A Robust Adaptive Pruning Algorithm Based on Tropical Geometry

Emmanouil Theodosis and Petros Maragos
School of Electrical and Computer Engineering
National Technical University of Athens
Athens, 15780, Greece
manostheodosis@mail.ntua.gr, maragos@cs.ntua.gr

***Abstract*—Decoding problems have always been extremely important in telecommunications and speech recognition. As data becomes more abundant, efficient and robust pruning variants of classic decoding algorithms become necessary in order to achieve real time decoding. However, fixed beam decoding algorithms fail to achieve both efficiency and robustness and have to sacrifice one in favor of the other. In this work we propose a new adaptive pruning algorithm inspired by the tropical geometrical analysis of the traditional Viterbi pruning algorithm to solve decoding problems. In particular, the proposed algorithm tries to adapt the beam width based on the internal volume and the entropy, indicating the relative change, of the solution space. We illustrate the robustness and efficiency of our algorithm over other adaptive algorithms in the literature through experimental validation.**

***Index Terms*—Adaptive pruning, Viterbi algorithm, tropical geometry**

I. Introduction

Pruning algorithms are ubiquitous in computer science and are applied in a vast array of problems, with notable applications in speech recognition and convolutional code decoding. Such algorithms are used in order to emphasize the speed of computation over the optimality of the solution. This is essential in applications where real time computation requirements are indispensable (as is the real time decoding of incoming codes). Several authors have proposed adaptive algorithms with varying motivations: some algorithms try to minimize the power consumption of the pruning procedure [16, 8, 9], while others aim to employ ideas from theoretical computer science to improve the accuracy of the algorithm [2, 10]. Some authors even use techniques from control theory to adaptively alter the pruning parameter [18].

Adaptive algorithms are a prime candidate for analysis using tropical geometry [11], which has been increasing in popularity. Many authors [17, 4] resort to using tropical geometry for its appealing properties; namely the piecewise linearity of the solution space, and the intuitive reasoning regarding that space. Tropical geometry allows for a layer of abstraction; instead of reasoning about the algorithm itself, we can reason about the solution space it produces, which often can lead to deductions about the possible solutions. This can be extremely useful in modern day, since the sheer size and dimensionality of the input data can make explicit remarks about the function of algorithms near impossible.

To the authors' best knowledge, this is the first work the utilizes the properties of tropical geometry in the design of an adaptive algorithm. Other authors have proposed (numerical) optimizations which can reduce the energy consumption of convolutional code decoding, and others have analyzed the structure of specific models in order to make deductions and predictions about the pruning parameter. In contrast, in this paper we try to adapt the pruning parameter based solely on the shape and state of the solution space, without assuming any specific structure of the applied model.

References [16, 8, 9] tackle pruning from a telecommunications' perspective, aiming to minimize the energy consumption during decoding in receivers. The authors of [18] aim to utilize metrics, derived from an assumed system structure, to predict the evolution of the leniency parameter. Reference [2] tries to exploit the inherent nature of speech recognition in order to speed up pruning. Finally, in [10] the authors try to efficiently understand the structure of the solution space by computing cliques in order to, subsequently, improve pruning. However, that approach is supervised; each specific application domain has to be analyzed and evaluated whether cliques can be computed, and thus the approach is not generalizable.

In this work we propose an adaptive variation of the Viterbi pruning that adapts the pruning parameter by exploiting the geometrical structure of the solution space. In particular, tropical polytopes can be defined during each step of the Viterbi algorithm. We utilize two metrics (defined in our previous work [17]) deriving from the tropical polytopes of the Viterbi pruning in order to design the new adaptive algorithm. The proposed algorithm computes the metrics' values at each time frame and then compares them with a previous history in order to decide whether to adapt the current value of the pruning parameter. In the case that pruning is indeed warranted, the algorithm tries to adapt the parameter to the effect of maintaining the volume enclosed in the solution space.

In Section II we introduce the background upon which this work is based. Section III presents the proposed adaptive algorithm and briefly analyzes its function. Finally, in Section IV we perform experiments by applying the proposed algorithm to a simulated attack on a network.

II. Background

A. Tropical Algebra and Geometry

Tropical algebra [7, 3, 6] is an algebraic body similar to linear algebra, where the pair of main operations is $(\wedge, +)$. It operates on the extended real multidimensional space $\mathbb{R}^n_{\min}$ ($\mathbb{R}_{\min} = \mathbb{R} \cup \{+\infty\}$). The min-plus matrix multiplication is denoted $\boxplus$, and its result between two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^n_{\min}$ is given by:

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

where $\wedge$ denotes the minimum (see [12] for details).

Tropical geometry [11] studies the objects of Euclidean geometry under the tropical prism. Similar to its Euclidean counterpart, a tropical polytope will be a closed intersection of a finite number of tropical halfspaces. Figure 1 offers visual examples of tropical halfspaces and polytopes.

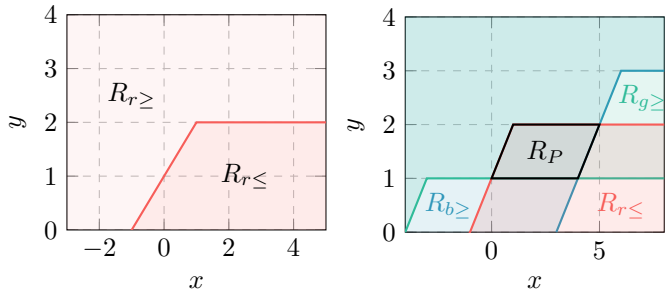

Fig. 1: (a) The tropical line $\min(x+1, 2)$ separates the space into two regions. (b) Three tropical halfspaces define a tropical polytope, R_p.

B. Tropical Viterbi

The Viterbi algorithm can be written in tropical algebra, as we proposed in [17], in the following closed form:

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

where $\mathbf{x}(t)$ is the state vector, $\mathbf{A}$ is the matrix of the transition weights, and $\mathbf{P}(\sigma_t)$ is a diagonal matrix containing the observation weights for the input symbol σ_t at each state.

In [17] we analyze the pruning variant of the Viterbi algorithm in tropical algebra and comment on its geometry. Consider variables $z_1, \ldots, z_n$, where n is the dimensionality of $\mathbf{x}(t)$. Each z_i expresses the range of the values for $x_i(t)$ that are between the Viterbi update law and the leniency threshold. Note that a particular z_i might not admit any values, if the Viterbi update for the corresponding $x_i(t)$ is greater than the leniency threshold. Formally, let $\mathbf{z}$ be a variable vector. That vector can be bounded by the Viterbi update law of (2) and the pruning vector $\boldsymbol{\eta} = \theta + \frac{1}{2}\left(\mathbf{x}(t)^T \boxplus \mathbf{x}(t)\right) + \mathbf{0}$, where θ is the leniency parameter. This defines a tropical polytope on the variable vector $\mathbf{z}$, which encloses all the possible assignments of the variables that satisfy the constraints (which, essentially, is the solution space for the pruning procedure). Then, at every iteration of the algorithm, two metrics are calculated based on that polytope:

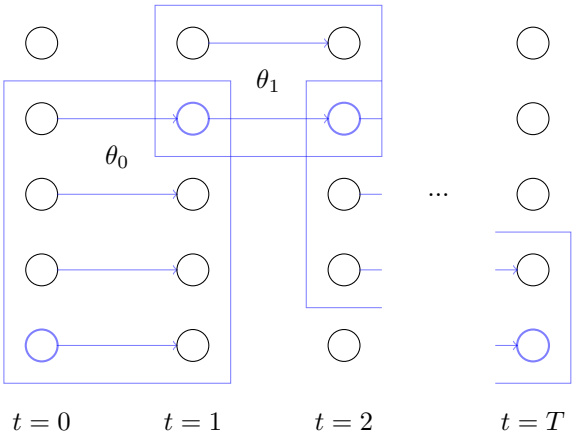

Fig. 2: The proposed algorithm calculates a new value θ_i for the leniency parameter θ of each time frame. Based on the entropy and the volume of the solution space at that time frame, the value θ_i is adapted to allow for the survival of more, or fewer, paths.

- a metric based on the normalized volume inside the polytope:

$$\nu = -\frac{1}{|\text{supp}(\mathbf{z})|} \sum_{i \in \text{supp}(\mathbf{z})} \frac{\log r_i}{\log(\max \mathbf{r})} \tag{3}$$

- a metric based on the entropy of the polytope:

$$\varepsilon = -\frac{1}{|\text{supp}(\mathbf{z})|} \sum_{i \in \text{supp}(\mathbf{z})} -z_i(t) \cdot e^{-z_i(t)} \tag{4}$$

where $r_i = \eta - z_i$. Essentially, r_i is the degree to which each dimension satisfies the Viterbi constraints.

C. Poisson distributions

It is very common for network requests in telecommunication applications to be modeled as Poisson distributions [1, 13]. Poisson distributions are controlled by the parameter λ, which can be interpreted as the mean amount of requests in a time frame. Alternatively, in queuing systems the λ parameter can be perceived as the average time units that a user will have to wait until he is serviced. Formally, the probability of witnessing exactly κ requests in the time frame (or waiting for κ time units) is given by:

$$P[X = \kappa] = e^{-\lambda} \frac{\lambda^\kappa}{\kappa!} \tag{5}$$

where X is the random variable modeling the number of requests in the time frame.

Algorithm 1 AdaptivePruning(data, I, A, α, β, τ, θ_0)

$t \leftarrow 0$
$E, N, L \leftarrow \emptyset$
$\theta \leftarrow \theta_0$
while $t < T$ **do**
  $\varepsilon, \nu \leftarrow \text{polytope}(n, I, A, \Lambda, \theta)$
  if $t \geq \tau$ **then**
    if $\frac{|\varepsilon - \text{mean}(E)|}{\text{mean}(E)} \geq \alpha$ **then**
      if $\nu \leq \text{mean}(N)$ **then**
        $\theta \leftarrow (1 + \beta) \times \theta$
      else
        $\theta \leftarrow (1 - \beta) \times \theta$
      end if
    end if
  end if
  update(E, N, L)
end while
seq $\leftarrow$ backtrack(L)
return seq, L

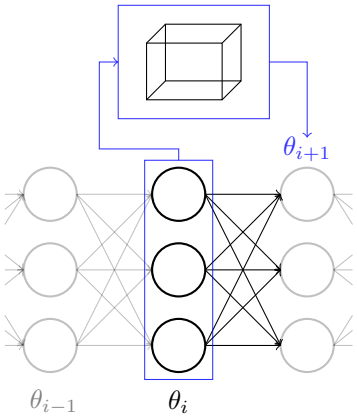

Fig. 3: Bounding the solution space of variable vector $\mathbf{z}$ by the Viterbi update law and the pruning vector $\boldsymbol{\eta}$ of Eq. (2) defines a tropical polytope. The adaptive algorithm exploits the properties of the polytope to adapt the value of the leniency parameter θ.

III. Algorithm

We propose a novel adaptive pruning algorithm that dynamically adapts the pruning parameter by consulting the metrics ν and ε of Eq. (3) and (4). The algorithm computes the metrics and then decides, based on a history of values, if the current iteration warrants an adaptation of the pruning parameter θ. In such a case, the θ parameter is increased or decreased accordingly, in order to maintain the volume enclosed in the polytope. In essence, the algorithm computes a new value θ_i for each time frame, based on the state of the solution space (Fig. 2). The main parameters of the algorithm are:

i. the parameter α, which is the percentage threshold for ε. If the current ε differs from the running history by a percentage more than α, then the algorithm will adapt the parameter θ.

ii. the parameter β, which is the percentage change for θ. If the algorithm proceeds to the adaptation of θ, the current value of ν is compared with the running history. If larger, then this means that the current θ allows for the inclusion of more paths than before, and thus the algorithm proceeds to decrease θ by a percentage of β. Similarly, if ν is smaller than the running history, the algorithm increases θ by a percentage of β, in order to allow for the survival of more paths.

iii. the parameter τ, which is the length of the running history. The algorithm first collects τ samples for the running history, and then compares the current interval's metrics with the average of the most recent τ entries in order to decide if an adaptation is warranted, and also whether to increase or decrease the parameter θ.

iv. the parameter θ_0, which is the initial value of the pruning parameter. θ_0 is used as the pruning parameter for the first τ iterations, in order to calculate the history of metrics ε and ν.

Algorithm 1 presents the proposed algorithm. Besides the parameters mentioned above, the algorithm also accepts other inputs required for the simulation and the Viterbi computation. In particular the initial, and transition costs I, A are required.

In essence, the algorithm operates as follows. First, the Viterbi computation is performed using the previous value for the parameter θ. Then, the geometrical metrics ε and ν are calculated from the polytope of the Viterbi computation. Then, the algorithm compares the current value of the metric ε with the running history. This is done because ε, essentially, calculates the entropy of the solution space. In information theory [14, 5, 15], entropy is a measure of surprise. In particular entropy is used to communicate if a new sample conveys a significant amount of information. For example, if a sample abides by the expectations of the current parametrization of the distribution, then the entropy will be low, indicating that the sample provides no new information regarding the understanding of the distribution (and thus we are not surprised to observe this sample when we are sampling our distribution). Conversely, if a sample is atypical for the current parametrization, then the entropy will be high, indicating that the sample provides new information regarding our understanding of the distribution, suggesting that the parametrization might be wrong. The algorithm tries to leverage this measure of surprise, by comparing the level of current entropy with a running history. If there is a significant difference, this indicates a change in the solution space; the previous parametrization of θ will no longer have similar effects. Thus, in such cases, the algorithm decides to update the value of θ in order to curb the levels of excitement.

When the algorithm decides there exists a need for adaptation, the metric ν is compared to its running history. This is done because, as already mentioned, ν is a measure of the volume of the solution space. Volume can convey quantitative information about the solution space. By maintaining that volume, we ensure that the solution space has some metric similarity throughout iterations. The normalized volume is

useful since the values that the state vector admits change over time. As the total costs get larger (or the probabilities get smaller), then the leniency parameter θ has less of an effect. By trying to maintain the normalized volume, we ensure that the leniency parameter θ will adapt to accommodate larger weights in the later iterations of the Viterbi pruning. The algorithm tries to leverage these observations by keeping a history of the metric ν. In cases where pruning is warranted (as indicated by the percentage change in ε), an effort is made to maintain the normalized volume metric ν, in order to keep the size of the solution space similar. Figure 3 further illustrates this process.

IV. Experimental Results

We performed experiments to evaluate Algorithm 1 using a simulated attack on a network. In particular, we assumed that a malevolent user has gained unauthorized access to the network and wishes to disrupt its function by performing a large number of requests, thus hogging the available resources, hindering the service of benevolent users. We would like to utilize Algorithm 1 in order to solve the attacker localization problem and thus determine that malevolent user. We also assume that user exhibits some intelligence, by varying the rate according to which he sends packets to the network at timed intervals. Thus, our adaptive algorithm will try to dynamically adapt the value of the pruning parameter θ across the iterations, in order to locate the attacker without wasting the system's resources (namely, without examining a large number of states). The number of requests for both the attacker and the benevolent users are modeled with Poisson distributions of varying parameters, as discussed in Section II.

We compare Algorithm 1 (`polytope`) with three other decoding algorithms:

a. non-adaptive Viterbi pruning (`constant`) with a good choice for the leniency parameter, which we use as a baseline.

b. a variation of the average algorithm (`average`) from [10] where we introduced a multiplicative constant to further limit the number of states. The actual proposed algorithm of [10] is not examined as it performed poorly compared to the other algorithms examined.

c. the control-based algorithm (`control`) from [18]. Note that we did not use the variant with the adaptive estimation for the number of steps (referred therein as the two-step control system) because it requires significant preprocessing, which introduces an unfair comparison with other algorithms that do not entail a learning component.

The parameters for the algorithms (a-c) were chosen such that all algorithms achieve the *exact same accuracy*.

In the modeling for the experiments, we assigned a lower Poisson parameter to the attacker. This choice was made because the original framework that was proposed for the metrics ε and ν was modeled in min-plus algebra, and thus the Viterbi algorithm aims to find the sequence of the *lowest cost*. In that vein, we decided to keep the modeling the same, to highlight the mathematical origin of the algorithm. At the same time, we wanted the experiment to maintain its immediately interpretable nature, meaning that the parameter of the Poisson distributions reflects the average number of requests by each user. Thus, we allowed a lower Poisson parameter to the attacker, and perform min-plus calculations.

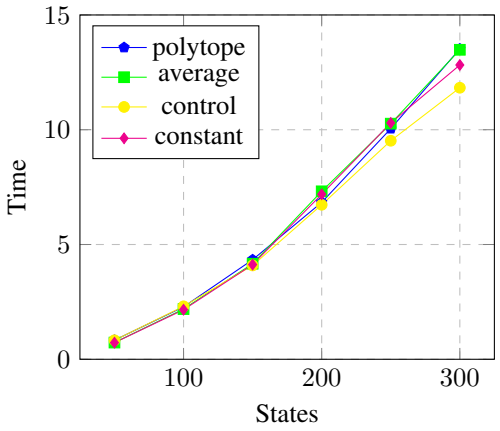

Fig. 4: Computation time in seconds versus the number of users in the network.

We performed two experiments: one where we set the parameter λ of the attacker equal to half of the benign users' and varied the number of total users in the network, and one where the number of users was constant and equal to 100, and we varied the parameter of the attacker. All results were averaged over three runs to ensure stability.

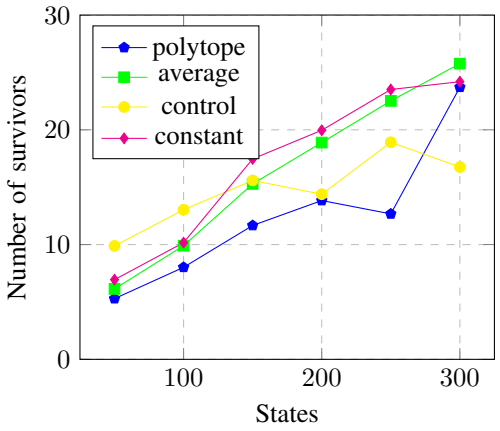

Fig. 5: Number of survivors versus the number of users in the network.

Regarding the parameters of the algorithms, our proposed algorithm has three input parameters; the percentage change in the enclosed volume α, the learning rate β, and the initial guess for the leniency θ_0, which is required for all the examined algorithms. The parameter of the percentage change was set to 0.25 the learning rate was set to 0.01 across all experiments, with the algorithm performing consistently great. However, that is not the case for the other algorithms. The "average" algorithm requires a considerate experimentation with the multiplicative factor of the leniency, which greatly

affects how the algorithm performs. The control based method, even though it can somewhat adapt, it still heavily relies on setting a number of states, or doing significant preprocessing to run. Finally, the constant leniency requires to pinpoint the best leniency (which is problematic).

As can be seen in Fig. 4, all algorithms perform similarly in the time domain. However, as Fig. 5 highlights, our proposed algorithm consistently examines fewer states, which measures the algorithm complexity. In the case of 300 users, all algorithms examine a higher number of states, except the control-based algorithm, whose parameter on the number of surviving states prevents it from adapting. Our proposed algorithm's robustness can also be highlighted in Fig. 6, where, when the difference between the attacker and the users is minuscule, a great number of states is examined due to uncertainty. However, it is evident that other, namely the variation of the average and the control-based algorithms, fail to adapt to this uncertainty, which has a high probability of pruning the best path. Conversely, when the distinction between the attacker and the users is clear from the data, our algorithm only expands a single state, where others waste resources due to their inability to adapt.

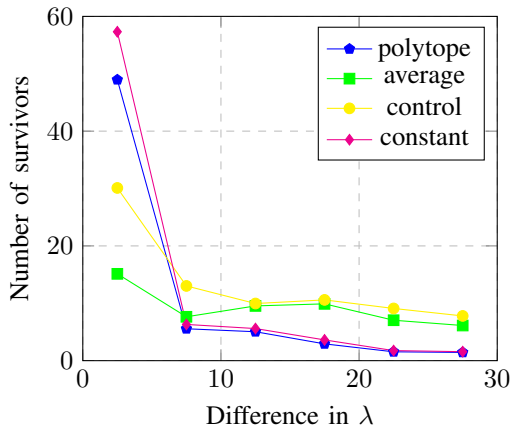

Fig. 6: Number of survivors versus the difference in packet rate λ.

V. Conclusion

In this work we proposed a robust variation of the Viterbi. In particular, we proposed an adaptive pruning algorithm inspired by the geometrical aspect of the tropical analysis of the Viterbi pruning. By analyzing the tropical geometry of the traditional pruning algorithm, we incorporate metrics into the proposed adaptive algorithm in order to evaluate the need for adaptation. In the case the algorithm deemed that the current iteration's metrics vary sufficiently from a previous history, then an adaptation is made to the effect of maintaining the previous levels of the enclosed volume. We experimented with various values for the parameters of the algorithm and presented numerical results of the application of the proposed algorithm in the task of locating a simulated attacker on a network.

References

[1] D. Bertsekas and J. Tsitsiklis. *Introduction to Probability*. Athena Scientific, 2008.

[2] J. Bilmes and H. Lin. "Online Adaptive Learning for Speech Recognition Decoding". In: *Proc. of the Intern. Conf. of the Intern. Speech Communication Assoc.* 2010.

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

[4] V. Charisopoulos and P. Maragos. "Morphological Perceptrons: Geometry and Training Algorithms". In: *Proc. of the Intern. Symp. of Mathematical Morphology*. 2017.

[5] T. Cover and J. Thomas. *Elements of Information Theory*. Series in Telecommunications and Signal Processing. Wiley-Interscience, 2006.

[6] R. A. Cuninghame-Green. *Minimax Algebra*. Lecture Notes in Economics and Mathematical Systems. Springer, 1979.

[7] M. Gondran and M. Minoux. *Graphs, Dioids and Semirings*. Operations Research/Computer Science Interfaces Series. Springer, 2008.

[8] R. Henning and C. Chakrabarti. "Low-Power Approach for Decoding Convolutional Codes with Adaptive Viterbi Algorithm Approximations". In: *Proc. of the Intern. Symp. on Low Power Electronics and Design*. 2002.

[9] R. Henning and C. Chakrabarti. "An Approach for Adaptively Approximating the Viterbi Algorithm to Reduce Power Consumption While Decoding Convolutional Codes". In: *IEEE Trans. on Signal Processing* 52.5 (2004), pp. 1443–1451.

[10] W. Liu and W. Han. "Improved Viterbi Algorithm in Continuous Speech Recognition". In: *Proc. of the Intern. Conf. on Computer Application and System Modeling*. 2010.

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

[12] P. Maragos. "Dynamical Systems on Weighted Lattices: General Theory". In: *Mathematics of Control, Signals, and Systems* 29.21 (2017), pp. 1–33.

[13] J. Pitman. *Probability*. Springer, 1999.

[14] C. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, 1998.

[15] J. Stone. *Information Theory: A Tutorial Introduction*. Sebtel Press, 2015.

[16] R. Tessier et al. "A Reconfigurable, Power-Efficient Adaptive Viterbi Decoder". In: *IEEE Trans. on Very Large Scale Integration Systems* (2005), pp. 484–488.

[17] E. Theodosis and P. Maragos. "Analysis of the Viterbi Algorithm Using Tropical Algebra and Geometry". In: *Proc. of the Intern. Workshop on Signal Processing Advances on Wireless Communications*. 2018.

[18] D. Zhang and L. Du. "Dynamic Beam Pruning Strategy Using Adaptive Control". In: *Proc. of the Intern. Conf. of the Intern. Speech Communication Assoc.* 2004.