Lattice-Based Minimum Error Rate Training Using Weighted Finite-State Transducers with Tropical Polynomial Weights

Rory Waite, Graeme Blackwood, Bill Byrne

Department of Engineering
University of Cambridge

FSMNLP 2012
Donostia - San Sebastian, Spain

25 July 2012

Minimum Error Rate Training – MERT

MERT is an iterative optimisation procedure for log-linear translation models [1].

- Optimizes model parameters directly against difficult-to-optimise metrics, such as BLEU [2]
- 'Tunes' grammars extracted from very large parallel text to small, in-domain, tuning sets
- Originally formulated over N-Best lists – Can be extended to WFSAs [3] and hypergraphs [4]

Previous work has noted that line optimisation over a lattice can be implemented as a semiring of sets of linear functions [5] [6]

---

[1] Och. 2003. Minimum error rate training in statistical machine translation. ACL'03

[2] Papineni et al.. BLEU: a method for automatic evaluation of machine translation. ACL'02

[3] Macherey et al. Lattice-based minimum error rate training for statistical machine translation. EMNLP'08

[4] Kumar et al. Efficient minimum error rate training and minimum Bayes-risk decoding for translation hypergraphs and lattices. ACL'09

[5] Dyer et al. cdec: A decoder, alignment, and learning framework for finite- state and context-free translation models. ACL'10

[6] Sokolov and Yvon. Minimum error rate training semiring. EAMT'11

Alternative MERT formulation

We describe an alternative formulation using the tropical polynomial semiring [7]

- Concise formalism for describing line optimisation
- 'Intuitive' explanation of the MERT shortest distance
- draws on techniques in the currently active field of Tropical Geometry [8]
- Straightforward, simple implementation using OpenFST [9] (see paper for discussion)

[7] Speyer & Sturmfels. Tropical mathematics. Mathematics Magazine. 2009

[8] Richter-Gebert, Sturmfels, Theobald. First steps in tropical geometry. Idempotent mathematics and mathematical physics. 2005

[9] Allauzen et al. OpenFst: A general and efficient weighted finite-state transducer library. Proc Ninth International Conference on Implementation and Application of Automata, 2007

Log-linear translation models

For a source language sentence $\mathbf{f}$, translations are produced via

$$\widehat{\mathbf{e}}(\mathbf{f}_s; \lambda_1^M) = \underset{\mathbf{e}}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m \, h_m(\mathbf{e}, \mathbf{f}) \right\}$$

- Model parameters: λ_m , $m = 1 \ldots M$
- Feature functions: $h_m(\mathbf{e}, \mathbf{f})$, $m = 1 \ldots M$
  - phrase pairs, language model scores, syntactic features, ...

Given a set of reference translations $\{(\mathbf{f_1}, \mathbf{r_1}), \ldots (\mathbf{f_1}, \mathbf{r_S})\}$, translation quality is measured as

$$E(\mathbf{r}_1^S, \mathbf{e}_1^S) = \sum_{s=1}^{S} E(\mathbf{r}_s, \mathbf{e}_s)$$

- E is typically sentence-level BLEU score
- The quality metric is additive over the individual sentence pairs

Minimum Error Rate Training

The general problem is to optimise model parameters over a tuning set $\{(\mathbf{f_1}, \mathbf{r_1}), \ldots (\mathbf{f_1}, \mathbf{r_S})\}$

$$\underset{\lambda_1^M}{\text{argmax}} \sum_s E(\mathbf{r}_s, \widehat{\mathbf{e}}(\mathbf{f_s}, \lambda_1^M))$$

Two simplifications to the general problem, to make it tractable:

1. For each f_s, constrain the possible translation candidates to a set $C_s = \{e_{s,1}, \ldots, e_{s,K}\}$ so that for any λ_1^M, the candidate translations are chosen as

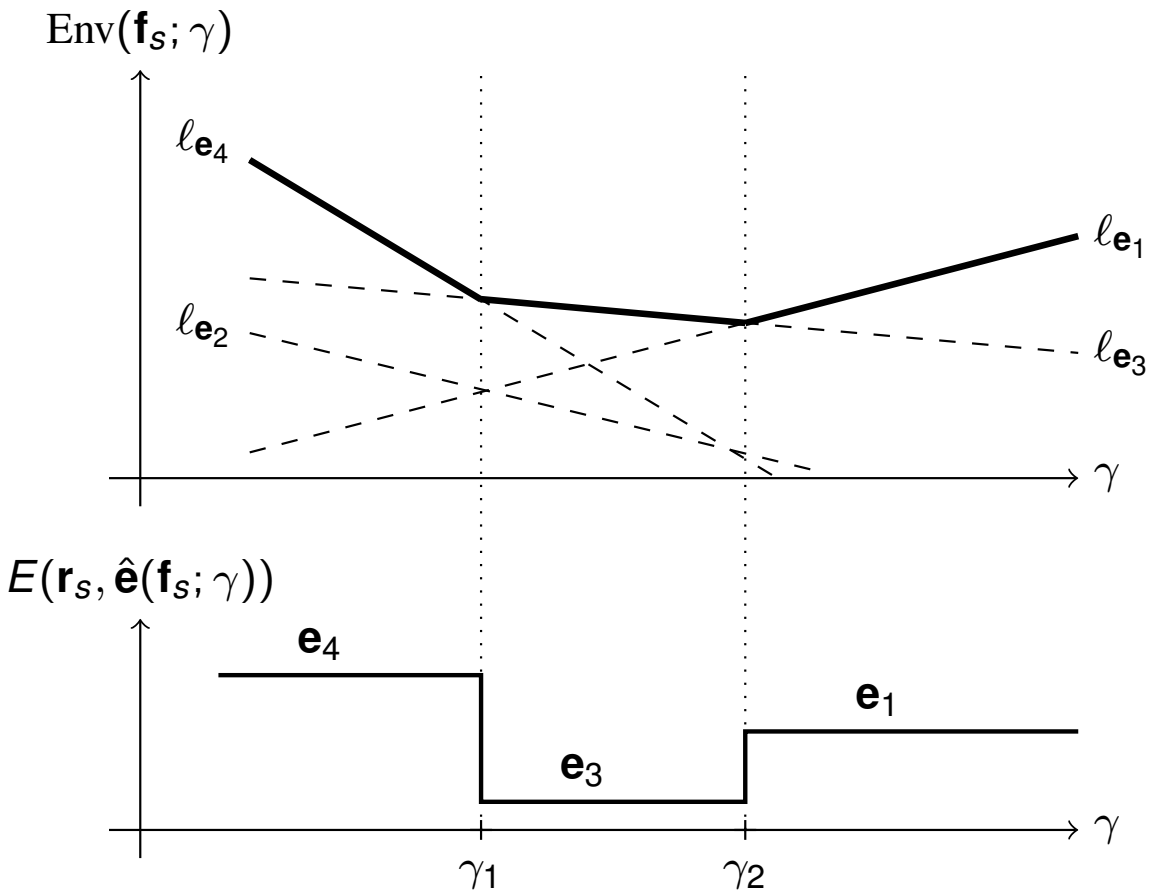
$$\hat{\mathbf{e}}(\mathbf{f}_s; \lambda_1^M) = \underset{\mathbf{e} \in \mathbf{C}_s}{\text{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(\mathbf{e}, \mathbf{f_s}) \right\}$$

2. Line search: constrain the optimisation to search over γ : $\lambda_1^M + \gamma\, d_1^M$

$$\underset{\gamma}{\text{argmax}} \sum_s E(\mathbf{r}_s, \underset{\mathbf{e} \in \mathbf{C}_s}{\text{argmax}} \left\{ \sum_{m=1}^{M} (\lambda_1^M + \gamma\, d_1^M)\, h_m(\mathbf{e}, \mathbf{f_s}) \right\})$$

Hypotheses and error surfaces are piece-wise linear functions of γ

$$\hat{\mathbf{e}}(\mathbf{f}_s;\gamma) = \underset{\mathbf{e}\in\mathbf{C}_s}{\operatorname{argmax}} \Big\{ \underbrace{\sum_m \lambda_m h_m(\mathbf{e},\mathbf{f}_s)}_{a(\mathbf{e},\mathbf{f}_s)} + \gamma \underbrace{\sum_m d_m h_m(\mathbf{e},\mathbf{f}_s)}_{b(\mathbf{e},\mathbf{f}_s)} \Big\} = \underset{\mathbf{e}\in\mathbf{C}_s}{\operatorname{argmax}} \{ \underbrace{a(\mathbf{e},\mathbf{f}_s) + \gamma b(\mathbf{e},\mathbf{f}_s)}_{\ell_{\mathbf{e}}(\gamma)} \}$$



Optimisation

Each $\mathbf{e} \in \mathbf{C}_s$ is associated with a linear function of γ:

$$\ell_{\mathbf{e}}(\gamma) = a(\mathbf{e}, \mathbf{f}_s) + \gamma b(\mathbf{e}, \mathbf{f}_s)$$

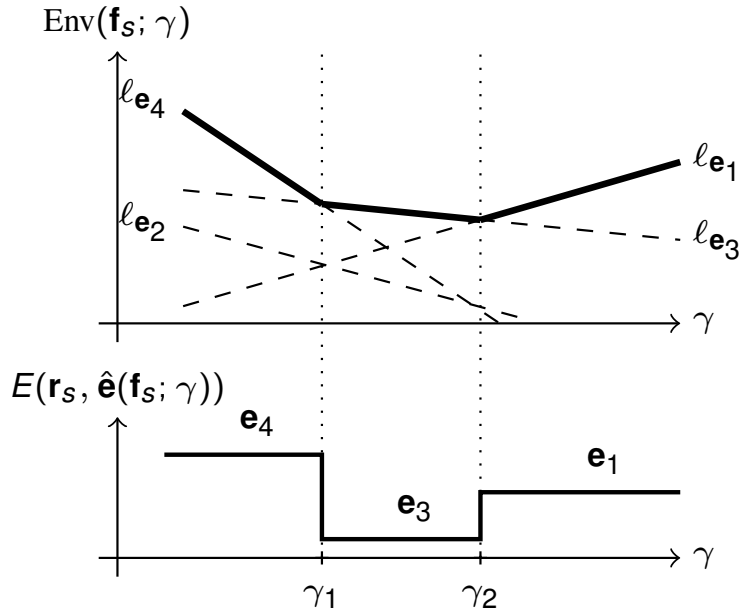
- $a(\mathbf{e}, \mathbf{f}_s)$ is the y-intercept
- $b(\mathbf{e}, \mathbf{f}_s)$ is the gradient.

The 'upper envelope' is specified by :

- linear functions which form line segments
- the values of γ at which they intersect.

These intersection points define regions associated with distinct hypotheses

- $\mathbf{e}_3$ is associated with the midpoint of the interval (γ_1, γ_2)



Upper envelope associated with four hypotheses is represented by three linear functions and two values γ_1 and γ_2.

The intervals and error values specified by the γ intersection points can easily be extracted when the $\mathbf{C}_s$ are N-Best hypothesis lists

- This information is enough to perform numerical optimisation via e.g. Powell's algorithm

Goal: An algorithm which extracts the upper envelope for $\mathbf{C}_s$ that are WFSAs

Tropical Polynomials

Motivation: polynomials can form rings and so can be used as weights in WFSTs

$$f(\gamma) = a_n\gamma^n + a_{n-1}\gamma^{n-1} + \cdots + a_2\gamma^2 + a_1\gamma + a_0$$

Extend classifical definition of polynomials through the use of tropical semiring operations

- Exponentiation: $\gamma^i = \gamma \otimes \cdots \otimes \gamma = i \times \gamma$
- A **tropical monomial** is a linear function with y-intercept a and gradient i

$$a \otimes \gamma^i = a + i \times \gamma$$

- A **tropical polynomial** is a sum ($\oplus$) of monomials

$$f(\gamma) = (a \otimes \gamma^i) \oplus (b \otimes \gamma^j) = \min(a + i \times \gamma\ ,\ b + j \times \gamma)$$

  In classical arithmetic this gives the minimum of a finite collection of linear functions
- Tropical polynomials can be multiplied by a monomial to form another tropical polynomial:

$$\begin{aligned} f(\gamma) &= [\ (a \otimes \gamma^i) \oplus (b \otimes \gamma^j)\ ] \otimes (c \otimes \gamma^k) \\ &= [(a + c) \otimes \gamma^{i+k}] \oplus [(b + c) \otimes \gamma^{j+k}] \\ &= \min(\ (a + c) + (i + k) \times \gamma\ ,\ (b + c) + (j + k) \times \gamma\ ) \end{aligned}$$

Integer Realisations using Tropical Monomials

Goal: represent the upper envelope over an WFST using tropical polynomials

- How to realise $\ell_{\mathbf{e}}(\gamma) = a(\mathbf{e}, \mathbf{f}_s) + \gamma\, b(\mathbf{e}, \mathbf{f}_s)$ using quantities $a \otimes \gamma^i = a + i \times \gamma$?
- The terms $b(\mathbf{e}, \mathbf{f}_s)$ are not integers

One solution: approximate the y-intercept and gradient of $\ell_{\mathbf{e}}(\gamma)$ to n decimal places

$$\begin{aligned}
\tilde{a}(\mathbf{e}, \mathbf{f}_s) &= [a(\mathbf{e}, \mathbf{f}_s) \cdot 10^n]_{\text{int}} \\
\tilde{b}(\mathbf{e}, \mathbf{f}_s) &= [b(\mathbf{e}, \mathbf{f}_s) \cdot 10^n]_{\text{int}} \\
\ell_{\mathbf{e}}(\gamma) \approx \tilde{\ell}_{\mathbf{e}}(\gamma) &= \frac{\tilde{a}(\mathbf{e}, \mathbf{f}_s)}{10^n} + \gamma \cdot \frac{\tilde{b}(\mathbf{e}, \mathbf{f}_s)}{10^n} \\
&= -\tilde{a}(\mathbf{e}, \mathbf{f}_s) \otimes \gamma^{-\tilde{b}(\mathbf{e}, \mathbf{f}_s)}
\end{aligned}$$

$\tilde{\ell}_{\mathbf{e}}(\gamma)$ is a scaled version of $\ell_{\mathbf{e}}(\gamma)$, but ordering over $\mathbf{e}$ is preserved

$$\text{if } \ell_{\mathbf{e}_i}(\gamma) > \ell_{\mathbf{e}_j}(\gamma) \text{ then } \tilde{\ell}_{\mathbf{e}_i}(\gamma) > \tilde{\ell}_{\mathbf{e}_j}(\gamma)$$

and boundary points are unchanged:

$$\text{if } \ell_{\mathbf{e}_i}(\gamma) = \ell_{\mathbf{e}_j}(\gamma) \text{ then } \tilde{\ell}_{\mathbf{e}_i}(\gamma) = \tilde{\ell}_{\mathbf{e}_j}(\gamma)$$

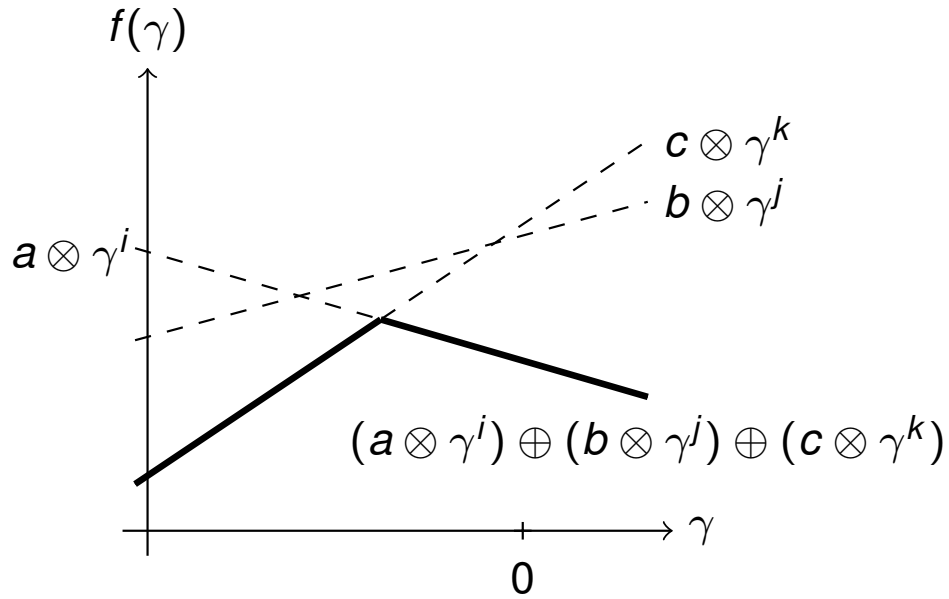
N.B. For large n the approximation errors are minimal

Canonical Forms of Tropical Polynomials

The **canonical form of a tropical polynomial** contains only the monomial terms needed to describe the piece-wise linear function it represents.

- Repeated $\oplus$ operations can introduce redundant monomial terms into tropical polynomials
- Multiple representations of the same polynomial causes problems for shortest distance algorithms
- We modify the generalised sum $\oplus$ to remove redundant terms from a tropical polynomial after every operation
  - Include the SweepLine algorithm[10]
- Only monomials which correspond to the relevant linear functions are kept

Tropical polynomial with redundant terms



$$(a \otimes \gamma^i) \oplus (b \otimes \gamma^j) \oplus (c \otimes \gamma^k) = (a \otimes \gamma^i) \oplus (c \otimes \gamma^k)$$

The canonical form of a tropical polynomial corresponds to a unique and minimal representation of the upper envelope.

[10] Bentley and Ottmann, 1979

The Tropical Polynomial Shortest Distance

The upper envelope corresponds to the shortest distance through a WFST weighted with tropical polynomial weights

- The tropical polynomial shortest distance covers many paths in the WFST
- The hypothesis represented by these paths have to be extracted to compute the error

Tropical polynomial weights can be transformed into regular tropical weights by evaluating the tropical polynomial for a specific value of γ.

$$\begin{aligned} f(1) &= -\tilde{a}(e, \mathbf{f}_s) \otimes 1^{-\tilde{b}(e,\mathbf{f}_s)} \\ &= -\tilde{a}(e, \mathbf{f}_s) - \tilde{b}(e, \mathbf{f}_s) \end{aligned}$$

For a given value of γ a WFST with a tropical polynomial semiring can be mapped to a corresponding WFST with a regular tropical semiring

- The mapping between semirings allows distinct hypotheses to be extracted
- The tropical polynomial shortest distance is a function that describes all possible tropical shortest distances in a WFST for all values of γ

TGMERT Algorithm

0. Pick a line specified by λ_1^M and d_1^M
1. Convert feature functions $h_1^M(e, \mathbf{f})$ to a linear function $\ell_e(\gamma)$
2. Convert $\ell_e(\gamma)$ to $\tilde{\ell}_e(\gamma)$ by approximating y-intercepts and gradients to n decimal places
3. Convert $\tilde{\ell}_e(\gamma)$ to tropical monomials

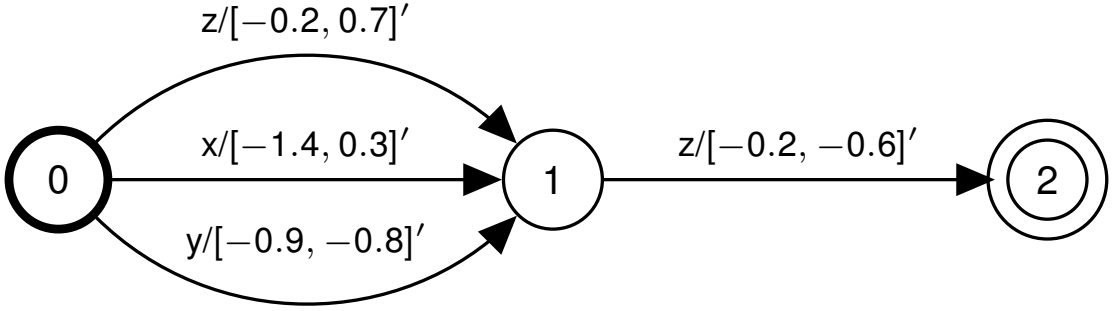
$$-\tilde{a}(e, \mathbf{f}_s) \otimes \gamma^{-\tilde{b}(e,\mathbf{f}_s)}$$

4. Compute the WFST shortest distance to the exit states with generalised sum $\oplus$ and generalised product $\otimes$ defined by the tropical polynomial semiring.
   The resulting tropical polynomial represents the upper envelope of the lattice.
5. Compute the intersection points of the linear functions corresponding to the monomial terms of the tropical polynomial shortest distance.
   These intersection points define intervals of γ in which the MAP hypothesis does not change.
6. Using the midpoint of each interval convert the tropical monomial

$$-\tilde{a}(e, \mathbf{f}_s) \otimes \gamma^{-\tilde{b}(e,\mathbf{f}_s)}$$

   to a regular tropical weight.
7. Find the MAP hypothesis for each interval by extracting the shortest path using a tropical semiring shortest path algorithm.
   These are the upper envelope.

TGMERT Example – two-dimensional feature weights

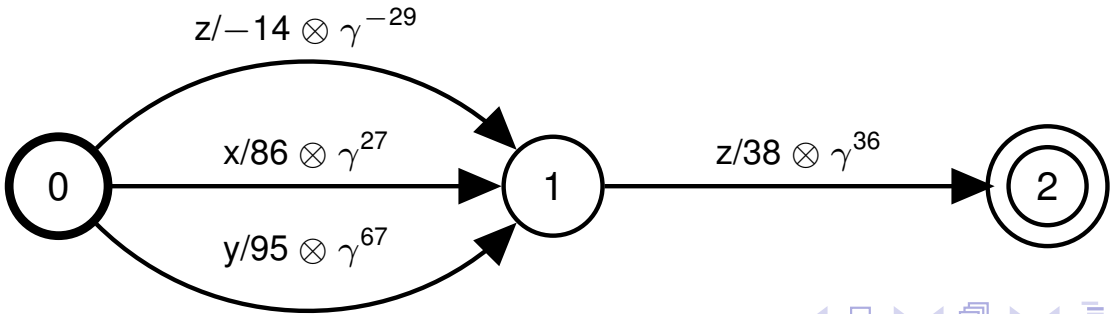


1. Line Search: Initial parameters are $\lambda_1^2 = [0.7, 0.4]$ and the direction is $d_1^2 = [0.3, 0.5]$
2. Map vector feature weights to tropical monomials, e.g. for arc $x/[-1.4, 0.3]'$

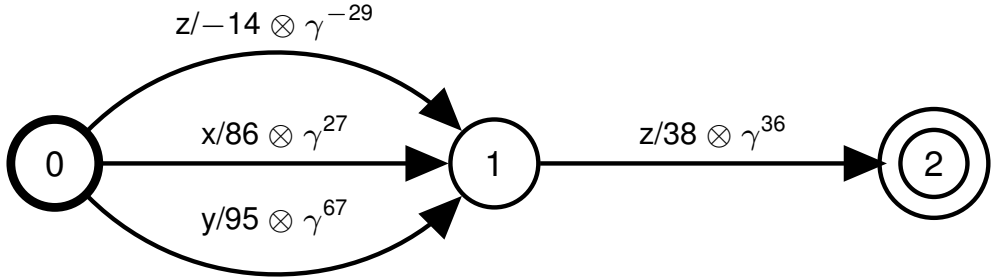
$$\ell_e(\gamma) = \underbrace{\sum_{m=1}^{2} \lambda_m h_1^M(e, \mathbf{f})}_{a(e,\mathbf{f})} + \gamma \underbrace{\sum_{m=1}^{2} d_m h_1^M(e, \mathbf{f_s})}_{b(e,\mathbf{f})} = \underbrace{0.7 \cdot -1.4 + 0.4 \cdot 0.3}_{a(e,\mathbf{f})} + \gamma \cdot \underbrace{0.3 \cdot -1.4 + 0.5 \cdot 0.3}_{b(e,\mathbf{f})}$$

$$= -0.86 - 0.27\gamma$$

3. Quantization at $n = 2$ (i.e. scale by 100), e.g. the weight for arc x becomes $86 \otimes \gamma^{27}$



TGMERT Example, continued



5. There are 3 paths through the lattice, with following costs

$$-14 \otimes \gamma^{-29} \otimes 38 \otimes \gamma^{36} = 24 \otimes \gamma^{7} \qquad \text{z z}$$

$$86 \otimes \gamma^{27} \otimes 38 \otimes \gamma^{36} = 122 \otimes \gamma^{63} \qquad \text{x z}$$

$$95 \otimes \gamma^{67} \otimes 38 \otimes \gamma^{36} = 133 \otimes \gamma^{103} \qquad \text{y z}$$

6. The shortest distance from the initial to the final state:

$$(24 \otimes \gamma^{7}) \oplus (133 \otimes \gamma^{103})$$

   The monomial term $122 \otimes \gamma^{63}$ corresponding to "x z" is dropped from the canonical form.
7. The shortest distance to the exit state is the minimum of two linear functions:

$$\min(24 + 7\gamma, 133 + 103\gamma)$$

   These lines intersect at $\gamma \approx -1.4$
8. The upper envelope contains the hyps z z and y z.
   These, and their associated parameter values γ, are fed to the optimisation procedure.

Three Feature Weight Optimisation Procedures for HiFST [12] – Hierarchical Phrase-Based SMT with Weighted Finite State Transducers

BLEU scores for DARPA Arabic/Chinese-to-English GALE P3 and P4 evaluations[11]

	Chinese-to-English					
	MERT		LMERT		TGMERT	
It.	Tune	Test	Tune	Test	Tune	Test
1	19.5 25.3	16.7	19.5 29.3	22.6	19.5 29.3	22.6
2	16.4 18.9	23.9	22.5 31.4	32.1	22.5 31.4	32.1
3	23.6 28.2	29.1	31.6 32.2	32.5	31.6 32.2	32.5
4	29.2 31.3	31.5	32.2 32.2	32.5	32.2 32.2	32.5
5	31.3 31.8	32.1				
6	32.1 32.4	32.3				
7	32.4 32.4	32.3				

	Arabic-to-English					
	MERT		LMERT		TGMERT	
It.	Tune	Test	Tune	Test	Tune	Test
1	36.2 42.1	40.9	36.2 39.7	38.9	36.2 39.7	38.9
2	42.0 45.1	43.2	44.5 45.8	44.3	44.5 45.8	44.3
3	44.5 45.5	44.1				
4	45.6 45.7	44.0				

- TGMERT and LMERT converge in fewer iterations and to a small gain over MERT
- TGMERT and LMERT are numerically identical

[11] http://projects.ldc.upenn.edu/gale/data/catalog.html

[12] de Gispert et al. Hierarchical phrase-based translation with weighted finite-state transducers and shallow-n grammars. Computational Linguistics, 2010

Conclusions

‘Simple’ implementation of parameter optimisation via lattice MERT

- Add a tropical polynomial weight class to OpenFST, with $\otimes$ and $\oplus$ defined to include the SweepLine algorithm
- ‘Standard’ shortest distance algorithm gives the canonical polynomial associated with the upper envelope
- Performance is identical to other published approaches
- integer approximation is not a problem

Close links to tropical geometry and optimisation theory