ALGORITHMS FOR DATA FLOWS1

L. Aslanyan, J. Castellanos, F. Mingo, H. Sahakyan, V. Ryazanov

Abstract: Data analysis is a regular massif task of applied sciences and businesses. A huge number of algorithms were developed for different kinds of data and for particular types of data analysis. Traditional theories work with traditional databases and data structures, although the paradigm of Internet doesn't want to wait, requiring novel technologies, able to work effectively with huge amounts of data, with data flows and uncertainties. The two current research projects, INTAS 397 and 626 are devoted to development of these issues. The paper gives the general statement, current results and examples of these researches.

Keywords: Data analysis, data flows, complexity, logic separation.

Introduction

The novel issues of advanced knowledge-based data analysis algorithmic frameworks are under the development. Traditional theories, such as pattern recognition and combinatorial algorithms, usually work with the given input data sets, delivering the appropriate knowledge in conditions of optimized computational resources – mostly the time and memory [Zhur, 1978]. The nowadays work environment often is the distributed databases and networks. Here the concept "input data set" is dynamic. Where appropriate, data might arrive by portions, with delays and uncertainties. Some algorithms, e.g. statistical, can treat this data, delivering the required estimates of parameters. But the main procedures of intelligent data analysis, which are known in mathematical statistics, pattern recognition theory, logic deduction theories and experimental and theoretical heuristics, are to be revised for the needs of these new conditions:

- Large and increasing input data amounts and dynamic flows (data warehouses, networks, uncertainties),
- Cross-cooperating algorithms and information theories, - converging to the general reference model of intelligent data analysis and data mining,
- The complementary group of tasks – development of newly theoretical postulations (distributed intelligence, accumulative (additive) knowledge scheme, complexity and approximation of algorithms), practical realization (searches as WEB surf, Internet automation, diverse business models over the Internet).

The practical part of research is in developing prototype systems with real applications, particularly, for e-business management and for intrusion detection problems [Pasic, 2000]. The technological components used include mobile and intelligent software agents, neural nets, logic-combinatorial and algebraic recognition, association rules induction and heuristics harmonization. The result is a convergence of technologies into the hybrid technologies of data analysis in conditions, mentioned above: – consequent input data flows by complementary portions, data sizes are very huge, the networked distributed and intelligent analysis provides the new and powerful business mechanisms [Asl, 2001, Gim, 2001]. Algorithm Incubator is one more description of research result which means theories and technologies to near-to-market state, providing ultimately leading edge technologies for wealth creation, thereby advancing cross-European competitiveness.

The described problem is studied for several particular models. A good description of these results is surveyed in [Prov, 2000] which discussed the possibility of scaling of inductive algorithms. In part these studies are related to data mining algorithms but some particular cases are also important, e.g. the work of algorithms analysing huge data in a limited operational memory space. Without going deep into the comparisons it is sufficient to note that our case is related to the situation, when some data is available, more data are possible but uncertain, and we have to have a temporal data analysis result in any time ready to be outputted by request.

[1] The research was supported by INTAS 00-397 and 00-626 Projects.

General Notes and the Simplest Examples

The simplest example of algorithms, working effectively with data flows is e.g., calculation of main values by the series. Given two blocks of numbers: $\alpha_1, \alpha_2, ..., \alpha_m$ and $\delta_1, \delta_2, ..., \delta_n$. Then the overall main value equals $M = \frac{\sum \alpha_i + \sum \delta_j}{m+n}$. If blocks are arriving consecuently, then we may have the main value $M_\alpha = \frac{\sum \alpha_i}{m}$ for the first block in arrival of the second block. So the calculation $M_\delta = \frac{\sum \delta_j}{n}$, and then $M = \frac{mM_\alpha + nM_\delta}{m+n}$ might be preferable of the direct calculation of M. Besides the computational counterpart, it is important, that in some cases and in some applications it is not known or certain, that after $\alpha_1, \alpha_2, ..., \alpha_m$ the new data block will arrive. This is why we suppose M_α calculated in arrival of the next block.

More meaningful is the following cluster analysis scheme [Duda, 1973]. Given a set $\Xi = \{x_1, x_2, ..., x_n\}$ of numbers (or vectors), the question is in partitioning it into the c clusters, minimizing the sum of square errors (distances) J_i in clusters. Let us consider a temporal (current) partitioning: $\Xi_1, \Xi_2, ..., \Xi_c$ and the reallocation scheme of search of a local optimum by the given criterion. If an element $\hat{x} \in \Xi_j$ is reallocated to the Ξ_i then the value of the optimization criteria in this part equals to $\sum_{x \in \Xi_i} \left\| x - m_i + \frac{\hat{x} - m_i}{n_i + 1} \right\|^2 + \left\| \frac{n_i}{n_i + 1} (\hat{x} - m) \right\|^2$. The reallocation strategy is based on consolidation of differences of these changes, which may use the simplified calculation $J_i + \frac{n_i}{n_i + 1} \| \hat{x} - m_i \|^2$, where $n_i = |\Xi_i|$ and m_i is the main value in Ξ_i [Duda, 1973]. Similar and still simple are the formulas of reallocating of a group of elements with elements – numbers or vectors. This algorithm also fits with the input data flow scheme, when new arrived data are to be integrated into a current partitioning (reallocation from outside). The idea is in easy update of the current (temporal) results and constructions, instead of analysing the whole input data set again, in each iteration. It is not evident that there are many algorithms suitable for this re-engineering strategy. For example, the discrete isoperimetry problem and its solutions are a specific case, not fully extendable in this sense.

Let E^n be the n-dimensional unit cube: $E^n = \{(x_1, x_2, \cdots, x_n) / x_i \in \{0,1\}, i = \overline{1,n}\}$, and let $A \subseteq E^n$. We say that $x \in A$ is an interior point of A, if $S_1^n(x) \subseteq A$, where $S_1^n(x)$ is the unit Hamming sphere of E^n, centered at x. We denote by a the size of A, and by $I(A)$ - the set of all interior points of A. A obeys the isoperimetry property, if $|I(A)| = \max_{B \subseteq E^n, |B| = a} |I(B)|$.

Let us agree to omit the trivial cases $a = 0$ and $a = 2^n$ and then $a = \sum_{t=0}^{k} C_n^t + \delta$, for some k and δ, where $0 \le k < n$ and $0 \le \delta < C_n^{k+1}$. Define the standard placement $L^n = \{x_1, x_2, ..., x_{2^n}\}$ of vertices, in the following way: for any vertices α and β of E^n $\alpha \prec \beta$ in L^n, iff

1) $\|\alpha\| < \|\beta\|$, or

2) $\|\alpha\| = \|\beta\|$ and α lexicographically precedes β (here $1 \prec 0$).

Let L_a^n denotes the initial a-segments of L^n. It is well known [Asl, 1979] that for a prefixed size a the L_a^n is a solution of the discrete isoperimetry problem. It is also known, that the arbitrary solutions are semi-spherical subsets, which are the slight modifications of the structure L_a^n [Asl, 1979]. So the algorithm of discrete isoperimetry is extendable for the basic solution through L^n, and can't be extendable in all cases, because of the sum of semi-spheres is not a semi-sphere [Asl, 1979]. That is, when we are given numbers a_1 and a_2, then the fusion of solutions for a_1 and a_2 into the solution for $a_1 + a_2$ is not evident. The same time, the direct construction of the basic solution for $a_1 + a_2$ is as simple, as the construction of the structure L_a^n.

Logical Separation

Let us follow by the Logical Separation (LS) pattern recognition model [Asl, 1976], which based on implementation of several logically expressed suppositions above the elements of the learning sets. These are some formalisms or additional properties (bias) of classification, expressed in terms of Boolean functions and especially – of the Reduced Disjunctive Normal Forms (RDNF) of Boolean functions.

Let us consider a set of logical variables (binary properties) $\chi_1, \chi_2, \cdots, \chi_n$, and the case of two disjoint classes K_1 and K_2. Let $\beta \in K_1$, $\gamma \in K_2$ and let α is an unknown object in the sense of classification. We say that γ is separated by the information of β for α if $\beta \oplus \gamma \leq \beta \oplus \alpha$ where $\oplus$ the bit-vice mod2 summation is. In simple words, this means that the information difference between β and α is larger than of β and γ (the first includes the second). As a consequence of this assumption we get, that the RDNFs of the pairs of complementary partially defined Boolean functions describe the complete structure of information enlargements, started from the learning set.

Let $L \subseteq E^n$ be the learning set; $L_1 = L \cap K_1 = \{\beta_1, ..., \beta_{l(n)}\}$ and $L_2 = L \cap K_2 = \{\gamma_1, ..., \gamma_{k(n)}\}$. Several structures serve the work of LS algorithms. Let us denote by $\mathrm{N}(\beta_i, L_2)$ the set of all maximal subcubes, containing β_i and - out of the set L_2. $\mathrm{N}(\gamma_i, L_1)$ is the similar structure for the second class. If f is the partial Boolean function, determined as "one" in L_1 and "zero" in L_2, then $N_1 = \bigcup_i \mathrm{N}(\beta_i, L_2)$ (in fact – it's simplification) is the RDNF of f. In terms of supervised classification N_1 is the area of the possible extensions of the learning subset L_1. The intersection $N_1 \cap N_2$ is logically reachable from both classes. These are the base structures of LS, which are related to different theoretical and applied problems.

In theories LS is the instrument for optimization [AslZhur, 2001]. It is given a partial Boolean function f. f might be evaluated arbitrarily on the area of indetermination. The aim is to minimize the complexity of the targeted extension function.

In pattern recognition LS is a powerful addition to the "compactness hypotheses"-based models [Zhur, 1978, Asl, 1979]. It is important the role of LS in advanced data mining solutions. LS is an universal interpreter of IREP class of data mining algorithms. While IREP is an approximate scheme through the rules growing and pruning stages, the LS is the exact construction of class separations by rules. We do not need to go now deep into the comparison of these models and it is to check whether the LS constructions are well suited to the data flow analysis algorithms.

The key structure is the set construction (bunch of subcubes) $\mathrm{N}(\beta_i, L_2)$ for an arbitrarily fixed β_i and for L_2. The question is in constructing the $\mathrm{N}(\beta_i, L_2)$ in conditions, when we are given the L_2 in the consecutive blocks $L_2^{'} + L_2^{''}$. The use of traditional algorithms of synthesis of RDNFs is one way. The complexity is evidently high for these algorithms, because of they are of growing type – they are going from vertices to subcubes. The natural way is the pronging approach. This is effective to apply consequently by the input vertices (L_2), starting from E^n. This is a monotone process of breaking E^n, and then - the consecutive results of pronging. When $\mathrm{N}(\beta_i, L_2^{'})$ and $\mathrm{N}(\beta_i, L_2^{''})$ are constructed, then their fusion - into the $\mathrm{N}(\beta_i, L_2)$ is through the intersection of pairs of subcubes from $\mathrm{N}(\beta_i, L_2^{'})$ and $\mathrm{N}(\beta_i, L_2^{''})$. If $S^{'} \in \mathrm{N}(\beta_i, L_2^{'})$ and $S^{''} \in \mathrm{N}(\beta_i, L_2^{''})$, then $S = S^{'} \cap S^{''}$ belongs to the $\mathrm{N}(\beta_i, L_2)$. This is evident, because of $\beta_i \in S$ and S is out of $L_2^{'} + L_2^{''}$.

Now, let S be an arbitrary subcube of $\mathrm{N}(\beta_i, L_2)$. First, it is clear that $\beta_i \in S$. Due to the mentioned monotony, the intersections $S \bigcap \mathrm{N}(\beta_i, L_2^{'})$ and $S \bigcap \mathrm{N}(\beta_i, L_2^{''})$ are none empty. Let $\delta \in S$ is the vertex, opposite to β_i δ must be covered by these bunches so that there exist $S^{'} \in \mathrm{N}(\beta_i, L_2^{'})$ and $S^{''} \in \mathrm{N}(\beta_i, L_2^{''})$ containing both β_i and δ. The subset, containing both - β_i and δ have to contain the whole S. This intersection gives the proof of the statement.

The conclusion of the above paragraph is that the $\mathrm{N}(\beta_i, L_2)$ may be constructed effectively through the intersections of pairs of elements of $\mathrm{N}(\beta_i, L_2^{'})$ and $\mathrm{N}(\beta_i, L_2^{''})$.

Conclusion

Data flows, as a special case of input information formation, are known and/but re-estimated by the appearance of Internet and other distributed databases. Data analysis algorithms, mainly constructed for use with fixed data sets, are to be revised for the new conditions. Some of the well known algorithms are easy to apply to data flows. Others are conditional or not optimal. The data mining algorithms, as the approximate branch of the Logical Separation pattern recognition model, are a special case of studies. The huge data sets imply to complex calculations, which is unavoidable, partially. The way is in systematization and it was demonstrated, that the general scheme of Logical Separation is extendable.

Bibliography

[Duda, 1973] C R. O. Duda, P. E. Hart, Pattern classification and scene analysis, Wiley, New York, (1973).

[Asl, 1976] L. H. Aslanyan, The pattern recognition algorithms with logical separators, in “Collection of works on Theoretical Cybernetics”, Computer Center of USSR Academy of Sciences, Moscow, (1976), 116-131.

[Zhur, 1978] Yu. I. Zhuravlev, On an algorithmic approach to the problems of recognition and classification, Problemi Kibernetiki, 33, (1978), 5-68.

[Asl, 1979] L. H. Aslanyan, The discrete isoperimetric problem and the related extremal problems in discrete spaces, Problemy Kibernetiki, 36, Moscow, (1979), 85-128.

[Pasic, 2000] Pasic A., Kulin A., Salmonsen G. and Aslanyan L., Security Policy Adaptation Reinforced Through Agents, International Seminar “Conversion Potential of Armenia and ISTC Programs”, 2-7 October, (2000), Yerevan.

[Prov, 2000] Provost F., Kolluri V., A survey of methods for scaling up inductive algorithms. Kluwer Academic Publishers, Boston, pp. 1-42.

[AslZhur, 2001] Aslanyan L. and Zhuravlev Yu., Logic Separation Principle, CSIT Conference, Yerevan, September 17-20, (2001), 151-156.

[Asl, 2001] Aslanyan L., Castellanos J., Georgiou P., Tsagas G. and Riazanov V., Concurrent heuristics in data analysis and prediction, Pattern Recognition and Image Analysis, vol. 11, No. 1, (2001), 8-10.

[Gim, 2001] Gimenez-Martinez V., Aslanyan L., Castellanos J., Riazanov V., Distribution Function as Attractors for Recurrent Neural Networks, Pattern recognition and image analysis, vol. 11, No. 3, (2001), 492-497.

Author information

Levon Aslanyan – Institute for Informatics and Automation Problems, NAS Armenia; P. Sevak 1 Str., Yerevan-14, Armenia; e-mail: lasl@sci.am

Juan Castellanos - Departamento de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo s/n, Boadilla del Monte, 28660, Madrid, jcastellanos@fi.upm.es

Fernando Mingo - Departamento de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo s/n, Boadilla del Monte, 28660, Madrid

Hasmik Sahakyan - Institute for Informatics and Automation Problems, NAS Armenia; P. Sevak 1 Str., Yerevan-14, Armenia; e-mail: hasmik@ipia.sci.am

Vladimir Ryazanov - *Computer Center of Russian Academy of Sciences, Vavilov, 40, Moscow, rvv@ccas.ru*