179.

[28] Hermann Kober: On the arithmetic and geometric means and Hölder's inequality, Proc. Amer. Math'l. Soc. 9,3 (1958) 452-459.

[29] E. Kohlberg & J.W. Prat: The contraction mapping approach to the Perron-Frobenius theory: why Hilbert's metric? Mathematics of Operations Research 7,2 (1982) 198-210.

[30] Thus despite the rather poor appearance of our bound $O\left(N^{5.01} + \mathrm{Matmul}(N) \log|\log\epsilon|\right)$, it may be optimal, or at least optimal is only conventional matrix multiplicaiton is used!

[30] E.L. Lawler: Combinatorial optimization: networks and matroids, Holt, Rinehart and Winston, New York, 1976; Dovwer reprint 2001.

[31] N.Linial, A.Samorodnitsky, A.Wigderson: A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents, Combinatorica 20,4 (2000) 545-568.

[32] C.R. MacCluer: The many proofs and applications of Perron's theorem, SIAM Rev. 42,3 (2000) 487-498.

[33] S.M.Macgill: Theoretical properties of biproportional matrix adjustments, Environment & Planning A9 (1977) 687-701.

[34] Maurice Mignotte: Mathematics for Computer Algebra, Springer 1992.

[35] Henryk Minc: Nonnegative matrices, Wiley 1988.

[36] Hervé Moulin: Condorcet's Principle Implies the No Show Paradox, J. Economic Theory 45 (1988) 53-64.

[37] Arkadi Nemirovski & Uriel Rothblum: On complexity of matrix scaling, Linear Algebra Appl. 302/303 (1999) 435-460.

[38] Victor Y. Pan: Solving a polynomial equation: some history and recent progress, SIAM Review, 39,2 (1997) 187-220.

[39] C.H.Papadimitriou & K.Steiglitz: Combinatorial optimization: Algorithms and Complexity, Prentice-Hall, Englewood Clifffs NJ 1982.

[40] B.N. Parlett & T.L. Landis: Methods for Scaling to Double Stochastic Form, LAA 48 (1982) 53-79.

[41] Hazel Perfect & Leon Mirsky: The distribution of positive elements in doubly stochastic matrices, J. London. Math'l. Soc. 40 (1965) 689-698.

[42] Michael H. Schneider; Matrix scaling, entropy minimization, and conjugate duality (II): The dual problem, Math'l. Programming B 48,1 (1990) 103-124

[43] Hans Schneider & Michael H. Schneider: Max-balancing weighted directed graphs and matrix scaling, Math'cs of Operations Research 16,1 (1991) 208-222.

[44] M.H. Schneider & S.A. Zenios: A comparative study of algorithms for matrix balancing, Operations Research 38,3 (1990) 439-455.

[45] J.T.Schwartz: Fast probabilistic algorithms for verification of polynomial identities, J. ACM 27 (1980) 701-717.

[46] E.Seneta: Nonnegative matrices and Markov chains, 2nd ed Springer 1981.

[47] R. Sinkhorn & P. Knopp: Concerning nonnegative matrices and doubly stochastic matrices, Pacific J. Math. 21 (1967) 343-348.

[48] Richard Sinkhorn: A relationship between arbitrary positive matrices and doubly stochastic matrices, Ann. Math'l. Statist. 35 (1964) 876-879.

[49] R. Sinkhorn: Diagonal equivalence to matrices with prescribed row and column sums, Amer. Math. Monthly 74 (1967) 402-405.

[50] G. W. Soules: The rate of convergence of Sinkhorn balancing, LAA 150 (1991) 3-40.

[51] Warren D. Smith: Range voting, #56 at `http://math.temple.edu/~wds/homepage/works.html`.

[52] Warren D. Smith: Reweighted range voting – new multiwinner voting method, #78 at `http://math.temple.edu/~wds/homepage/works.html`.

[53] Warren D. Smith:"Asset voting" scheme for multiwinner elections #77 at `http://math.temple.edu/~wds/homepage/works.html`.

[54] Warren D. Smith: Rating Systems for Gameplayers, and Learning, `http://math.temple.edu/~wds/homepage/works.html` #20.

[55] Dustin J. Stewart: Domination and matrix properties in tournaments and generalized tournaments, PhD thesis, University of Colorado at Denver, Applied math 2005.

[56] R.E. Tarjan: Depth first search and linear graph algorithms, SIAM J. Computing 1 (1972) 146-160.

[57] Pravin M. Vaidya: A new algorithm for minimizing convex functions over convex sets, Math'l. Programming 73A,3 (1996) 291-341.