Analysis of the Viterbi Algorithm Using Tropical Algebra and Geometry

Emmanouil Theodosis and Petros Maragos
School of Electrical and Computer Engineering, National Technical University of Athens, 15773 Athens, Greece
manostheodosis@gmail.com, maragos@cs.ntua.gr

***Abstract*—The Viterbi algorithm and its pruning variant, are some of the most frequently used algorithms in communications and speech recognition. There has been extended research on improving the algorithms' computational complexity, however work trying to interpret their nonlinear structure and geometry has been limited. In this work we analyse the Viterbi algorithm in the field of tropical (min-plus) algebra, and we utilize its pruning variant in order to define a polytope. Then, we interpret certain faces of the polytope as the most probable states of the algorithm. This also provides a useful geometrical interpretation of the Viterbi algorithm.**

***Index Terms*—Min-plus algebra, Viterbi, algorithm optimisation, geometry.**

I. Introduction

The field of optimisation tries to find optimal points in functions, minimize costs, and find optimal paths in graphs. A common application of the latter can be found on Weighted Finite State Transducers (WFSTs), which are mathematical structures that extend the traditional automata to support outputs and weights. WFSTs have found great use in Automatic Speech Recognition (ASR) ([1], [2], [3]) and Natural Language Processing (NLP) ([3]). An example of a WFST is given in Fig. 1. The Viterbi algorithm ([4], [5], [6]) has been instrumental in the field of digital communications and speech recognition, and its pruning variant is necessary in order to have a feasible computational complexity. Several authors have researched their computational complexity ([6], [7], [8], [9]), and many have tried to introduce new architectures, or provide optimisations in certain settings ([10], [11], [12], [13]). The authors of [14] first noted the tropical structure of WFSTs, and developed a multitude of algorithms for them ([15], [5]).

However, to the authors' best knowledge, no extensive work has been done on the geometric interpretation of the algorithm. While numerical optimisations have been studied extensively, its geometry hasn't been thoroughly explored. In this paper we examine its properties from a theoretical standpoint and try to deduct a geometric structure, in addition to advancing its algebraic understanding using tropical analysis.

Reference [6] offers various ways researchers have tried to reduce the complexity of decoding. These vary from operations unrelated to the algorithm itself (such as minimization and weight pushing on the WFST), to alternations in the algorithm (such as pruning and rescoring). In [8] the authors propose using a precompiled automaton to efficiently compute surviving states, and thus speed up the decoding process. In [9] authors utilize Viterbi, alongside heuristics, to solve

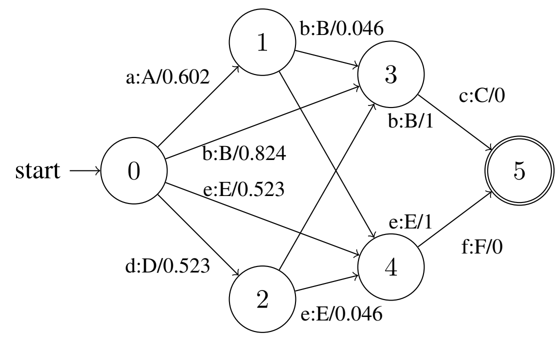

Figure 1: A simple Weighted Finite State Transducer. The initial state is denoted with an arrow with the label start, and accepting states are marked with a double circle. The transition labels are of the form "input label:output label/weight". The weights on the edges are negative logarithms of probabilities.

efficiently the problem of computing graph trajectories. In [7] the authors unveil a connection between decoding and the maximum clique algorithm. Regarding new architectures, [11] and [10] introduce more sizeable architectures that allow us to decode strings faster. In [12] the authors take advantage of recent advances in GPU processing to significantly speed up algorithms on WFSTs. Reference [13] is the result of a hackathon, which utilizes heuristics and optimisations in order to speed up decoding. Pachter and Strumfels ([16]) studied aspects of the geometry of the sum-product algorithm in HMMs. In [17] the author proposes that all Viterbi sequences have a part with a periodic structure, and divides Markov chains in "regions" of similar Viterbi paths, which he interprets as vertices of a Newton polytope. Finally, [18] is an effort to use tropical geometry in neural networks (max-plus perceptron).

In this paper, we analyse the tropical structure of the algorithm. First we model it and its pruning variant in tropical algebra, which is a min-plus algebra ([19]). Then, we analyse its geometry. We consider a relaxation of the update law that allows us to define a polytope, enabling the geometrical interpretation of the problem. We relate the vertices and the faces of the polytope with the N-best paths and the pruned vector of the algorithm. Finally, we present a numerical example to highlight the geometrical structure. This provides a useful geometrical interpretation of the Viterbi algorithm.

In Section II we present the required background and define the notation that we will use throughout this paper. Section III models the Viterbi algorithm and its pruning variant in tropical algebra, and then considers a relaxation in order to define a polytope. In Section IV we suggest that the space enclosed in the polytope can be indicative of the effectiveness of the pruning procedure, and we present a connection between the faces and the vertices of the polytope with the N-best paths. Section V contains the numerical example and visualizations.

II. Background & Notation

A. Notation

We use $\mathbb{R}$ to refer to the line of real numbers $(-\infty, +\infty)$, and let $\mathbb{R}_{\min} = \mathbb{R} \cup \{+\infty\}$. We denote:

- scalars with lowercase symbols.
- column vectors with lowercase boldfaced symbols.
- matrices with uppercase boldfaced symbols.

When referring to elements of matrices or vectors we use the non-boldfaced symbol with the corresponding subscript.

B. Tropical Algebra

Min-plus (tropical) algebra ([20], [21], [22], [19]) is an algebraic body similar to linear algebra in which the pair of operations $(+, \times)$ is replaced by the pair $(\min, +)$ of a generalized "addition" and a generalized "multiplication", corresponding to the minimum and the standard addition. We use $\wedge$ to denote the minimum. For matrix and vector "multiplication" we use the notation proposed in [23]. In particular, we use $\boxplus$ to denote the min-plus multiplication between vectors/matrices. Formally, let $\mathbf{A} \in \mathbb{R}_{\min}^{m\times n}$ and $\mathbf{B} \in \mathbb{R}_{\min}^{n\times r}$. Then:

$$(\mathbf{A} \boxplus \mathbf{B})_{ij} = \bigwedge_{k=1}^{n} A_{ik} + B_{kj}$$

(In [23] the author uses $\boxplus$ to denote the max-plus multiplication, and reserves $\boxplus'$ for the min-plus multiplaction. Since our main focus is tropical algebra, we reverse the notation.)

The neutral elements for each of the operations are:

- $+\infty$ for the the generalized "addition".
- 0 for the generalized "multiplication".

We present some example calculations for illustration:

- $$\begin{bmatrix}3\\7\end{bmatrix} \wedge \begin{bmatrix}9\\2\end{bmatrix} = \begin{bmatrix}\min(3,9)\\\min(7,2)\end{bmatrix} = \begin{bmatrix}3\\2\end{bmatrix}$$
  is an example of a generalized vector "addition".
- $$\begin{bmatrix}2 & 4\\-6 & 11\end{bmatrix} \boxplus \begin{bmatrix}7\\3\end{bmatrix} = \begin{bmatrix}\min(2+7, 4+3)\\\min(-6+7, 11+3)\end{bmatrix} = \begin{bmatrix}7\\1\end{bmatrix}$$
  is an example of a generalized "multiplication" between a matrix and a vector.

We denote the $n \times n$ identity matrix as $\mathbf{I}_n$, whose entries are $+\infty$, except the elements in the diagonal, which are 0.

C. Geometry and Polytopes

Tropical geometry ([21]) applies the ideas of polyhedra and polytopes ([24], [25]) in the tropical setting. Polyhedra are fundamental objects in geometry, defined by halfspaces.

Definition 1. *Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}_{\min}^{n+1}$. An* affine tropical halfspace *is a subset $\mathbb{R}_{\min}^{n}$ defined by:*

$$T(\mathbf{a},\mathbf{b}) := \{\mathbf{x} \in \mathbb{R}_{\min}^{n} : \quad \left(\bigwedge_{i=1}^{n} a_i + x_i\right) \wedge a_{n+1} \geq \left(\bigwedge_{i=1}^{n} b_i + x_i\right) \wedge b_{n+1}\}$$

Having defined halfspaces, we can now define polyhedra.

Definition 2. *The intersection of a finite number of affine tropical halfspaces is called a* tropical polyhedron.

Polytopes are polyhedra that are bounded.

Definition 3. *If a tropical polyhedron is bounded, then it is a* tropical polytope.

III. Tropical Viterbi

A. Analysis

In this section we will examine how Viterbi pruning can be written compactly in min-plus algebra. The Viterbi algorithm can be written as:

$$q_i(t) = \left(\max_j w_{ji} q_j(t-1)\right) \cdot b_i(\sigma_t) \tag{1}$$

where w_{ji} is the transition probability from state j to state i, $b_i(\sigma_t)$ is the probability of observing the input symbol σ_t (at the time frame t), while on state i, and $q_i(t)$ is the best score (highest probability) along a single path which, optimized over the previous $t-1$ time frames, ends in state i at time frame t and accounts for the observed symbols $\sigma_1, ..., \sigma_t$. By taking the negative logarithm of the above, we get:

$$-\log q_i(t) = \left(\min_j \left(-\log w_{ji} q_j(t-1)\right)\right) - \log b_i(\sigma_t) \tag{2}$$

Now, setting $\mathbf{x}(t) = -\log \mathbf{q}(t)$, $\mathbf{A} = -\log \mathbf{W}$, and $\mathbf{p}(\sigma_t) = -\log \mathbf{b}(\sigma_t)$ we can write this in matrix notation:

$$\mathbf{x}(t) = \mathbf{A}^T \boxplus \mathbf{x}(t-1) + \mathbf{p}(\sigma_t) \tag{3}$$

In order to be strict about using only min-plus notation, we can define the matrix $\mathbf{P}(\sigma_t)$ as follows:

$$\mathbf{P}(\sigma_t) = \begin{bmatrix} p_1(\sigma_t) & \cdots & \infty \\ \vdots & \ddots & \vdots \\ \infty & \cdots & p_n(\sigma_t) \end{bmatrix} \tag{4}$$

Then (3) can be written compactly:

$$\mathbf{x}(t) = \mathbf{P}(\sigma_t) \boxplus \mathbf{A}^T \boxplus \mathbf{x}(t-1) \tag{5}$$

In the pruning version of the algorithm, we go through each vector $\mathbf{x}(t)$ and we set values that are greater than a threshold to $+\infty$. We claim that the indices that should be pruned are indicated by the *Cuninghame-Green inverse* ([19], [23]).

Proposition 1. *Let*

$$\mathbf{X}(t) = \begin{bmatrix} x_1(t) & \infty & \cdots & \infty \\ \infty & x_2(t) & \cdots & \infty \\ \vdots & \vdots & \ddots & \vdots \\ \infty & \infty & \cdots & x_n(t) \end{bmatrix}$$

where $x_i(t)$ represents the i-th element of the vector $\mathbf{x}(t)$, and let $\boldsymbol{\eta} = \eta + \frac{1}{2}\left(\mathbf{x}(t)^T \boxplus \mathbf{x}(t)\right) + \mathbf{0}$, where $\mathbf{0}$ is a vector that comprises of 0 and η is the leniency variable. Finally, let $\boxplus'$ denote the max-plus matrix multiplication and $\mathbf{X}^{\#}(t) := -\mathbf{X}^T(t)$. Then the negative elements of

$$\overline{\mathbf{y}} = \mathbf{X}^{\#}(t) \boxplus' \boldsymbol{\eta} \tag{6}$$

indicate which indices of $\mathbf{x}(t)$ need to be pruned.

Proof. The *Cuninghame-Green inverse* $\overline{\mathbf{y}}$ provides the smallest feasible solution to the inequality $\mathbf{X}(t) \boxplus \mathbf{y} \geq \boldsymbol{\eta}$. Since $\mathbf{X}(t)$ is diagonal, then each element y_i corresponds to $x_i(t)$. The sum of each y_i and $x_i(t)$ needs to be bigger than the threshold values. However, $\overline{\mathbf{y}}$ is the smallest solution, and thus if $x_i(t)$ is already greater than the threshold, then $\overline{y}_i$ will admit a negative value, indicating that $x_i(t)$ should be pruned. □

An example is given below. Let

$$\mathbf{x} = \begin{bmatrix} 1 & 7 & 4 \end{bmatrix}^T$$

and suppose that the leniency is $\eta = 5$. Then, $\boldsymbol{\eta} = 5 + \frac{1}{2}(2 \times \min(1, 7, 4)) + \mathbf{0} = \begin{bmatrix} 6 & 6 & 6 \end{bmatrix}^T$. The optimal solution then is given by (6):

$$\overline{\mathbf{y}} = \begin{bmatrix} -1 & -\infty & -\infty \\ -\infty & -7 & -\infty \\ -\infty & -\infty & -4 \end{bmatrix} \boxplus' \begin{bmatrix} 6 \\ 6 \\ 6 \end{bmatrix} = \begin{bmatrix} 5 \\ -1 \\ 2 \end{bmatrix}$$

As $\overline{y}_2$ is negative, it gets pruned, and the resulting vector is:

$$\mathbf{x}_{\text{pruned}} = \begin{bmatrix} 1 & \infty & 4 \end{bmatrix}^T$$

B. Geometry

1) Viterbi equations: We will extract some geometric structure from the Viterbi equations. Let $\mathbf{z}$ be a vector of variables. We will bound the possible values of $\mathbf{z}$ using the update law of (5). Consider the relaxation:

$$\mathbf{z} \geq \mathbf{b}, \quad \mathbf{b} = \mathbf{P}(\sigma_t) \boxplus \mathbf{A}^T \boxplus \mathbf{x}(t-1) \tag{7}$$

By letting $\mathbf{G} = \mathbf{I}_n$, (7) can be written in the form:

$$\mathbf{G} \boxplus \mathbf{z} \geq \mathbf{b} \tag{8}$$

The reason why we choose the relaxation of $\geq$ over $\leq$ is because of its significance in the min-plus algebraic setting. The equation $\mathbf{G} \boxplus \mathbf{z} \leq \mathbf{b}$ is translated as "*The minimum element of the i-th row of the product between $\mathbf{G}$ and $\mathbf{z}$ needs to be **less** than b_i.*" However, that doesn't provide meaningful constraints on the vector $\mathbf{z}$. It affects a *single* element of the vector, and we

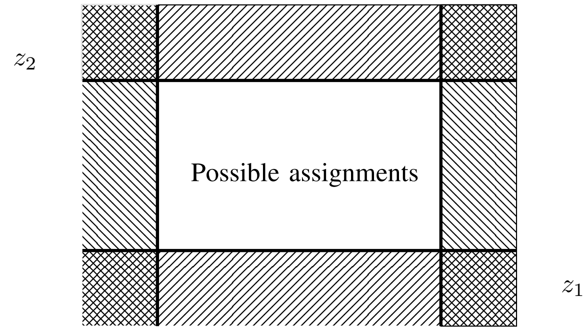

Figure 2: Possible assignments constrained by the Viterbi update law (lower and leftmost constraints) and by pruning operation (upper and rightmost constraints).

can't know apriori which one. Instead, consider the equation $\mathbf{G} \boxplus \mathbf{z} \geq \mathbf{b}$. This translates to "*The minimum element of the i-th row of the product between $\mathbf{G}$ and $\mathbf{z}$ needs to be **greater** than b_i.*" Since the minimum is greater than a value, it follows that every element in the product is greater than that value.

2) Pruning equations: Before, we treated pruning as an estimation problem, since we had no way of enforcing the Viterbi update law of (5) on $\mathbf{y}$. However, the variable vector $\mathbf{z}$ has been constrained by the relaxation of (7). Thus, we can see pruning as another set of constraints on the vector $\mathbf{z}$:

$$\mathbf{z} \leq \boldsymbol{\eta}, \quad \boldsymbol{\eta} = \eta + \frac{1}{2}\left(\mathbf{b}^T \boxplus \mathbf{b}\right) + \mathbf{0} \tag{9}$$

where $\mathbf{b}$ was defined in (7). Following the same trick we did in (8), let $\mathbf{H} = \mathbf{I}_n$. Then (9) can be written in the form:

$$\mathbf{H} \boxplus \mathbf{z} \leq \boldsymbol{\eta} \tag{10}$$

The combination of (8) and (10) defines a polytope. Indeed:

$$T(\mathbf{G}^b, \mathbf{H}^\eta) = \{\mathbf{z} \in \mathbb{R}^n_{\min} : \mathbf{G}^b \boxplus \mathbf{z} \geq \mathbf{0}, \mathbf{H}^\eta \boxplus \mathbf{z} \leq \mathbf{0}\} \tag{11}$$

where $\mathbf{G}^b$ is identical to $\mathbf{G}$, except that every row i has been decreased by b_i, and $\mathbf{H}^\eta$ is identical to $\mathbf{H}$, except that every row i has been decreased by η_i.

The space enclosed inside the polytope contains all the assignments of $\mathbf{z}$ that satisfy the constraints. An example of the constrained space is visualized in Fig. 2.

IV. Geometry of the Viterbi

In this section we examine how these geometric properties can be interpreted, and how the polytope and its structure can give us information about the efficiency of the pruning.

We propose that the "volume" (referring to d-dimensional volume, for example a segment in 1-D, area in 2-D, conventional volume in 3-D, etc.) enclosed in the interior of the polytope can provide a measure of the efficiency of the pruning procedure. However, the volume scales with the amount of participating nodes. Thus, it would be wise to incorporate into the metrics the support of the pruned vector $\mathbf{z}$.

Definition 4. *The* support *of a vector $\mathbf{x}$, denoted by $\operatorname{supp}(\mathbf{x})$ is the set of the indices corresponding to finite entries in $\mathbf{x}$.*

Let $r_i = (\min(\mathbf{z}) + \eta) - z_i$. Then $r_i \in [0, \eta]$, and indicates to what capacity each index satisfies the constraints of (8) and (10). Then:

- Let ν be the negative logarithm of the normalized volume, divided by the support of the pruned vector:

$$\begin{aligned}\nu &= -\frac{1}{\mathrm{supp}(\mathbf{z})} \log \left(\prod_{i \in \mathrm{supp}(\mathbf{z})} \frac{r_i}{\max \mathbf{r}} \right) \\ &= -\frac{1}{\mathrm{supp}(\mathbf{z})} \sum_{i \in \mathrm{supp}(\mathbf{z})} \frac{\log r_i}{\log (\max \mathbf{r})}\end{aligned}$$

- Let ε be:

$$\varepsilon = -\frac{1}{\mathrm{supp}(\mathbf{z})} \sum_{i \in \mathrm{supp}(\mathbf{z})} -z_i(t) \cdot e^{-z_i(t)}$$

  In essence, this metric is the entropy of the probability vector $\mathbf{q}(t)$. Remember that $\mathbf{x} = -\log \mathbf{q}$, so:

$$-\sum_{i \in \mathrm{supp}(\mathbf{z})} -z_i(t) \cdot e^{-z_i(t)} = -\sum_{i \in \mathrm{supp}(\mathbf{z})} q_i(t) \cdot \log q_i(t)$$

These metrics provide insights about the efficiency of pruning and can help designers choose values for η. Both ν and ε can be used to quantify the tradeoff between computational complexity and the probability of finding the best path. Minimizing the derivative of ε, while on the same time maximizing ν can aid in choosing the leniency values.

Of interest is how the support of $\mathbf{x}$ is related to the support of $\mathbf{z}$. In particular:

- depending on the size of $\mathrm{supp}(\mathbf{z})$ w.r.t. $\mathrm{supp}(\mathbf{x})$, we can dynamically adjust the leniency variable η between frames.
- we can examine the set V of states for which it is true that:

$$V = \{i : i \in \mathrm{supp}(\mathbf{x}), i \notin \mathrm{supp}(\mathbf{z})\}$$

  It is possible to find similarities in those states (and their acceptable paths), and thus simplify the WFSTs.

A final thing to consider is how the vertices of the polytope are connected with the N-best paths that pruning algorithms usually produce. In fact, certain $n-1$-faces of the polytope appear in the N-best paths. These $n-1$-faces are produced by the lower constraint for each dimension, and thus act on the space produced by the other dimensions. The vertex where these hyperplanes intersect is the vector that is returned by the algorithm, containing all the best paths.

V. Numerical Example and Experimentation

Let us present a numerical example. Consider again the WFST of Fig. 1. The transition matrix implied by the edges' weights is:

$$\mathbf{A} = \begin{bmatrix} \infty & 0.602 & 0.523 & 0.824 & 0.523 & \infty \\ \infty & \infty & \infty & 0.046 & 1 & \infty \\ \infty & \infty & \infty & 1 & 0.046 & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

We assume that each state has a higher probability for its entry symbol (the one indicated in Fig. 1) than for the rest of the symbols. Let the probability for the entry symbol be 0.30, and 0.175 for the rest. Thus, state 1 has a high probability to observe a, state 2 a high probability to observe b, and so on. By taking the negative logarithm of these probabilities we construct the matrix $\mathbf{P}(\sigma_t)$ for each time frame. Suppose we observe the symbols a, b, and c, in the first three time frames. The observation matrix for the first time frame is:

$$\mathbf{P}(a) = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 0.523 & \infty & \infty & \infty & \infty \\ \infty & \infty & 0.757 & \infty & \infty & \infty \\ \infty & \infty & \infty & 0.757 & \infty & \infty \\ \infty & \infty & \infty & \infty & 0.757 & \infty \\ \infty & \infty & \infty & \infty & \infty & 0.757 \end{bmatrix}$$

Finally, since state 0 can be the only starting state, let:

$$\mathbf{x}(0) = \begin{bmatrix} 0 & \infty & \infty & \infty & \infty & \infty \end{bmatrix}^T$$

Using (5), the three vectors outputed by the algorithm are:

$$\begin{aligned}\mathbf{x}(1) &= \begin{bmatrix} \infty & 1.125 & 1.28 & 1.581 & 1.28 & \infty \end{bmatrix}^T \\ \mathbf{x}(2) &= \begin{bmatrix} \infty & \infty & \infty & 1.694 & 2.083 & 2.037 \end{bmatrix}^T \\ \mathbf{x}(3) &= \begin{bmatrix} \infty & \infty & \infty & \infty & \infty & 2.217 \end{bmatrix}^T\end{aligned}$$

Let the leniency be $\eta = 0.347$ (the negative logarithm of 0.45). Using (9), the threshold vectors comprise of the elements 1.472 and 2.041 respectively. The polytopes defined by the combination of the constraints of the Viterbi update law and the constraints of the the pruning variant for the vectors $\mathbf{x}(1)$ and $\mathbf{x}(2)$ can be seen in Fig. 3.

Let's see a more substantial experiment. We used data from an NLP transliteration task; in particular, we used a WFST that tranduces Greek text written using Latin characters to the most likely Greek "translation". The experiments were performed on a laptop with a 2.5GHz Intel i7 processor. Table 1 notes the execution time, the average values of ν and ε, and the minimum and maximum number of surviving nodes across time frames for different leniency values. The derivative of ε for both the presented examples is minimized for values of η around 10 (while also maximizing ν). For such leniency values, at most 30% of the total states survive pruning at each time frame.

VI. Conclusion

In this work we analysed the Viterbi algorithm using min-plus algebra and geometry. First, we formulated the algorithm in the context of tropical algebra. Then, we modeled pruning as a problem of optimally solving min-plus inequalities in tropical algebra. Using the constraints from the update law and the pruning variant, we defined a tropical polytope. We presented its $n-1$-faces as hyperplanes of the N-best paths. Finally, we provided a numerical and visual example to highlight the geometrical structure. Our ongoing research seeks the application of these ideas in NLP and machine learning.

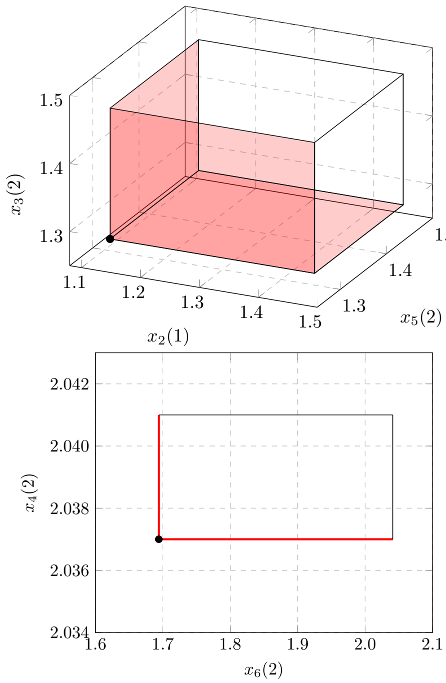

Figure 3: Resulting polytopes of the numerical example. The highlighted edges are the $n-1$-faces of the N-best paths, and the black dot indicates the vector outputed by the algorithm. Top (3D, $x_2(1), x_3(1), x_5(1)$). Bottom (2D, $x_4(2), x_6(2)$).

Transliteration from latin to greek characters						
input	η	time (s)	ε	ν	min	max
\ELLIPEIS\	0	89.5	0.0248	0	1	1
(Latin text	5	121.7	0.0018	1.558	1	1444
for the	10	201.9	0.0013	2.094	101	3829
Greek word	15	533.0	0.0001	1.630	5145	10333
ΕΛΛΙΠΕΙΣ)	∞	580.3	0.0001	0	10333	10333
\ALLA\	0	77.6	0.0616	0	1	1
(Latin text	5	93.3	0.0039	1.435	1	1215
for the	10	175.2	0.0026	2.072	153	5431
Greek word	15	481.8	0.0003	1.765	7088	14246
ΑΛΛΑ)	∞	562.9	0.0002	0	14246	14246

Table 1: Execution time, ν and ε values, and minmum and maximum number of surviving states across time frames for different leniency values η for two different input words.

References

[1] L. R. Rabiner and R. W. Schafer, *Theory and Applications of Digital Speech Processing*. Pearson, 2010.

[2] L. R. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. Prentice Hall, 1993.

[3] D. Jurafsky and J. Martin, *Speech and Language Processing*. Prentice Hall, 2009.

[4] G. D. Forney, "The Viterbi algorithm", *Proceedings of the IEEE*, vol. 61, pp. 268–278, 1973.

[5] M. Mohri, "Weighted Automata Algorithms", in *Handbook of Weighted Automata*, Springer, 2009, pp. 213–254.

[6] T. Hori and A. Nakamura, *Speech Recognition Algorithms Using Weighted Finite-State Transducers*. Morgan & Claypool, 2013.

[7] A. Backurs and C. Tzamos, "Improving Viterbi is hard: Better runtimes imply faster clique algorithms", *Proc. Intl. Conf. on Mach. Learning*, 2017.

[8] R. W. Luk and R. I. Damper, "Computational complexity of a fast viterbi decoding algorithm for stochastic letter-phoneme transduction", *IEEE Trans. on Speech and Audio Processing*, vol. 6, pp. 217–225, 1998.

[9] Y. Yang, S. Chen, M. A. Maddah-Ali, P. Grover, S. Kar, and J. Kovacevic, "Fast path localization on graphs via multiscale Viterbi decoding", *Proc. ICASSP*, 2017.

[10] I. Lee and J. L. Sonntag, "A new architecture for the fast viterbi algorithm", *IEEE Transactions on Communications*, vol. 51, pp. 1624–1628, 2003.

[11] J. Feldman, I. Abou-Faycal, and M. Frigo, "A fast maximum-likelihood decoder for convolutional codes", *Proc. 56th Vehicular Technology Conference*, 2002.

[12] A. Argueta and D. Chiang, "Decoding with Finite-State Transducers on GPUs", *Proc. Conf. European Chapter of the ACL*, 2017.

[13] H. Wei, Y. Wang, G. Forman, Y. Zhu, and H. Guan, "Fast Viterbi map matching with tunable weight functions", *Proc. Intl. Conf. on Advances in Geographic Information Systems*, 2012.

[14] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition", *Computer Speech & Language*, vol. 16, pp. 69–88, 2002.

[15] M. Mohri, "Minimization algorithms for sequential transducers", *Th. Comp. Sci.*, vol. 234(177–201), 2000.

[16] L. Pachter and B. Sturmfels, "Tropical geometry of statistical models", *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 46, pp. 16 132–16 137, 2004.

[17] E. H. Kuo, "Viterbi sequences and polytopes", *Journal of Symbolic Computation*, vol. 41, pp. 151–163, 2006.

[18] V. Charisopoulos and P. Maragos, "Morphological Perceptrons: Geometry and Training Algorithms", in *Mathematical Morphology and Its Applications to Signal and Image Processing (ISMM)*, Springer, 2017.

[19] R. Cuninghame-Green, *Minimax Algebra*, ser. Lecture Notes in Economics and Mathematical Systems. Springer, 1979, vol. 166.

[20] M. Gondran and M. Minoux, *Graphs, Dioids and Semirings*. Springer, 2008, vol. 41.

[21] D. Maclagan and B. Sturmfels, *Introduction to Tropical Geometry*. American Mathematical Society, 2015.

[22] P. Butkovič, *Max-linear Systems: Theory and Algorithms*. Springer, 2010.

[23] P. Maragos, "Dynamical systems on weighted lattices: General theory", *Mathematics of Control, Signals, and Systems*, vol. 29:21, 2017.

[24] G. M. Ziegler, *Lectures on Polytopes*. Springer, 2012.

[25] S. Gaubert and R. D. Katz, "Minimal half-spaces and external representation of tropical polyhedra", *J. Algebraic Combinatorics*, vol. 33, pp. 325–348, 2011.