Analysis of the Viterbi Algorithm Using Tropical Algebra and Geometry

Emmanouil Theodosis and Petros Maragos

ICCS, School of ECE, National Technical University of Athens, Greece

SPAW 2018

For more information: cvsp.cs.ntua.gr

Introduction

Motivation

- **Tropical geometry** [4] is an emerging and interesting field.
- Geometrical analysis of algorithms allows for **intuition**.
- Pruning naturally defines **polytopes** which enables geometrical analysis.

Contributions

- Analysing Viterbi and pruning in **tropical algebra**.
- Pruning occurs from the **Cuninghame-Green inverse**.
- Utilising objects of **tropical geometry** to better understand pruning.
- **Metrics** on polytopes.

Background

Tropical Algebra

- Similar to **linear algebra**, but the pair $(+, \times)$ is replaced by $(\wedge, +)$ (where $\wedge = \min$).
- Matrix/vector multiplication [6] (elements from $\mathbb{R}_{\min} = (-\infty, \infty]$):

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

- Neutral elements are ∞ for the minimum and 0 for the addition.
- Example:

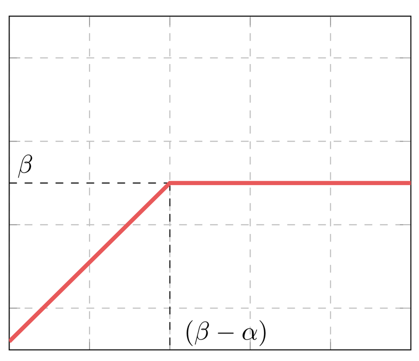
$$\begin{bmatrix} 2 & 4 \\ -6 & 11 \end{bmatrix} \boxplus \begin{bmatrix} 7 \\ 3 \end{bmatrix} = \begin{bmatrix} \min(2+7, 4+3) \\ \min(-6+7, 11+3) \end{bmatrix} = \begin{bmatrix} 7 \\ 1 \end{bmatrix}$$

Tropical Geometry

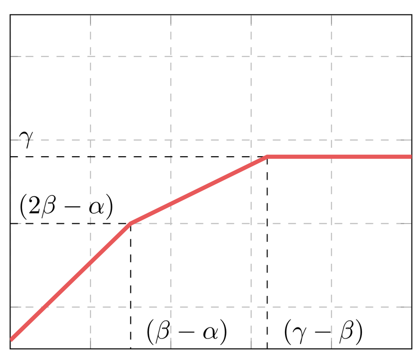
Definition 1: *Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}_{\min}^{n+1}$. An* **affine tropical half-space** *is a subset of $\mathbb{R}_{\min}^{n}$ defined by:*

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

- **Tropical polyhedra** are intersections of affine tropical half-spaces.
- **Tropical polytopes** are bounded tropical polyhedra.



$y = \alpha + x \wedge \beta = \min(\alpha + x, \beta)$



$y = \alpha + 2x \wedge \beta + x \wedge \gamma = \min(\alpha + 2x, \beta + x, \gamma)$

Tropical Viterbi

- Viterbi algorithm [3]:

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

- Negative logarithm of (1) and $\mathbf{x}(t)$ = -log $\mathbf{q}(t)$, $\mathbf{A}$ = -log $\mathbf{W}$, $\mathbf{p}(\sigma_t)$ = -log $\mathbf{b}(\sigma_t)$:

$$\mathbf{x}(t) = \mathbf{A}^T \boxplus \mathbf{x}(t-1) + \mathbf{p}(\sigma_t) \tag{2}$$

- Define $\mathbf{P}(\sigma_t)$:

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

- Viterbi in **tropical algebra**:

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

- **Pruning**: go through $\mathbf{x}(t)$ and set values greater than a threshold to $+\infty$.
- Indices that should be pruned ⟶ **Cuninghame-Green inverse**

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 $\boldsymbol{\eta} = \theta + \frac{1}{2}\left(\mathbf{x}(t)^T \boxplus \mathbf{x}(t)\right) + \mathbf{0}$, where $\mathbf{0}$ is a vector that comprises of 0 and θ 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

$$\bar{\mathbf{y}} = \mathbf{X}^\#(t) \boxplus' \boldsymbol{\eta} \tag{4}$$

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

Geometry of the Viterbi

- Variable vector $\mathbf{z}$
- **Bind** $\mathbf{z}$:
  - from below:

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

  - from above:

$$\mathbf{z} \leq \boldsymbol{\eta}, \quad \boldsymbol{\eta} = \theta + \frac{1}{2}\left(\mathbf{b}^T \boxplus \mathbf{b}\right) + \mathbf{0} \tag{6}$$

- (5) + (6) ⟶ **polytope**
- $(n-1)$-faces ⟶ **best paths**

Definition 2: *The* **support** *of a vector $\mathbf{x}$, denoted by $\mathrm{supp}(\mathbf{x})$ is the set of the indices corresponding to finite entries in $\mathbf{x}$.*

- $r_i = (\min(\mathbf{z}) + \eta) - z_i$
- **Metrics**:

$$\nu = -\frac{1}{\mathrm{supp}(\mathbf{z})} \sum_{i \in \mathrm{supp}(\mathbf{z})} \frac{\log r_i}{\log(\max \mathbf{r})}, \quad \varepsilon = -\frac{1}{\mathrm{supp}(\mathbf{z})} \sum_{i \in \mathrm{supp}(\mathbf{z})} -z_i(t) \cdot e^{-z_i(t)}$$

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

- ν is based on **volume**.
- ε is based on **entropy**.
- Tradeoff between **complexity** and **accuracy**.

Example and Experimentation

Numerical Example for Weighted Finite State Transducers

- **Transition** matrix $\mathbf{A}$, **observation** matrix $\mathbf{P}(a)$:

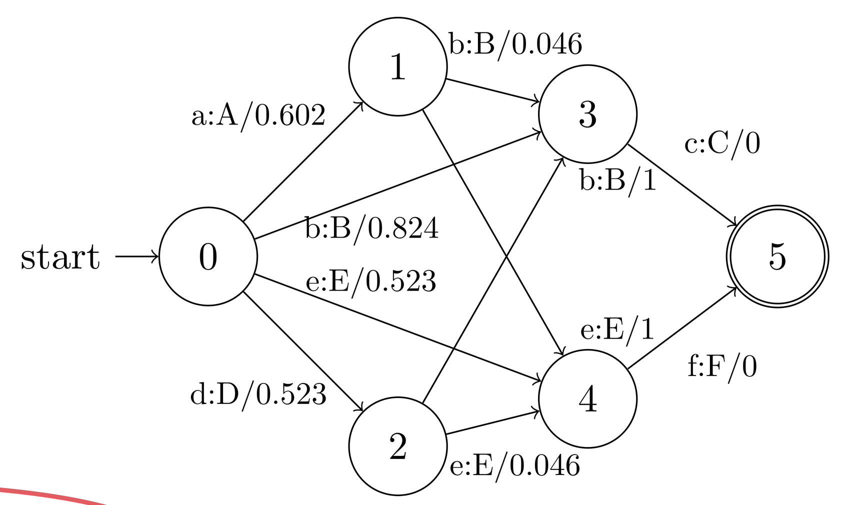
$$\mathbf{A} = \begin{bmatrix} \infty & 0.602 & 0.523 & 0.824 & 0.523 & \infty \\ \infty & \infty & \infty & 0.046 & 1 & \infty \\ \infty & \infty & \infty & 1 & 0.046 & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}, \mathbf{P}(a) = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 0.523 & \infty & \infty & \infty & \infty \\ \infty & \infty & 0.757 & \infty & \infty & \infty \\ \infty & \infty & \infty & 0.757 & \infty & \infty \\ \infty & \infty & \infty & \infty & 0.757 & \infty \\ \infty & \infty & \infty & \infty & \infty & 0.757 \end{bmatrix}$$

- **Starting** state $\mathbf{x}(0)$:

$$\mathbf{x}(0) = [0 \quad \infty \quad \infty \quad \infty \quad \infty \quad \infty]^T$$

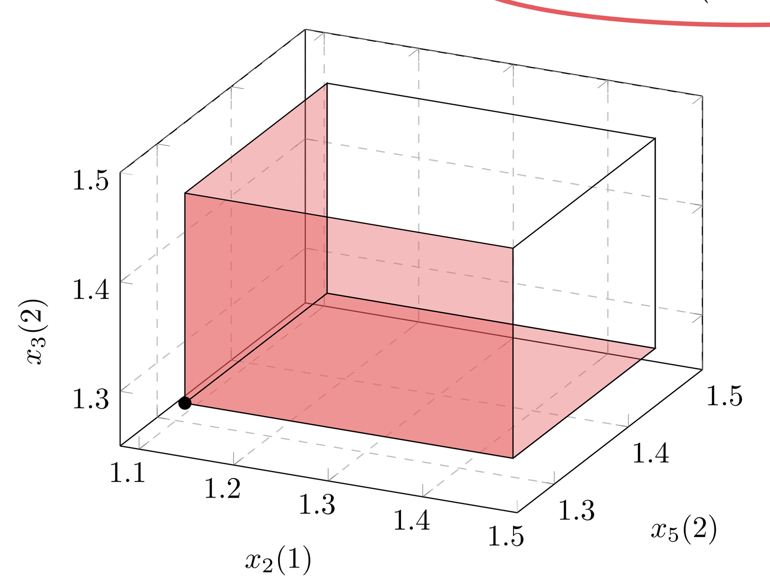
- **Outputs**:

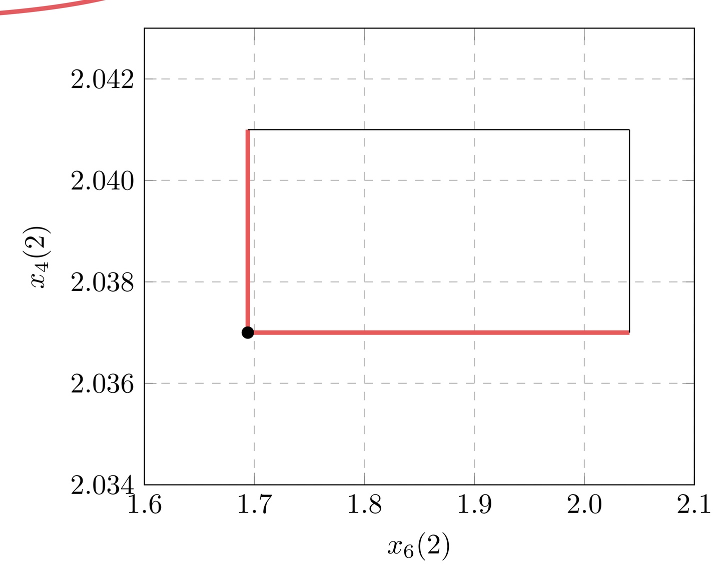
$$\mathbf{x}(1) = [\infty \quad 1.125 \quad 1.28 \quad 1.581 \quad 1.28 \quad \infty]^T$$
$$\mathbf{x}(2) = [\infty \quad \infty \quad \infty \quad 1.694 \quad 2.083 \quad 2.037]^T$$
$$\mathbf{x}(3) = [\infty \quad \infty \quad \infty \quad \infty \quad \infty \quad 1.842]^T$$



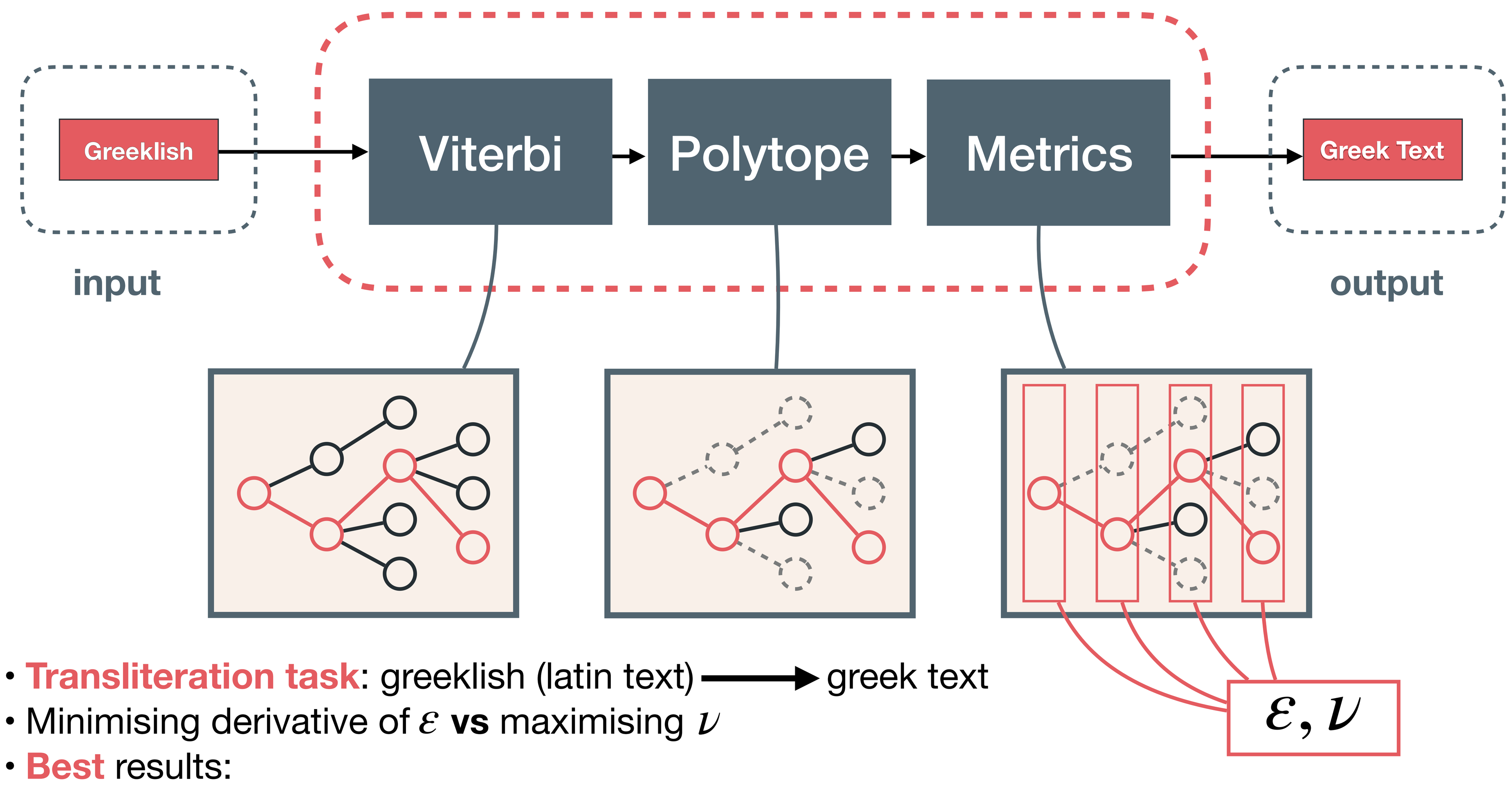
- **Polytopes** for $\mathbf{x}(1)$ and $\mathbf{x}(2)$:

$\theta = 0.35 \; (= -\log 0.45)$





NLP Experiment & Application



- **Transliteration task**: greeklish (latin text) ⟶ greek text
- Minimising derivative of ε **vs** maximising ν
- **Best** results:

$\theta = 10$

- $<$ 30% states **survive**

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

References

[1] P. Butkovic, "Max-linear Systems: Theory and Algorithms", Springer, 2010.

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

[3] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proc. of the IEEE*, vol. 72, 1989.

[4] D. Maclagan and B. Strumfels, "Introduction to Tropical Geometry", American Mathematical Society, 2015.

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

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