Sparse Approximate Solutions to Max-Plus Equations

Nikos Tsilivis[1], Anastasios Tsiamis[2], and Petros Maragos[1]

[1] School of ECE, National Technical University of Athens, Greece
ntsilivis96@gmail.com, maragos@cs.ntua.gr
[2] ESE Department, SEAS, University of Pennsylvania, USA
atsiamis@seas.upenn.edu

Abstract. In this work, we study the problem of finding approximate, with minimum support set, solutions to matrix max-plus equations, which we call sparse approximate solutions. We show how one can obtain such solutions efficiently and in polynomial time for any ℓ_p approximation error. Subsequently, we propose a method for pruning morphological neural networks, based on the developed theory.

Keywords: Sparsity · Max-plus Algebra · Submodular Optimization

1 Introduction

In the last decades, the areas of signal and image processing had been greatly benefited from the advancement of the theory of sparse representations [10]. Given a few linear measurements of an object of interest, sparse approximation theory provides efficient tools and algorithms for the acquisition of the sparsest (most zero elements) solution of the corresponding underdetermined linear system [10,19]. Based on the sparsity assumption of the initial signal, this allows perfect reconstruction from little data. Ideas stemming from this area had also given birth to *compressed sensing* techniques [9,5] that allow accurate reconstructions from limited random projections of the initial signal, with wide-ranging applications in photography, magnetic resonance imaging and others.

Yet, there is a variety of problems in areas such as scheduling and synchronization [7,2], morphological image and signal analysis [21,13,17] and optimization and optimal control [2,1,12] that do not admit linear representations. Instead, these problems share the ability to be described as a system of nonlinear equations, which involve maximum operations together with additions. The relevant theoretical framework has initially been developed in [7,2,4] and the appropriate algebra for this kind of problems is called *max-plus* algebra. Motivated by the sparsity in the linear setting, [22] introduced the notion of sparsity (signals with many $-\infty$ values, i.e. the identity element of this algebra) in max-plus algebra. Herein, we contribute to this theory, by studying the problem of sparse approximate solutions to matrix max-plus equations allowing the approximation error to be measured by any l_p norm. Indicatively, we also present a preliminary application of the theory to the pruning of morphological neural networks.

In particular, we make the following contributions: a) We pose a *generalized* problem of finding the sparsest approximate solution to matrix max-plus equations under a constraint which makes the problem more tractable, also known as the "lateness constraint". The approximation error is in terms of any ℓ_p norm, for $p < \infty$. b) We prove that for any ℓ_p norm, $p < \infty$, the problem has supermodular properties, which allows us to solve it approximately but efficiently via a greedy algorithm, with a derived approximation ratio. c) We investigate the ℓ_∞ case without the "lateness constraint", reveal its hardness and propose a heuristic method for solving it. d) We demonstrate how one may prune whole neurons from morphological neural networks using the developed theory.

2 Background Concepts

For max and min operations we use the well-established lattice-theoretic symbols of $\vee$ and $\wedge$, respectively. We use roman letters for functions, signals and their arguments and greek letters mainly for operators. Also, boldface roman letters for vectors (lowcase) and matrices (capital). If $\mathbf{M} = [m_{ij}]$ is a matrix, its (i, j) element is also denoted as m_{ij} or as $[\mathbf{M}]_{ij}$. Similarly, $\mathbf{x} = [x_i]$ denotes a column vector, whose i-th element is denoted as $[\mathbf{x}]_i$ or simply x_i.

2.1 Max-plus algebra

Max-plus arithmetic consists of the idempotent semiring $(\mathbb{R}_{\max}, \max, +)$, where $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$ is equipped with the standard maximum and sum operations, respectively. *Max-plus algebra* consists of vector operations that extend max-plus arithmetic to $\mathbb{R}^n_{\max}$. They include the pointwise operations of partial ordering $\mathbf{x} \leq \mathbf{y}$ and pointwise supremum $\mathbf{x} \vee \mathbf{y} = [x_i \vee y_i]$, together with a class of vector transformations defined below. Max-plus algebra is isomorphic to the *tropical algebra*, namely the min-plus semiring $(\mathbb{R}_{\min}, \min, +)$, $\mathbb{R}_{\min} = \mathbb{R} \cup \{\infty\}$ when extended to $\mathbb{R}^n_{\min}$ in a similar fashion. Vector transformations on $\mathbb{R}^n_{\max}$ (resp. $\mathbb{R}^n_{\min}$) that distribute over max-plus (resp. min-plus) vector superpositions can be represented as a max-plus $\boxplus$ (resp. min-plus $\boxplus'$) product of a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}_{\max}(\mathbb{R}^{m\times n}_{\min})$ with an input vector $\mathbf{x} \in \mathbb{R}^n_{\max}(\mathbb{R}^n_{\min})$:

$$[\mathbf{A} \boxplus \mathbf{x}]_i \triangleq \bigvee_{k=1}^{n} a_{ik} + x_k, \; [\mathbf{A} \boxplus' \mathbf{x}]_i \triangleq \bigwedge_{k=1}^{n} a_{ik} + x_k \tag{1}$$

More details about general algebraic structures that obey those arithmetics can be found in [18]. In the case of a max-plus matrix equation $\mathbf{A} \boxplus \mathbf{x} = \mathbf{b}$, there is a solution if and only if the vector

$$\hat{\mathbf{x}} = (-\mathbf{A})^\intercal \boxplus' \mathbf{b} \tag{2}$$

satisfies it [7,4,18]. We call this vector the *principal solution* of the equation. It also satisfies the inequality $\mathbf{A} \boxplus \hat{\mathbf{x}} \leq \mathbf{b}$. Lastly, a vector $\mathbf{x} \in \mathbb{R}^n_{\max}$ is called *sparse* if it contains many $-\infty$ elements and we define its *support set*, $\text{supp}(\mathbf{x})$, to be the set of positions where vector $\mathbf{x}$ has finite values, that is $\text{supp}(\mathbf{x}) = \{i \mid x_i \neq -\infty\}$.

2.2 Submodularity

Let U be a universe of elements. A set function $f : 2^U \to \mathbb{R}$ is called *submodular* [16] if $\forall A \subseteq B \subseteq U,\ k \notin B$ holds:

$$f(A \cup \{k\}) - f(A) \geq f(B \cup \{k\}) - f(B). \tag{3}$$

A set function f is called *supermodular* if $-f$ is submodular. Submodular functions occur as models of many real world evaluations in a number of fields and allow many hard combinatorial problems to be solved fast and with strong approximation guarantees [15,3]. It has been suggested that their importance in discrete optimization is similar to convex functions' in continuous optimization [16].

The following definition captures the idea of how far a given function is from being submodular and generalizes the notion of submodularity.

Definition 1. *[8] Let U be a set and $f : 2^U \to \mathbb{R}^+$ be an increasing, non-negative, function. The submodularity ratio of f is*

$$\gamma_{U,k}(f) \triangleq \min_{L \subseteq U, S : |S| \leq k, S \cap L = \emptyset} \frac{\sum_{x \in S} f(L \cup \{x\}) - f(L)}{f(L \cup S) - f(L)} \tag{4}$$

Proposition 1. *[8] An increasing function $f : 2^U \to \mathbb{R}$ is submodular if and only if $\gamma_{U,k}(f) \geq 1,\ \forall\, U, k$.*

In [8], the authors used the submodularity ratio to analyze the properties of greedy algorithms in discrete optimization problems with functions that are only approximately submodular ($\gamma \in (0, 1)$). They proved that the performance of the algorithms degrade gradually as a function of γ, thus allowing guarantees for a wider variety of objective functions.

3 Sparse approximate solutions to max-plus equations

We consider the problem of finding the sparsest approximate solution to the max-plus matrix equation $\mathbf{A} \boxplus \mathbf{x} = \mathbf{b}, \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^m$. Such a solution should i) have minimum support set $\text{supp}(\mathbf{x})$, and ii) have small enough approximation error $\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}\|_p^p$, for some $\ell_p, p < \infty$ norm. For this reason, given a prescribed constant ϵ, we formulate the following optimization problem:

$$\begin{aligned} \arg \min_{\mathbf{x} \in \mathbb{R}^n_{\max}} |\text{supp}(\mathbf{x})|, \text{ s.t. } \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}\|_p^p \leq \epsilon,\ p < \infty \\ \mathbf{A} \boxplus \mathbf{x} \leq \mathbf{b}. \end{aligned} \tag{5}$$

Note that we add an additional constraint $\mathbf{A} \boxplus \mathbf{x} \leq \mathbf{b}$, also known as the "lateness" constraint. This constraint makes problem (5) more tractable; it enables the reformulation of problem (5) as a set optimization problem in (13). In many applications this constraint is desirable–see [22]. However, in other situations, it might lead to less sparse solutions or higher residual error. A possible way to overcome this constraint is explored in Section 3.1.

Even with the additional lateness constraint, problem (5) is very hard to solve. For example, when $\epsilon = 0$, solving (5) is an $\mathcal{NP}$-hard problem [22]. Thus, we do not expect to find an efficient algorithm which solves (5) exactly. Instead, we will prove next there is a polynomial time algorithm which finds an approximate solution, by leveraging its supermodular properties. First, let us show that the above problem can be formed as a discrete optimization problem over a set. We follow a similar procedure to [22], where the case $p = 1$ was examined. For the rest of this section, let $J = \{1, .., n\}$.

Lemma 1. ***(Projection on the support set, ℓ_p case)*** *Let $T \subseteq J$,*

$$X_T = \{\mathbf{x} \in \mathbb{R}^n_{max} : supp(\mathbf{x}) = T,\ \mathbf{A} \boxplus \mathbf{x} \leq \mathbf{b}\}. \tag{6}$$

and $\mathbf{x}|_T$ be defined as $\hat{\mathbf{x}}$ inside T and $-\infty$ otherwise, where $\hat{\mathbf{x}}$ is the principal solution defined in (2). Then, it holds:

- $\mathbf{x}|_T \in X_T$.
- $\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}|_T\|_p^p \leq \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}\|_p^p \ \forall\, \mathbf{x} \in X_T$.

Proof.

- It suffices to show that $\mathbf{A} \boxplus \mathbf{x}|_T \leq \mathbf{b}$. For $j \in T$ it is $[\mathbf{x}|_T]_j = \hat{x}_j$ and for $j \in J \setminus T, [\mathbf{x}|_T]_j = -\infty \leq \hat{x}_j$. Thus,

$$\mathbf{x}|_T \leq \hat{\mathbf{x}} \iff \mathbf{A} \boxplus \mathbf{x}|_T \leq \mathbf{A} \boxplus \hat{\mathbf{x}} \implies \mathbf{A} \boxplus \mathbf{x}|_T \leq \mathbf{b}. \tag{7}$$

  Hence, $\mathbf{x}|_T \in X_T$.
- Let $\mathbf{x} \in X_T$, then $\mathbf{A} \boxplus \mathbf{x} \leq \mathbf{b} \iff \mathbf{x} \leq \hat{\mathbf{x}}$, which implies (since both $\mathbf{x}, \mathbf{x}|_T$ have $-\infty$ values outside of T):

$$\mathbf{x} \leq \mathbf{x}|_T \iff \mathbf{b} - \mathbf{A} \boxplus \mathbf{x}|_T \leq \mathbf{b} - \mathbf{A} \boxplus \mathbf{x}. \tag{8}$$

  Hence:

$$\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}|_T\|_p^p = \sum_{j \in T} (\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}|_T)_j^p \leq \sum_{j \in T} (\mathbf{b} - \mathbf{A} \boxplus \mathbf{x})_j^p = \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}\|_p^p. \tag{9}$$

□

The previous lemma informs us that we can fix the finite values of a solution of Problem (5) to be equal to those of the principal solution $\hat{\mathbf{x}}$. Indeed,

Proposition 2. *Let $\mathbf{x}_{OPT}$ be an optimal solution of (5), then we can construct a new one with values inside the support set equal to those of the principal solution $\hat{\mathbf{x}}$.*

Proof. Define

$$\mathbf{z} = \begin{cases} \hat{x}_j, & j \in \text{supp}(\mathbf{x}_{\text{OPT}}) \\ -\infty, & \text{otherwise} \end{cases}, \tag{10}$$

then $\text{supp}(\mathbf{x}_{\text{OPT}}) = \text{supp}(\mathbf{z})$ and, from Lemma 1, $\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{z}\|_p^p \leq \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}_{\text{OPT}}\|_p^p$ and $\mathbf{A} \boxplus \mathbf{z} \leq \mathbf{b}$. Thus, $\mathbf{z}$ is also an optimal solution of (5). □

Therefore, the only variable that matters in Problem (5) is the support set. To further clarify this, let us proceed with the following definitions:

Definition 2. *Let $T \subseteq J$ be a candidate support and let $\mathbf{A}_j$ denote the j-th column of $\mathbf{A}$. The* error vector $\mathbf{e} : 2^J \to \mathbb{R}^m$ *is defined as:*

$$\mathbf{e}(T) = \begin{cases} \mathbf{b} - \bigvee_{j\in T}(\mathbf{A}_j + \hat{x}_j), & T \neq \emptyset \\ \bigvee_{j\in J} \mathbf{e}(\{j\}), & T = \emptyset. \end{cases} \tag{11}$$

Observe that for any T, it holds $\bigvee_{j\in T}(\mathbf{A}_j + \hat{x}_j) \leq \bigvee_{j\in J}(\mathbf{A}_j + \hat{x}_j) \leq \mathbf{b}$, which means that the above vector $\mathbf{e}(T) = (e_1(T), e_2(T), .., e_m(T))^\intercal$ is always non-negative. We also define the corresponding error function $E_p : 2^J \to \mathbb{R}$ as:

$$E_p(T) = \|\mathbf{e}(T)\|_p^p = \sum_{i=1}^{m}(e_i(T))^p. \tag{12}$$

Problem (5) can now be written as:

$$\begin{aligned} &\arg\min_{T\subseteq J} |T| \\ &\quad \text{s.t. } E_p(T) \leq \epsilon \end{aligned} \tag{13}$$

The main results of this section are based on the following properties of E_p.

Theorem 1. *Error function E_p is decreasing and supermodular.*

Proof. Regarding the monotonicity, let $\emptyset \neq C \subseteq B \subset J$, then

$$\bigvee_{j\in C}(\mathbf{A}_j + \hat{x}_j) \leq \bigvee_{j\in B}(\mathbf{A}_j + \hat{x}_j) \iff \mathbf{e}(B) \leq \mathbf{e}(C), \tag{14}$$

thus raising the, non-negative, components of the two vectors to the p-th power and adding the inequalities together yields $E_p(B) \leq E_p(C)$. The case for $C = \emptyset$ easily follows from the definition of $\mathbf{e}$.

Let $S, L \subseteq U \subseteq J$, with $|S| \leq K$, $S \cap L = \emptyset$ and define $f(U) = -E_p(U), \forall\, U$. Then:

$$\gamma_{U,K}(f) = \min_{L,S} \frac{\sum_{s_k\in S} f(L \cup \{s_k\}) - f(L)}{f(L \cup S) - f(L)}, \tag{15}$$

where $f(L) = \sum_{i=1}^{m}[b_i - \bigvee_{j\in L}(A_{ij} + \hat{x}_j)]^p$. Let now I_1 be the set:

$$I_1 = \{i \mid \bigvee_{j\in L\cup S}(A_{ij} + \hat{x}_j) = \bigvee_{j\in L}(A_{ij} + \hat{x}_j)\} \tag{16}$$

and for each $s_k \in S$, we define two sets of indices:

$$I_2(s_k) = \{i \mid \bigvee_{j\in L\cup\{s_k\}}(A_{ij} + \hat{x}_j) = \bigvee_{j\in L\cup S}(A_{ij} + \hat{x}_j) > \bigvee_{j\in L}(A_{ij} + \hat{x}_j)\} \tag{17}$$

and:

$$I_3(s_k) = \{i \mid \bigvee_{j \in L \cup S} (A_{ij} + \hat{x}_j) > \bigvee_{j \in L \cup \{s_k\}} (A_{ij} + \hat{x}_j) > \bigvee_{j \in L} (A_{ij} + \hat{x}_j)\}. \tag{18}$$

Then, if

$$\Sigma_1(L,S) = \sum_{s_k \in S} \sum_{i \in I_1, I_2(s_k)} \{-[b_i - \bigvee_{j \in L \cup \{s_k\}} (A_{ij} + \hat{x}_j)]^p + [b_i - \bigvee_{j \in L} (A_{ij} + \hat{x}_j)]^p\} \tag{19}$$

and

$$\Sigma_2(L,S) = \sum_{s_k \in S} \sum_{i \in I_3(s_k)} - [b_i - \bigvee_{j \in L \cup \{s_k\}} (A_{ij} + \hat{x}_j)]^p + [b_i - \bigvee_{j \in L} (A_{ij} + \hat{x}_j)]^p, \tag{20}$$

the ratio becomes:

$$\gamma_{U,K}(f) = \min_{L,S} \frac{\Sigma_1(L,S) + \Sigma_2(L,S)}{\Sigma_1(L,S)} \geq 1, \ \forall \ U, K \tag{21}$$

meaning (Proposition 1) that f is submodular or, equivalently, $E_p = -f$ is supermodular. □

Algorithm 1: Approximate solution of problem (5)

Input: $\mathbf{A}, \mathbf{b}$
Compute $\hat{\mathbf{x}} = (-\mathbf{A})^\intercal \boxplus' \mathbf{b}$
if $E_p(J) > \epsilon$ **then**
  return Infeasible
Set $T_0 = \emptyset, k = 0$
while $E_p(T_k) > \epsilon$ **do**
  $j = \arg\min_{s \in J \setminus T_k} E_p(T_k \cup \{s\})$
  $T_{k+1} = T_k \cup \{j\}$
  $k = k + 1$
end
$x_j = \hat{x}_j, j \in T_k$ and $x_j = -\infty$, otherwise
return $\mathbf{x}, T_k$

Setting $\tilde{E}_p(T) = \max(E_p(T), \epsilon)$ [1] and leveraging the previous theorem, we are able to formulate problem (13), and thus the initial one (5), as a cardinality minimization problem subject to a supermodular equality constraint [23], which allows us to approximately solve it by the greedy Algorithm 1. The calculation of the principal solution requires $\mathcal{O}(nm)$ time and the greedy selection of the

[1] The new, truncated, error function remains supermodular; see [15].

support set of the solution costs $\mathcal{O}(n^2)$ time. We call the solutions of problem (5) *Sparse Greatest Lower Estimates* of $\mathbf{b}$. Regarding the approximation ratio between the optimal solution and the output of Algorithm 1, the following proposition holds.

Proposition 3. *Let* $\mathbf{x}$ *be the output of Algorithm 1 after* $k > 0$ *iterations of the inner while loop and* T_k *the respective support set. Then, if* T^* *is the support set of the optimal solution of (5), then the following inequality holds:*

$$\frac{|T_k|}{|T^*|} \leq 1 + \log\left(\frac{m\Delta^p - \epsilon}{E_p(T_{k-1}) - \epsilon}\right), \tag{22}$$

where $\Delta = \bigvee_{i,j}(b_i - A_{ij} - \hat{x}_j)$.

Proof. From [23], the following bound holds for the cardinality minimization problem subject to a supermodular and decreasing constraint, defined as function $f : 2^J \to \mathbb{R}$, by the greedy algorithm:

$$\frac{|T_k|}{|T^*|} \leq 1 + \log\left(\frac{f(\emptyset) - f(J)}{f(T_{k-1}) - f(J)}\right) \tag{23}$$

For our problem, it is $f = \tilde{E}_p$. Observe now that, since $k > 0$, $\tilde{E}_p(\emptyset) = E_p(\emptyset) \leq m\Delta^p$, $0 \leq \tilde{E}_p(J) = \epsilon$ and $\tilde{E}_p(T_{k-1}) > \epsilon$. Therefore, the result follows. □

The ratio warn us to expect less optimal and, thus, less sparse vectors when increasing the norm p that we use to measure the approximation. It also hints towards an inapproximability result when $p \to \infty$, which is formalised next.

3.1 Sparse vectors with minimum ℓ_∞ errors

Although in some settings the $\mathbf{A} \boxplus \mathbf{x} \leq \mathbf{b}$ constraint is needed [22], in other cases it could disqualify potentially sparsest vectors from consideration. Omitting the constraint, on the other hand, makes it unclear how to search for minimum error solutions for any ℓ_p ($p < \infty$) norm. For instance, it has recently been reported that it is $\mathcal{NP}$-hard to determine if a given point is a local minimum for the ℓ_2 norm [14]. For that reason, we shift our attention to the case of $p = \infty$. It is well known [7,4] that problem $\min_{\mathbf{x} \in \mathbb{R}^n_{\max}} \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}\|_\infty$ has a closed form solution; it can be calculated in $\mathcal{O}(nm)$ time by adding to the principal solution element-wise the half of its ℓ_∞ error. Note that this new vector does not necessarily satisfy $\mathbf{A} \boxplus \mathbf{x} \leq \mathbf{b}$, so it shows a way to overcome the aforementioned limitation.

First, let us demonstrate that problem (5), when considering the ℓ_∞ norm, becomes harder than before and non-approximable by the greedy Algorithm 1. Hence, consider now the following optimization problem:

$$\begin{aligned} \arg \min_{\mathbf{x} \in \mathbb{R}^n_{\max}} \ & |\mathrm{supp}(\mathbf{x})| \\ \text{s.t. } & \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}\|_\infty \leq \epsilon. \end{aligned} \tag{24}$$

Thanks to a similar construction as in the previous section, this problem can be recast as a set-search problem.

Lemma 2. ***(Projection on the support set, ℓ_∞ case)*** *Let $T \subseteq J$, $\mathbf{x}|_T$ defined as $\hat{\mathbf{x}}$ inside T and $-\infty$ otherwise and $\mathbf{x}^* = \mathbf{x}|_T + \frac{\|\mathbf{b}-\mathbf{A}\boxplus\mathbf{x}|_T\|_\infty}{2}$. Then $\forall\, \mathbf{z} \in \mathbb{R}^n_{max}$ with $supp(\mathbf{z}) = T$, it holds:*

$$\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{z}\|_\infty \geq \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}^*\|_\infty = \frac{\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}|_T\|_\infty}{2}. \tag{25}$$

Proof. (Sketch) By fixing the support set of the considered vectors equal to T, equivalently we omit the columns and indices of $\mathbf{A}$ and $\mathbf{x}$, respectively, that do not belong in T (since they will not be considered at the evaluation of the maximum). By doing so, we get a new equation with same vector $\mathbf{b}$ and restricted $\mathbf{A}, \mathbf{x}$. The vector $\mathbf{x}^*$ that minimizes the ℓ_∞ error of this equation is obtained from its principal solution plus the half of its ℓ_∞ error. But now observe that the new principal solution shares the same values with the original principal solution (follows from Lemma 1) inside T, which is exactly vector $\mathbf{x}|_T$. Extending $\mathbf{x}^*$ back to $\mathbb{R}^n_{\max}$ yields the result. □

So, a similar result to Proposition 2 holds.

Proposition 4. *Let $\mathbf{x}_{OPT}$ be an optimal solution of (24), then we can construct a new one with values inside the support set equal to those of the principal solution $\hat{\mathbf{x}}$ plus the half of its ℓ_∞ error.*

By defining $E_\infty(T) = \frac{\|\mathbf{b}-\mathbf{A}\boxplus\mathbf{x}|_T\|_\infty}{2}$, (24) becomes:

$$\begin{aligned} &\arg\min_{T\subseteq J} |T| \\ &\quad \text{s.t. } E_\infty(T) \leq \epsilon \end{aligned} \tag{26}$$

Unfortunately this problem does not admit an approximate solution by the greedy Algorithm 1 (to be precise, the modified version of Algorithm 1 when E_p becomes E_∞), as its error function, although decreasing, is not supermodular. The following example also reveals that the submodularity ratio (4) of E_∞ is 0. Therefore, it is not even approximately supermodular and a solution by Algorithm 1 can be arbitrarily bad [8].

Example 1. Let $A = \begin{pmatrix} 0 & 5 & 2 \\ 4 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix}$, then principal solution $\hat{\mathbf{x}}$ is:

$$\hat{\mathbf{x}} = \begin{pmatrix} 0 & -4 & 0 \\ -5 & -1 & -1 \\ -2 & 0 & 0 \end{pmatrix} \boxplus' \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -3 \\ -2 \\ 0 \end{pmatrix}.$$

We calculate now the error function on different sets:

- When $T = \{3\}$, then $\hat{\mathbf{x}}|_{\{3\}} = \big(-\infty, -\infty, 0\big)^\intercal$ and
  $E_\infty(\{3\}) = \frac{1}{2}\|\mathbf{b} - \bigvee_{j\in\{3\}}(\mathbf{A}_j + \hat{x}|_{\{3\},j})\|_\infty = \frac{1}{2}\| \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \|_\infty = \frac{1}{2}.$

– Likewise, when $T = \{1,3\}$, $E_\infty(\{1,3\}) = \frac{1}{2}\| \begin{pmatrix}3\\1\\0\end{pmatrix} - \begin{pmatrix}-3\\1\\-3\end{pmatrix} \bigvee \begin{pmatrix}2\\0\\0\end{pmatrix} \|_\infty = \frac{1}{2}$.

– $T = \{2,3\}$, $E_\infty(\{2,3\}) = \frac{1}{2}\| \begin{pmatrix}3\\1\\0\end{pmatrix} - \begin{pmatrix}3\\-1\\-1\end{pmatrix} \bigvee \begin{pmatrix}2\\0\\0\end{pmatrix} \|_\infty = \frac{1}{2}$.

– $T = \{1,2,3\}$, $E_\infty(\{1,2,3\}) = \frac{1}{2}\| \begin{pmatrix}3\\1\\0\end{pmatrix} - \begin{pmatrix}-3\\1\\-3\end{pmatrix} \bigvee \begin{pmatrix}3\\-1\\-1\end{pmatrix} \bigvee \begin{pmatrix}2\\0\\0\end{pmatrix} \|_\infty = 0$.

Let now $f = -E_\infty, L = \{3\}, S = \{1,2\}$, then, by (4), we have:

$$\frac{f(\{3\}\cup\{1\} - f(\{3\}) + f(\{3\}\cup\{2\}) - f(\{3\})}{f(\{3\}\cup\{1,2\}) - f(\{3\})} = \frac{-1/2 + 1/2 - 1/2 + 1/2}{0 + 1/2} = 0, \tag{27}$$

meaning that f has submodularity ratio 0 or E_∞ is not even approximately supermodular.

Although the previous discussion denies from problem (24) a greedy solution with any guarantees, we propose next a practical alternative to get a sparse enough vector. We first obtain a sparse vector $\mathbf{x}_{p,\epsilon}$ by solving problem (5). Then, we add to this vector element-wise half of its ℓ_∞ error $\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}_{p,\epsilon}\|_\infty/2$. Interestingly, this new solution minimizes the ℓ_∞ error among all vectors with the same support, as formalized in the following result.

Proposition 5. *Let $\mathbf{x}_{SMMAE} \in \mathbb{R}^n_{max}$ be defined as:*

$$\mathbf{x}_{SMMAE} = \mathbf{x}_{p,\epsilon} + \frac{\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}_{p,\epsilon}\|_\infty}{2}, \tag{28}$$

where $\mathbf{x}_{p,\epsilon}$ is a solution of problem (5) with fixed (p,ϵ). Then $\forall\ \mathbf{z} \in \mathbb{R}^n_{max}$ with $supp(\mathbf{z}) = supp(\mathbf{x}_{p,\epsilon})$, it holds

$$\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{z}\|_\infty \geq \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}_{SMMAE}\|_\infty = \frac{\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}_{p,\epsilon}\|_\infty}{2} \tag{29}$$

and, also,

$$\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}_{SMMAE}\|_\infty \leq \frac{\sqrt[p]{\epsilon}}{2}. \tag{30}$$

Proof. Observe that $\mathbf{x}_{p,\epsilon}$ is equal to the principal solution $\hat{\mathbf{x}}$ inside $\text{supp}(\mathbf{x}_{p,\epsilon})$. So the first inequality holds from Lemma 2. Regarding the second one, we have:

$$\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}_{SMMAE}\|_\infty = \frac{\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}_{p,\epsilon}\|_\infty}{2} = \frac{\bigvee_i (b_i - [\mathbf{A} \boxplus \mathbf{x}_{p,\epsilon}]_i)}{2}. \tag{31}$$

But, notice that:

$$(\bigvee_i b_i - [\mathbf{A} \boxplus \mathbf{x}_{p,\epsilon}]_i)^p = \bigvee_i (b_i - [\mathbf{A} \boxplus \mathbf{x}_{p,\epsilon}]_i)^p \leq \sum_i (b_i - [\mathbf{A} \boxplus \mathbf{x}_{p,\epsilon}]_i)^p \leq \epsilon, \tag{32}$$

so

$$\bigvee_i (b_i - [\mathbf{A} \boxplus \mathbf{x}_{p,\epsilon}]_i) \leq \sqrt[p]{\epsilon} \tag{33}$$

and the result follows from (31). Note that the bound tightens, as p increases. □

The above method provides sparse vectors that are approximate solutions of the equation with respect to the ℓ_∞ norm without the need of the lateness constraint. After computing $\mathbf{x}_{p,\epsilon}$, $\mathbf{x}_{SMMAE}$ requires $\mathcal{O}(m|\text{supp}(\mathbf{x}_{p,\epsilon})| + |\text{supp}(\mathbf{x}_{p,\epsilon})|) = \mathcal{O}((m+1)|\text{supp}(\mathbf{x}_{p,\epsilon})|)$ time. We call $\mathbf{x}_{\text{SMMAE}}$ *Sparse Minimum Max Absolute Error (SMMAE)* estimate of $\mathbf{b}$.

4 Application in neural network pruning

Recently, there has been a renewed interest in Morphological Neural Networks [20,6,24,11] which consist of neural networks with layers performing morphological operations (dilations or erosions). While they are theoretically appealing because of the success that morphology operations had in traditional computer vision tasks and the universal approximation property that these networks possess, they have also shown an ability to be pruned and produce interpretable models. Herein, we propose a way to do this systematically, by formulating the pruning problem as a system of max-plus equations.

Let a morphological network be a multi-layered network that contains layers of linear transformations followed by max-affine operations. The authors of [24] call this sequence of layers as a *Max-plus block*. If $\mathbf{x} \in \mathbb{R}^d$ represents the input and k is the output's dimension, then a simple network of 1 max-plus block (see Fig. 1) performs the following operations:

$$\mathbf{z} = \mathbf{W}\mathbf{x} \text{ and } \mathbf{y} = \mathbf{A} \boxplus \mathbf{z}, \tag{34}$$

where $\mathbf{W} \in \mathbb{R}^{n \times d}$ and $\mathbf{A} \in \mathbb{R}_{\max}^{k \times n}$. Suppose now that this network has been trained successfully, possibly with a redundant number n of neurons and we wish to maintain its accuracy while minimizing its size. For each training sample $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$, it holds $\tilde{\mathbf{y}}^{(i)} = \mathbf{A} \boxplus \mathbf{z}^{(i)}$, where $\tilde{\mathbf{y}}^{(i)}$ is the network's prediction. We keep now fixed the prediction (that we wish to maintain) and the matrix $\mathbf{A}$ and we find a sparse approximate solution of this equation with respect to vector $\mathbf{z}^{(i)}$. Observe that if a value of $\mathbf{z}$ equals $-\infty$, then equivalently we can set the corresponding column of $\mathbf{A}$ to -∞, thus pruning the whole unit. Of course, this naive technique would prune units that are important for other training samples. We propose overcoming this by finding sparse solutions for each sample, counting how many times each index $j \in \{1, \ldots, n\}$ has been found inside the support set of a solution and then keeping only the k most frequent values.

The proposed method enables one to fully prune neurons from any layer that performs a max-affine operation, without harming its performance, and produce compact, interpretable networks. We support the above analysis by providing an experiment on MNIST and FashionMNIST datasets. Both datasets are balanced and contain 10 different classes.

Example 2. We train 2 networks for each dataset, containing 1 max-plus block with 64 and 128 neurons, respectively, inside the hidden layer, for 20 epochs with Stochastic Gradient Descent optimizing the Cross Entropy Loss. After the training, we pick at random 10000 samples from the training dataset (which account to 17% of the whole training data), we perform a forward pass over the network for each one of them to obtain predictions and then run Algorithm 1 with $p = 20$ and $\epsilon = 2^{20}$, so that we acquire sparse vectors $\mathbf{z}$ (and their support sets). Then, we simply find the 10 (same as the number of classes) most frequent indices inside the support sets of the solutions, keep the units that correspond to those indices and prune the rest of them. As can be seen in Table 1, all of the pruned networks record the same test accuracy as the full models, while having 54 and 118 *less* neurons, respectively. Note that trying to train from scratch networks with $n = 10$, under the same training setting, produces significantly worse results (around 60% for both datasets).

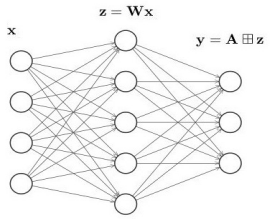

Fig. 1: A simple Max-plus block with $d = 4, n = 5, k = 3$.

	MNIST		FashionMNIST	
	64	128	64	128
Full model	92.21	92.17	79.27	83.37
Pruned ($n = 10$)	92.21	92.17	79.27	83.37

Table 1: Test set accuracy before and after pruning.

5 Conclusions and Future Work

In this work, we developed the theory of sparsest approximate solutions to max-plus equations, tackled the hardness of finding one by exploiting problem's submodular structure and provided efficient algorithms for any ℓ_p approximation error. We briefly presented then a usage of the developed algorithms in a representative area of applications, the pruning of Morphological Neural Networks. It is a subject of future work to investigate the applications of sparsity in more areas of applications, perform further experiments on the proposed pruning technique in deeper and more general networks and develop a theory of sparsity in general nonlinear vector spaces called Complete Weighted Lattices [18].

References

1. Akian, M., Gaubert, S., Guterman, A.: Tropical Polyhedra Are Equivalent To Mean Payoff Games. Int'l J. Algebra and Computation **22**(1) (2012)

2. Baccelli, F., Cohen, G., Olsder, G.J., Quadrat, J.P.: Synchronization and Linearity: An Algebra for Discrete Event Systems. J. Wiley & Sons (1992)
3. Bach, F.: Learning with submodular functions: A convex optimization perspective (2013)
4. Butkovič, P.: Max-linear Systems: Theory and Algorithms. Springer (2010)
5. Candès, E., Romberg, J., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Communications on Pure and Applied Mathematics **59**(8), 1207–1223 (2006)
6. Charisopoulos, V., Maragos, P.: Morphological Perceptrons: Geometry and Training Algorithms. In: Angulo, J., et al. (eds.) Proc. Int'l Symp. Mathematical Morphology (ISMM). LNCS, vol. 10225, pp. 3–15. Springer, Cham (2017)
7. Cuninghame-Green, R.: Minimax Algebra. Springer-Verlag (1979)
8. Das, A., Kempe, D.: Approximate submodularity and its applications: Subset selection, sparse approximation and dictionary selection. Journal of Machine Learning Research **19**(1), 74–107 (1 2018)
9. Donoho, D.: Compressed sensing. IEEE Transactions on Information Theory **52**, 1289–1306 (2006)
10. Elad, M.: Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. Springer, 1st edn. (2010)
11. Franchi, G., Fehri, A., Yao, A.: Deep morphological networks. Pattern Recognition **102**, 107246 (2020)
12. Gaubert, S., McEneaney, W., Qu, Z.: Curse of dimensionality reduction in max-plus based approximation methods: Theoretical estimates and improved pruning algorithms. In: Proc. IEEE Conf. on Decision and Control and Eur. Control Conf. (2011)
13. Heijmans, H.: Morphological Image Operators. Acad. Press, Boston (1994)
14. Hook, J.: Max-plus linear inverse problems: 2-norm regression and system identification of max-plus linear dynamical systems with gaussian noise (2019)
15. Krause, A., Golovin, D.: Submodular function maximization. In: Tractability (2014)
16. Lovász, L.: Submodular functions and convexity. Mathematical Programming The State of the Art. Springer, Berlin, Heidelberg (1983)
17. Maragos, P.: Morphological filtering for image enhancement and feature detection. The Image and Video Processing Handbook, Second Edition pp. 135–156 (2005)
18. Maragos, P.: Dynamical systems on weighted lattices: General theory. Math. Control Signals Syst. **29**(21) (2017)
19. Natarajan, B.K.: Sparse approximate solutions to linear systems. SIAM J. Comput. **24**(2), 227–234 (1995)
20. Ritter, G.X., Urcid, G.: Lattice algebra approach to single-neuron computation. IEEE Trans. Neural Netw. **14**(2), 282–295 (2003)
21. Serra, J.: Image Analysis and Mathematical Morphology. Acad. Press (1982)
22. Tsiamis, A., Maragos, P.: Sparsity in Max-plus Algebra. Discrete Events Dynamic Systems **29**, 163–189 (2019)
23. Wolsey, L.: An analysis of the greedy algorithm for the submodular set covering problem. Combinatorica **2**, 385–393 (1982)
24. Zhang, Y., Blusseau, S., Velasco-Forero, S., Bloch, I., Angulo, J.: Max-Plus Operators Applied to Filter Selection and Model Pruning in Neural Networks. In: Burgeth, B., et al. (eds.) Proc. Int'l Symp. Mathematical Morphology (ISMM). LNCS, vol. 11564, pp. 310–322. Springer Nature (2019)