

Finding Optimal Parameters for Edit Distance Based Sequence Classification is NP-Hard

Vlado Kešelj, Haibin Liu, Norbert Zeh,
Christian Blouin, and Chris Whidden

{vlado, haibin, nzeh, cblouin, whidden}@cs.dal.ca

Faculty of Computer Science, Dalhousie University

June 28, 2009

Outline

- 1 Introduction
- 2 Related Work
- 3 Background
 - Edit Distance
 - Parametric Edit Distance for Sequence Classification
- 4 OED-Class Problem
- 5 NP-hardness of OED-Class Problem
- 6 Conclusions and Future Work

Outline

- 1 Introduction
- 2 Related Work
- 3 Background
 - Edit Distance
 - Parametric Edit Distance for Sequence Classification
- 4 OED-Class Problem
- 5 NP-hardness of OED-Class Problem
- 6 Conclusions and Future Work

Introduction

- Motivational problem: Identifying interaction-describing sentences or passages in biomedical text
- Typical interactions: protein-protein, protein-DNA, gene regulations etc.
- First phase: extended POS tagging (BIO tag)
- Second phase: machine decision whether sentence is interaction or non-interaction
- The use of edit distance was proposed (Huang et al.)
- Similar to the use of edit distance in analysis of biological sequences

Outline

- 1 Introduction
- 2 Related Work**
- 3 Background
 - Edit Distance
 - Parametric Edit Distance for Sequence Classification
- 4 OED-Class Problem
- 5 NP-hardness of OED-Class Problem
- 6 Conclusions and Future Work

Related Work

- General reference: Ananiadou and Mcnaught — general book on text mining in biomedicine
- Detecting interaction patterns in text Jang et al., Skusa et al.
- BioCreative II challenge: extracton of interaction information
- Edit distance is well-studied topic in computer science
- Huang *et al.* propose use of patterns and edit distance
- Parameters for edit distance are heuristically determined

Outline

- 1 Introduction
- 2 Related Work
- 3 Background**
 - Edit Distance
 - Parametric Edit Distance for Sequence Classification
- 4 OED-Class Problem
- 5 NP-hardness of OED-Class Problem
- 6 Conclusions and Future Work

Edit Distance

- Similarity distance between two strings
- = minimal number of elementary edit operations to transform one string to another
- Levenshtein distance is one particular example
 - ▶ insert or delete of a letter
 - ▶ reversal of two adjacent letters
- Original motivaton: correction of transmission errors
- Efficiently calculated using dynamic programming: $O(mn)$ time

Edit Distance Variations

- Standard mathematical metric properties:

$$d(x, y) = 0 \Leftrightarrow x = y$$

$$d(x, y) = d(y, x)$$

$$d(x, y) + d(y, z) \geq d(x, z)$$

- Variations: not just operation count, but sum of operation costs depending on letters
- Even the same letter has a “matching” cost
- Common in analysis of biological sequences
- Downside: metric properties lost, in general

ED based Sequence Classification

- Sequence classification, e.g., sentence classification
- A sentence is either interactive or non-interactive
- Some preprocessing is applied: POS tagging, text chunking, etc.
- P is a set of typical interactive patterns
- I_S — interaction sentences
- N_S — non-interaction sentences

Examples of Tags

Tag name	Tag description	Tag type
<i>BIO</i>	Unified tag for biological entities	System
<i>NP</i>	Noun phrases	System
<i>VB</i>	Verbal unit	System
<i>IN</i>	Preposition, subordinating conjunction	POS
<i>CC</i>	Coordinating conjunction	POS
<i>TO</i>	to	POS
<i>PPC</i>	Punctuation comma	POS
<i>PRP</i>	Determiner of possessive second	POS
<i>DET</i>	Determiner	POS
<i>POS</i>	Possessive	POS

Illustrative Example

- Pattern example: *BIO VB IN BIO*
- Test sentence 1:
“Protein A interacts with protein B.”
tag representation: *BIO VB IN BIO*
- Test sentence 2:
“Studies of protein A identified protein B.”
tag representation: *NP IN BIO VB BIO*

Classification Algorithm

- Distance from test sentence to each pattern is calculated
- If there is a close pattern, the sentence is interaction sentence
- in other words

$$s \text{ is interaction sentence} \Leftrightarrow \min_{p \in P} d(s, p) < t \quad (1)$$

- where P is the set of patterns, and t is a given threshold

Edit Distance Based Classification

- Similar to kNN classification ($k = 1$)
- The class of interaction sentences s described as a union of spheres around patterns as centers
- Edit distance parameters:
 - ▶ $D(a)$ = deletion of letter a
 - ▶ $I(a)$ = insertion of letter a ,
 - ▶ $M(a, b)$ = replacement of a with b , including $a = b$
- Metric properties not necessarily satisfied (not even symmetricity)
- Convention: $d(s, p)$ = sum of costs of operations applied to sentence s to obtain pattern p

More About Parameters

- Negative costs are allowed, considered awards
- Sequence of operations with cost sum = $-\infty$ is not allowed
- Simplifying assumption: disjoint pattern and sentence alphabets
- Clear sets of deleted and inserted letters, and pairs of matched letters

Finding Optimal Parameters

If we are given a set I_s of interaction sentences, set N_s of non-interaction sentences, and set P of patterns, how can we determine optimal values for costs I , D , and M , as well as the threshold t , so that the classification accuracy is maximized, i.e., the number of sentences from $S = I_s \cup N_s$ that are correctly classified using formula (1) is maximal?

- parameters are also known as *Scoring Scheme*

Example of a Scoring Scheme

Tag name	Delete/Insert cost	Match cost	Mismatch cost
<i>BIO</i>	10	-8	3
<i>NP</i>	8	-6	3
<i>VB</i>	7	-7	3
<i>IN</i>	6	-5	1
<i>CC</i>	6	-5	1
<i>TO</i>	1	-5	1
<i>PPC</i>	1	-3	1
<i>PRP</i>	1	-3	1
<i>DET</i>	1	-3	1
<i>POS</i>	1	-3	1

Finding Optimal Parameters

- 1 Heuristically, experimentally, ad hoc
- 2 Greedy algorithms
- 3 Machine learning approaches
- 4 Genetic algorithms
- 5 Provably optimal solution?

Outline

- 1 Introduction
- 2 Related Work
- 3 Background
 - Edit Distance
 - Parametric Edit Distance for Sequence Classification
- 4 OED-Class Problem**
- 5 NP-hardness of OED-Class Problem
- 6 Conclusions and Future Work

OED-Class Problem

Find parameters (M, D, I, t) for a given six-tuple $(\Sigma_P, P, \Sigma_S, I_S, N_S, S)$, such that the number of correctly classified sentences from $S = I_S \cup N_S$ according to formula (1) is maximal.

Alternative decision problem: Is there a set of parameters (M, D, I, t) for a given $(\Sigma_P, P, \Sigma_S, I_S, N_S, S)$ which give a 100% accurate classification of the sentences S .

Outline

- 1 Introduction
- 2 Related Work
- 3 Background
 - Edit Distance
 - Parametric Edit Distance for Sequence Classification
- 4 OED-Class Problem
- 5 NP-hardness of OED-Class Problem**
- 6 Conclusions and Future Work

A Lemma

A note on notation: Adding a scalar c to a matrix of vector V , as in $V + c$, means adding scalar to each element of the matrix.

Lemma

If all patterns are of a fixed length n , and interactive and non-interactive sentences are of a fixed length k , where n and k are non-negative integers, then for any set of parameters (M, D, I, t) , and any constant c , the set of parameters $(M + 2c, D + c, I + c, t + nc + kc)$ will produce the same classification result in an OED-Class problem.

Proof of the Lemma

- If n_m , n_i , and n_d are the numbers of matches, inserts and deletes, than $2n_m + n_i + n_d$ is always equal $n + k$.
- Therefore, for any distance d between a pattern and a sentence using parameters (M,D,I,t) , the corresponding distance with the parameter set $(M + 2c, D + c, I + c, t + nc + kc)$ will be $d + n_m \cdot 2c + n_d c + n_i c = d + nc + kc$, so the classification will be the same according to the threshold $t + nc + kc$.

Corollary

- If all patterns and sentences are of the same length, then we can assume that all parameters are positive, since we can always choose sufficiently large constant c .

Claim

The OED-Class problem is NP-hard.

- Prove by reduction from 3-SAT problem
- 3-SAT is a satisfiability problem of a formula

$$\bigwedge_{i=1}^n (x_i \vee y_i \vee z_i),$$

- where each x_i , y_i , and z_i is a literal, and
- Each literal is either a variable p_j , or a negation of a variable $\neg p_j$, from a set of variables $\{p_j\}_{j=1}^k$
- Design OED-Class problem whose solution would give a 3-SAT solution

- A 3-SAT problem is $\mathcal{F} = \bigwedge_{i=1}^n (x_i \vee y_i \vee z_i)$
- Define OED-Class problem:
 - ▶ Pattern alphabet is $\Sigma_P = \{ B, E, t, f \}$,
 - ▶ Pattern set is $P = \{ BtffffE, BffttffE, BffffttE, BttttffE, BtfffttE, BffttttE, BttttttE \}$,
 - ▶ Sentence alphabet is $\Sigma_S = \{ B, E \} \cup \{ p, P \mid \text{for all variables } p \text{ in } \mathcal{F} \}$.
 - ▶ For any literal x , we define $\bar{x} = p$ if $x = p$, or $\bar{x} = P$ if $x = \neg p$ for a variable p .
 - ▶ Interactive sentences are $I_S = \{ B\bar{x}\bar{x}\bar{y}\bar{y}\bar{z}\bar{z}E \mid \text{for each clause } x \vee y \vee z \text{ in } \mathcal{F} \}$, and
 - ▶ Non-interactive sentences are $N_S = \{ BpPpPpPE \mid \text{for all variables } p \} \cup \{ B\bar{x}\bar{x}\bar{y}\bar{y}\bar{z}\bar{z}E\bar{z}, B\bar{x}\bar{x}\bar{y}\bar{z}\bar{z}E\bar{y}, B\bar{x}\bar{y}\bar{y}\bar{z}\bar{z}E\bar{x}, \bar{x}\bar{x}\bar{y}\bar{y}\bar{z}\bar{z}EB, EB\bar{x}\bar{x}\bar{y}\bar{y}\bar{z}\bar{z} \mid \text{for each clause } x \vee y \vee z \text{ in } \mathcal{F} \}$.

3-SAT \Rightarrow 100% acc. in OED-Class

- $p = \top \Rightarrow M(p, \top) = 0, M(\mathbb{P}, \top) = 1,$
 $M(p, \perp) = 1, M(\mathbb{P}, \perp) = 0$
- otherwise, for $p = \perp \Rightarrow M(p, \top) = 1,$
 $M(\mathbb{P}, \top) = 0, M(p, \perp) = 0, M(\mathbb{P}, \perp) = 1.$
- The letters \mathbb{B} and \mathbb{E} match only themselves with cost 0, otherwise the cost is 1.
- All insertions and deletions have cost 1, and the threshold t is 1.
- Interactive sentences are at distance 0 from their truth-value pattern.

- Non-interactive sentences not accepted since $BpPpPpPE$ has distance 3 from any pattern, and the distance of other non-interactive sentences, which do not start with B or do not end with E , will be larger than 1 since all patterns start with a B and end with an E .

100% acc. in OED-Class \Rightarrow 3-SAT

- 100% classification accuracy is achievable for some parameters.
- All patterns and sentences have the same length, so assume that parameters are positive
- Non-interactive sentences $B_p P_p P_p P E$ are rejected, hence at most one of parameters $M(p, t)$ or $M(P, t)$ is $\leq t_1/6$, where $t_1 = t - M(B, B) - M(E, E)$.
- If $M(p, t) \leq t_1/6$ we assign $p = \top$, if $M(P, t) \leq t_1/6$, we assign $p = \perp$ (false), and otherwise we can assign either true or false to p .

- For each clause in formula \mathcal{F} , the corresponding interactive sentence is accepted by one of the patterns.
- Only matchings were used; otherwise, if there were any insertions or deletions, then one of the characters that were inserted could be inserted last (or first if it is \mathbb{E}), and one of the non-interactive sentences that does not start with \mathbb{B} and ends with \mathbb{E} would be accepted.
- This implies that at least for one of the literals x in the clause, which is matched with a \mathfrak{t} in the pattern, we must have $M(\bar{x}, \mathfrak{t}) \leq t_1/6$, which implies that $p = \top$ if $x = p$, or $p = \perp$ if $x = \neg p$.

Outline

- 1 Introduction
- 2 Related Work
- 3 Background
 - Edit Distance
 - Parametric Edit Distance for Sequence Classification
- 4 OED-Class Problem
- 5 NP-hardness of OED-Class Problem
- 6 **Conclusions and Future Work**

Conclusions

- OED-Class problem is NP-hard
- Use of heuristic methods is justified, such as heuristic search and GA
- One-letter problem can be solved in $O(n \log n)$
- What about more, but limited number of letters, e.g. four ($\{ A, C, G, T \}$)