SIAM J. MATRIX ANAL. APPL.
Vol. 38, No. 4, pp. 1410–1433

MAX-PLUS ALGEBRAIC STATISTICAL LEVERAGE SCORES*

JAMES HOOK†

Abstract. The statistical leverage scores of a matrix $A \in \mathbb{R}^{n\times d}$ record the degree of alignment between $\operatorname{col}(A)$ and the coordinate axes in $\mathbb{R}^n$. These scores are used in random sampling algorithms for solving certain numerical linear algebra problems. In this paper we present a max-plus algebraic analogue of statistical leverage scores. We show that max-plus statistical leverage scores can be used to calculate the exact asymptotic behavior of the conventional statistical leverage scores of a generic radial basis function network (RBFN) matrix. We also show how max-plus statistical leverage scores can provide a novel way to approximate the conventional statistical leverage scores of a fixed, nonparametrized matrix.

Key words. randomized numerical linear algebra, max-plus algebra, least squares

AMS subject classification. 65F30

DOI. 10.1137/16M1097596

1. Introduction.

The *statistical leverage scores* of $A \in \mathbb{R}^{n\times d}$ are the vector $p(A) \in \mathbb{R}^n$, with

$$p_i(A) = \left(\max_{x\in\mathbb{R}^n} \frac{|(Ax)_i|}{\|Ax\|_2}\right)^2 \quad \text{for } i = 1,\dots,n. \tag{1}$$

The ith statistical leverage score of A is equal to the square of the cosine of the angle between $\operatorname{col}(A)$ and the unit vector $\underline{e}_i$. To calculate $p(A)$ we take a decomposition that provides an orthogonal basis for $\operatorname{col}(A)$. For example, suppose that A has rank k; then if we take the QR decomposition, we obtain $A = QR$, with $Q \in \mathbb{R}^{n\times k}$ and

$$p_i(A) = \|Q_{i\cdot}\|_2^2 \quad \text{for } i = 1,\dots,n, \tag{2}$$

where $Q_{i\cdot}$ denotes the ith row of Q. Note that $\sum_{i=1}^n p_i(A)/k = 1$, so that the vector $p(A)/k \in \mathbb{R}^n$ is a probability distribution on $\{1,\dots,n\}$.

Statistical leverage score distributions are used in random sampling algorithms for solving certain numerical linear algebra problems [7, 8, 12, 17, 20]. For example, Algorithm 1 approximates the least squares solution $x^* = \arg\min_{x\in\mathbb{R}^d} \|Ax - y\|_2$ by examining a randomly selected sample of the rows of $[A, y]$. The $r \times n$ random matrix M samples r rows from the least squares problem with replacement, according to the probability distribution p. The sampled rows are scaled by one over the square root of their sampling probability to ensure that the approximate solution is unbiased. Next we compute the solution $\hat{x}$ that is optimal for the sampled rows and then use it as an approximate solution for the full problem. This is similar to taking a poll to predict an election result, and just like taking a poll, it is crucial that our sample set reflect the statistical properties of the full set of rows. Theorem 1.1 shows that sampling with respect to statistical leverage scores is one way of achieving this. The full result

*Received by the editors October 7, 2016; accepted for publication (in revised form) by P. Drineas September 11, 2017; published electronically November 16, 2017.
http://www.siam.org/journals/simax/38-4/M109759.html

Funding: The work of the author was supported by the University of Bath, 50th Anniversary Prize fellowship in the Bath Institute of Mathematical Innovation.

†Department of Mathematical Sciences, University of Bath, Bath BA2 7AY, UK (james.l.hook@gmail.com).

presented in [10] also shows how approximate statistical leverage scores can also be used for sampling. Note that Theorem 1.1 holds for an arbitrary matrix $A \in \mathbb{R}^{n\times d}$; in particular, there is no assumed statistical model for the rows of A.

Algorithm 1 Given $A \in \mathbb{R}^{n\times d}$, $y \in \mathbb{R}^n$, a probability distribution $p \in \mathbb{R}^n$, and $r \in \{d, \ldots, n\}$, compute $\hat{x} \approx x^* = \arg\min_{x\in\mathbb{R}^d} \|Ax - y\|_2$.

```
for $i = 1, \ldots, r$ independently do
    set $M_{i\cdot} = \underline{e}_j^T/\sqrt{p_j}$, with probability $p_j$, for $j = 1, \ldots, n$
end for
set $\hat{x} = \arg\min_{x\in\mathbb{R}^d} \|(MA)x - My\|_2$
```

Theorem 1.1 (see [10, Theorem 3.1]). *Let $A \in \mathbb{R}^{n\times d}$, $y \in \mathbb{R}^n$. Let $x^* = \arg\min_{x\in\mathbb{R}^d} \|Ax - y\|_2$, let $p \in \mathbb{R}^n$ be the statistical leverage scores of $[A, y]$, and let $\hat{x}$ be the output of Algorithm* 1; *then for*

$$r \geq 8\frac{d+1}{\epsilon^2} \ln \frac{(d+1)}{\delta}$$

we have

$$\mathbb{P}\left\{\|A\hat{x} - y\|_2 \leq (1+2\epsilon)\|Ax^* - y\|_2\right\} > 1 - 3\delta.$$

In practice if we are considering solving an $n \times d$ least squares problem using a random sampling method, then we must be in a scenario where $\mathcal{O}(nd^2)$ computations are too costly and we are therefore unable to use (2) to calculate the statistical leverage scores of the matrix $[A, y]$. Thus there is interest in developing efficient methods for approximating the statistical leverage scores of a matrix. Drineas et al. present such a method in [9]. Their approach uses random projections and can be tuned to provide approximations with a desired accuracy for a desired reliability probability. While the exact cost of computing this approximation depends on the chosen accuracy and probability, it is $\mathcal{O}\big(nd\log(n)\big)$ for moderate values.

Clarkson and Woodruff present an algorithm based on sparse subspace embeddings, which for $\epsilon > 0$ returns an x' for which

$$\|Ax' - y\|_2 \leq (1+\epsilon) \min_{x\in\mathbb{R}^d} \|Ax - y\|_2,$$

with cost $\mathcal{O}\big(\text{nnz}(A) + d^2\epsilon^{-1}\big)$, where $\text{nnz}(A)$ is the number of nonzeros in the matrix A. They call this a regression in sparsity time algorithm [7]. The same techniques enable them to approximate all of the statistical leverage scores of A up to a constant relative error with cost $\mathcal{O}\big(\text{nnz}(A)\log(n)\big)$. They are able to solve the regression problem faster than approximating all of the statistical leverage scores up to a constant factor because their regression algorithm is able to use a less precise statistical leverage score approximation.

Cohen et al. use iterative row resampling to achieve another regression in sparsity time algorithm [8]. Using this technique they are able to approximate all of the statistical leverage scores of A up to a multiplicative factor of d^θ with cost $\mathcal{O}\big(\text{nnz}(A)\theta^{-1}\big)$, which yields an approximation up to a constant factor with cost $\mathcal{O}\big(\text{nnz}(A)\log(d)\big)$, by setting $\theta^{-1} = 1/\log(d)$.

Although these recent works have achieved optimal worst case complexity for solving tall skinny regression problems, they can suffer from large constant factors, and since developing fast methods for solving huge linear systems is such an important area of research, it is worthwhile to explore alternatives. In this paper we present a max-plus algebraic analogue of statistical leverage scores. In section 2 we show that max-plus statistical leverage scores can be used to calculate the exact asymptotic behavior of the statistical leverage scores of a radial basis function network (RBFN) matrix. In section 5 we present an algorithm for computing max-plus statistical leverage scores. Our algorithm uses quickselect, which has good average case complexity but poor worst case complexity. As a result our algorithm also has good average case complexity but poor worst case complexity. We justify in section 5 the claim that with high probability our algorithm can compute all of the max-plus statistical leverage scores of an $n \times d$ max-plus matrix $\mathcal{A}$ with cost $\mathcal{O}\big(\mathrm{nf}(\mathcal{A}) + d^3\big)$, where $\mathrm{nf}(\mathcal{A})$ is the number of finite entries in $\mathcal{A}$; note that finite entries play the role of nonzero entries in a max-plus matrix. Alternatively, by using a heap-based algorithm instead of quickselect, we achieve a worst case complexity of $\mathcal{O}(\mathrm{nf}(\mathcal{A})\log(d) + d^3)$, but with the attractive feature that the algorithm requires only two passes of the input matrix.

In section 3 we show how these max-plus scores can be used to approximate the conventional statistical leverage scores of a matrix $A \in \mathbb{R}^{n\times d}$. Our approximation method has three promising features: first, that it agrees with the asymptotic behavior of an RBFN matrix (in the sense of Theorem 3.2); second, that it is quick to compute; and third, that it performs well on the randomly generated problems presented in section 4. However, we must also point out that since the max-plus approximation depends only on the moduli of the entries in the matrix, there are certain highly structured problems for which it is very inaccurate. Developing a more reliable alternative approximation method, which still uses some of the ideas surrounding the max-plus statistical leverage scores, but which also uses sign information to provide a more reliable approximation, is a promising avenue for future research.

Throughout this paper real matrices will be denoted by capital letters, with their entries denoted by the corresponding lower case letter in the usual way, e.g., $A = (a_{ij}) \in \mathbb{R}^{n\times d}$. Max-plus matrices will be denoted by calligraphic capital letters, and their entries by the corresponding lower case calligraphic letter, e.g., $\mathcal{A} = (\mathscr{a}_{ij}) \in \mathbb{R}^{n\times d}_{\max}$.

For an $n \times d$ matrix A, we use the notation $A([i_1,\dots,i_m],[j_1,\dots,j_k])$ to denote the $m\times k$ matrix formed from the $\{i_1,\dots,i_m\}$ rows and $\{j_1,\dots,j_k\}$ columns of A. We also use the notation $A([i_1,\dots,i_m]^c,[j_1,\dots,j_k]^c)$ to denote the $(n-m)\times(d-k)$ matrix formed from the $\{1,\dots,n\}/\{i_1,\dots,i_m\}$ rows and $\{1,\dots,d\}/\{j_1,\dots,j_k\}$ columns of A.

1.1. Quick introduction to max-plus algebra. Max-plus algebra concerns the max-plus semiring $\mathbb{R}_{\max} = (\mathbb{R}\cup\{-\infty\},\oplus,\otimes)$, where

$$\mathscr{a} \oplus \mathscr{b} = \max\{a,b\}, \quad \mathscr{a}\otimes\mathscr{b} = \mathscr{a}+\mathscr{b} \quad \text{for all } \mathscr{a},\mathscr{b}\in\mathbb{R}_{\max}. \tag{3}$$

Akian, Bapat, and Gaubert showed that max-plus algebra can be used to calculate the exact asymptotic growth rates of the eigenvalues of generic matrices whose entries are Puiseux series [1, 2]. Gaubert and Sharify were the first to exploit this idea to develop max-plus algebraic methods for approximating the order of magnitude of the eigenvalues of a fixed complex matrix polynomial [11]. This approach has since been adapted and expanded to approximate matrix singular values and LU factors [15, 16]. In this paper we extend the approach further to include statistical leverage scores. We introduce a definition for max-plus statistical leverage scores, which enables us to calculate the exact asymptotic growth rates of the statistical leverage scores of generic

RBFN matrices, and to approximate the statistical leverage scores of a fixed complex matrix. We provide all of the necessary background material in this section. For a more thorough introduction to max-plus algebra, see [6].

A max-plus matrix $\mathcal{A} \in \mathbb{R}_{\max}^{n\times d}$ is simply an $n\times d$ array of elements from $\mathbb{R}_{\max}$. Max-plus matrix multiplication is defined in analogy to the classical case: for $\mathcal{A} \in \mathbb{R}_{\max}^{n\times d}$ and $\mathcal{B} \in \mathbb{R}_{\max}^{d\times m}$, the product $(\mathcal{A} \otimes \mathcal{B}) \in \mathbb{R}_{\max}^{n\times m}$ is the max-plus matrix with

$$(\mathcal{A} \otimes \mathcal{B})_{ij} = \bigoplus_{k=1}^{d} a_{ik} \otimes b_{kj} = \max_{k=1}^{d}(a_{ik} + b_{kj}). \tag{4}$$

For clarity we will often display equations using max-plus algebraic notation alongside equivalent expressions that only use standard notation.

For $\mathcal{A} \in \mathbb{R}_{\max}^{n\times d}$, with $n \geq d$, we define the *max-plus permanent* of $\mathcal{A}$ by

$$\operatorname{perm}(\mathcal{A}) = \bigoplus_{\phi\in\Phi(d,n)} \bigotimes_{j=1}^{d} a_{\phi(j)j} = \max_{\phi\in\Phi(d,n)} \sum_{j=1}^{d} a_{\phi(j)j}, \tag{5}$$

where $\Phi(d, n)$ is the set of all injections from $\{1, \dots, d\}$ to $\{1, \dots, n\}$. We also define the set of *optimal assignments* of $\mathcal{A}$ by

$$\operatorname{oas}(\mathcal{A}) = \arg\max_{\phi\in\Phi(d,n)} \sum_{j=1}^{d} a_{\phi(j)j}. \tag{6}$$

Computing $\operatorname{perm}(\mathcal{A})$ and $\operatorname{oas}(\mathcal{A})$ is commonly referred to as the *optimal assignment problem.* For $\phi \in \operatorname{oas}(\mathcal{A})$ and $i \in \{1, \dots, n\}$, if $\phi(j) = i$ for some $j \in \{1, \dots, d\}$, then we say that ϕ *assigns* row i to column j and vice versa. Note that for a square matrix $\mathcal{A} \in \mathbb{R}_{\max}^{n\times n}$, the formula (6) looks very much like the max-plus version of the determinant, except for the alternating sign term, which has no max-plus algebraic analogue since $\oplus$ is not invertible. We exploit this connection between max-plus permanents and conventional determinants extensively in section 3.

Optimal assignments have a neat operational research interpretation. Suppose that we have n jobs and d workers and that we must assign each worker to a distinct job. Let $\mathcal{A} \in \mathbb{R}_{\max}^{n\times d}$ be the max-plus matrix with a_{ij} equal to the benefit of assigning worker j to job i. Then $\operatorname{perm}(\mathcal{A})$ is the maximum possible total benefit, and $\operatorname{oas}(\mathcal{A})$ is the set of optimal assignments of workers to jobs.

For $\mathcal{A} \in \mathbb{R}_{\max}^{n\times d}$ and for $i = 1, \dots, n$, $j = 1, \dots, d$, we define the (i, j)-*obligated permanent* of $\mathcal{A}$ by

$$\operatorname{perm}(\mathcal{A}, j \mapsto i) = \bigoplus_{\phi\in\Phi(d,n;j\mapsto i)} \bigotimes_{k=1}^{d} a_{\phi(k)k} = \max_{\phi\in\Phi(d,n;j\mapsto i)} \sum_{k=1}^{d} a_{\phi(k)k}, \tag{7}$$

where $\Phi(d, n; j \mapsto i)$ is the set of all injections ϕ from $\{1, \dots, d\}$ to $\{1, \dots, n\}$, with $\phi(j) = i$. In terms of the operational research interpretation, $\operatorname{perm}(\mathcal{A}, j \mapsto i)$ is the maximum possible total benefit if we are obliged to assign worker j to job i.

We can expand the permanent of a square matrix along a row or column in the same way as a determinant.

PROPOSITION 1.2. *Let $\mathcal{A} \in \mathbb{R}_{\max}^{n\times n}$; then for $i = 1, \dots, n$*

$$\operatorname{perm}(\mathcal{A}) = \bigoplus_{j=1}^{n} a_{ij} \otimes \operatorname{perm}\big(\mathcal{A}([i]^c, [j]^c)\big) = \max_{j=1}^{n} \big(a_{ij} + \operatorname{perm}\big(\mathcal{A}([i]^c, [j]^c)\big)\big).$$

Proof. For a square matrix (5) can be written as

$$\operatorname{perm}(\mathcal{A}) = \max_{\phi\in\Pi(n)} \sum_{k=1}^{d} a_{k\pi(k)},$$

where $\Pi(n)$ is the set of permutations on $\{1,\dots,n\}$. Now for $i,j = 1,\dots,n$, set $\Pi(n)_{ij} = \{\pi \in \Pi(n) \ : \ \pi(i) = j\}$. Then for $i = 1,\dots,n$ we have

$$\begin{aligned}\operatorname{perm}(\mathcal{A}) &= \max_{j=1}^{n}\left(\max_{\pi\in\Pi(n)_{ij}} \sum_{k=1}^{d} a_{k\pi(k)}\right)\\ &= \max_{j=1}^{n}\left(a_{ij} + \max_{\pi\in\Pi(n)_{ij}} \sum_{k\neq i} a_{k\pi(k)}\right)\\ &= \max_{j=1}^{n}\left(a_{ij} + \operatorname{perm}\big(\mathcal{A}([i]^c,[j]^c)\big)\right).\end{aligned}$$ □

1.2. Max-plus statistical leverage scores. We define the *max-plus statistical leverage scores* of $\mathcal{A}$ to be the vector $p(\mathcal{A}) \in \mathbb{R}^n_{\max}$ with

$$p_i(\mathcal{A}) = 2\Big(\max_{j=1}^{d} \operatorname{perm}(\mathcal{A}, j \mapsto i) - \operatorname{perm}(\mathcal{A})\Big) \quad \text{for } i = 1,\dots,n. \tag{8}$$

We can understand this definition of max-plus statistical leverage scores in terms of the previously outlined jobs-to-workers interpretation of the optimal assignment problem.

Operational research interpretation. The score of the ith row, $p_i(\mathcal{A})$, is equal to minus two times the smallest nonnegative bonus that needs to be applied to the benefit of job i in order for there to exist an optimal assignment of workers to jobs that assigns a worker to job i. Thus if there exists an optimal assignment that assigns job i, then row i will have score zero, and otherwise row i will have a strictly negative score.

Example 1.3. Consider

$$\mathcal{A} = \begin{bmatrix} 3 & 3\\ 0 & 2\\ 1 & 0\end{bmatrix}, \quad \operatorname{perm}(\mathcal{A}, j \mapsto i) = \begin{bmatrix} 5 & 4\\ 3 & 5\\ 4 & 3\end{bmatrix}, \quad \mathcal{A}+\Delta\mathcal{A} = \begin{bmatrix} 3 & 3\\ 0 & 2\\ 1+x & x\end{bmatrix}.$$

To compute the max-plus statistical leverage scores of $\mathcal{A}$ we need to compute the permanent of $\mathcal{A}$. This is given by $\operatorname{perm}(\mathcal{A}) = 3 + 2 = 5$, which is attained by $\phi = (1,2)$. Since ϕ assigns rows 1 and 2, these rows have score zero. The $(3,1)$-obligated permanent is given by $\operatorname{perm}(\mathcal{A}, 1 \mapsto 3) = 3 + 1 = 4$, which is attained by $\phi(3,1)$. Since this is the maximally weighted assignment that assigns row 3, row 3 has score $-2(4-3) = -2$. Therefore the max-plus statistical leverage scores of $\mathcal{A}$ are given by $p(\mathcal{A}) = [0,0,-2]$.

In terms of the operational research interpretation, suppose that we apply a bonus of x to the benefit of carrying out job 3. This adjusted benefit matrix is given by $\mathcal{A}+\Delta\mathcal{A}$. Clearly the least nonnegative x such that $\mathcal{A}+\Delta\mathcal{A}$ has an optimal assignment that assigns row 3 is given by $x = 1$, and hence $p_3(\mathcal{A}) = -2$.

2. Asymptotics of generic radial basis function network (RBFN) matrices.

An RBFN matrix $K \in \mathbb{R}^{n\times d}$ is the form

$$k_{ij} = \exp(a_{ij}/\sigma) \quad \text{for } i = 1,\dots,n,\ j = 1,\dots,d, \tag{9}$$

for some $\mathcal{A} \in \mathbb{R}_{\max}^{n\times d}$ and $\sigma \in \mathbb{R}_+$. We will use the notation $K = \exp(\mathcal{A}/\sigma)$ to mean (9). Such matrices arise in the formulation of RBFNs [5] as follows. Suppose that we have a set of data points $\mathbf{x}_1,\dots,\mathbf{x}_n \in \mathbb{R}^m$, with target values $y_1,\dots,y_n \in \mathbb{R}$, and that we wish to fit an RBFN to this data using d radial basis functions, centered at the locations $\mathbf{c}_1,\dots,\mathbf{c}_d \in \mathbb{R}^m$. Then for $\mathbf{x} \in \mathbb{R}^m$, the output of the RBFN is given by

$$\varphi(\mathbf{x}) = \sum_{i=1}^{d} w_i \exp\big(-\|\mathbf{x}-\mathbf{c}_j\|_2^2/\sigma\big), \tag{10}$$

where $w \in \mathbb{R}^d$ is a vector of weights that needs to be learned. If we wish to minimize the 2-norm difference between the target values and the output of the RBFN, then the optimal weights are given by

$$w = \underset{w'\in\mathbb{R}^d}{\arg\min} \|Kw' - y\|_2,$$

where $K = \exp(\mathcal{A}/\sigma)$, with $\mathcal{A} \in \mathbb{R}_{\max}^{n\times d}$ given by

$$a_{ij} = -\|\mathbf{x}_i - \mathbf{c}_j\|_2^2. \tag{11}$$

In machine learning applications, n can be very large, so we may wish to solve (10) approximately using Algorithm 1.

Understanding the behavior of an RBFN matrix in the limit $\sigma \to 0$ can give useful insight into its behavior for more moderate values. Proposition 2.1 (below) tells us that, provided the centers are evenly distributed among the data points, the RBFN matrix will have orthogonal columns in the limit $\sigma \to 0$. From this result we might expect that RBFNs, with centers evenly distributed among their data points, will be close to orthogonal for small but nonzero values of σ, which would imply that we can accurately approximate their statistical leverage scores using row norms. Using max-plus algebra we can study the asymptotic behavior of RBFN matrices with more general distribution of centers. Theorems 2.8 and 2.9, which are the main results of this section, extend the result of Proposition 2.1 and show that the asymptotic exponential growth rates of the statistical leverage scores of an RBFN matrix $K = \exp(\mathcal{A}/\sigma)$ are exactly equal to the max-plus statistical leverage scores of the matrix $\mathcal{A}$.

Proposition 2.1. *Let $\mathbf{x}_1,\dots,\mathbf{x}_n \in \mathbb{R}^m$ and $\mathbf{c}_1,\dots,\mathbf{c}_d \in \mathbb{R}^m$, and let $\mathcal{A} \in \mathbb{R}_{\max}^{n\times d}$ be as defined in* (11)*; then no two centers have a common nearest data point if and only if*

$$\lim_{\sigma\to 0} (K\operatorname{diag}(s)^{-1})^T (K\operatorname{diag}(s)^{-1}) = I,$$

where the diagonal scaling $s \in \mathbb{R}^d$ is given by $s_j = \|K_{\cdot j}\|_2$ for $j = 1,\dots,d$ and $I \in \mathbb{R}^{d\times d}$ is the identity matrix.

Proof. First note that

$$\lim_{\sigma\to 0} \|K_{\cdot j}\|_2 / \exp\Big(-\min_{i=1}^{n} \|\mathbf{x}_i - \mathbf{c}_j\|_2^2/\sigma\Big) = \sqrt{c_j},$$

where c_j is the number of distinct elements in $\arg\min_{i=1}^{n} \|\mathbf{x}_i - \mathbf{c}_j\|_2$. Thus

$$\lim_{\sigma\to 0}(K\mathrm{diag}(s)^{-1})_{ij} = \begin{cases} 1/\sqrt{c_j} & \text{for } i \in \arg\min_{i'=1}^{n} \|\mathbf{x}_{i'} - \mathbf{c}_j\|_2, \\ 0 & \text{otherwise,} \end{cases}$$

and therefore

$$\lim_{\sigma\to 0}\langle (K\mathrm{diag}(s)^{-1})_{\cdot j'}, (K\mathrm{diag}(s)^{-1})_{\cdot j}\rangle \begin{array}{ll} = 1 & \text{if } j' = j, \\ > 0 & \text{if } j' \neq j \text{ and } \mathbf{c}_{j'} \text{ and } \mathbf{c}_j \text{ have a common} \\ & \text{nearest data point,} \\ = 0 & \text{otherwise.} \end{array} \qquad \square$$

It will be useful for us to distinguish a subset of nondegenerate matrices to work with. We say that $\mathcal{A} \in \mathbb{R}_{max}^{n\times d}$ is a *generic distance matrix* if for all subsets $\{i_1,\dots,i_m\} \subset \mathbb{N}_n$ and $\{j_1,\dots,j_k\} \subset \mathbb{N}_d$, with $k \leq m$, the submatrix $\mathcal{A}([i_1,\dots,i_m], [j_1,\dots,j_k])$ has a finite permanent and a unique optimal assignment.

LEMMA 2.2. *Suppose that* $\mathbf{x}_1,\dots,\mathbf{x}_n \in \mathbb{R}^m$ *and* $\mathbf{c}_1,\dots,\mathbf{c}_d \in \mathbb{R}^m$ *are points in general position; then the matrix* $\mathcal{A}$ *defined in* (11) *is a generic distance matrix.*

Proof. By general position we mean that there is an open and dense set $G \subset \mathbb{R}^{m\times n}\times\mathbb{R}^{m\times d}$ such that $\mathcal{A}$ is a generic distance matrix whenever $(\mathbf{x}_1,\dots,\mathbf{x}_n,\mathbf{c}_1,\dots,\mathbf{c}_d) \in G$.

For any two subsets $\{i_1,\dots,i_m\} \subset \mathbb{N}_n$, $\{j_1,\dots,j_k\} \subset \mathbb{N}_d$ the submatrix $\mathcal{A}_{IJ} = \mathcal{A}([i_1,\dots,i_m],[j_1,\dots,j_k])$ will have a finite permanent, since (11) is finite for all $(\mathbf{x}_1,\dots,\mathbf{x}_n,\mathbf{c}_1,\dots,\mathbf{c}_d) \in \mathbb{R}^{m\times n}\times\mathbb{R}^{m\times d}$. Now consider two distinct assignments $\phi_1,\phi_2 \in \Phi(k,m)$, and let $G(\phi_1,\phi_2) \subset \mathbb{R}^{m\times n}\times\mathbb{R}^{m\times d}$ be the subset of data points and centers for which $f(\mathbf{x}_1,\dots,\mathbf{x}_n,\mathbf{c}_1,\dots,\mathbf{c}_d) = w(\phi_1) - w(\phi_2) = 0$, where

$$w(\phi) = \sum_{\ell=1}^{k} a_{i_{\phi(\ell)},j_\ell}.$$

The function f is a polynomial in the coefficients of $(\mathbf{x}_1,\dots,\mathbf{x}_n,\mathbf{c}_1,\dots,\mathbf{c}_d)$, and it is therefore identically equal to zero or only zero on some lower dimensional manifold. Now consider $(\mathbf{x}_1,\dots,\mathbf{x}_n,\mathbf{c}_1,\dots,\mathbf{c}_d) \in \mathbb{R}^{m\times n}\times\mathbb{R}^{m\times d}$, with $\mathbf{c}_1 = \cdots = \mathbf{c}_d = \mathbf{x}_{i_{\phi_1(1)}} = \cdots = \mathbf{x}_{i_{\phi_1(k)}} = \underline{0}$ and $\mathbf{x}_i = \underline{1}$ for the remaining data points; then $w(\phi_1) = 0$ and $w(\phi_2) \neq 0$. Thus f is not identically equal to zero, and its zero set $G(\phi_1,\phi_2)$ forms a lower dimensional manifold whose complement is open and dense in $\mathbb{R}^{m\times n}\times\mathbb{R}^{m\times d}$.

The set of data points and centers for which no two assignments have the same weight is a subset of the set for which the maximally weighted assignment is unique. Thus

$$\bigcap_{\phi_1,\phi_2} G(\phi_1,\phi_2)^C \subset G,$$

where the intersection is taken over all pairs of assignments of submatrices of $\mathcal{A}$. Since a finite intersection of open and dense sets is open and dense, and since a set containing a set that is open and dense is itself open and dense, this completes the proof. $\square$

LEMMA 2.3. *For a generic distance matrix* $\mathcal{B} \in \mathbb{R}_{\max}^{d\times d}$, *there exists* $\epsilon > 0$, *such that* $\det(M) \neq 0$ *for all* $0 < \sigma \leq \epsilon$ *and*

$$\lim_{\sigma\to 0} -\sigma\log|\det(M)| = \mathrm{perm}(\mathcal{B}),$$

where $M = \exp(\mathcal{B}/\sigma)$.

Proof. We have

$$\det(M) = \sum_{\pi\in\Pi_d} \operatorname{sgn}(\pi) \prod_{i=1}^{d} m_{i\pi(i)} = \sum_{\pi\in\Pi_d} \operatorname{sgn}(\pi) \exp\left(\sum_{i=1}^{d} \mathscr{b}_{i\pi(i)}/\sigma \right).$$

Let π' be the unique optimal assignment of $\mathcal{B}$; then $g(\pi) = \sum_{i=1}^{d} \mathscr{m}_{i\pi(i)} - \operatorname{perm}(\mathcal{B}) < 0$ for all $\pi \neq \pi'$, and

$$\det(M) = \exp\big(\operatorname{perm}(\mathcal{B})/\sigma\big) \left(\operatorname{sgn}(\pi') + \sum_{\pi\neq\pi'} \operatorname{sgn}(\pi) \exp\big(g(\pi)/\sigma\big) \right). \tag{12}$$

For the first result let $g = \max_{\pi\neq\pi'} g(\pi)$; then

$$\left| \sum_{\pi\neq\pi'} \operatorname{sgn}(\pi) \exp\big(g(\pi)/\sigma\big) \right| < 1,$$

and $\det(M) \neq 0$, whenever $\sigma < -g/(\log n!)$. The second result then follows from taking the limit $\sigma \to 0$ in (12). ◻

Max-plus linear systems, i.e., equations of the form $\mathcal{A} \otimes x = b$, have many applications in scheduling and dynamical systems [6, 13]. Such systems are better understood by studying the symmetrization of max-plus algebra $\mathbb{S}$, which is an extension of $\mathbb{R}_{\max}$, that allows for a kind of max subtraction operation (see [3, section 3.4] for an introduction). In this setting it is possible to either solve or determine that no solution exists to certain max-plus linear equations using a max-plus analogue of Cramer's rule (see [3, section 3.5.2]). This approach uses an expression for the max-plus inverse of a max-plus matrix, which looks exactly like the conventional Cramer's rule inverse, only with permanents instead of determinants. Typically this max-plus inverse does not provide a functional inverse in the usual sense, as only a special subset of all max-plus matrices are invertible. In Lemma 2.4 we show how to use this same max-plus inverse expression to calculate the asymptotic growth rates of the entries in the inverse of an RBFN matrix.

For $\mathcal{B} \in \mathbb{R}_{\max}^{d\times d}$, define the *max-plus inverse* $\mathcal{B}^{\otimes -1} \in \mathbb{R}_{\max}^{d\times d}$ by

$$(\mathcal{B}^{\otimes -1})_{ij} = \operatorname{perm}\big(\mathcal{B}([j]^c, [i]^c)\big) - \operatorname{perm}(\mathcal{B}) \quad \text{for } i, j = 1, \ldots, d. \tag{13}$$

Lemma 2.4. *For a generic distance matrix $\mathcal{B} \in \mathbb{R}_{\max}^{n\times d}$, there exists $\epsilon > 0$, such that $M = \exp(\mathcal{B}/\sigma)$ is invertible for all $0 < \sigma \leq \epsilon$ and*

$$\lim_{\sigma\to 0} \sigma \log |(M^{-1})_{ij}| = (\mathcal{B}^{\otimes -1})_{ij}.$$

Proof. From Cramer's rule we have

$$(M^{-1})_{ij} = (-1)^{i+j} \det\big(M([j]^c, [i]^c)\big) / \det(M) \quad \text{for } i, j = 1, \ldots, d.$$

Applying the result of Lemma 2.3, we have that each $(M^{-1})_{ij}$ is finite for sufficiently small σ and that

$$\begin{aligned} \lim_{\sigma\to 0} \sigma \log |(M^{-1})_{ij}| &= \operatorname{perm}\big(\mathcal{B}([j]^c, [i]^c)\big) - \operatorname{perm}(\mathcal{B}) \\ &= (\mathcal{B}^{\otimes -1})_{ij} \quad \text{for } i, j = 1, \ldots, d. \end{aligned}$$ ◻

The following three lemmas are technical results which are needed to prove Theorems 2.8 and 2.9.

LEMMA 2.5. *Let $\mathcal{A} \in \mathbb{R}_{\max}^{n\times d}$, let $\phi \in \operatorname{oas}(\mathcal{A})$, and let $j \in \{1,\dots,d\}$; then*

$$\operatorname{perm}\big(\mathcal{A}([1,\dots,n],[j]^c)\big) = \operatorname{perm}\big(\mathcal{A}([\phi(1),\dots,\phi(d)],[j]^c)\big).$$

Proof. First note that, since the left-hand side (LHS) is the maximum over a set of assignments that includes all of the assignments in the right-hand side (RHS), we have

$$\operatorname{perm}\big(\mathcal{A}([1,\dots,n],[j]^c)\big) \geq \operatorname{perm}\big(\mathcal{A}([\phi(1),\dots,\phi(d)],[j]^c)\big).$$

To prove the reverse inequality we will need the following results (14), (15):

(i) For $\mathcal{B} \in \mathbb{R}_{\max}^{m\times \ell}$ with $m \geq \ell$, $\phi \in \operatorname{oas}(\mathcal{B})$, and $i \in \{1,\dots,m\}$, either ϕ does not assign row i, or it assigns row i to some column $j \in \{1,\dots,\ell\}$. This yields

$$\begin{aligned}\text{(14)}\qquad \operatorname{perm}(\mathcal{B}) = {} & \operatorname{perm}\big(\mathcal{B}([i]^c,[1,\dots,k])\big)\\ & \oplus \max_{j=1}^{\ell}\Big(b_{ij} + \operatorname{perm}\big(\mathcal{B}([i]^c,[j]^c)\big)\Big).\end{aligned}$$

The expression in the first line of the RHS of (14) is the maximum over all assignments that do not assign row i, and the expression in the second line is the maximum over $j \in \{1,\dots,\ell\}$, of the maximum over all assignments that assign row i to column j.

(ii) For $\mathcal{B} \in \mathbb{R}_{\max}^{m\times \ell}$ with $m \geq \ell$ and $\phi \in \operatorname{oas}(\mathcal{B})$ we have

$$\text{(15)}\qquad \operatorname{perm}\big(\mathcal{B}([\phi(j_1),\dots,\phi(j_k)]^c,[j_1,\dots,j_k]^c)\big) = \sum_{t\neq j_1,\dots,j_k} b_{\phi(t)t}.$$

First note that by restricting ϕ to the rows and columns of the submatrix on the LHS of (15), we obtain an assignment with weight equal to the expression on the RHS, so that $LHS \geq RHS$. Now suppose that $LHS > RHS$; then there exists an assignment ϕ' of the submatrix with weight strictly greater than that of ϕ. But we can extend ϕ' to an assignment of the full matrix $\mathcal{B}$ by assigning j_t to $\phi(j_t)$ for $t = 1,\dots,k$. This results in an assignment of $\mathcal{B}$ with weight strictly greater than that of ϕ, which is a contradiction.

We construct a sequence $j_1,\dots,j_k$ as follows. Set $j_1 = j$, as in the statement of the lemma; then from (14) we have

$$\begin{aligned}\operatorname{perm}\big(\mathcal{A}([1,\dots,n],[j_1]^c)\big) = {} & \operatorname{perm}\big(\mathcal{A}([\phi(j_1)]^c,[j_1]^c)\big)\\ & \oplus \max_{t\neq j_1}\Big(a_{\phi(j_1)t} + \operatorname{perm}\big(\mathcal{A}([\phi(j_1)]^c,[j_1,t]^c)\big)\Big).\end{aligned}$$

If the expression in the first line of the RHS attains the maximum, we stop; otherwise we set j_2 to be a value of t that attains the maximum in the second line of the RHS. After $k-1$ steps we have $j_1,\dots,j_k$, a sequence of distinct elements of $\{1,\dots,d\}$. From (14) we have

$$\begin{aligned}\operatorname{perm}\big(\mathcal{A}([\phi(j_1),\dots,\phi(j_{k-1})]^c,[j_1,\dots,j_k]^c)\big) = {} & \operatorname{perm}\big(\mathcal{A}([\phi(j_1),\dots,\phi(j_k)]^c,[j_1,\dots,j_k]^c)\big)\\ \oplus \max_{t\neq j_1,\dots,j_k}\Big(a_{\phi(j_k)t} + \operatorname{perm}\big(\mathcal{A}([\phi(j_1),\dots,\phi(j_{k-1})]^c,[j_1,\dots,j_k,t]^c)\big)\Big).\end{aligned}$$

If the expression in the first line of the RHS attains the maximum, we stop; otherwise we set j_{k+1} to be a value of t that attains the maximum in the second line. Continuing in this way either we generate a sequence of length d, in which case

$$\operatorname{perm}\big(\mathcal{A}([1,\dots,n],[j_1]^c)\big) = \sum_{t=1}^{d-1} a_{\phi(j_t)j_{t+1}}, \tag{16}$$

or we stop after $k < d$ steps, in which case

$$\operatorname{perm}\big(\mathcal{A}([1,\dots,n],[j_1]^c)\big) = \sum_{t=1}^{k-1} a_{\phi(j_t)j_{t+1}} + \operatorname{perm}\big(\mathcal{A}([\phi(j_1),\dots,\phi(j_k)]^c,[j_1,\dots,j_k]^c)\big). \tag{17}$$

The expression on the RHS of (16) is the weight of an assignment of $\mathcal{A}$ that only assigns the rows $\{\phi(1),\dots,\phi(d-1)\}$ and does not assign the column $j_1 = j$, so that

$$\operatorname{perm}\big(\mathcal{A}([1,\dots,n],[j_1]^c)\big) = \sum_{t=1}^{d} a_{\phi(j_t)j_{t+1}} \leq \operatorname{perm}\big(\mathcal{A}([\phi(1),\dots,\phi(d)],[j_1]^c)\big).$$

Applying result (15) to (17) yields

$$\operatorname{perm}\big(\mathcal{A}([1,\dots,n],[j_1]^c)\big) = \sum_{t=1}^{k-1} a_{\phi(j_t)j_{t+1}} + \sum_{t=k+1}^{d} a_{\phi(j_t)j_t}.$$

The expression on the RHS is the weight of an assignment of $\mathcal{A}$ that only assigns the rows $\{\phi(1),\dots,\phi(d)\}/\phi(k)$ and does not assign the column j, so that

$$\begin{aligned}\operatorname{perm}\big(\mathcal{A}([1,\dots,n],[j_1]^c)\big) &= \sum_{t=1}^{k-1} a_{\phi(j_t)j_{t+1}} + \sum_{t=k+1}^{d} a_{\phi(j_t)j_t} \\ &\leq \operatorname{perm}\big(\mathcal{A}([\phi(1),\dots,\phi(d)],[j]^c)\big).\end{aligned}$$ □

Operational research interpretation. The result of Lemma 2.5 can also be understood in terms of the jobs-to-workers interpretation of the optimal assignment outlined in section 1.1. Suppose that we have an optimal assignment of workers to jobs and that one of the workers quits. Then we can find a new optimal assignment of workers to jobs that only assigns jobs which were assigned under the previous optimal assignment, and we do not need to consider any previously unassigned jobs.

LEMMA 2.6. *For a generic distance matrix $\mathcal{A} \in \mathbb{R}_{\max}^{n\times d}$, let ϕ be the optimal assignment of $\mathcal{A}$ and let $\mathcal{B} \in \mathbb{R}_{\max}^{d\times d}$ be the matrix formed from the d rows of $\mathcal{A}$ that are assigned by ϕ:*

$$\mathcal{B} = \mathcal{A}([\phi(1),\dots,\phi(d)],[1,\dots,d]).$$

Then the following hold:

1. $$\operatorname{perm}(\mathcal{A}, j \mapsto i) = a_{ij} + \big(\mathcal{B}^{\otimes -1} \otimes \underline{0}\big)_j + \operatorname{perm}(\mathcal{A})$$

   for all i such that row i is not assigned by ϕ, and for all $j = 1,\dots,d$.
2. *The max-plus statistical leverage scores of $\mathcal{A}$ are given by*

   $$p_i(\mathcal{A}) = \begin{cases} 0 & \textit{if } i \textit{ is an assigned row,} \\ 2\big(\mathcal{A} \otimes \mathcal{B}^{\otimes -1} \otimes \underline{0}\big)_i & \textit{otherwise.}\end{cases}$$

Proof. For result 1: Recall that the (i,j)-obligated permanent of $\mathcal{A}$ is the weight of the maximally weighted assignment that assigns column j to row i. Clearly this weight is attained by the optimal assignment of the remaining rows and columns as follows:

$$\operatorname{perm}(\mathcal{A}, j \mapsto i) = a_{ij} + \operatorname{perm}\big(\mathcal{A}([i]^c, [j]^c)\big).$$

From Lemma 2.5 and using the fact that i is not assigned by ϕ, we have

$$\begin{aligned}\operatorname{perm}(\mathcal{A}, j \mapsto i) &= a_{ij} + \operatorname{perm}\big(\mathcal{A}([\phi(1), \dots, \phi(d)], [j]^c)\big)\\ &= a_{ij} + \operatorname{perm}\big(\mathcal{B}([1, \dots, d], [j]^c)\big)\\ &= a_{ij} + \max_{k=1}^{d} \operatorname{perm}\big(\mathcal{B}([k]^c, [j]^c)\big).\end{aligned}$$

From (13) we have

$$\operatorname{perm}(\mathcal{A}, j \mapsto i) = a_{ij} + \max_{k=1}^{d} \big(\mathcal{B}^{\otimes -1}_{jk} + \operatorname{perm}(\mathcal{B})\big).$$

Finally using the fact that $\operatorname{perm}(\mathcal{B}) = \operatorname{perm}(\mathcal{A})$ and the definition of max-plus matrix-vector multiplication (4), we have

$$\operatorname{perm}(\mathcal{A}, j \mapsto i) = a_{ij} + \big(\mathcal{B}^{\otimes -1} \otimes \underline{0}\big)_j + \operatorname{perm}(\mathcal{A}).$$

For result 2: For the assigned rows $i = \phi(1), \dots, \phi(d)$, we have $p_i(\mathcal{A}) = 0$, which matches the definition of $p(\mathcal{A})$ given in (8). For the remaining unassigned rows recall from (8) that

$$p_i(\mathcal{A}) = 2\Big(\max_{j=1}^{d} \operatorname{perm}(\mathcal{A}, j \mapsto i) - \operatorname{perm}(\mathcal{A})\Big).$$

Then using result 1, we have

$$\begin{aligned}p_i(\mathcal{A}) &= 2\Big(\max_{j=1}^{d} a_{ij} + \big(\mathcal{B}^{\otimes -1} \otimes \underline{0}\big)_j\Big)\\ &= 2\big(\mathcal{A} \otimes \mathcal{B}^{\otimes -1} \otimes \underline{0}\big)_i.\end{aligned}$$ □

LEMMA 2.7. *For a generic distance matrix $\mathcal{A} \in \mathbb{R}^{n\times d}_{\max}$, let ϕ be the optimal assignment of $\mathcal{A}$ and let $\mathcal{B} \in \mathbb{R}^{d\times d}_{\max}$ be the matrix formed from the d rows of $\mathcal{A}$ that are assigned by ϕ. Then*

$$\lim_{\sigma\to 0} \sigma \log |(KM^{-1})_{ij}| = \begin{cases} \log \delta_{i\phi(j)} & \text{if } i \text{ is an assigned row,}\\ \big(\mathcal{A} \otimes \mathcal{B}^{\otimes -1}\big)_{ij} & \text{otherwise.}\end{cases}$$

Proof. Since the matrix M consists of the assigned rows of K that are assigned by the optimal assignment ϕ, we have

$$(KM^{-1})_{\phi(j)\cdot} = \underline{e}_j^T \quad \text{for } j = 1, \dots, d. \tag{18}$$

For an unassigned row i, from Cramer's rule we have

$$(KM^{-1})_{ij} = \det\big(M(K_{i\cdot}, j)\big) / \det(M), \tag{19}$$

where $M(K_{i\cdot}, j)$ is the matrix formed by replacing the jth row of M with the ith row of K. Note that $M(K_{i\cdot}, j) = \exp\big(\mathcal{C}(i,j)/\sigma\big)$, where

$$\mathcal{C}(i,j) = \mathcal{A}([\phi(1), \dots, \phi(j-1), i, \phi(j+1), \phi(d)], [1, \dots, d]).$$

Applying the result of Lemma 2.3 to (19), we obtain

$$\lim_{\sigma\to 0}\sigma\log|(KM^{-1})_{ij}| = \operatorname{perm}\big(\mathcal{C}(i,j)\big) - \operatorname{perm}(\mathcal{B}). \tag{20}$$

By Proposition 1.2 we can expand $\operatorname{perm}\big(\mathcal{C}(i,j)\big)$ along its jth row:

$$\begin{aligned}\operatorname{perm}\big(\mathcal{C}(i,j)\big) &= \max_{k=1}^{d} c_{jk} + \operatorname{perm}\big(\mathcal{C}(i,j)([k]^c,[j]^c)\big)\\ &= \max_{k=1}^{d} a_{ik} + \operatorname{perm}\big(\mathcal{B}([k]^c,[j]^c)\big).\end{aligned}$$

Substituting back into (20), we have

$$\begin{aligned}\lim_{\sigma\to 0}\sigma\log|(KM^{-1})_{ij}| &= \max_{k=1}^{d} a_{ik} + \operatorname{perm}\big(\mathcal{B}([k]^c,[j]^c)\big) - \operatorname{perm}(\mathcal{B})\\ &= \max_{k=1}^{d} a_{ik} + \left(\mathcal{B}^{\otimes -1}\right)_{kj}\\ &= \left(\mathcal{A}\otimes\mathcal{B}^{\otimes -1}\right)_{ij}.\end{aligned}$$ □

THEOREM 2.8. *For a generic distance matrix $\mathcal{A}\in\mathbb{R}_{\max}^{n\times d}$, let ϕ be the optimal assignment of $\mathcal{A}$ and let $\mathcal{B}\in\mathbb{R}_{\max}^{d\times d}$ be the matrix formed from the d rows of $\mathcal{A}$ that are assigned by ϕ. Then*

$$\lim_{\sigma\to 0}(KM^{-1})^T(KM^{-1}) = I,$$

where $K=\exp(\mathcal{A}/\sigma)$ and $M=\exp(\mathcal{B}/\sigma)$.

Proof. From the proof of Lemma 2.7 we have (18) for assigned rows. For unassigned rows we have (20), where since $\mathcal{B}$ is formed from the rows that are assigned by the optimal assignment of $\mathcal{A}$, we have $\operatorname{perm}(\mathcal{A})=\operatorname{perm}(\mathcal{B})$. Also since $\mathcal{C}(i,j)$ is formed from a different subset of the rows of $\mathcal{A}$, we must have $\operatorname{perm}\big(\mathcal{C}(i,j)\big)<\operatorname{perm}(\mathcal{B})$, because $\mathcal{A}$ has a unique optimal assignment. Therefore

$$\lim_{\sigma\to 0}(KM^{-1})_{ij} = 0$$

for all $i=1,\dots,n$ unassigned by ϕ and all $j=1,\dots,d$. □

THEOREM 2.9. *For a generic distance matrix $\mathcal{A}\in\mathbb{R}_{\max}^{n\times d}$, we have*

$$\lim_{\sigma\to 0}\sigma\log p_i(K) = p_i(\mathcal{A}) \quad \textit{for } i=1,\dots,n,$$

where $K=\exp(\mathcal{A}/\sigma)$, $p_i(K)$ is the ith statistical leverage score of K, and $p_i(\mathcal{A})$ is the ith max-plus statistical leverage score of $\mathcal{A}$.

Proof. Let ϕ be the optimal assignment of $\mathcal{A}$, and let $\mathcal{B}\in\mathbb{R}_{\max}^{d\times d}$ be the matrix formed from the d rows of $\mathcal{A}$ that are assigned by ϕ. Then from Lemma 2.3 there exists $\epsilon>0$ such that $M=\exp(\mathcal{B}/\sigma)$ is invertible for all $0<\sigma\le\epsilon$ and therefore such that KM^{-1} has statistical leverage scores identical to those of K.

Since KM^{-1} has a subset of rows which form the $d\times d$ identity matrix, we have

$$\|KM^{-1}x\|_2 \ge \|x\|_2 \quad \text{for all } x\in\mathbb{R}^d. \tag{21}$$

From Theorem 2.8 we have

$$\lim_{\sigma\to 0}\frac{\|KM^{-1}x\|_2^2}{\|x\|_2^2} \le \lim_{\sigma\to 0}\frac{\sum_{j=1}^{d}\|KM^{-1}\underline{e}_j\|_2^2 x_j^2}{\|x\|_2^2} = 1, \tag{22}$$

uniformly for all $x \in \mathbb{R}^d$. Therefore

$$\lim_{\sigma\to 0} \frac{\|KM^{-1}x\|_2^2}{\|x\|_2^2} = 1,$$

uniformly for all $x \in \mathbb{R}^d$.

Using the above results, the asymptotic growth rate of the ith statistical leverage score of KM^{-1} is given by

$$\begin{aligned}
\lim_{\sigma\to 0} \sigma \log p_i(KM^{-1}) &= \lim_{\sigma\to 0} \sigma \log \left(\max_{x\in\mathbb{R}^d} \left(\frac{|(KM^{-1}x)_i|}{\|KM^{-1}x\|_2} \right)^2 \right) \\
&= \lim_{\sigma\to 0} \max_{x\in\mathbb{R}^d} \left(\sigma \log \left(\frac{|(KM^{-1}x)_i|^2}{\|x\|_2^2} \right) - \lim_{\sigma\to 0} \sigma \log \left(\frac{\|KM^{-1}x\|_2^2}{\|x\|_2^2} \right) \right) \\
&= \lim_{\sigma\to 0} \sigma \log \left(\|\underline{e}_i KM^{-1}\|_2^2 \right) \\
&= \lim_{\sigma\to 0} 2\sigma \log \left(\max_{j=1}^{d} |KM^{-1}|_{ij} \right) = p_i(\mathcal{A}),
\end{aligned}$$

where the final step uses the results of Lemmas 2.6 and 2.7. □

Example 2.10. We randomly generate an RBFN matrix by sampling $n = 5$ data points and $d = 2$ centers from a standard 2-variate Gaussian; see Figure 1(a). We compute the distance matrix $\mathcal{A} \in \mathbb{R}^{n\times d}$ and the max-plus statistical leverage scores $p(\mathcal{A})$. Next we compute the statistical leverage scores $p(K)$ of $K = \exp(\mathcal{A}/\sigma)$ for a range of values of σ. Figure 1(b) shows $\sigma \log p(K)$ converging to $p(\mathcal{A})$ as required by Theorem 2.9.

In this example each center $\mathbf{c_j}$ has a distinct closest data point $\mathbf{x}_{\phi(\mathbf{j})}$, where ϕ is the optimal assignment of $\mathcal{A}$, so that Proposition 2.1 applies. In this case we have

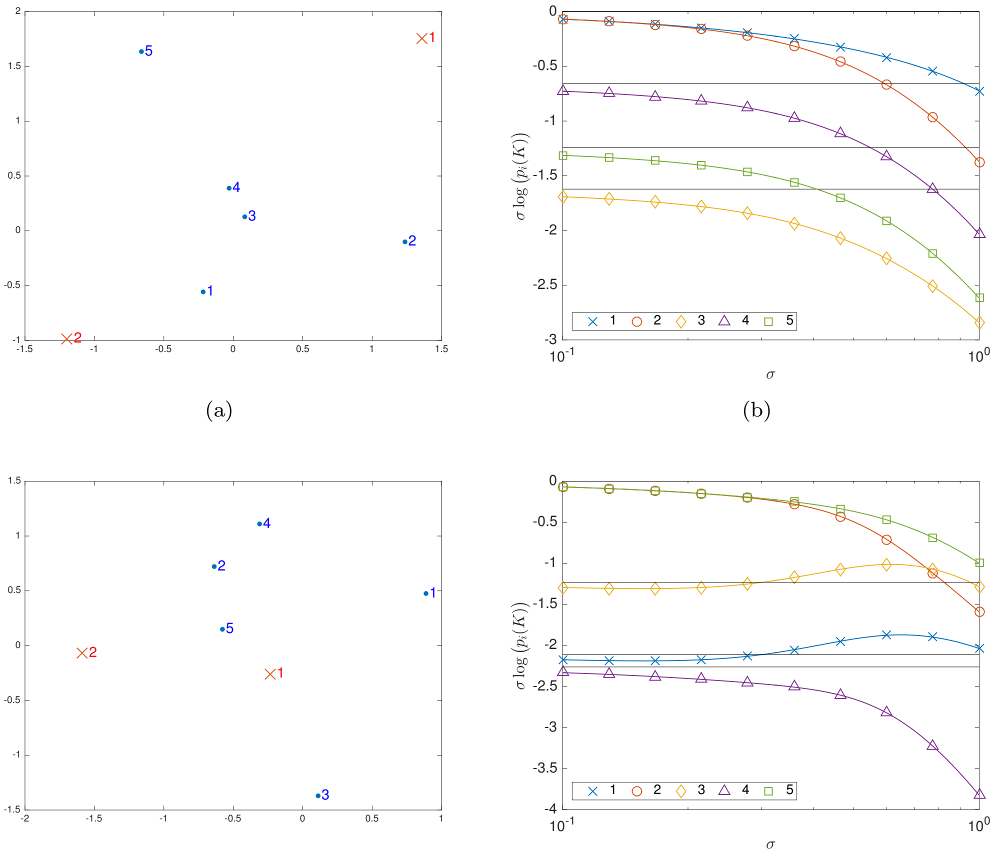
$$\lim_{\sigma\to 0} K \begin{bmatrix} \exp(\|\mathbf{x_2} - \mathbf{c_1}\|_2^2/\sigma) & 0 \\ 0 & \exp(\|\mathbf{x_1} - \mathbf{c_2}\|_2^2/\sigma) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Example 2.11. We repeat the experiment of Example 2.10. See Figures 1(c) and 1(d). In this example the property that each center $\mathbf{c_j}$ has a distinct closest data point $\mathbf{x}_{\phi(\mathbf{j})}$ is not satisfied, since both centers have the same closest data point $\mathbf{x_5}$. Consequently Proposition 2.1 does not apply. In this case we have

$$\lim_{\sigma\to 0} K \begin{bmatrix} \exp(\|\mathbf{x_2} - \mathbf{c_1}\|_2^2/\sigma) & 0 \\ 0 & \exp(\|\mathbf{x_1} - \mathbf{c_2}\|_2^2/\sigma) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}.$$

However, Theorem 2.8 does apply, and we have

$$\lim_{\sigma\to 0} K \begin{bmatrix} \exp(-\|\mathbf{x_5} - \mathbf{c_1}\|_2^2/\sigma) & \exp(-\|\mathbf{x_5} - \mathbf{c_1}\|_2^2/\sigma) \\ \exp(-\|\mathbf{x_2} - \mathbf{c_1}\|_2^2/\sigma) & \exp(-\|\mathbf{x_2} - \mathbf{c_2}\|_2^2/\sigma) \end{bmatrix}^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}.$$



(a) (b)

(c) (d)

Fig. 1. *Left plots: Data points in blue and RBF centers in red with crosses. Right plots: Convergence of statistical leverage scores.*

3. Approximation of statistical leverage scores of a fixed matrix.

In this section we show how we can use max-plus statistical leverage scores, or rather the obligated permanent scores, to approximate the scores of a fixed, nonparametrized matrix.

Let $A \in \mathbb{R}^{n\times d}$, and define $M \in \mathbb{R}^{n\times d}$ by

$$m_{ij} = \exp\big(\mathrm{perm}(\log|A|, j \mapsto i)\big).$$

Then the max-plus *statistical leverage score approximation* $q(A) \in \mathbb{R}^n$ is given by

$$q_i(A) = \|C_{i\cdot}\|_2^2 \quad \text{for } i = 1, \ldots, n, \tag{23}$$

where $C = M\mathrm{diag}(s)^{-1}$, with $s \in \mathbb{R}^d$ given by $s_j = \|M_{\cdot j}\|_2$ for $j = 1, \ldots, d$.

Heuristic 3.1. *For $A \in \mathbb{R}^{n\times d}$, we have*

$$p(A) \approx q(A),$$

where $p(A)$ are the exact statistical leverage scores and $q(A)$ are the max-plus approximate scores of A. The approximation $\approx$ should be interpreted as meaning an order of magnitude level approximation.

The intuition behind this approximation is as follows. The obligated permanent $\operatorname{perm}(\log|A|, j \mapsto i)$ scores how "strong" row i is in column j. These scores are then exponentiated and normalized to give the matrix C, where c_{ij} also scores how "strong" row i is in column j, except that now we also have $\sum_i c_{ij}^2 = 1$. The approximate scores $q(A)$ are then given by the squared row norms of C. Thus if we add new copies of an existing row, then this will reduce the original row's score in the natural way. This approximation is compatible with the asymptotic results of section 2 in the following sense.

Theorem 3.2. *Let $K = \exp(\mathcal{A}/\sigma) \in \mathbb{R}^{n\times d}$ be an RBFN matrix, and let $q(K)$ be the max-plus approximated scores of K. Then*

$$\lim_{\sigma\to 0} \sigma \log q_i(K) = p_i(\mathcal{A}),$$

where $p(\mathcal{A})$ are the max-plus scores of $\mathcal{A}$.

Proof. For $K = \exp(\mathcal{A}/\sigma)$, the matrix $M \in \mathbb{R}^{n\times d}$, in Heuristic 3.1, is given by

$$m_{ij} = \exp\big(\operatorname{perm}(\mathcal{A}, j \mapsto i)/\sigma\big).$$

The diagonal scaling $s \in \mathbb{R}^d$ is given by the column norms of M and satisfies

$$\exp\big(\operatorname{perm}(\mathcal{A})/\sigma\big) \le s_j \le n \exp\big(\operatorname{perm}(\mathcal{A})/\sigma\big),$$

so that

$$\sum_{j=1}^{d} (R_{ij}/n)^2 \le q_i(K) \le \sum_{j=1}^{d} R_{ij}^2,$$

where

$$R_{ij} = \exp\Big(\big(\operatorname{perm}(\mathcal{A}, j \mapsto i) - \operatorname{perm}(\mathcal{A})\big)/\sigma\Big).$$

Thus

$$\begin{aligned}\lim_{\sigma\to 0} \sigma \log q_i(K) &= \lim_{\sigma\to 0} \sigma \log\Big(\max_{j=1}^{d} R_{ij}^2\Big)\\ &= 2\max_{j=1}^{d}\big(\operatorname{perm}(\mathcal{A}, j\mapsto i) - \operatorname{perm}(\mathcal{A})\big) = p_i(\mathcal{A}).\end{aligned}$$ □

Theorem 3.2 tells us that the asymptotic behavior of the max-plus approximation given in Heuristic 3.1 will match the max-plus statistical leverage scores, which, as we showed in section 2, match the asymptotic behavior of the exact scores.

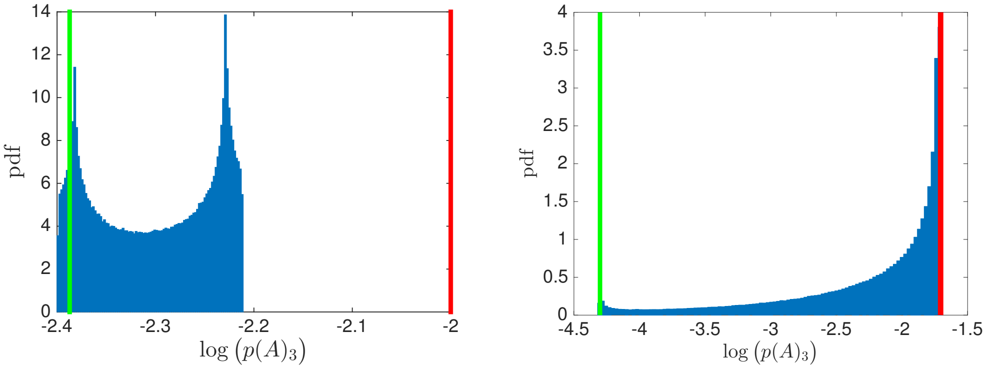
Example 3.3. Consider

$$A = \begin{bmatrix} 1000 & 1000 \\ 1 & 100 \\ 10 & 1 \end{bmatrix}, \quad \mathcal{A} = \log|A| = \begin{bmatrix} 3 & 3 \\ 0 & 2 \\ 1 & 0 \end{bmatrix}, \quad \operatorname{perm}(\mathcal{A}, j \mapsto i) = \begin{bmatrix} 5 & 4 \\ 3 & 5 \\ 4 & 3 \end{bmatrix}.$$

We have $p(A) = [0.9999, 0.9918, 0.0083]$ and $q(A) = [0.9999, 0.9901, 0.0100]$, which captures the order of magnitude of all of the scores.

Example 3.4. Consider

$$A = \begin{bmatrix} 1000 & 1000 \\ 1 & 100 \\ 10 & 10 \end{bmatrix}, \quad \mathcal{A} = \log|A| = \begin{bmatrix} 3 & 3 \\ 0 & 2 \\ 1 & 1 \end{bmatrix}, \quad \operatorname{perm}(\mathcal{A}, j \mapsto i) = \begin{bmatrix} 5 & 4 \\ 3 & 5 \\ 4 & 4 \end{bmatrix}.$$



(a)

(b)

FIG. 2. *Distribution of* $\log\big(p(A)_3\big)$ *for randomly generated matrices based on* (a) *Example* 1.3 *and* (b) *Example* 3.4*. Max-plus approximation marked in red, and score of example problem in green.*

We have $p(A) = [0.9999, 1, 0.00009]$ and $q(A) = [0.9998, 0.9805, 0.0197]$. In this example the max-plus approximation fails to capture the order of magnitude of the score of row 3.

To understand why the max-plus approximation works better for Example 3.3 than Example 3.4, it is important to consider the fact that the max-plus approximation is "blind to sign" in the sense that it does not depend on the sign or complex argument of the entries in A. To illustrate this point further we randomly generate many matrices with the same sized entries as the previous example problems but with independent, uniformly distributed complex arguments. For each of these randomly generated matrices we compute the log of the statistical leverage score of row 3 and plot a histogram of the results. See Figure 2. Note that for Example 1.3 the scores are always confined to a narrow band, which is within an order of magnitude of the max-plus approximation. But note that for Example 3.4 the scores have a light lower tail, so that there is a set of small measure for which the max-plus approximation is not accurate to within an order of magnitude. The matrix A in Example 3.4 belongs to this small measure set.

The only direct support for Heuristic 3.1 comes from the empirical evidence presented in section 4. We have no theorem saying that the accuracy of the approximation should be within a certain factor for an arbitrary matrix. This is because there are certain problems for which the max-plus approximation is very inaccurate. Whenever we have such a problem we find that applying a random perturbation to the complex arguments of its entries results in a matrix A' whose statistical leverage scores are well approximated by the max-plus approximation. In this sense we say that the max-plus approximation provides an order of magnitude approximation for the statistical leverage scores for all but a small measure set of "nasty" matrices.

Of course this is cold comfort if we are interested in approximating the scores of a particular matrix A that happens to fall in this nasty set. Ultimately the success of the max-plus approximation in practice will depend on identifying domains of problems and compatible preprocessing techniques that give rise to matrices where these problems do not occur, or on finding new approximation strategies that also satisfy a result like Theorem 3.2, but which are more robust in practice. We should note that issues of this sort are common to other methods which use max-plus algebra to approximate classical linear algebra objects including eigenvalues [11] and LU factors [16]. In these other applications we find that max-plus methods tend to work well on large sparse matrices from practical problems, particularly highly unstructured problems with a large range of entry sizes.

4. Numerical experiments.

In this section we apply the max-plus approximation of Heuristic 3.1 to some larger numerical examples and compare using different sampling distributions in Algorithm 1. For each matrix $A \in \mathbb{R}^{n\times d}$, we compute the exact statistical leverage score probability distribution $p(A)/d$, using (2). We use Algorithm 3 to compute the max-plus approximation $q(A)/d$. For comparison we also compute an alternative statistical leverage score approximation which, like the max-plus approximation, depends only on the size of the entries in A. The *column normalized row norm* (CNRN) scores of A are given by

$$r_i(A) = \|C_{i\cdot}\|_2^2 \quad \text{for } i = 1, \dots, n, \tag{24}$$

where $C = A\ \mathrm{diag}(s)^{-1}$, with $s \in \mathbb{R}^d$ given by $s_j = \|A_{\cdot j}\|_2$ for $j = 1, \dots, d$. We compute the CNRN probability distribution $r(A)/d$ for each matrix. Note that, although their computation is far more straightforward, the cost of computing the CNRB scores is of the same order as the max-plus scores.

For each numerical example we formulate and solve the least squares problem $x^* = \arg\min_{x\in\mathbb{R}^{d-1}} \|Ax - y\|_2$. We then compute approximate solutions using Algorithm 1. For each different sampling distribution and a range of values of r, we run 100 independent instances of Algorithm 1.

4.1. RBFN matrices.

For each of the following example problems we construct the matrix $K = \exp(\mathcal{A})$ from the data points and centers as described in section 2. For each example we choose data points and centers that lie in $\mathbb{R}^{25}$.

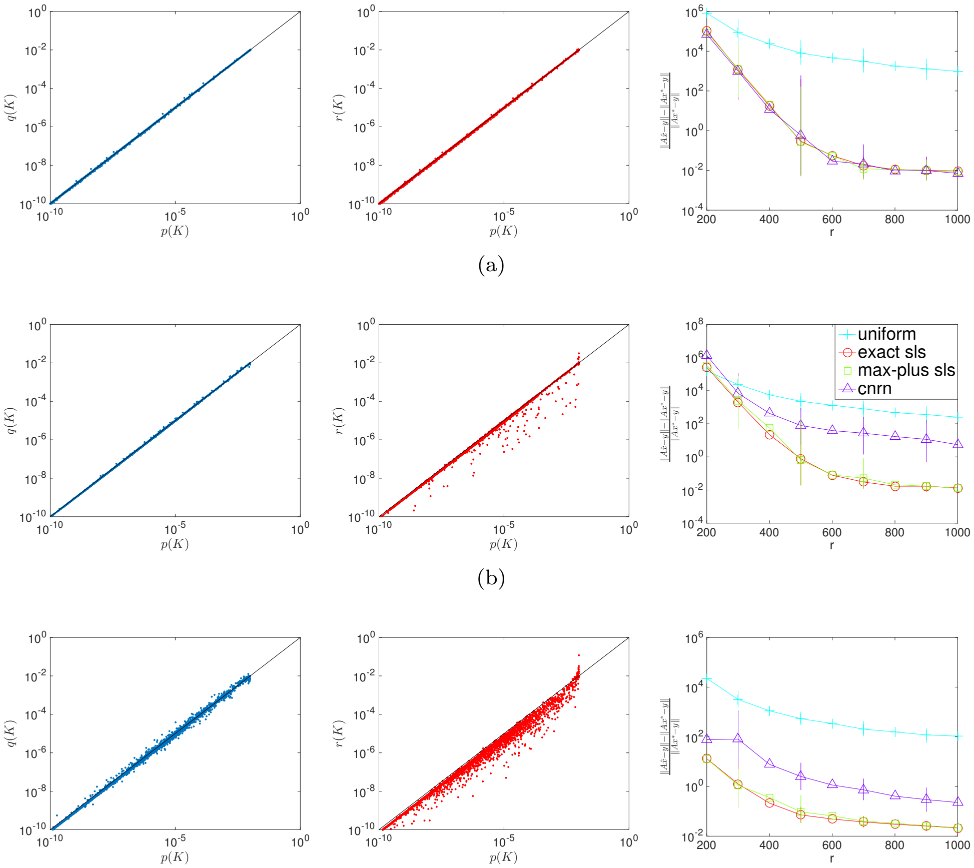
Tall skinny example. We set $n = 10000$ and $d = 100$; each data point and center is sampled independently from $\mathcal{N}(\underline{0}, I)$.

Moderate aspect ratio example. We set $n = 2500$ and $d = 100$; each data point and center is sampled independently from $\mathcal{N}(\underline{0}, I)$.

Tall and skinny with clustered centers example. We set $n = 10000$ and $d = 100$; each data point is sampled independently from $\mathcal{N}(\underline{0}, I)$, and each center is sampled independently from $\mathcal{N}(\underline{0}, I/2)$.

For each matrix we sample a vector $\mathbf{h}$ from $\mathcal{N}(\underline{0}, I)$ and use it to construct the target vector $y \in \mathbb{R}^n$, with $y_i = \sin\big(\langle \mathbf{x}_i, \mathbf{h}\rangle/\sqrt{k}\big)\exp\big(-\|\mathbf{x}_i\|^2/k\big)$, where $\mathbf{x}_i \in \mathbb{R}^{25}$ is the ith data point for $i = 1, \dots, n$. We then compute approximate solutions to $\min_x \|Kx - y\|_2$, using Algorithm 1 as described above. The results of these experiments are displayed in Figure 3.

All of the RBFN example problems have highly nonuniform scores. For the tall skinny problem both of the approximation methods capture the order of magnitude of all of the scores. The uniform sampling method performs poorly, while all of the nonuniform methods perform well. For the moderate aspect ratio problem the max-plus approximation captures the order of magnitude of all of the scores, while the CNRN approximation underapproximates many of the scores by more than a factor of 10. The uniform and CNRN sampling methods both perform poorly on this problem, while the exact statistical leverage scores and max-plus approximation both perform well. For the tall and skinny with clustered centers problem the max-plus



(a)

(b)

(c)

Fig. 3. *Left: Scatterplot of exact statistical leverage scores versus max-plus approximation. Middle: Scatterplot of exact statistical leverage scores versus CNRN approximation. Right: Error versus sample size for sampled least squares approximations. Errors plotted are the geometric mean of* 100 *independent trials; vertical bars show* 90% *range. Plots are for* (a) *tall and skinny,* (b) *moderate aspect ratio, and* (c) *tall and skinny with clustered centers. Plots of statistical leverage score approximations have their axes clipped to focus on detail of higher scores.*

approximation captures the order of magnitude of all of the scores, but the CNRN approximation is inaccurate by more than a factor of 10 for many of the rows. The uniform and CNRN sampling methods both perform poorly on this problem, while the exact statistical leverage scores and max-plus approximation both perform well.

The performance of the different approximation methods can be understood by examining the behavior of $K = \exp(\mathcal{A}/\sigma)$ in the limit $\sigma \to 0$. Theorems 2.9 and 3.2 guarantee that the max-plus approximation will match the exact statistical leverage scores in the limit $\sigma \to 0$, so we expect the approximation to be fairly accurate for moderate values of σ for all of the example problems. For the tall skinny example there are many more data points than centers, so with high probability the centers will be distributed evenly among the data points and each center will have a distinct closest data point. Therefore from Proposition 2.1 we expect the matrix K to have close to orthogonal columns and hence the CNRN approximation to be accurate. For the moderate aspect ratio problem there are fewer data points per center, so there is a greater likelihood that centers will share nearest data points. In this case, as in Example 2.11, the matrix K does not converge to a matrix with orthogonal columns, so the CNRN method performs poorly. Similarly for the tall skinny with clustered centers problem, since the centers are distributed unevenly among the data points, there is a greater likelihood that centers will share nearest data points, and again, this is why the CNRN method performs poorly on this example.

4.2. Varying coherence. The next test set of matrices that we use is randomly generated using the scheme set out in [17, section 4.1]. The *coherence* of a matrix $A \in \mathbb{R}^{n\times d}$ is equal to its largest individual statistical leverage score. A coherent matrix, with a large coherence value, will have a wide range of statistical leverage scores. An incoherent matrix, with a small coherence value, will have more uniform statistical leverage scores. We set $n = 10^5$, $d = 50 + 1$, and $\Sigma \in \mathbb{R}^{d\times d}$, with $\Sigma_{ij} = 2 \times 0.5^{|i-j|}$. The example matrices are generated as follows.

Incoherent example. Each row of $A \in \mathbb{R}^{n\times d}$ is chosen independently from a multivariate Gaussian distribution $\mathcal{N}(\underline{1}, \Sigma)$, where $\underline{1} \in \mathbb{R}^d$ is a vector of ones.

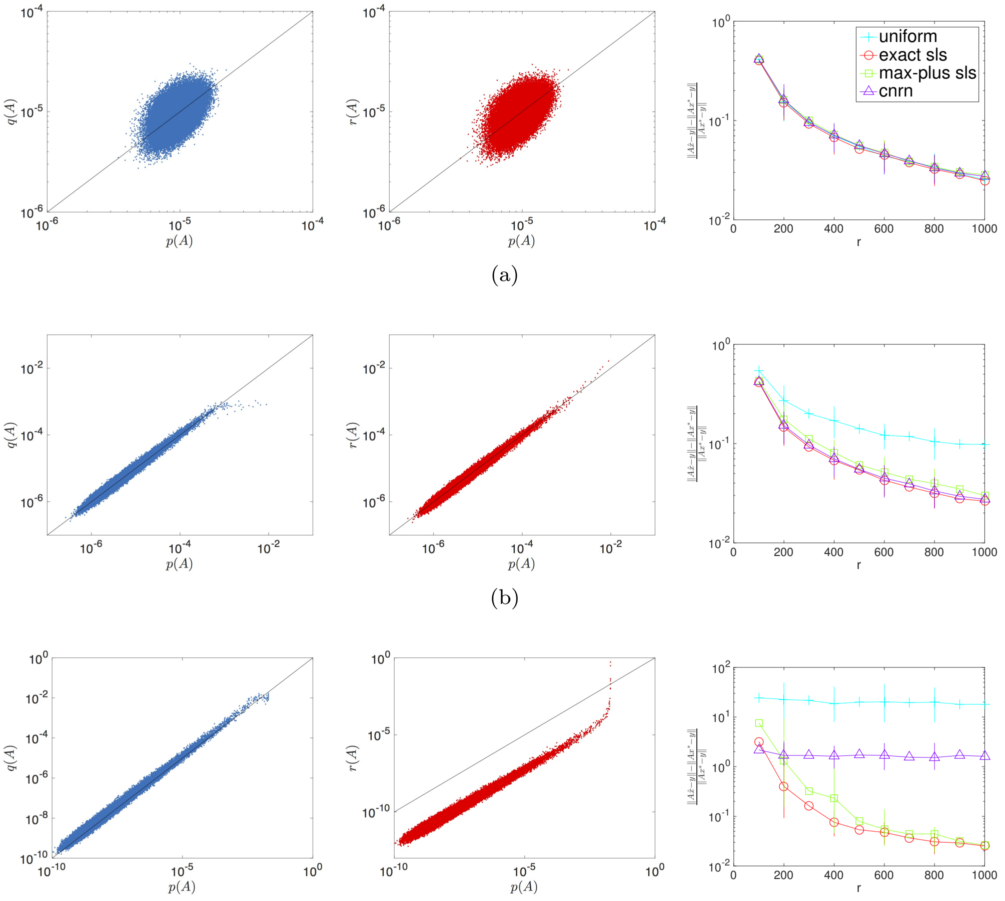
Semicoherent example. Each row of $A \in \mathbb{R}^{n\times d}$ is chosen independently from a multivariate t-distribution $t_3(\underline{1}, \Sigma)$, with three degrees of freedom.

Coherent example. Each row of $A \in \mathbb{R}^{n\times d}$ is chosen independently from a multivariate t-distribution $t_1(\underline{1}, \Sigma)$, with one degree of freedom.

For each example matrix we formulate and solve the least squares problem $x^* = \arg\min_{x\in\mathbb{R}^{d-1}} \|Bx - y\|_2$, where $B = A([1, \dots, n], [1, \dots, d-1])$ and $y = A([1, \dots, n], [d])$. We then compute approximate solutions using Algorithm 1. The results of these experiments are displayed in Figure 4.

For the incoherent example the exact statistical leverage scores are nearly uniform. Both approximation methods capture the order of magnitude of all of the scores, and all of the different sampling methods have the same performance. For the semicoherent example the exact statistical leverage scores range between 10^{-2} and 10^{-6}. Both approximation methods capture the order of magnitude of all of the scores, but the CNRN approximation is slightly more accurate. The rows with the largest exact scores are underapproximated by the max-plus scores, but never by more than a factor of 10. The uniform sampling method does not perform as well as the other methods in this example. For the coherent example the exact statistical leverage scores range between 10^{-2} and 10^{-10}. The max-plus approximation captures the order of magnitude of all of the scores, but the CNRN scores overapproximate the largest scores and underapproximate many of the smaller scores. The uniform sampling method performs very poorly on this problem, and the CNRN method does not perform as well as the exact statistical leverage scores method or the max-plus approximation method, which both perform well.

4.3. Structured matrices from practical problems. The previous examples are all randomly generated and so avoid the sort of degenerate behavior that cannot be detected by our "blind to sign" method. We did attempt to run our approximation on some more structured problems from practical applications, but the results were inconsistent. We found many examples where both the max-plus approximation and the CNRN approximation worked well, as well as examples where the max-plus approximation worked well, but the CNRN method failed. However, we also found several examples where both approximation methods failed and even some examples where the CNRN approximation worked well, but the max-plus method failed.



(a)

(b)

(c)

FIG. 4. *Left: Scatterplot of exact statistical leverage scores versus max-plus approximation. Middle: Scatterplot of exact statistical leverage scores versus CNRN approximation. Right: Error versus sample size for sampled least squares approximations. Errors plotted are the geometric mean of* 100 *independent trials; vertical bars show* 90% *range. Plots are for* (a) *incoherent example,* (b) *semicoherent example, and* (c) *coherent example.*

5. Max-plus statistical leverage score algorithm. Lemma 2.6 shows us how to calculate the max-plus statistical leverage scores $p(\mathcal{A})$ for $\mathcal{A} \in \mathbb{R}_{\max}^{n\times d}$. First we compute an optimal assignment $\phi \in \mathrm{oas}(\mathcal{A})$. Next we set $\mathcal{B} = \mathcal{A}([\phi(1), \dots, \phi(d)], [1, \dots, d])$ and compute $\mathcal{B}^{\otimes -1}$. Then

$$p_i(\mathcal{A}) = \begin{cases} 0 & \text{if } i \text{ is assigned by } \phi, \\ 2\big(\mathcal{A} \otimes \mathcal{B}^{\otimes -1} \otimes \underline{0}\big)_i & \text{otherwise.} \end{cases} \tag{25}$$

See Algorithm 3. We treat the computations of the optimal assignment and max-plus inverse separately below. The multiplication on line 4 has cost $\mathcal{O}(d^2)$. For $i = 1, \dots, n$, setting $p_i(\mathcal{A})$ has cost $\mathcal{O}(d)$, so that the total cost of setting $p(\mathcal{A})$ is $\mathcal{O}(nd)$. Note that each row can be treated independently in parallel and that if $\mathcal{A}$ is a sparse matrix,[1] then the total cost of setting $p(\mathcal{A})$ is $\mathcal{O}\big(\mathrm{nf}(\mathcal{A})\big)$, where $\mathrm{nf}(\mathcal{A})$ is the number of finite entries in $\mathcal{A}$.

To compute an optimal assignment $\phi \in \mathrm{oas}(\mathcal{A})$, we can use the Hungarian algorithm [18], the successive shortest paths algorithm [19], or the auction algorithm [4]. Applied directly to $\mathcal{A} \in \mathbb{R}_{\max}^{n\times d}$, all of these algorithms have cost $\mathcal{O}(nd^2)$. However, we can reduce this cost considerably by noting that the optimal assignment of a tall skinny $n \times d$ matrix depends only on the d largest entries in each column. For each row of $\mathcal{A}$ we select the d largest entries, which we then sort in decreasing order. This results in a sparse matrix with at most d entries per column and a known sorting order for each column. We then pass this matrix to the successive shortest paths algorithm, which is able to compute the optimal assignment with cost $\mathcal{O}(d^3)$. An efficient implementation which exploits the fact that the d largest entries in each column have been sorted is essential to achieve this lower cost. To select the d largest entries in each column we use quickselect [14]. Like quicksort this algorithm has poor worst case cost but good average case cost. For quickselect, finding the d largest entries in a column of length n has worst case cost $\mathcal{O}(n^2)$ but average case cost $\mathcal{O}(n)$. This means that it is possible for the operation of finding the d largest entries in each column to have cost total $\mathcal{O}(n^2d)$. However, for a large problem, if we think of each column as being randomly ordered, independently of the other columns, then with high probability the total cost is $\mathcal{O}(nd)$. Using this approach the total average case cost of computing the optimal assignment ϕ is $\mathcal{O}(nd+d^3)$. Clearly each column can be treated independently in parallel, and if $\mathcal{A}$ is a sparse matrix, then the total cost is $\mathcal{O}(\mathrm{nf}(\mathcal{A})+d^3)$.

To compute the max-plus inverse $\mathcal{B}^{\otimes -1}$ we adapt the approach taken in [16, Appendix A], where the authors present an algorithm for computing max-plus LU factors. For $\mathcal{B} \in \mathbb{R}_{\max}^{d\times d}$ and $\pi \in \mathrm{oas}(\mathcal{B})$, let $\mathcal{P}_\pi \in \mathbb{R}_{\max}^{d\times d}$ be the max-plus permutation matrix with

$$(\mathcal{P}_\pi)_{ij} = \begin{cases} 0 & \text{if } i=\pi(j), \\ -\infty & \text{otherwise.} \end{cases} \tag{26}$$

There exist max-plus diagonal matrices[2] such that

$$\mathcal{H} = \mathcal{P}_\pi \otimes \mathcal{D}_1 \otimes \mathcal{B} \otimes \mathcal{D}_2 \tag{27}$$

satisfies $\mathcal{h}_{ij} \le 0$ and $\mathcal{h}_{ii} = 0$ for all $i,j = 1,\dots,d$. We say that $\mathcal{H}$ is a *Hungarian scaling* of $\mathcal{B}$. The coefficients of the diagonal scaling matrices are given by the dual variables in the LPP form of the optimal assignment problem. Primal dual algorithms for computing the optimal assignment of a matrix, like those listed above, will also produce these scaling coefficients a byproduct. Ordinarily we would need to apply one of these algorithms to $\mathcal{B}$ with worst case cost $\mathcal{O}(d^3)$, but in this setting we can use the results from the previous computation applied to $\mathcal{A}$.

To compute the max-plus inverse of $\mathcal{B}$ we use the formula

$$\mathcal{B}^{\otimes -1} = \mathcal{D}_1 \otimes \mathcal{H}^{\otimes -1} \otimes D_2, \tag{28}$$

where the entries in the inverse of the Hungarian matrix $\mathcal{H}$ can be calculated as follows. Let $G(\mathcal{H})$ be the graph with vertices $\{1,\dots,d\}$ and an edge $i \mapsto j$ with

[1] A sparse max-plus matrix is one with many entries equal to minus infinity. If $A \in \mathbb{R}^{n\times d}$ is a conventional sparse matrix, then $\log|A| \in \mathbb{R}_{\max}^{n\times d}$ is a sparse max-plus matrix.

[2] A max-plus diagonal matrix is one whose off-diagonal entries are all equal to minus infinity.

weight $\mathcal{h}_{ij}$ whenever $\mathcal{h}_{ij} \neq -\infty$. Then

$$\left(\mathcal{H}^{\otimes -1}\right)_{ij} = \text{weight of the maximally weighted path through } G(\mathcal{H}) \text{ from } i \text{ to } j. \tag{29}$$

Each row of $\mathcal{H}^{\otimes -1}$ can be computed by independently using Dijkstra's algorithm, with a total worst case cost of $\mathcal{O}(d^3)$ for a dense matrix.

The total average case cost of Algorithm 3 is therefore $\mathcal{O}(nd+d^3)$ or $\mathcal{O}(\mathrm{nf}(\mathcal{A})+d^3)$ in the sparse case. Of course, since quickselect has worst case cost $\mathcal{O}(n^2)$, Algorithm 3 has worst case cost $\mathcal{O}(n^2d + d^3)$, which is greater than the cost of computing the statistical leverage scores exactly. However, this worst case cost is extremely unlikely to be attained in practice unless the problem matrix has been carefully constructed for this purpose. If we want to minimize the worst case cost at the expense of the average case cost, we can instead use a heap-based algorithm to select the d largest entries in each column. This has worst case cost $\mathcal{O}(n \log(d)$ per column and results in a total worst case cost of $\mathcal{O}(nd \log(d) + d^3)$ or $\mathcal{O}(\mathrm{nf}(\mathcal{A}) \log(d) + d^3)$ in the sparse case. This approach results in a two-pass algorithm which is also appealing.

5.1. Max-plus statistical leverage score approximation algorithm. To compute the max-plus statistical leverage score approximation (23) for a complex matrix $A \in \mathbb{C}^{n\times d}$, we need to compute the full matrix of obligated permanents $R \in \mathbb{R}^{n\times d}_{\max}$, with $r_{ij} = \mathrm{perm}(\log|A|, j \mapsto i)$. Let $\mathcal{A} = \log|A|$, let ϕ be an optimal assignment of $\mathcal{A}$, and let $\mathcal{B} = \mathcal{A}([\phi(1), \dots, \phi(d)], [1, \dots, d])$. Then from Lemma 2.6, we have

$$r_{ij} = \mathcal{a}_{ij} + \left(\mathcal{B}^{\otimes -1} \otimes \underline{0}\right)_j + \mathrm{perm}(\mathcal{A})$$

for all i such that row i is unassigned by ϕ and for all $j = 1, \dots, d$. Computing $\mathrm{perm}(\mathcal{A})$ and $\mathcal{B}^{\otimes -1}$ as described above, the total cost of computing the obligated permanents for all of the unassigned rows is $\mathcal{O}(\mathrm{nnz}(A) + d^3)$.

Lemma 2.6 does not apply to assigned rows, so we must treat these separately. Note that the (i, j)-obligated permanent of a tall skinny $n \times d$ matrix can only depend on the (i, j) entry and the remaining $d + 1$ largest entries in each column (one more than for the basic permanent). As before we use quickselect to form the sparse matrix containing only the d+1 largest entries in each column, and then compute the optimal assignment of this matrix using the successive shortest paths algorithm. The obligated permanents are then computed using the approach taken in [16, Appendix A]. First we augment the associated bipartite graph with respect to the optimal assignment. Then, for each assigned row i, we calculate all of the (i, j)-obligated permanents through a single application of Dijkstra's algorithm, with an initial weight of zero on the ith row vertex. The total cost of computing the obligated permanents for the assigned rows in this way is $\mathcal{O}(\mathrm{nnz}(A) + d^3)$.

The max-plus statistical leverage score approximation is then a simple function of the obligated permanents. See Algorithm 2. The total cost of computing the max-plus statistical leverage score approximation is $\mathcal{O}(\mathrm{nnz}(A) + d^3)$.

Conclusion. We presented a max-plus algebraic analogue of statistical leverage scores. We showed that these scores could be used to calculate the exact asymptotic behavior of the statistical leverage scores of an RBFN matrix. We also showed how the max-plus scores could be used to approximate the scores of a conventional matrix. This approximation could be useful in practice since the max-plus scores can be computed very quickly. However, it is clear from our experiments that while the max-plus approximation given in Heuristic 3.1 works very well on randomly generated problems, it is unreliable on highly structured problems, which are more typical in applications. Developing a more reliable alternative approximation method, which still uses some of the ideas surrounding the max-plus statistical leverage scores, but which also uses sign information to provide a more reliable approximation, is a promising avenue for future research.

Algorithm 2 Given a max-plus matrix $\mathcal{A} \in \mathbb{R}_{\max}^{n \times d}$, compute $p(\mathcal{A})$.

```
compute an optimal assignment $\phi \in \mathrm{oas}(\mathcal{A})$
set $\mathcal{B} = \mathcal{A}([\phi(1), \dots, \phi(d)], [1, \dots, d])$
compute $\mathcal{B}^{\otimes -1}$
set $\chi = \mathcal{B}^{\otimes -1} \otimes \underline{0}$
for $i = 1, \dots, n$ do
    if $i$ assigned by $\phi$ then
        set $p_i(\mathcal{A}) = 0$
    else
        set $p_i(\mathcal{A}) = 2(\mathcal{A} \otimes \chi)_i$
    end if
end for
```

Algorithm 3 Given a max-plus matrix $A \in \mathbb{C}^{n \times d}$, compute $q(A)$.

```
set $\mathcal{A} = \log |A|$
compute an optimal assignment $\phi \in \mathrm{oas}(\mathcal{A})$
set $\mathcal{B} = \mathcal{A}([\phi(1), \dots, \phi(d)], [1, \dots, d])$
compute $\mathcal{B}^{\otimes -1}$
set $\chi = \mathcal{B}^{\otimes -1} \otimes \underline{0}$
for $i = 1, \dots, n$ do
    if $i$ assigned by $\phi$ then
        compute $R_{i,\cdot}$ by Dijkstra's algorithm
    else
        set $r_{ij} = a_{ij} + \chi_j + \mathrm{perm}(\mathcal{A})$ for $j = 1. \dots, d$
    end if
end for
for $j = 1, \dots, d$ do
    set $s_j = \sqrt{\sum_{i=1}^{n} \exp(2r_{ij})}$
end for
for $i = 1, \dots, n$ do
    set $q_i = \sum_{j=1}^{d} (\exp(r_{ij})/s_j)^2$
end for
```

REFERENCES

[1] M. Akian, R. Bapat, and S. Gaubert, *Generic asymptotics of eigenvalues using min-plus algebra*, in Proceedings of the Satellite Workshop on Max-Plus Algebras, IFAC SSSC01, Elsevier, New York, 2001.

[2] M. Akian, R. Bapat, and S. Gaubert, *Perturbation of eigenvalues of matrix pencils and the optimal assignment problem*, C. R. Math. Acad. Sci. Paris, 339 (2004), pp. 103–108.

[3] F. L. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat, *Synchronization and Linearity: An Algebra for Discrete Event Systems*, John Wiley & Sons, Chichester, UK, 2001.

[4] D. P. Bertsekas and D. A. Castanon, *The auction algorithm for the transportation problem*, Ann. Oper. Res., 20 (1989), pp. 67–96.

[5] D. S. Broomhead and D. Lowe, *Radial basis functions, multi-variable functional interpolation and adaptive networks*, Technical report, DTIC Document, 1988.

[6] P. Butkovič, *Max-Linear Systems: Theory and Algorithms*, Springer-Verlag, London, 2010.

[7] K. L. Clarkson and D. P. Woodruff, *Low rank approximation and regression in input sparsity time*, in Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing, STOC '13, ACM, New York, 2013, pp. 81–90, https://doi.org/10.1145/2488608.2488620.

[8] M. B. Cohen, Y. T. Lee, C. Musco, C. Musco, R. Peng, and A. Sidford, *Uniform sampling for matrix approximation*, in Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS '15, ACM, New York, 2015, pp. 181–190, https://doi.org/10.1145/2688073.2688113.

[9] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff, *Fast approximation of matrix coherence and statistical leverage*, J. Mach. Learn. Res., 13 (2012), pp. 3475–3506.

[10] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, *Sampling algorithms for ℓ_2 regression and applications*, in Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2006, pp. 1127–1136.

[11] S. Gaubert and M. Sharify, *Tropical scaling of polynomial matrices*, in Positive Systems: Proceedings of the Third Multidisciplinary International Symposium on Positive Systems: Theory and Applications (POSTA 2009), Springer, Berlin, 2009, pp. 291–303.

[12] A. Gittens and M. W. Mahoney, *Revisiting the Nyström method for improved large-scale machine learning*, J. Mach. Learn. Res., 17 (2016), 117.

[13] B. Heidergott, G. J. Olsder, and J. van der Woude, *Max Plus at Work. Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications*, Princeton University Press, Princeton, NJ, 2006.

[14] C. A. R. Hoare, *Algorithm* 65: *Find*, Commun. ACM, 4 (1961), pp. 321–322.

[15] J. Hook, *Max-plus singular values*, Linear Algebra Appl., 486 (2015), pp. 419–442.

[16] J. Hook and F. Tisseur, *Incomplete LU Preconditioner Based on Max-Plus Approximation of LU Factorization*, MIMS EPrint 2016.47, University of Manchester, Manchester, UK, 2016.

[17] P. Ma, M. W. Mahoney, and B. Yu, *A statistical perspective on algorithmic leveraging*, J. Mach. Learn. Res., 16 (2015), pp. 861–911.

[18] J. Munkres, *Algorithms for the assignment and transportation problems*, J. Soc. Indust. Appl. Math., 5 (1957), pp. 32–38, https://doi.org/10.1137/0105003.

[19] J. B. Orlin and Y. Lee, *Quickmatch—A Very Fast Algorithm for the Assignment Problem*, Working Paper 3547-93, Sloan School of Management, MIT, Cambridge, MA, 1993.

[20] D. P. Woodruff, *Sketching as a tool for numerical linear algebra*, Found. Trends Theor. Comput. Sci., 10 (2014), pp. 1–157.