More Speed and More Pattern Variations for Knowledge Discovery System BONSAI

Hideo Bannai [1] bannai@ims.u-tokyo.ac.jp
Keisuke Iida [2] k-ihda@i.kyushu-u.ac.jp
Ayumi Shinohara [2] ayumi@i.kyushu-u.ac.jp
Masayuki Takeda [2] takeda@i.kyushu-u.ac.jp
Satoru Miyano [1] miyano@ims.u-tokyo.ac.jp

[1] Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan
[2] Department of Informatics, Kyushu University 33, Fukuoka 812-8581, Japan

Keywords: pattern matching, knowledge discovery, decision tree, alphabet indexing

1 Introduction

BONSAI is a machine learning system for knowledge acquisition from positive and negative examples of strings [3]. A hypothesis generated by the system is a pair of a classification of symbols called an *alphabet indexing*, and a *decision tree over regular patterns*, which classifies given examples (strings) to either positive or negative. The algorithm of the system consists of two parts: a learning algorithm for constructing a decision tree over regular patterns, and a local search algorithm for finding a good alphabet indexing for the production of the decision tree. Our focus here is in the improvement of the former, increasing both the speed of hypothesis construction, and the descriptional strength of the generated hypotheses.

It has been reported that the system has discovered knowledge which can classify amino acid sequences of trans-membrane domains and randomly chosen amino acid sequences located in other parts of the PIR database, with over 90% accuracy [3]. However, in the current implementation, only substring patterns (i.e. whether or not a string pattern appears as a substring of the data string) are searched for, and such patterns may not be powerful enough for distinguishing between positive and negative data of a more complex nature. In this paper, we present a new version of the BONSAI system which implements several, more powerful variations of patterns, namely, *subsequence patterns*, *episode patterns*, and *approximate patterns* [1, 4, 2]. We also implement an efficient branch-and-bound algorithm for finding the best pattern which distinguishes between the positive and negative data sets [1].

2 Pattern Variations

Let Σ be a finite *alphabet* and let Σ^* be the set of all *strings* over Σ. For a string w, let $|w|$ denote the length of w. A string $w = w_1 \cdots w_p \in \Sigma^*$ is a *substring* of a string $t = t_1 \cdots t_n \in \Sigma^*$ if there exists $1 \leq i \leq (n - p + 1)$ such that $w_j = t_{i+j-1}$ for $1 \leq j \leq p$. A string $w = w_1 \cdots w_p \in \Sigma^*$ is a *subsequence* of a given string $t = t_1 \cdots t_n \in \Sigma^*$ if there exists $q_1, \ldots, q_p$ $(1 \leq q_1 < \cdots < q_n \leq n)$ such that $w_i = t_{q_i}$ for all $1 \leq i \leq p$. e.g.: *abba* is a substring of *abaaabbaa*. *abbbb* is a subsequence of *abaaabbab*.

Definition 1 (Substring Pattern) A *substring pattern* is a string $w \in \Sigma^*$. A substring pattern *matches* a given string $t \in \Sigma^*$ if w is a substring of t. □

Definition 2 (Subsequence Pattern) A *subsequence pattern* is a string $w \in \Sigma^*$. A subsequence pattern *matches* a given string $t \in \Sigma^*$ if w is a subsequence of t. □

Definition 3 (Episode Pattern) An *episode pattern* is a pair (w, l) where $w \in \Sigma^*$, and l is a non-negative integer ($l \geq |w|$). An episode pattern (w, l) *matches* a given string $t \in \Sigma^*$ if there exists a substring v of t, where $|v| \leq l$ and w is a subsequence of v. (e.g.: $(banzai, 8)$ will match *ban*b*ona*n*zai*, whereas, $(bannai, 8)$ will not.) □

Definition 4 (Approximate Pattern) An *approximate pattern* is a triplet (w, k, F) where $w \in \Sigma^*$ is a string, k is a non-negative integer, and $F \subseteq \{\text{insertion}, \text{deletion}, \text{substitution}\}$. An approximate pattern (w, k, F) *matches* a given string $t \in \Sigma^*$ if a substring of t can be made from w with k or less transformations contained in F. (e.g.: (*b*a*n*n*ai*, 2, {substitution}) matches *b*o*n*s*ai*, but does not match *banana*, whereas, (*bannai*, 2, {insertion, substitution}) will match *ban*a*na*. □

Efficient matching algorithms for the pattern matching of each of the pattern variations can be found in [1, 4, 2].

3 Efficient Search

For each node in the decision tree, the pattern which best distinguishes between the positive and negative examples, in terms of matches, is searched for: i.e. a pattern matches most of the positive examples, but does not match most of the negative examples, or vice versa, is desired. All pattern variations we consider satisfy the condition of [1], that is, for a pattern of some variation based on the string $w \in \Sigma^*$, a pattern based on any longer string containing w results in a smaller number of matches against a given set of strings. For such patterns and a conic score function, an upper bound of the score for the longer string may be calculated, and the search can be pruned if the upper bound is less than the current maximum score. For episode patterns, the algorithm of [2] is used to efficiently find the optimal threshold l at the same time. A similar algorithm is also applicable for finding a suboptimal mismatch number k in approximate patterns, and is implemented.

4 Conclusion

The new BONSAI system has been implemented in the Objective Caml language [5], a simple but powerful functional language. The source code for BONSAI will be available and distributed at http://biocaml.org/bonsai/, under the GNU General Public License.

References

[1] Hirao, M., Hoshino, H., Shinohara, A., Takeda, M., and Arikawa, S., A practical algorithm to find the best subsequences patterns, *Theoretical Computer Science*, 2001, to appear.

[2] Hirao, M., Inenaga, S., Shinohara, A., Takeda, M., and Arikawa., S., A practical algorithm to find the best episode patterns. *Proceedings of the Fourth International Conference on Discovery Science (DS2001)*, LNAI 2226. Springer-Verlag, 2001, to appear.

[3] Shimozono, S., Shinohara, A., Shinohara, T., Miyano, S., Kuhara, S., and Arikawa, S., Knowledge acquisition from amino acid sequences by machine learning system BONSAI, *Trans. Information Processing Society of Japan*, 35(10):2009–2018, 1994.

[4] Wu, S. and Manber, U., Fast text searching allowing errors, *Commun. ACM*, 35:83–91, 1992.

[5] Objective Caml - http://www.ocaml.org/.