On stochastic tree distances and their training via expectation-maximisation

Martin Emms

February 6, 2012

Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

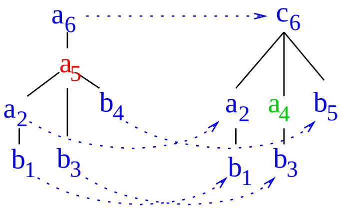
Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

example Tai mapping σ:



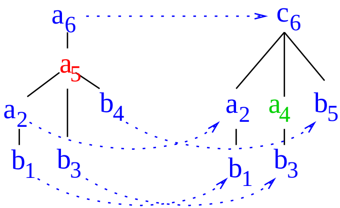
Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

example Tai mapping σ:



Cost of a mapping given by cost of

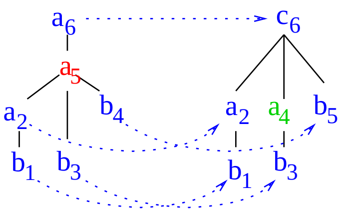
Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

example Tai mapping σ:



Cost of a mapping given by cost of

deletions eg. a_5 has no image

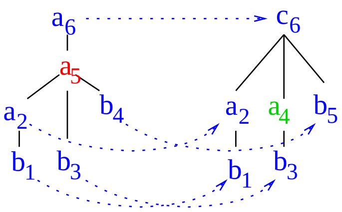
Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

example Tai mapping σ:



Cost of a mapping given by cost of

deletions eg. a_5 has no image

insertions eg. a_4 has no source

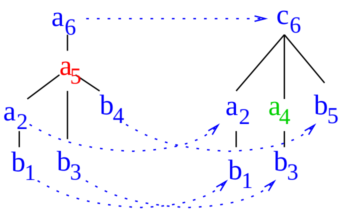
Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

example Tai mapping σ:



Cost of a mapping given by cost of

deletions eg. a_5 has no image

insertions eg. a_4 has no source

match/swaps eg. a_6 goes to c_6

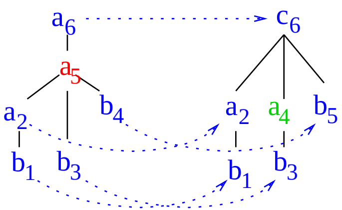
Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

example Tai mapping σ:



given cost table $\mathcal{C}$:

	λ	*a*	*b*	*c*
λ		•	1	•
a	1	0	•	1
b	•	•	0	•
c	•	•	•	•

Cost of a mapping given by cost of

deletions eg. a_5 has no image

insertions eg. a_4 has no source

match/swaps eg. a_6 goes to c_6

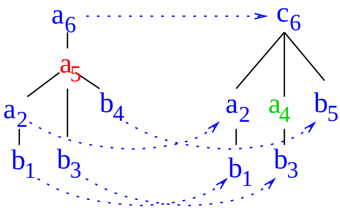
Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

example Tai mapping σ:



given cost table $\mathcal{C}$:

	λ	a	b	c
λ		•	1	•
a	1	0	•	1
b	•	•	0	•
c	•	•	•	•

Cost of a mapping given by cost of

deletions eg. a_5 has no image

insertions eg. a_4 has no source

match/swaps eg. a_6 goes to c_6

total cost of σ is sum on non-zero costs

$\mathcal{C}[\lambda][a] + \mathcal{C}[a][\lambda] + \mathcal{C}[a][c] = 3$

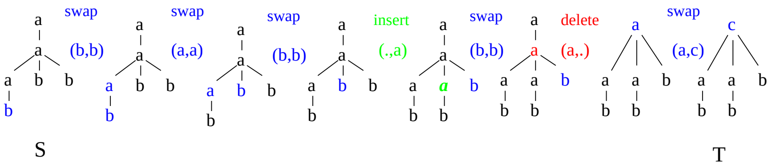
this is also a least cost mapping for this table

Script-based definition Tree Distance

- Can also consider sequence of ’edits’ turning a source tree S into a target tree T

Script-based definition Tree Distance

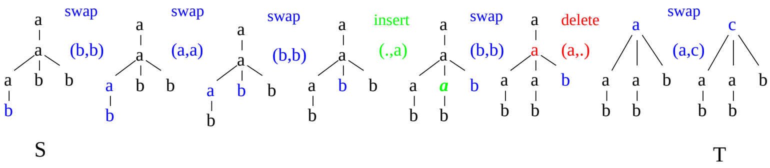
- Can also consider sequence of 'edits' turning a source tree *S* into a target tree *T*



and seek to define distance via least-cost script

Script-based definition Tree Distance

- Can also consider sequence of 'edits' turning a source tree S into a target tree T



  and seek to define distance via least-cost script
- The mapping- and script-based definitions are known to be equivalent, with a script serving as a serialised representation of a mapping. (Tai 79, Kuboyama 07)

Stochastic string distances

- for the case of strings (linear trees), a stochastic variant was first proposed by Ristand and Yianilos (98)

Stochastic string distances

- for the case of strings (linear trees), a stochastic variant was first proposed by Ristand and Yianilos (98)
- where Σ is an alphabet, let *edit operation identifiers*, *EdOp*, be:

$$EdOp = ((\Sigma \cup \{\lambda\}) \times (\Sigma \cup \{\lambda\})) \backslash \langle \lambda, \lambda \rangle$$

  and represent a script with $op_1 \ldots op_n \#$, with each $op_i \in EdOp$.
- assuming a prob distribution p on $EdOp \cup \{\#\}$, define a script probability as

$$P(e_1 \ldots e_n) = \prod_i p(e_i)$$

Stochastic string distances

- for the case of strings (linear trees), a stochastic variant was first proposed by Ristand and Yianilos (98)
- where Σ is an alphabet, let *edit operation identifiers*, *EdOp*, be:

$$EdOp = ((\Sigma \cup \{\lambda\}) \times (\Sigma \cup \{\lambda\})) \backslash \langle \lambda, \lambda \rangle$$

  and represent a script with $op_1 \ldots op_n \#$, with each $op_i \in EdOp$.
- assuming a prob distribution p on $EdOp \cup \{\#\}$, define a script probability as

$$P(e_1 \ldots e_n) = \prod_i p(e_i)$$

- Can think of a script as *yielding* a pair of strings (s, t). If $E(s, t)$ is all scripts which yield (s, t), they defined

  all-paths stochastic edit distance:
  the sum of the probabilities of all scripts $e \in E(s, t)$

  viterbi stochastic edit distance:
  prob. of the most probable $e \in E(s, t)$

Stochastic tree distances

this can be adapted to the case of trees (first proposed by Boyer et al 2007)

Stochastic tree distances

this can be adapted to the case of trees (first proposed by Boyer et al 2007)

Definition (All-scripts stochastic Tai similarity/distance)

The all-scripts stochastic Tai similarity, $\Theta_s^A(S,T)$, is the sum of the probabilities of all edit-scripts which represent a *Tai*-mapping from S to T. The all-scripts stochastic Tai distance, $\Delta_s^A(S,T)$, is its negated logarithm, ie.

$$2^{-\Delta_s^A(S,T)} = \Theta_s^A(S,T)$$

Stochastic tree distances

this can be adapted to the case of trees (first proposed by Boyer et al 2007)

Definition (All-scripts stochastic Tai similarity/distance)

The all-scripts stochastic Tai similarity, $\Theta_s^A(S,T)$, is the sum of the probabilities of all edit-scripts which represent a *Tai*-mapping from S to T. The all-scripts stochastic Tai distance, $\Delta_s^A(S,T)$, is its negated logarithm, ie.

$$2^{-\Delta_s^A(S,T)} = \Theta_s^A(S,T)$$

Definition (Viterbi-script stochastic Tai similarity/distance)

The Viterbi-script stochastic Tai similarity, $\Theta_s^V(S,T)$, is the probability of the most probable edit-script which represents a *Tai*-mapping from S to T. The Viterbi-script stochastic Tai distance, $\Delta_s^V(S,T)$, is its negated logarithm, ie.

$$2^{-\Delta_s^V(S,T)} = \Theta_s^V(S,T)$$

EM Cost adaptation

EM Cost adaptation

- a possible use of a distance is k-NN classifier:

  $cat(S)$ = *VOTE*($\{$*categories of k* **nearest neighbours** *of S* $\}$)

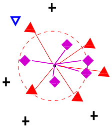
EM Cost adaptation

- a possible use of a distance is k-NN classifier:

  $$cat(S) = VOTE(\{categories\ of\ k\ \textbf{nearest neighbours}\ of\ S\ \})$$

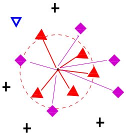
- change cost table $\Rightarrow$ change nearest neighbours $\Rightarrow$ change categorisation:

neighbourhood with
one cost table



infer

neighbourhood with
another cost table



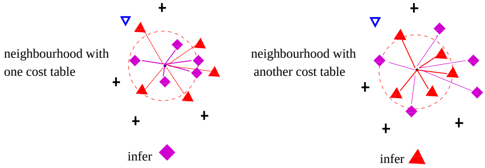
infer

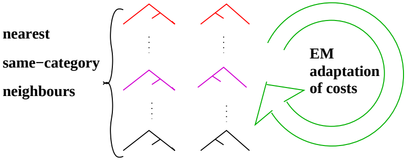
EM Cost adaptation

- a possible use of a distance is k-NN classifier:

  $$cat(S) = VOTE(\{categories\ of\ k\ \textbf{nearest neighbours}\ of\ S\ \})$$

- change cost table $\Rightarrow$ change nearest neighbours $\Rightarrow$ change categorisation:





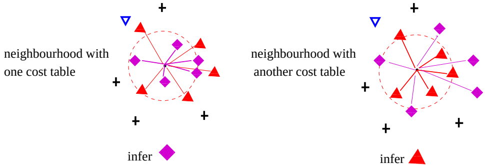
scripts between between *same-category neighbours should have distinctive probs*

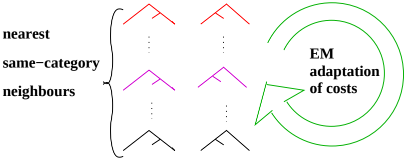
EM Cost adaptation

- a possible use of a distance is k-NN classifier:

  $$cat(S) = VOTE(\{categories\ of\ k\ \textbf{nearest neighbours}\ of\ S\ \})$$

- change cost table $\Rightarrow$ change nearest neighbours $\Rightarrow$ change categorisation:





scripts between between *same-category neighbours should have distinctive probs* $\Rightarrow$ perhaps can use *Expectation-Maximisation techniques to adapt edit-probs from a corpus of same-category nearest neighbours*

Outline

Brute force All-paths EM (infeasible)

Let the *brute-force all-scripts EM algorithm*, EM^{A}_{bf}, be iterations of pair of steps

Brute force All-paths EM (infeasible)

Let the *brute-force all-scripts EM algorithm*, EM^{A}_{bf}, be iterations of pair of steps

(Exp)$_A$ *generate a virtual corpus of scripts by treating each training pair* (S, T) *as standing for* all *the edit-scripts* σ*, which can relate* S *to* T*, weighting each by its conditional probability* $P(\sigma / \Theta^{A}_{s}(S, T)$*), under current probabilities* C^{Θ}

Brute force All-paths EM (infeasible)

Let the *brute-force all-scripts EM algorithm*, EM^A_{bf}, be iterations of pair of steps

(Exp)$_A$ *generate a virtual corpus of scripts by treating each training pair* (S, T) *as standing for* all *the edit-scripts* σ*, which can relate* S *to* T*, weighting each by its conditional probability* $P(\sigma / \Theta^A_s(S, T))$*, under current probabilities* C^Θ

(Max) *apply maximum likelihood estimation to the virtual corpus to derive a new probability table.*

Brute force All-paths EM (infeasible)

Let the *brute-force all-scripts EM algorithm*, EM^A_{bf}, be iterations of pair of steps

(Exp)$_A$ *generate a virtual corpus of scripts by treating each training pair (S, T) as standing for* all *the edit-scripts σ, which can relate S to T, weighting each by its conditional probability $P(\sigma / \Theta^A_s(S, T))$, under current probabilities C^Θ*

(Max) *apply maximum likelihood estimation to the virtual corpus to derive a new probability table.*

A virtual count or expectation $\gamma_{S,T}(op)$ contributed by S, T for an operaton *op* can be defined by

$$\gamma_{S,T}(op) = \sum_{\sigma : S \mapsto T} [\frac{P(\sigma)}{\Theta^A_s(S, T)} \times freq(op \in \sigma)]$$

Brute force All-paths EM (infeasible)

Let the *brute-force all-scripts EM algorithm*, EM^A_{bf}, be iterations of pair of steps

(Exp)$_A$ *generate a virtual corpus of scripts by treating each training pair* (S,T) *as standing for* all *the edit-scripts* σ*, which can relate* S *to* T*, weighting each by its conditional probability* $P(\sigma / \Theta^A_s(S,T))$*, under current probabilities* C^Θ

(Max) *apply maximum likelihood estimation to the virtual corpus to derive a new probability table.*

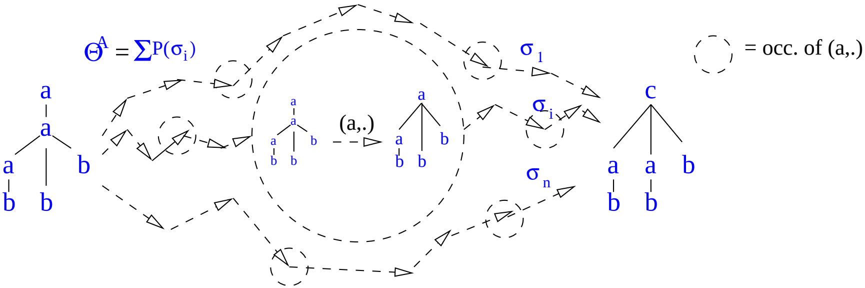
A virtual count or expectation $\gamma_{S,T}(op)$ contributed by S,T for an operaton *op* can be defined by

$$\gamma_{S,T}(op) = \sum_{\sigma : S \mapsto T} [\frac{P(\sigma)}{\Theta^A_s(S,T)} \times freq(op \in \sigma)]$$

(Exp)$_A$ accumulates the $\gamma_{S,T}(op)$ for all *op*'s, for all (S,T)

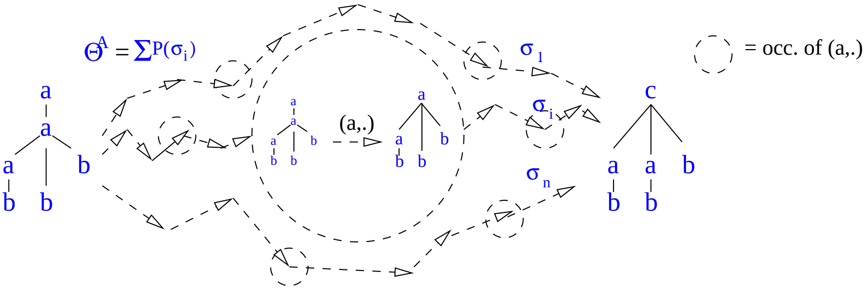
Brute force All-paths EM (infeasible)

Brute force All-paths EM (infeasible)



$$\gamma_{S,T}(op) = \sum_{\sigma:S\mapsto T} \left[\frac{P(\sigma)}{\Theta^A_s(S,T)} \times freq(op \in \sigma)\right]$$

Brute force All-paths EM (infeasible)

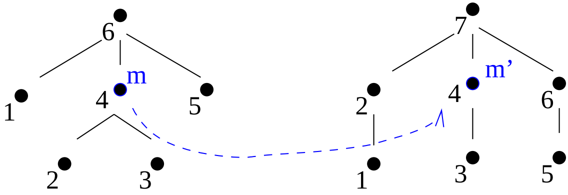


$$\gamma_{S,T}(op) = \sum_{\sigma:S\mapsto T} [\frac{P(\sigma)}{\Theta^A_s(S,T)} \times freq(op \in \sigma)]$$

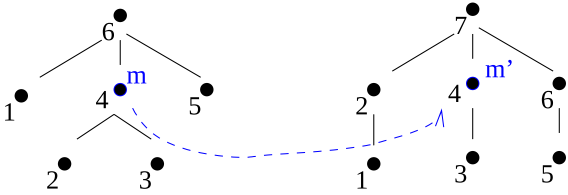
- **infeasible**

for HMMs and stochastic string distance the key to making feasible algorithm is to split the expectations $\gamma_{(S,T)}$ into position specific version $\gamma_{(S,T)}[i,j]$

for HMMs and stochastic string distance the key to making feasible algorithm is to split the expectations $\gamma_{(S,T)}$ into position specific version $\gamma_{(S,T)}[i,j]$



for HMMs and stochastic string distance the key to making feasible algorithm is to split the expectations $\gamma_{(S,T)}$ into position specific version $\gamma_{(S,T)}[i,j]$



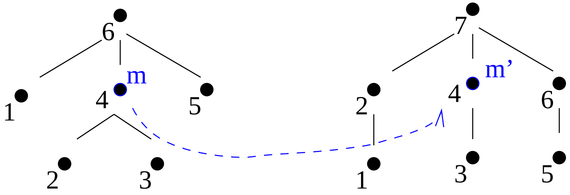
$\gamma_{(S,T)}[4][4](m,m')$, the expectation for a swap (m,m') at $(4,4)$ has the semantics

$$\begin{aligned}\gamma_{(S,T)}[4,4](m,m') &= \sum_{\sigma \in E(S,T),(m_4,m'_4)\in\sigma} \left[\frac{p(\sigma)}{\Theta^A_s(S,T)}\right] \\ &= \frac{1}{\Theta^A_s(S,T)} \times \sum_{\sigma \in E(S,T),(m_4,m'_4)\in\sigma} [p(\sigma)]\end{aligned}$$

in words,

> the sum over the conditional probabilities of any script σ containing a m_4, m'_4 substitution, given that it is a script between S and T

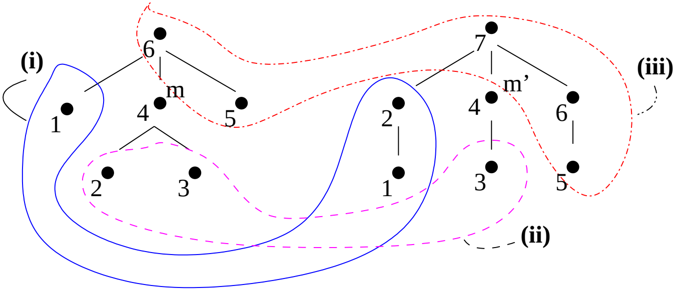
Efficient calculation of $\gamma_{(S,T)}[i][j](op)$?



So how to efficiently calculate:

$$\frac{1}{\Theta^A_s(S,T)} \times \sum_{\sigma \in E(S,T),(m_4,m'_4)\in\sigma} [p(\sigma)]$$

Efficient calculation of $\gamma_{(S,T)}[i][j](op)$?



Boyer et al (2007) suggest the fectorisation

$$\sum_{e_1 \in E([\cdot_1],[\cdot_2(\cdot_1)])} [p(e_1)] \times \sum_{e_2 \in E([\cdot_2 \cdot_3],[\cdot_3])} [p(e_2)] \times p(m,m') \times \sum_{e_3 \in E([\cdot_6(\cdot_5)],[\cdot_7(\cdot_6(\cdot_5))])} [p(e_3)]$$

but we can show that this is not a sound factorisation

Unsoundness

$$\sum_{\sigma \in E(S,T),(m_4,m_4')} p(\sigma)$$

Unsoundness

$$\sum_{\sigma \in E(S,T),(m_4,m_4')} p(\sigma)$$

means sum $p(\sigma)$ for scripts which represent a mapping containing (m_4, m_4')

Unsoundness

$$\sum_{\sigma \in E(S,T),(m_4,m_4')} p(\sigma)$$ means sum $p(\sigma)$ for scripts which represent a mapping containing (m_4, m_4')

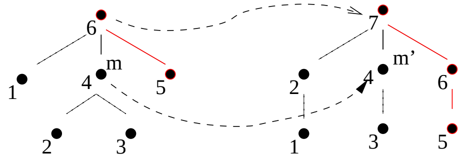
$\Rightarrow$ if an ancestor of m_4 is in the mapping (ie. not deleted) then its image under the mapping must be an ancestor of m_4' also

Unsoundness

$$\sum_{\sigma \in E(S,T),(m_4,m'_4)} p(\sigma)$$ means sum $p(\sigma)$ for scripts which represent a mapping containing (m_4,m'_4)

$\Rightarrow$ if an ancestor of m_4 is in the mapping (ie. not deleted) then its image under the mapping must be an ancestor of m'_4 also

so $\cdot_6$ of S being mapped to $\cdot_7$ of T is consistent with (m_4, m'_4)

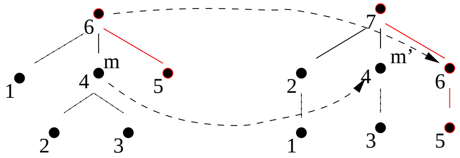


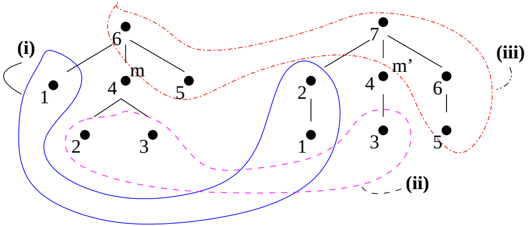
Unsoundness

$$\sum_{\sigma \in E(S,T),(m_4,m_4')} p(\sigma)$$ means sum $p(\sigma)$ for scripts which represent a mapping containing (m_4,m_4')

$\Rightarrow$ if an ancestor of m_4 is in the mapping (ie. not deleted)
then its image under the mapping must be an ancestor of m_4' also

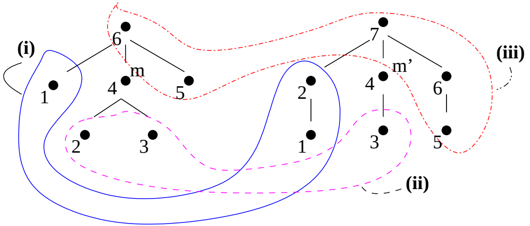
but $\cdot_6$ of S being mapped to $\cdot_6$ of T is not consistent with (m_4, m_4')





so the problem with the factorisation

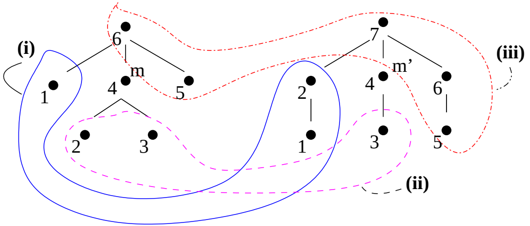
$$\sum_{e_1 \in E([\cdot_1],[\cdot_2(\cdot_1)])} [p(e_1)] \times \sum_{e_2 \in E([\cdot_2 \cdot_3],[\cdot_3])} [p(e_2)] \times p(m, m') \times \sum_{e_3 \in E([\cdot_6(\cdot_5)],[\cdot_7(\cdot_6(\cdot_5))])} [p(e_3)]$$



so the problem with the factorisation

$$\sum_{e_1 \in E([\cdot_1],[\cdot_2(\cdot_1)])} [p(e_1)] \times \sum_{e_2 \in E([\cdot_2 \cdot_3],[\cdot_3])} [p(e_2)] \times p(m,m') \times \sum_{e_3 \in E([\cdot_6(\cdot_5)],[\cdot_7(\cdot_6(\cdot_5))])} [p(e_3)]$$

is the third term sums both things consistent and inconsistent with (m_4, m_4')

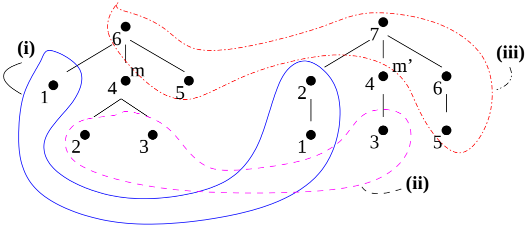


so the problem with the factorisation

$$\sum_{e_1 \in E([\cdot_1],[\cdot_2(\cdot_1)])} [p(e_1)] \times \sum_{e_2 \in E([\cdot_2 \cdot_3],[\cdot_3])} [p(e_2)] \times p(m, m') \times \sum_{e_3 \in E([\cdot_6(\cdot_5)],[\cdot_7(\cdot_6(\cdot_5))])} [p(e_3)]$$

is the third term sums both things consistent and inconsistent with (m_4, m'_4)

$\sum_{e_3 \in E([\cdot_6(\cdot_5)],[\cdot_7(\cdot_6(\cdot_5))])} [p(e_3)] =$

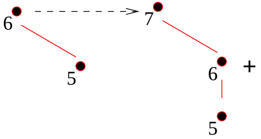


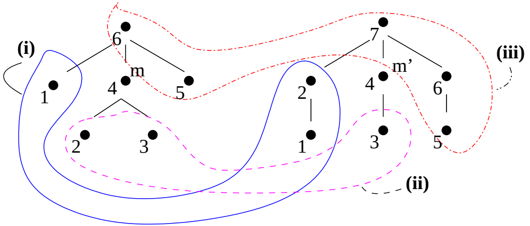
so the problem with the factorisation

$$\sum_{e_1\in E([\cdot_1],[\cdot_2(\cdot_1)])}[p(e_1)] \times \sum_{e_2\in E([\cdot_2\cdot_3],[\cdot_3])}[p(e_2)] \times p(m,m') \times \sum_{e_3\in E([\cdot_6(\cdot_5)],[\cdot_7(\cdot_6(\cdot_5))])}[p(e_3)]$$

is the third term sums both things consistent and inconsistent with (m_4, m_4')

$\sum_{e_3\in E([\cdot_6(\cdot_5)],[\cdot_7(\cdot_6(\cdot_5))])}[p(e_3)] =$



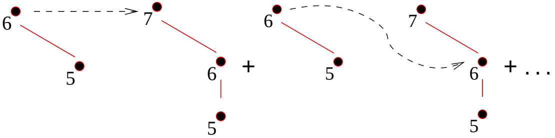


so the problem with the factorisation

$$\sum_{e_1\in E([\cdot_1],[\cdot_2(\cdot_1)])}[p(e_1)] \times \sum_{e_2\in E([\cdot_2\cdot_3],[\cdot_3])}[p(e_2)]\times p(m,m')\times \sum_{e_3\in E([\cdot_6(\cdot_5)],[\cdot_7(\cdot_6(\cdot_5))])}[p(e_3)]$$

is the third term sums both things consistent and inconsistent with (m_4, m'_4)

$\sum_{e_3\in E([\cdot_6(\cdot_5)],[\cdot_7(\cdot_6(\cdot_5))])}[p(e_3)] =$



For general trees, a feasible equivalent to the brute-force EM^{bf}_{A} remains an unsolved problem.

Outline

Let *the Viterbi EM algorithm* EM^V, be iterations of pair of steps

(Exp)$_V$ *generate a virtual corpus of scripts by treating each training pair* (S, T) *as standing for the* best *edit-script* σ*, which can relate* S *to* T*, weighting it by its conditional probability* $P(\sigma)/\Theta^A_s(S, T)$*, under current costs* $\mathcal{C}$

Let *the Viterbi EM algorithm* EM^V, be iterations of pair of steps

(Exp)$_V$ *generate a virtual corpus of scripts by treating each training pair* (S, T) *as standing for the* best *edit-script* σ*, which can relate* S *to* T*, weighting it by its conditional probability* $P(\sigma)/\Theta^A_s(S, T)$*, under current costs* $\mathcal{C}$

(Max) *apply maximum likelihood estimation to the virtual corpus to derive a new probability table.*

Where $\mathcal{V}$ is the best-script, the virtual count or expectation $\gamma_{S,T}(op)$ contributed by S, T for the operation *op* is defined by

$$\gamma_{(S,T)}(op) = \frac{\Theta^V_s(S, T)}{\Theta^A_s(S, T)} \times freq(op \in \mathcal{V})$$

Let *the Viterbi EM algorithm* EM^V, be iterations of pair of steps

(Exp)$_V$ *generate a virtual corpus of scripts by treating each training pair* (S, T) *as standing for the* best *edit-script* σ, *which can relate* S *to* T, *weighting it by its conditional probability* $P(\sigma)/\Theta^A_s(S, T)$, *under current costs* $\mathcal{C}$

(Max) *apply maximum likelihood estimation to the virtual corpus to derive a new probability table.*

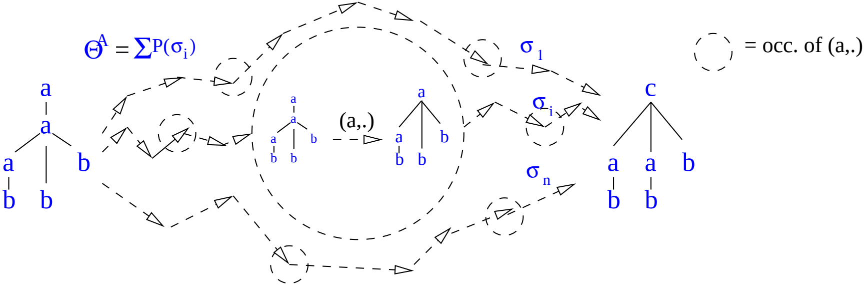
Where $\mathcal{V}$ is the best-script, the virtual count or expectation $\gamma_{S,T}(op)$ contributed by S, T for the operation *op* is defined by

$$\gamma_{(S,T)}(op) = \frac{\Theta^V_s(S, T)}{\Theta^A_s(S, T)} \times freq(op \in \mathcal{V})$$

(Exp)$_V$ accumulates the $\gamma_{S,T}(op)$ for all *op*'s, for all (S, T)

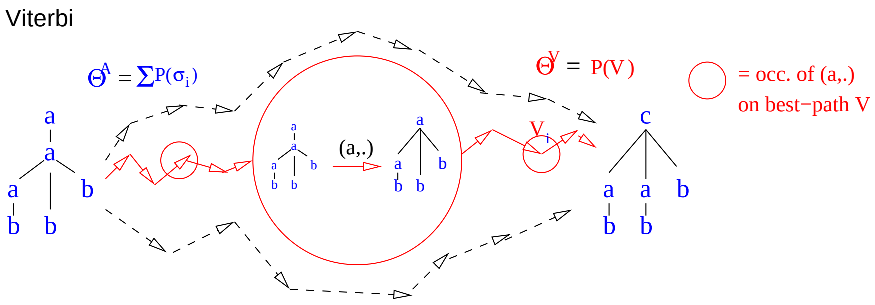
Viterbi approximation EM^V (feasible)

All paths



$$\gamma_{S,T}(op) = \sum_{\sigma:S\mapsto T}\left[\frac{P(\sigma)}{\Theta_s^A(S,T)} \times freq(op \in \sigma)\right]$$

Viterbi approximation EM^V (feasible)



$$\gamma_{(S,T)}(op) = \frac{\Theta_s^V(S,T)}{\Theta_s^A(S,T)} \times freq(op \in \mathcal{V})$$

Data set: QuestionBank

Data set: QuestionBank

2755 syntactically analysed and semantically categorised questions

Data set: QuestionBank

2755 syntactically analysed and semantically categorised questions

Cat	Example
NUM	When was London 's Docklands Light Railway constructed ? (SBARQ (WHADVP (WRB When))(SQ (VBD was)(NP (NP (NNP London)(POS 's))(NNPS Docklands) (JJ Light)(NN Railway))(VP (VBN constructed)))(. ?))
LOC	What country is the biggest producer of tungsten ? (SBARQ (WHNP (WDT What)(NN country))(SQ (VBZ is)(NP (NP (DT the)(JJS biggest)(NN producer)) (PP (IN of)(NP (NN tungsten)))))(. ?))
HUM	What is the name of the managing director of Apricot Computer ? (WHNP (WP What))(SQ (VBZ is)(NP (NP (DT the)(NN name))(PP (IN of)(NP (NP (DT the)(JJ managing)(NN director))

Data set: QuestionBank

2755 syntactically analysed and semantically categorised questions

Cat	Example
NUM	When was London 's Docklands Light Railway constructed ? (SBARQ (WHADVP (WRB When))(SQ (VBD was)(NP (NP (NNP London)(POS 's))(NNPS Docklands) (JJ Light)(NN Railway))(VP (VBN constructed)))(. ?))
LOC	What country is the biggest producer of tungsten ? (SBARQ (WHNP (WDT What)(NN country))(SQ (VBZ is)(NP (NP (DT the)(JJS biggest)(NN producer)) (PP (IN of)(NP (NN tungsten)))))(. ?))
HUM	What is the name of the managing director of Apricot Computer ? (WHNP (WP What))(SQ (VBZ is)(NP (NP (DT the)(NN name))(PP (IN of)(NP (NP (DT the)(JJ managing)(NN director))

Intuitiion: in scripts between between same-category neighbours *should* have distinctive probs eg. . $P(\textit{who}/\textit{when}) << P(\textit{state}/\textit{country})$.

k-NN categorisation

- experiments make 9:1 split into Examples vs Testing and evaluate a distance measure in k-NN classifier

k-NN categorisation

- experiments make 9:1 split into Examples vs Testing and evaluate a distance measure in k-NN classifier

  so for each T in Testing, assign a category based on the categories of its k nearest neighbours in Examples

k-NN categorisation

- experiments make 9:1 split into Examples vs Testing and evaluate a distance measure in k-NN classifier

  so for each T in Testing, assign a category based on the categories of its k nearest neighbours in Examples

  $$cat(T) = VOTE(\{categories\ of\ k\ \textbf{nearest neighbours}\ of\ T\})$$

- compare

k-NN categorisation

- experiments make 9:1 split into Examples vs Testing and evaluate a distance measure in k-NN classifier

  so for each T in Testing, assign a category based on the categories of its k nearest neighbours in Examples

  $$cat(T) = VOTE(\{categories\ of\ k\ \textbf{nearest neighbours}\ of\ T\})$$

- compare

  tree-distance with standard unit costs

k-NN categorisation

- experiments make 9:1 split into Examples vs Testing and evaluate a distance measure in k-NN classifier

  so for each T in Testing, assign a category based on the categories of its k nearest neighbours in Examples

  $$cat(T) = VOTE(\{categories\ of\ k\ \textbf{nearest neighbours}\ of\ T\})$$

- compare

  tree-distance with standard unit costs

  stochastic tree-distance with untrained costs

k-NN categorisation

- experiments make 9:1 split into Examples vs Testing and evaluate a distance measure in k-NN classifier

  so for each T in Testing, assign a category based on the categories of its k nearest neighbours in Examples

  $$cat(T) = VOTE(\{categories\ of\ k\ \textbf{nearest neighbours}\ of\ T\})$$

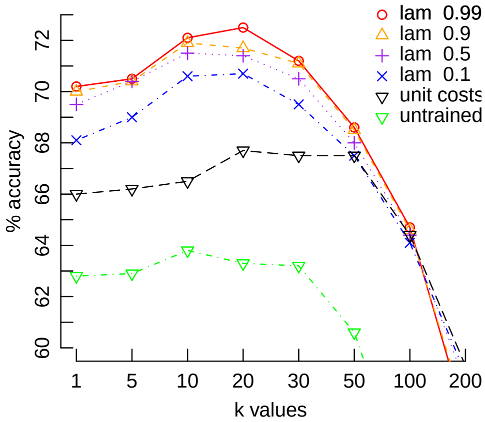
- compare

  tree-distance with standard unit costs

  stochastic tree-distance with untrained costs

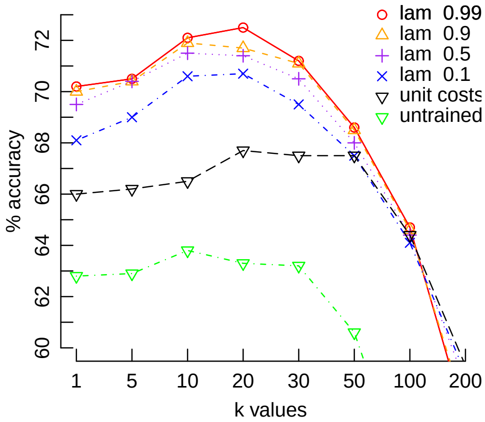
  stochastic tree distance with trained costs
  training by EM^V on same-category neighbours from the Example set

Experimental outcome (brief)



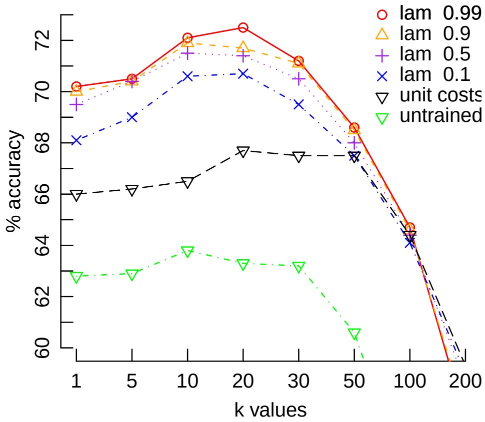
Experimental outcome (brief)

- standard unit-costs $\triangledown$, max. 67.7%



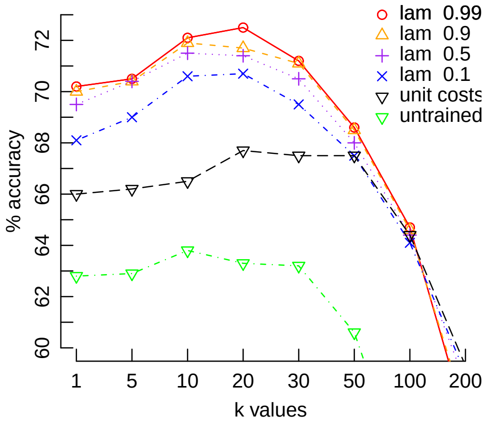
Experimental outcome (brief)

- standard unit-costs
  ▽, max. 67.7%
- initial stochastic costs
  ▽ max. 63.8%
  worse than unit costs



Experimental outcome (brief)

- standard unit-costs
  ▽, max. 67.7%
- initial stochastic costs
  ▽ max. 63.8%
  worse than unit costs
- best EM^V-adapted costs
  ○, max. 72.5%
  about 5% better than unit-costs
  (▽, max. 67.7%)



Stochastic cost initialisation

EM^V needs an initialisation of its parameters.

we used a basically uniform initialisation except

Stochastic cost initialisation

EM^V needs an initialisation of its parameters.

we used a basically uniform initialisation except

diagonal entries are d times more probable than non-diagonal.

Stochastic cost initialisation

EM^V needs an initialisation of its parameters.

we used a basically uniform initialisation except

diagonal entries are d times more probable than non-diagonal.

examples for $d = 3, 10, 100$, and 1000 are:

3	λ	a	b
λ	3.7	3.7	3.7
a	3.7	2.115	3.7
b	3.7	3.7	2.115

10	λ	a	b
λ	4.755	4.755	4.755
a	4.755	1.433	4.755
b	4.755	4.755	1.433

100	λ	a	b
λ	7.693	7.693	7.693
a	7.693	1.05	7.693
b	7.693	7.693	1.05

1000	λ	a	b
λ	10.97	10.97	10.97
a	10.97	1.005	10.97
b	10.97	10.97	1.005

NOTE: diagonal entries are not insignificant

Smoothing

We used a *smoothing* option on a table C^{Δ} derived by EM^{V}, interpolating it with the stochastic initialisation $C^{\Delta}{}_{u}(d)$ as follows:

Smoothing

We used a *smoothing* option on a table C^{Δ} derived by EM^{V}, interpolating it with the stochastic initialisation $C^{\Delta}{}_{u}(d)$ as follows:

$$2^{-C^{\Delta}{}_{\lambda}[x][y]} = \lambda(2^{-C^{\Delta}[x][y]}) + (1-\lambda)(2^{-C^{\Delta}{}_{u}(d)[x][y]})$$

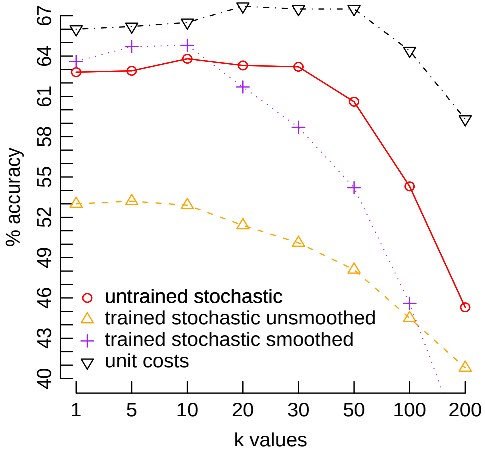
with $0 \leq \lambda \leq 1$

$\lambda = 1$ gives all the weight to the derived table

$\lambda = 0$ gives all the weight to the initial table

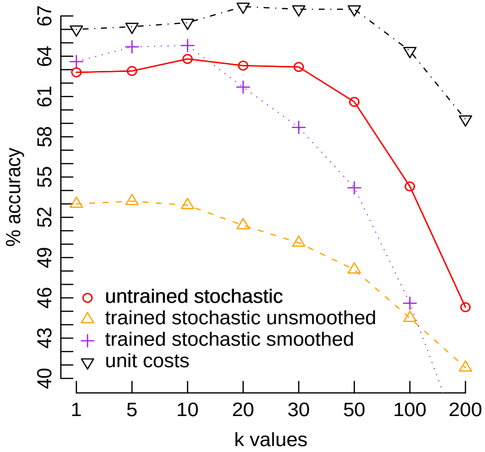
Outline

Experiment One



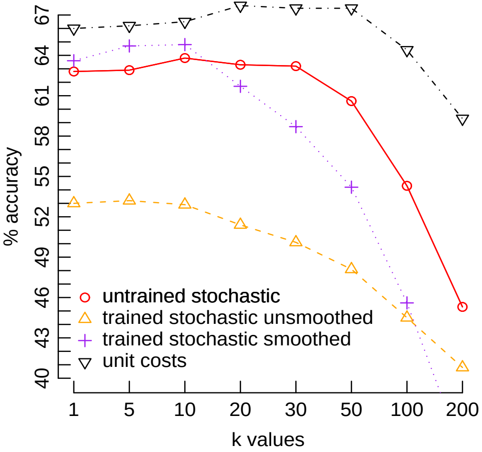
Experiment One

- unit-costs (▽, max. 67.7%) exceeds non-adapted $C^{\Delta}{}_u(3)$ costs (○, max. 63.8%)



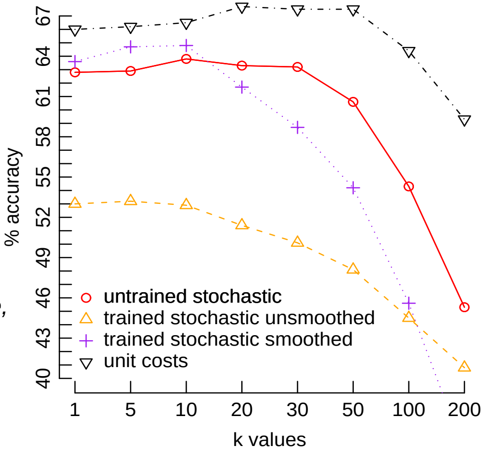
Experiment One

- unit-costs (▽, max. 67.7%) exceeds non-adapted $C^{\Delta}{}_{u}(3)$ costs (○, max. 63.8%)
- unsmoothed EM^{V}-adapted costs (△, max. 53.2%) worse than initial, stochastic costs (○, max. 63.8%)



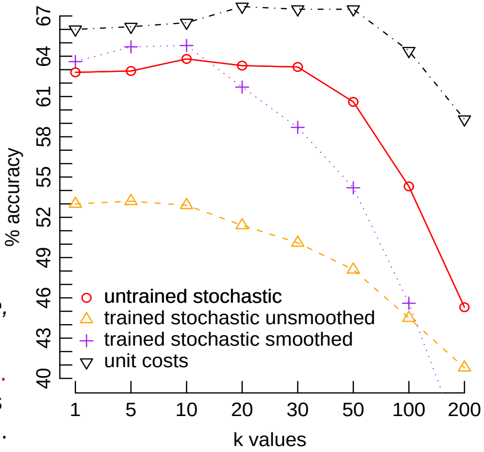
Experiment One

- unit-costs (▽, max. 67.7%) exceeds non-adapted $C^{\Delta}{}_{u}(3)$ costs (○, max. 63.8%)
- unsmoothed EM^V-adapted costs (△,max. 53.2%) worse than initial, stochastic costs (○, max. 63.8%)
- EM^V-adapted costs on training set gives 95% accuracy: $\Rightarrow EM^V$ makes training pairs *too probable, and over-fits*.



Experiment One

- unit-costs (▽, max. 67.7%) exceeds non-adapted $C^{\Delta}{}_{u}(3)$ costs (○, max. 63.8%)
- unsmoothed EM^{V}-adapted costs (△,max. 53.2%) worse than initial, stochastic costs (○, max. 63.8%)
- EM^{V}-adapted costs on training set gives 95% accuracy: $\Rightarrow EM^{V}$ makes training pairs *too probable, and over-fits*.
- *smoothing* adapted costs (+,max. 64.8%) improves over initial costs (○) but is still below unit costs (▽).



Despite poor performace of the EM^V-adapted costs, some of the adapted costs seem intuitive. Here is a sample from top 1% of adapted swap costs, which are plausibly discounted relative to others:

8.50	?	.	12.31	The	the
8.93	NNP	NN	12.65	you	I
9.47	VBD	VBZ	13.60	can	do
9.51	NNS	NN	13.83	many	much
9.78	a	the	13.92	city	state
11.03	was	is	13.93	city	country
11.03	’s	is			

Outline

- Recall: For the stochastic distance Δ^V_s cost-table entries represent probabilities via

$$2^{-C^{\Delta}(x,y)} = p(x,y)$$

- Recall: For the stochastic distance Δ_s^V cost-table entries represent probabilities via

$$2^{-C^{\Delta}(x,y)} = p(x,y)$$

- a single 0 entry in C^{Δ} implies *infinite* cost entries everywhere else.

- Recall: For the stochastic distance Δ_s^V cost-table entries represent probabilities via

$$2^{-C^{\Delta}(x,y)} = p(x,y)$$

- a single 0 entry in C^{Δ}implies *infinite* cost entries everywhere else.
  $\Rightarrow$ a stochastically valid cost table cannot have zero costs on the diagonal

- Recall: For the stochastic distance Δ^V_s cost-table entries represent probabilities via

$$2^{-C^{\Delta}(x,y)} = p(x,y)$$

- a single 0 entry in C^{Δ} implies *infinite* cost entries everywhere else.
  $\Rightarrow$ a stochastically valid cost table cannot have zero costs on the diagonal
- perhaps this impedes good categorisation; note also the unit-cost setting, which is clearly 'uniform' in a sense, out-performs the 'uniform' stochastic initialisations

- Recall: For the stochastic distance Δ_s^V cost-table entries represent probabilities via

$$2^{-C^{\Delta}(x,y)} = p(x,y)$$

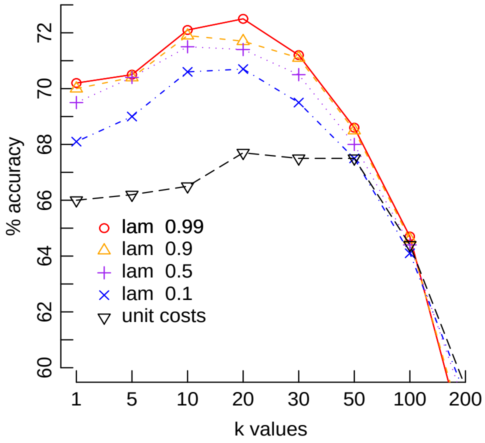
- a single 0 entry in C^{Δ} implies *infinite* cost entries everywhere else.
  $\Rightarrow$ a stochastically valid cost table cannot have zero costs on the diagonal
- perhaps this impedes good categorisation; note also the unit-cost setting, which is clearly 'uniform' in a sense, out-performs the 'uniform' stochastic initialisations
- suggests final step in which all the entries on the cost-table's diagonal are zeroed.

- Recall: For the stochastic distance Δ_s^V cost-table entries represent probabilities via

$$2^{-C^{\Delta}(x,y)} = p(x,y)$$

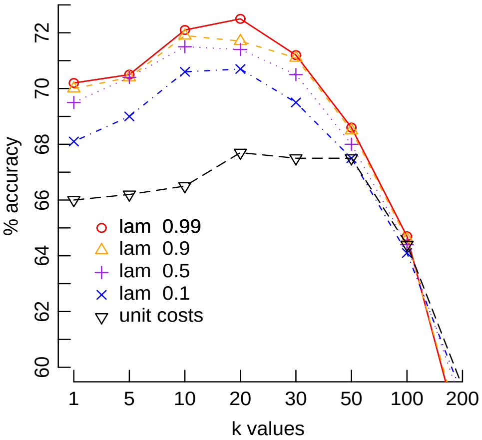
- a single 0 entry in C^{Δ}implies *infinite* cost entries everywhere else. $\Rightarrow$ a stochastically valid cost table cannot have zero costs on the diagonal
- perhaps this impedes good categorisation; note also the unit-cost setting, which is clearly 'uniform' in a sense, out-performs the 'uniform' stochastic initialisations
- suggests final step in which all the entries on the cost-table's diagonal are zeroed.
- Bilenko et al 2003 does essentially this in work on stochastic string distance

Experiment Two



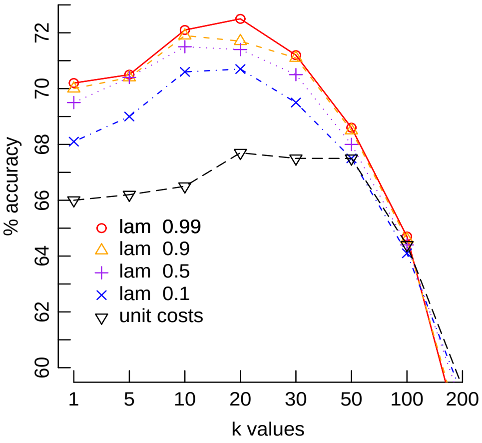
Experiment Two

- now with smoothing at varius levels of interpolation ($\lambda \in \{0.99, 0.9, 0.5, 0.1\}$) and with the diagonal zeroed, the EM^V-adapted costs clearly out-perform the unit-costs case ($\triangledown$).



Experiment Two

- now with smoothing at varius levels of interpolation ($\lambda \in \{0.99, 0.9, 0.5, 0.1\}$) and with the diagonal zeroed, the EM^V-adapted costs clearly out-perform the unit-costs case ($\bigtriangledown$).
- the best result being 72.5% ($k = 20$, $\lambda = 0.99$), as compared to 67.5% for unit-costs ($k = 20$)



Conclusions

Conclusions

- evidence to show that Viterbi EM cost-adaptation can increase the performance of a tree-distance based classifier, and improve it to above that attained in the unit-cost setting,

Conclusions

- evidence to show that Viterbi EM cost-adaptation can increase the performance of a tree-distance based classifier, and improve it to above that attained in the unit-cost setting,
- experiments on further data-sets is required: one possibility is the NLP-related tasks of question-answering, where the need is to assess pairs of sentences for their likelihood to be a question-answer pairs. A training set of such pairs could also serve as potential input to the cost adaptation algorithm.