Combine Vector Quantization and Support Vector Machine for Imbalanced Datasets

Ting Yu, John Debenham, Tony Jan and Simeon Simoff
Institute for Information and Communication Technologies
Faculty of Information Technology, University of Technology, Sydney,
PO Box 123, Broadway, NSW 2007, Australia
Capital Markets Cooperative Research Centre, Australia
{yuting, debenham, jant, simeon}@it.uts.edu.au

Abstract. In cases of extremely imbalanced dataset with high dimensions, standard machine learning techniques tend to be overwhelmed by the large classes. This paper rebalances skewed datasets by compressing the majority class. This approach combines Vector Quantization and Support Vector Machine and constructs a new approach, VQ-SVM, to rebalance datasets without significant information loss. Some issues, e.g. distortion and support vectors, have been discussed to address the trade-off between the information loss and undersampling. Experiments compare VQ-SVM and standard SVM on some imbalanced datasets with varied imbalance ratios, and results show that the performance of VQ-SVM is superior to SVM, especially in case of extremely imbalanced large datasets.

1 Introduction

The class imbalance problem typically occurs when, in classification problem, there are many more instances of some classes than other. In cases of extremely imbalanced (skewed) dataset with high dimensions, standard classifier tends to be overwhelmed by the large classes and ignore the small ones. Therefore, machine learning becomes an extremely difficult task, and performances of normal machine learning techniques decline dramatically. In practical applications, the ratio of the small to the large classes can be drastic such as 1 to 100, or 1 to 1000 [1] .

The recent Sigkdd explorations published a special issue on learning from imbalanced data sets [1], which summarized some well-known methods for dealing with problems with the imbalanced data: at the data level, undersampling and oversampling; at the algorithms level, one-class learning (cost-sensitive learning) and boasting etc.

Random undersampling can potentially remove certain important examples, and random oversampling can lead to overfitting. In addition, oversampling can introduce an additional computational task if the data set is already fairly large but imbalance.

A few researches of combining data compression techniques and machine learning have been done: Jiaqi Wang el at [2] combines K-means clustering and SVM to speed up the real-time learning; Smola el at [3] discussed the combination between VQ and SVM in their book.

At the algorithms level, for SVMs, cost-sensitive learning [5, 6, 7] aims to incorporate into the SVMs the prior knowledge of the risk factors of false positives and false negatives. Gang Wu el at [8] implemented KBA, Kernel Boundary Alignment to imbalanced datasets. Rehan Akbani el at [4] implemented SMOTE, a derivative of Support Vector Machine, to imbalanced datasets and discussed the drawbacks of random undersampling. Being different from the random undersampling, VQ compresses datasets by clustering them instead of simply eliminating instances.

2. Support Vector Machine

Support Vector Machine and other kernel methods were maturated and implemented broadly in 1990s, after Vapnik [9]. Support Vector Machine transforms (approximates) the nonlinear problem within a lower dimension space (input space) into a linear problem within a higher dimension space (feature space). Within this linear feature space, SVM could be treated as a linear learning machine, which finds a maximum margin hyper-plane to separate the given data with some tolerance (slack variables) to the noise. Vapnik-Chervonenkis (VC) dimension restricts the degree of approximations (generalization).

Decision Function of support vector classification (pattern reorganization):

$$f(x) = \operatorname{sgn}(\sum_{i=1}^{m} y_i \alpha_i \left\langle \Phi(x), \Phi(x_i) \right\rangle + b) = \operatorname{sgn}(\sum_{i=1}^{m} y_i \alpha_i k(x, x_i) + b) \qquad (1)$$

and the following quadratic program:

$$\max_{\alpha \in R} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \qquad (2)$$

subject to $\alpha_i \geq 0$ for all I=1, …, m, and $\sum_{i=1}^{m} \alpha_i y_i = 0$

3. Vector Quantization

Vector quantization (VQ) is a lossy data compression method based on the principle of block coding. According to Shannon's theory of data compression, lossy data compression, better known as rate-distortion theory, the decompressed data does not have to be exactly the same as the original data. Instead, some amount of distortion, D, is tolerated. Moreover the lossless compression is no distortion, i.e. D=0.

In 1980, Linde, Buzo, and Gray (LBG) [10] proposed a VQ design algorithm based on a training sequence. The use of a training sequence bypasses the need for multi-dimensional integration required by previous VQ methods. A VQ that is designed using this algorithm are referred to in the literature as an LBG-VQ, which can be stated as follows. Given a vector source with its statistical properties known, given a distortion measure, and given the number of codevectors, find a codebook (the set of all red stars) and a partition (the set of blue lines) which result in the smallest average distortion [11].

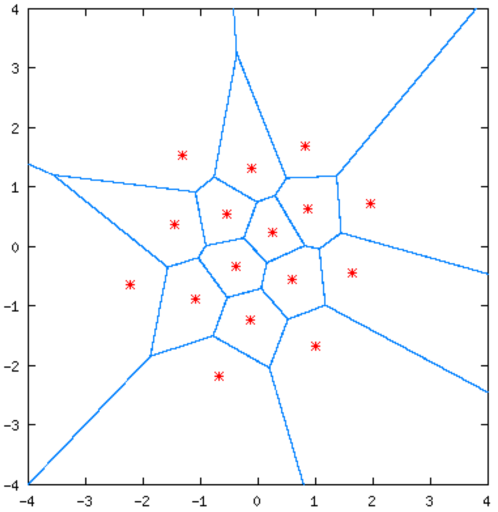

Fig. 1. A simple example of two-dimensional LBG-VQ [11]

Suppose a training sequence consisting of M vectors: $T=\{x_1,x_2,...,x_M\}$, and N codevectors $C=\{c_1,c_2,...,c_N\}$, then the whole region is partitioned by the codevectors into a set of sub-regions, so-called "Voronoi Region", $P=\{S_1,S_2,...,S_N\}$. Vectors within a region S_n are represented by their codevector $Q(x_m)=c_n$ if $x_m \in S_n$, and the average distortion can be given by:

$$D_{ave}=\frac{1}{Mk}\sum_{m=1}^{M}\|\mathrm{x}_{\mathrm{m}}-Q(\mathrm{x}_{\mathrm{m}})\|$$

which measures the information loss. Thus, the design problem can be stated as: to find C and P such that D is minimized.

If C ad P are a solution to the above minimization problem, then is must satisfy two criteria: Nearest Neighbour Condition, $S_n=\{x:\|x-c_n\|^2\le\|x-c_{n'}\|^2 \ \forall n'=1,2,..N\}$, and Centroid Condition:

$$c_n=\frac{\sum_{x_m\in S_n}x_m}{\sum_{x_m\in S_n}1}\text{, n=1, 2, .., N.}$$

The LBG VQ design algorithm is an iterative algorithm, which alternatively solves the above two optimality criteria. The algorithm requires an initial codebook. This initial codebook is obtained by the *splitting* method. In this method, an initial codevector is set as the average of the entire training sequence. This codevector is then split into two. The iterative algorithm is run with these two vectors as the initial codebook. The final two codevectors are splitted into four and the process is repeated until the desired number of codevectors is obtained [11].

4. VQ-SVM to Rebalance Dataset

VQ-SVM combines Vector Quantization and Support Vector Machine for dealing with extreme imbalance datasets, in which standard Support Vector Machine losses its accuracy dramatically. Here Vector Quantization could be treated as another way for incorporating domain knowledge into Support Vector Machine. In this case, the domain knowledge is the imbalance ratio and distribution of the majority group. Similar research can be found at authors' other papers [12, 13].

Pseudo-code of the algorithm VQ-SVM:

```
/* Step 1: Set parameters of VQ-SVM */
Float g;  //the kernel parameter g
Int number_of_undersampling;  // the number of code-vectors

/* LBG-VQ compresses the majority group and reduces the number of instances down to the given number, and then the new group and the minority group are combined to construct a balanced training dataset */
Balanced_Majority = LBGvq(Majority, number_of_undersampling);
New_training_data = combine(Balanced_Majority, Minority);

/* SVM based on the new balanced data*/
Model = SVM (New_training_data, g);
```

Under-sampling the frequency of the majority class, e.g. random undersampling, has its drawbacks and results in information loss. Support Vector Machine selects a subset of instances along the hyper-plane, so-called support vectors, and used them as the set of x_i within the decision function (1). These support vectors lie within the margin, and their α_i s are non-zero, $0 < \alpha_i < C$. That is: as the hyperplane is completely determined by the instances closest to it, the solution should no depend on the other examples [3].

The random undersampling inevitably reduces the number of support vectors, and thus potentially losses information with these removed support vectors. According to the theory of data compression, vector quantization is superior to random undersampling in term of the information loss, but both of them suffer from another risk of information loss within the majority group: Vector Quantization replaces some original SVs by their corresponding codevectors, which become new SVs and push the optimal hyperplane away from the original one trained by imbalanced data (cf. figure 2). Rehan Akbani el at [4] and Gang Wu el at [8] found that in case of imbalanced dataset, SVM always pushes the hyperplane towards minority group, which causes that the learning machine is overwhelmed by the majority group and minority group losses its information completely.

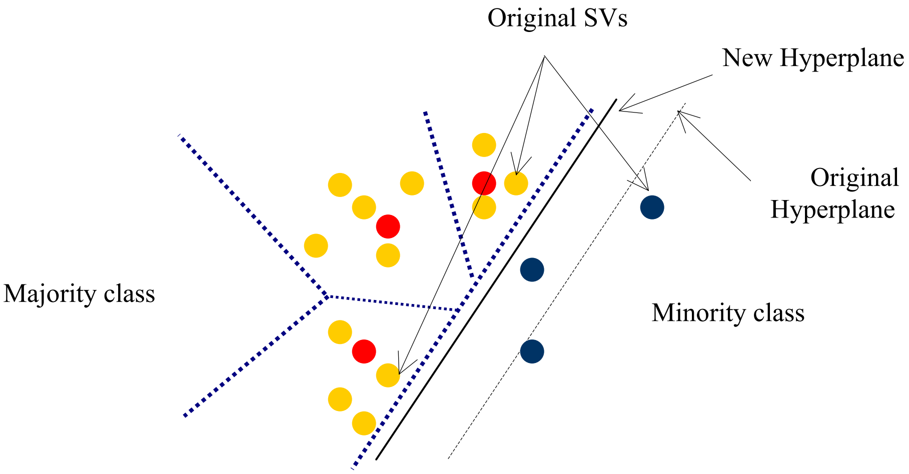

Fig. 2. VQ replaces the observations (yellow points) of majority group by codevectors (red points), the number of which is more equal to the number of observations (grey points) of the minority group. However, the imbalanced maximum margin hyperplane (dashed line) is pushed towards a new position (solid line), which is much closer to the majority group.

Throughout reducing the number of SVs of the majority group, VQ-SVM pulls the biased hyperplane away from the minority group. That is more close to the underlying "real" boundary.

VQ-SVM sacrifices the information held by the majority group to retrieval the information contained by the minority group. This is very important in many real life practices, which focus on the minority group. On the other hand, the compression ratio is tuned by the VQ-SVM to minimize the information loss of majority group. Therefore the optimal model is a trade-off between the compression rate and improved data balance, classification accuracy (i.e. g-means).

5. Experiments

For this evaluation, we used four UCI datasets. Those UCI datasets we experimented with are *abalone* (abalone19), *yeast* (yeast5) *glass* (g7), and letter (letter26). The number in the parentheses indicates the target class we chose. Table 1 shows the characteristics of these six datasets organized according to their negative-positive training-instance ratios. The top dataset (abalone 19) is the most imbalanced (the ratio is about 1:130). The four datasets mostly consist of continuous data instead of categorical data.

Table 1. Four UCI datasets with different compression rates: in the letter (26) dataset, the results of two compression rates demonstrate the effect of "over-compression". Through the initial exploration, the minority class is not linearly separated, and that is said, the minority class randomly scatters within the majority class.

Dataset	Positive Insts	Negative Insts	Imbalance Ratio	Insts after under-sampling
Abalone (19)	32	4145	1:129.54	32
Yeast (5)	47	1437	1:30.5	64
Letter (26)	734	19266	1:26.28	299 (1024) 550(2048)
Glass (7)	29	185	1:6.3	32

Because it is expected that undersampling at high rates generate a trade-off between improved data balance and loss of important information, we examined whether different compression rate could lead to a further enhancement of results.

The machine learning community use two metrics the sensitivity and the specificity, when evaluating the performance of various tests. Sensitivity can be defined as the accuracy on the positive instances: True Positives /(True Positives + False Positives), while specificity can be defined as the accuracy on the negative instances: True Negatives / (True Negatives + False Positives) [4]. Kubat et al [14] suggest the g-means, $g = \sqrt{acc^{+} acc^{-}}$, which combines specificity and sensitivity. In our experiments, the g-means replaces the standard accuracy rate, which losses its functions in imbalanced datasets.

Table 2. Test Result

Dataset	SVM			VQ-SVM		
	Se	Sp	G-means	Se	Sp	G-means
Abalone (19)	0	1	0	1	0.88356	0.93998
Yeast (5)	0	1	0	0.9	0.7561	0.8249
Letter (26)	0.3537	1	0.5948	1	0.1871	0.4326
				0.7007	0.9995	0.8368
Glass (7)	0.6667	1	0.8165	0.6667	1	0.8165

These experiments use LibSVM C code [15] to test the performance of c-SVM with RBF of the gamma values from 0.5 to 2. VQ-SVM consists of the Support Vector Classification by the Spider Machine Learning toolbox [16] and the Vector Quantization by the DCPR Matlab toolbox [17].

The results of experiments show that the g-means of VQ-SVM are rather better or equal to ones of standard SVM. In detail, the specificities of SVM are better than VQ-SVM, but SVM predicts all of instances as negative. Thus the specificities of standard SVM do not make any sense. In the dataset, Letter (26), while VQ-SVM compresses the number of negative instances to an extremely low level, a new imbalanced dataset is produced, and the predictive results of this dataset show that the positive group overwhelms the learning machine.

In case that the imbalance ratio is not high and rather small dataset (e.g. Glass (7) 1:6.3 and 185 instances), the impact of VQ is not significant, e.g. almost equal value of g-means between SVM and VQ-SVM.

6. Conclusion

The results of experiments have proved the theoretic part: SVM is highly sensitive to the balance ratio between the numbers of the vectors of classes, and majority group often overwhelms the learning machine. In case of the large amount of training data with imbalance classes, oversampling increases the number of minority class, but at the same time introduces more computation costs, especially with respect to SVM. Instead of a step of the data pre-process, within the VQ-SVM, VQ optimises directly the predictor performance in case of imbalanced datasets. The previous results show the significant improvement in case of binary classification.

In the further works, more precious controls and methods are necessary to be investigated. Especially the compression to only support vectors instead of all of vectors may enhance the controllability of the algorithm of VQ-SVM and manage the information loss caused by compression.

Acknowledge:

The work has been supported by Capital Markets CRC, and the authors would like to thank Prof Donald Stokes for his support and datasets.

Reference:

1. Chawla, N.V., N. Japkowics, and A. Kolcz, *Editorial: special issue on learning from imbalanced data sets.* SIGKDD Explorations, 2004. **6**(1).
2. Wang, J., X. Wu, and C. Zhang, *Support vector machines based on K-means clustering for real-time business intelligence systems.* International Journal of Business Intelligence and Data Mining, 2005. **1**(1).
3. Scholkopf, B. and A.J. Smola, *Learning with Kernels, Support Vector Machines, Regularization, Optimization, and Beyond*. 2002: MIT Press.
4. Akbani, R., S. Kwek, and N. Japkowicz. *Applying Support Vector Machines to Imbalanced Datasets*. in *Proceedings of the 15th European Conference on Machine Learning (ECML)*. 2004.
5. Veropoulos, K., C. Campbell, and N. Cristianini. *Controlling the Sensitivity of Support Vector Machines*. in *the International Joint Conference on Artificial Intelligence (IJCAI99), Workshop ML3*. 1999. Stockholm, Sweden.
6. Karakoulas, G. and J. Shawe-Taylor. *Optimizing classifiers for imbalanced training sets*. in *Advances in neural information processing systems*. 1998: MIT Press, Cambridge, MA, USA.
7. Lin, Y., Y. Lee, and G. Wahba, *Support Vector Machines for Classification in Nonstandard Situations.* Machine Learning, 2002. **46**(1-3): p. 191 - 202.
8. Wu, G. and E.Y. Chang, *KBA: kernel boundary alignment considering imbalanced data distribution.* IEEE Transactions on Knowledge and Data Engineering, 2005. **17**(6): p. 786 - 795.
9. Vapnik, V., *The Nature of Statistical Learning Theory*. 1995, New York: Springer-Verlag.
10. Linde, Y., A. Buzo, and R.M. Gray, *An Algorithm for Vector Quantizer Design.* IEEE Transactions on Communications, pp., 1980: p. 702--710.

11. Gersho, A. and R.M. Gray, *Vector Quantization And Signal Compression*. 1992: Kluwer Academic Publishers.
12. Yu, T., T. Jan, J. Debenham, and S. Simoff. *Incorporating Prior Domain Knowledge in Machine Learning: A Review*. in *AISTA 2004: International Conference on Advances in Intelligence Systems - Theory and Applications in cooperation with IEEE Computer Society*. 2004. Luxembourg.
13. Yu, T., T. Jan, J. Debenham, and S. Simoff. *Incorporate Domain Knowledge into Support Vector Machine to Classify Price Impacts of Unexpected News*. in *The 4th Australasian Conference on Data Mining*. 2005. Sydney, Australia.
14. Kubat, M. and S. Matwin. *Addressing the Curse of Imbalanced Data Sets: One-Sided Sampling*. in *Proceedings of the Fourteenth International Conference on Machine Learning*. 1997.
15. Chang, C.-C. and C.-J. Lin, *LIBSVM: a Library for Support Vecter Machine*. 2004, Department of Computer Sicence and Information Engineering, National Taiwan University.
16. Weston, J., A. Elisseeff, G. BakIr, and F. Sinz, *SPIDER: object-orientated machine learning library*. 2005.
17. Jang, J.-S.R., *DCPR MATLAB Toolbox*. 2005.

Ontology Support for Translating Negotiation Primitives

Maricela Bravo[1], Máximo López[1], Azucena Montes[1], René Santaolaya[1],
Raúl Pinto[1], and Joaquín Pérez[1]
[1]Centro Nacional de Investigación y Desarrollo Tecnológico
Interior Internado Palmira S/N, Cuernavaca, Mor. 62490, México
{mari_clau, maximo, amr, rene, rpinto, jperezo}@cenidet.edu.mx,
WWW home page: http://www.cenidet.edu.mx

Abstract. In this paper we present an ontology solution to solve the problem of language heterogeneity among negotiating agents during the exchange of messages over Internet. Traditional negotiation systems have been implemented using different syntax and semantics. Our proposal offers a novel solution incorporating an ontology, which serves as a shared vocabulary of negotiation messages; and a translation module that is executed on the occurrence of a misunderstanding. We implemented a service oriented architecture for executing negotiations and conducted experiments incorporating different negotiation messages. The results of the tests show that the proposed solution improves the interoperability between heterogeneous negotiation agents.

1 Introduction

Negotiation plays a fundamental role in electronic commerce activities, allowing participants to interact and take decisions for mutual benefit. Recently there has been a growing interest in conducting negotiations over Internet, and constructing large-scale agent communities based on emergent Web service architectures. The challenge of integrating and deploying negotiation agents in open and dynamic environments is to achieve effective communications.

Traditional negotiation systems have been implemented in multi-agent systems (MAS), where agents exchange messages using an agent communication language (ACL) based on a specification like KQML [1] or FIPA [2]. These specifications provide a set of negotiation primitives based on speech act theory, and provide semantics for these primitives usage during communication. In order to facilitate effective communication, agents must be designed to be compliant with one of these

ACL specifications. But the implementations of these negotiation primitives in real systems, differs in syntax and usage, because is based on proprietary program code produced by developers.

The problem of communication between negotiation agents is that even if two agents are following the same ACL, they may still suffer misunderstandings due to the different syntax and semantics of their vocabularies. In table 1, we can see that some of the reported communication languages in negotiation systems are based on FIPA, and some use a different ACL not compliant with any particular specification.

Table 1. Negotiation primitives used in different systems

Authors	ACL	Negotiation Primitives
Jin Baek Kim, Arie Segev [7]	FIPA	Initial_offer RFQ Accept Reject Offer Counter-offer
Stanley Y. W. Su, Chunbo Huang, Joachim Hammer [8]	FIPA	CFP Propose Accept Terminate Reject Acknowledge Modify Withdraw
Anthony Chavez, Pattie Maes [10]	Uses a predefined set of methods, not compliant with any ACL specification.	accept-offer?(agent, from-agent, offer) what-is-price?(agent, from-agent) what-is-item?(agent, from-agent) add-sell-agent add-buy-agent add-potential-customers(sell-agent, potential-customers) add-potential-sellers(buy-agent, potential-sellers) agent-terminated(marketplace, agent) deal-made(marketplace, sell-agent, buy-agent, item, price)

Sonia V. Rueda, Alejandro J. García, Guillermo R. Simari [11]	Based on speech act theory, not compliant with any ACL specification.	Requests_Add(s, h, p)
		Authorize_Add(s, h, p)
		Require(s, h, p)
		Demand(s, h, p)
		Accept(s, h, p)
		Reject(s, h, p)
		Unable(s, h, p)
		Require-for(s, h, p, q)
		Insist_for(s, h, p, q)
		Demand_for(s, h, p, q)

Haifei Li, Chunbo Huang and Stanley Y.W Su [12]	Superset of FIPA	Call for proposal Propose proposal Reject proposal Withdraw proposal	Accept proposal Modify proposal Acknowledge message Terminate negotiation	
Jürgen Müller [6]	Based on speech act theory, not compliant with any ACL specification	Initiators: Propose, Arrange, Request, Inform, Query, Command, Inspect	Reactors: Answer, Refine, Modify, Change, Bid, Send, Reply, Refuse, Explain	Completers: Confirm, Promise, Commit, Accept, Reject, Grant, Agree.

To solve the communication problem between heterogeneous agents, we selected a translation approach based on the implementation of a shared ontology. In this ontology we explicitly describe and classify negotiation primitives in a machine interpretable form. Negotiation agents should not be forced to commit to a specific syntax. Instead, the ontology provides a shared and public vocabulary that the translator module uses to help agents to communicate during negotiation processes. We have implemented a negotiation system based on Web services technologies, into which we have incorporated the translator module and the shared ontology. Our approach acknowledges that agents may use different negotiating languages.

The rest of the document is organized as follows. In section 2, we present the translator architecture. In section 3, we describe the design of the ontology. In section 4, the general architecture of the system for executing negotiation processes is presented. In section 5, we describe the results of experiments. Finally in section 6, we present conclusions.

2 Architecture of the Translator

The translator acts as an interpreter of different negotiation agents. In figure 1, we present the architectural elements involved in translation. This architecture consists of the following elements: multiple negotiation agents, the message transport, the translator module, and the shared ontology. Each negotiation agent in turn consists of a local ACL, decision making strategies to determine the preferences, and the negotiation protocol.

For example, suppose that agents *A* and *B* initiate a negotiation process, using their own local ACL, sending messages over the message transport. If happens that agent *A* misunderstands a message from agent *B*, it invokes the semantic translator module sending the message parameters (sender, receiver, message). The translator interprets the message based on the definitions of the sender agent and converts the message into an interlingua. Then the translator converts the interlingua representation to the target ACL based on the receiver agent definitions. Finally sends the message back to the invoking agent *A* and they continue with execution of negotiation. The translator is invoked only in the occurrence of a misunderstanding, assuring interoperability at run time.

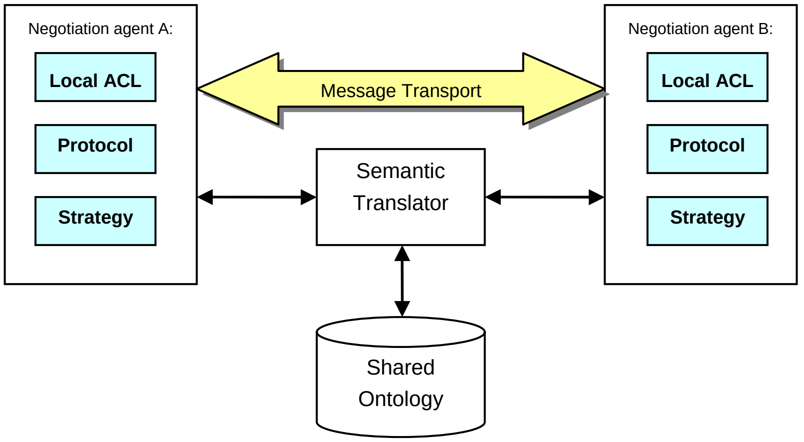

Fig. 1. Translator architecture

3 Shared Ontology

The principal objective in designing the ontology was to serve as an *interlingua* between agents during exchange of negotiation messages. According to Müller [6], negotiation messages are divided into three groups: initiators, if they initiate a negotiation, reactors, if they react on a given statement and completers, whether they complete a negotiation. We selected this classification to allow the incorporation of new negotiation primitives from the local agent ACL. Figure 2 shows the general structure of our ontology.

Based on the concepts and negotiation primitives we built our ontology. To code the ontology we decided to use OWL as the ontological language, because it is the most recent development in standard ontology languages from the World Wide Web Consortium (W3C)[1]. An OWL ontology consists of classes, properties and individuals. We developed the ontology using Protégé [14, 15], an open platform for ontology modeling and knowledge acquisition. Protégé has an OWL Plugin, which can be used to edit OWL ontologies, to access description logic reasoners, and to acquire instances of semantic markup.

[1] http://www.w3.org

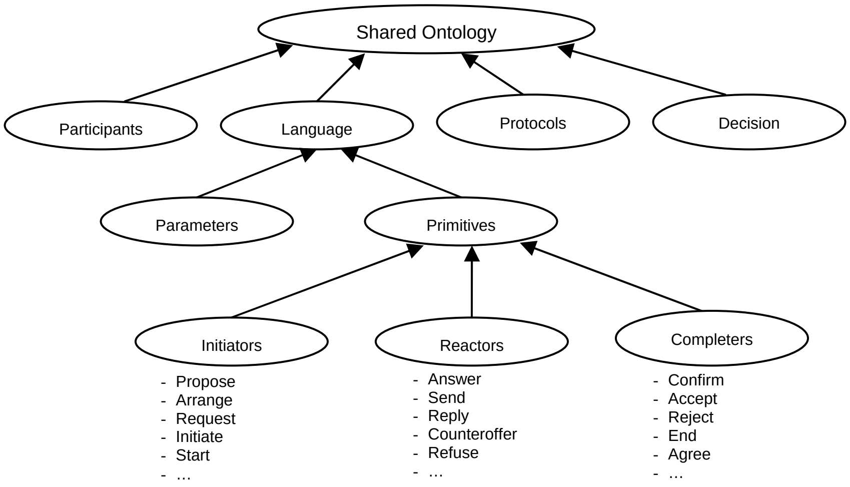

Fig. 2. General structure of the negotiation ontology

4 Implementation of the Negotiation System

The general architecture for the execution of negotiation processes is illustrated in figure 4. In this section we briefly describe the functionality and implementation techniques for each component.

a. The matchmaker is a Java module which is continuously browsing buyer registries and seller descriptions, searching for coincidences.

b. The negotiation process module is a BPEL4WS-based engine that controls the execution of negotiation processes between multiple agents according to the predefined protocols. BPEL4WS provides a language for the formal specification of business processes and business interaction protocols. The interaction with each partner occurs through Web service interfaces, and the structure of the relationship at the interface level is encapsulated in what is called a partner link.

c. Seller and buyer agents are software entities used by their respective owners to program their preferences and negotiation strategies. For example, a seller agent will be programmed to maximize his profit, establishing the lowest acceptable price and the desired price for selling. In contrast, a buyer agent is seeking to minimize his payment. On designing the negotiation agents, we identified three core elements, strategies, the set of messages and the protocol for executing the negotiation process. The requirements for these elements were specified as follows:

- Strategies should be private to each agent, because they are competing and they should not show their intentions.
- Messages should be generated privately.
- The negotiation protocol should be public or shared by all agents participating, in order to have the same set of rules for interaction. The negotiation protocol establishes the rules that agents have to follow for interaction.

d. The translator module is invoked whenever the agent misunderstands a negotiation message from another agent. The translator module was implemented using Jena2, a framework for building Semantic Web applications. It provides a programmatic environment for OWL, including a rule-based inference engine.

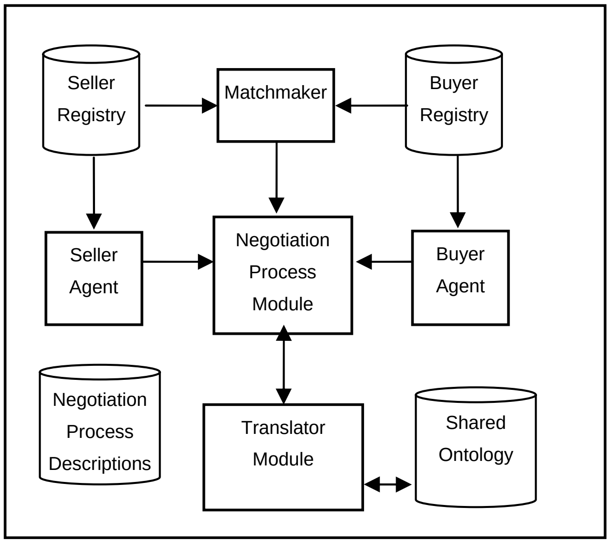

Fig. 3. Architecture of the negotiation system

5 Experimentation

In this section we describe the methodological steps that we followed for the execution of experiments.

a. Identify and describe negotiation agent's characteristics

Table 2 shows the characteristics of agents A and B, specifying their language definitions: names of primitives and a description.

[2] http://jena.sourceforge.net

Table 2. Characteristics of agents A and B

Agent A	Language definitions
	(CFP, “Initiate a negotiation process by calling for proposals”), (Propose, “Issue a proposal or a counterproposal”), (Accept, “Accept the terms specified in a proposal without further modifications”), (Terminate, “Unilaterally terminate the current negotiation process”), (Reject, “Reject the current proposal with or without an attached explanation”), (Acknowledge, “Acknowledge the receipt of a message”), (Modify, “Modify the proposal that was sent last”), (Withdraw, “Withdraw the last proposal”)
Agent B	**Language definitions**
	(Initial_offer, “Send initial offer”), (RFQ, “Send request for quote”), (Accept, “Accept offer”), (Reject, “Reject offer”), (Offer, “Send offer”), (Counter-offer, “Send counter offer”) (Withdraw, “Withdraw the last proposal”)

b. Classify negotiation primitives in the ontology classes

For each negotiation primitive we need to analyze its semantics and usage. According to this description we can identify to which class it belongs. Table 3 shows the classification of the primitives provided by agents A and B.

Table 3. Classification of negotiation primitives

Agent	Starter	Reactor	Completer
A (Buyer)	CFP	Propose	Accept
		Modify	Reject
		Withdraw	Terminate
		Acknowledge	NotUnderstood
B (Seller)	RFQ	Initial_Offer	Accept
		Offer	Reject
		Counter_Offer	NotUnderstood

c. Align primitives in a finite state machine

Alignment is necessary to verify and clarify the intended usage of negotiation primitives.

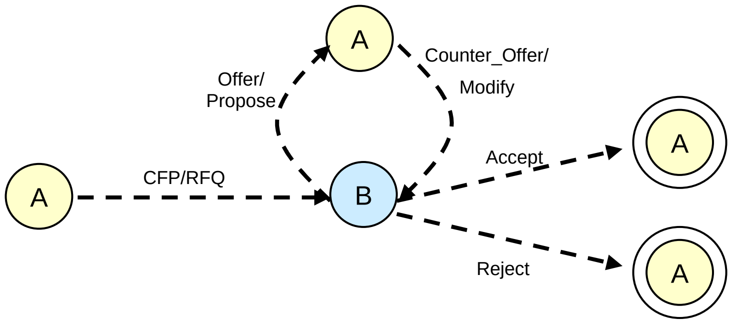

Fig. 4. Finite state machine

d. Identify and establish the relations between different primitives

Based on the classification of primitives and their allocation in the finite state machine, we can identify the relations between negotiation primitives.

A		**B**
CFP	isSynonymOf	RFQ
Propose	isSynonymOf	Offer
Propose	isSynonymOf	Inicial_Offer
Modify	isSynonymOf	Counter_Offer
Withdraw	isSynonymOf	Counter_Offer
Terminate	isSimilarOf	Reject

e. Publish and code primitives in the ontology

This step consists of populating the ontology with the primitive's definitions and relations.

f. Execute negotiation

When primitives have been published, the process of negotiation between these agents can be started. We executed 15 negotiation tests with these agents. The results of these experiments were registered in a log file. Table 4 shows the results.

Table 4. Experimental results

Test	LastPrice	MaxPay	Iterations	Quantity	FinalPrice	Result
1	$ 1,750.00	$ 1,000.00	12	1500	$ -	Reject
2	$ 774.00	$ 1,760.00	3	887	$ 1,674.00	Accept
3	$ 1,788.00	$ 128.00	12	1660	$ -	Reject
4	$ 1,058.00	$ 110.00	12	1270	$ -	Reject
5	$ 761.00	$ 77.00	2	1475	$ -	**NotUnderstood**
6	$ 621.00	$ 446.00	12	56	$ -	Reject
7	$ 114.00	$ 704.00	7	8	$ 614.00	Accept

Test	LastPrice	MaxPay	Iterations	Quantity	FinalPrice	Result
8	$ 1,837.00	$ 2,199.00	9	53	$ 2,137.00	Accept
9	$ 1,665.00	$ 2,047.00	9	56	$ 1,965.00	Accept
10	$ 1,920.00	$ 286.00	12	81	$ -	Reject
11	$ 172.00	$ 1,553.00	2	41	$ 1,172.00	Accept
12	$ 980.00	$ 1,541.00	2	67	$ -	**NotUnderstood**
13	$ 1,276.00	$ 500.00	2	43	$ -	Reject
14	$ 1,500.00	$ 1,108.00	2	110	$ -	**NotUnderstood**
15	$ 1,400.00	$ 1,520.00	3	4	$ 1,452.00	Accept

The results of experiments showed that there were some negotiations that ended the process with a *NotUnderstood* message. This was due to the emission of an *Acknowledge* message form agent A, which agent B does not recognize. Although, the experiment results show good evidence that the two agents are communicating efficiently even when their language definitions are quite different.

6 Conclusions

In this paper we have presented how an ontology approach can improve interoperability between heterogeneous negotiation agents. In particular we incorporated a translator solution for the problem of lack of understanding among seller and buyer agents during the exchange of messages at run time. We evaluated the ontology in the target application, and described the system architecture into which the negotiation processes are executed. We believe that semantic interoperability of ACL is an important issue that can be solved by incorporating a shared ontology. The experimental tests showed that the proposed architecture improves the continuity of the execution of negotiation processes, resulting in more agreements.

References

1. T. Finning, R. Fritzon, and R. McEntire: KQML as an agent communication language, in *Proceedings of the 3rd International Conference on Information and Knowledge Management*, November 1994.
2. FIPA – Foundation for Intelligent Physical Agents. FIPA Specifications, 2003; available at http://www.fipa.org/specifications/index.html.
3. Uschold, M. and King M., Towards a Methodology for Building Ontologies, *Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995.
4. Grüninger, M. and Fox, M., The Role of Competency Questions in Enterprise Engineering, *IFIP WG 5.7 Workshop on Benchmarking*. Theory and Practice, Trondheim, Norway, 1994.
5. Fernández, M., Gómez-Pérez, A., and Juristo, N., METHONTOLOGY: From Onthological Art towards Ontological Engineering, *Proceedings of AAAI Spring Symposium Series*, AAAI Press, Menlo Park, Calif., pp. 33-40, 1997.

6. Müller, H. J., Negotiation Principles, *Foundations of Distributed Artificial Intelligence*, in G.M.P. O´Hare, and N.R. Jennings, New York: John Wiley & Sons.
7. Jin Baek Kim, Arie Segev, A Framework for Dynamic eBusiness Negotiation Processes, *Proceedings of IEEE Conference on E-Commerce*, New Port Beach, USA, 2003.
8. Stanley Y. W. Su, Chunbo Huang, Joachim Hammer, Yihua Huang, Haifei Li, Liu Wang, Youzhong Liu, Charnyote Pluempitiwiriyawej, Minsoo Lee and Herman Lam, An Internet-Based Negotiation Server For E-Commerce, *the VLDB Journal*, Vol. 10, No. 1, pp. 72-90, 2001.
9. Patrick C. K. Hung, WS-Negotiation: An Overview of Research Issues, *IEEE Thirty-Seventh Hawaii International Conference on System Sciences* (HICSS-37), Big Island, Hawaii, USA, January 5-8, 2004.
10. Anthony Chavez, Pattie Maes, Kasbah: An Agent Marketplace for Buying and Selling Goods, *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, UK, April 1996.
11. Sonia V. Rueda, Alejandro J. García, Guillermo R. Simari, Argument-based Negotiation among BDI Agents, *Computer Science & Technology*, 2(7), 2002.
12. Haifei Li, Chunbo Huang and Stanley Y.W Su, Design and Implementation of Business Objects for Automated Business Negotiations, *Group Decision and Negotiation*, Vol. 11; Part 1, pp. 23-44, 2002.
13. Dignum, Jan Dietz, Communication Modeling – The language/Action Perspective, *Proceedings of the Second International Workshop on Communication Modeling*, Computer Science Reports, Eindhoven University of Technology, 1997.
14. J. Gennari, M. Musen, R. Fergerson, W. Grosso, M. Crubézy, H. Eriksson, N. Noy, and S. Tu: The evolution of Protégé-2000: An environment for knowledge-based systems development, *International Journal of Human-Computer Studies*, 58(1): 89-123, 2003.
15. H. Knublauch: An AI tool for the real world: Knowledge modeling with Protégé, *JavaWorld*, June 20, 2003.

Statistical Method of Context Evaluation for Biological Sequence Similarity

Alina Bogan-Marta[11], Ioannis Pitas[1], and Kleoniki Lyroudia[2]
1 Aristotle University of Thessaloniki, Department of Informatics,
Artificial Intelligence and Information Analysis Laboratory, Box 451,
54124 Thessaloniki, Greece, pitas@aiia.csd.auth.gr
2 Aristotle University of Thessaloniki, Department of Endodontology,
Dental School, Greece, lyroudia@aiia.csd.auth.gr

Abstract. Within this paper we are proposing and testing a new strategy for detection and measurement of similarity between sequences of proteins. Our approach has its roots in *computational linguistics* and the related techniques for quantifying and comparing content in strings of characters. The pairwise comparison of proteins relies on the content regularities expected to uniquely characterize each sequence. These regularities are captured by *n*-gram based modelling techniques and exploited by cross-entropy related measures. In this new attempt to incorporate theoretical ideas from computational linguistics into the field of bioinformatics, we experimented using two implementations having always as ultimate goal the development of practical, computationally efficient algorithms for expressing protein similarity. The experimental analysis reported herein provides evidence for the usefulness of the proposed approach and motivates the further development of linguistics-related tools as a means of analysing biological sequences.

1 Introduction

The practice of comparing gene or protein sequences with each other, in the hope of elucidating similarity conveying functional and evolutionary significance, is a subject of primary research interest in bioinformatics. The application of this type of analysis to complete genomes greatly expands its utility and implications. The rewards range from the purely technical, such as the identification of contaminated sequence phases, to the most fundamental ones, such as finding how many different domains define the tree of life. Proteins are large, complex molecules composed of

[1] Currently, she works at University of Oradea, Computer Science Department, Universitatii 1, Oradea, Romania.

amino acids and their comparison and clustering according to similarity, require dedicated algorithms.

The most frequently used methods for measuring protein similarities are based on tedious algorithmic procedures for sequence alignment. Smith-Waterman algorithm [1] remains the standard reference method for pairwise sequence similarity due to the accuracy of the obtained results. Other heuristic algorithms, like BLAST[1], FASTA[2] or CLUSTAL[3] provide higher computational efficiency at the expense of accuracy. Algorithms characterized by the computation of profiles for whole protein families are based on hidden Markov models [2], [3]. All the above mentioned methods are built over sequence alignment but a variety of new alternative methods has already become available for expressing similarity between biological sequences for use in different applications. In Sjolander's work [4] are used Dirichlet mixtures while the authors of [5] apply discriminative methods using the approach of support vector machines (SVMs). Latent semantic analysis (LSA) is another method used in the work of Ganapathiraju [6] and the universal similarity metric (USM) for structural similarity between pairs of proteins is proposed by Krasnogor and Pelta [7].

Despite the maturity of the developed methodologies working towards this direction, the derivation of protein similarity measures is still an active research area. The interest is actually renewed, due to the continuous growth in size of the widely available proteomic databases that call for alternative cost-efficient algorithmic procedures. They should reliably quantify protein similarity without resorting to any kind of alignment. Apart from efficiency, a second specification of equal importance for the establishment of similarity measures is the avoidance of parameters that need to be set by the user (a characteristic inherent in the majority of the above mentioned methodologies). It is often the case with the classical similarity approaches that the user faces a lot of difficulties in the choice of a suitable search algorithm, scoring matrix or function as well as set of optional parameters whose optimum values correspond to the most reliable similarity.

A new approach for measuring the similarity between two protein sequences is introduced in this paper. It is inspired by the successful use of the entropy concept for information retrieval in the field of statistical language modeling (Manning and Schütze [8], Jurafsky and Martin [9]). Although the *n*-gram concept has been used in earlier works, e.g. [10], [11], the presented work is following a first attempt to adopt this dual step for comparing biological sequences [12]. Therefore, some experiments were necessary in order to discover the most effective way in which these ideas could be applied in the specific domain. For a complete validation of the suggested similarity measure, we built an annotated database by selecting proteins from Astral SCOP genetic domain sequences (http://astral.berkeley.edu). Using standard procedures, well-known in the field of ***exploratory data analysis*** and ***information retrieval***, we evaluated the performance of our measure and contrasted with the performance of a relevant similarity score obtained by applying the popular CLUSTAL W method to the same database. CLUSTAL W method performs multiple sequence alignment and generates pairwise similarity scores using the identification of conserved sequence regions. We show that our method provides an effective way for capturing the common characteristics of the compared sequences,

[1] http://www.ncbi.nlm.nih.gov/BLAST/

[2] http://www.ebi.ac.uk/fasta33/

[3] http://www.ebi.ac.uk/clustalw/

while avoiding the annoying task of choosing parameters, additional functions or evaluation methods. The high performance of the new method and the ready-to-plug-in character, taken together with its computational efficiency, make our approach a promising alternative to the well-known, sophisticated protein similarity measurements.

2 Methods

2.1 Theoretical background

There are various kinds of language models that can be used to capture different aspects of regularities in natural language [13]. Markov chains are generally considered among the more fundamental concepts for building language models. In this approach, the dependency of the conditional probability of observing a word w_k at a position k in a given text is assumed to depend only upon its immediate n predecessor words $w_{k-n} \ldots w_{k-1}$. The resulting stochastic models, usually referred as ***n-grams***, constitute an heuristic approach for building language grammars and their linguistic justification has often been questioned in the past. However, in practice they have turned out to be extremely powerful theoretical tools. Nowadays *n*-gram language modeling stands out as superior to any formal linguistic approach [13] and has gained high popularity due to its simplicity.

Close related with the design of models for textual data are the algorithmic procedures used to validate them. Apart from the justification of a single model, they can facilitate the selection of the specific one (among competing alternatives) most faithfully representing the available data. ***Entropy*** is a key concept for this kind of procedures. In general, its estimation is considered to provide a quantification of the information in a text and has strong connections to probabilistic language modeling [14]. As described in [8] and [15], the entropy of a random variable X that ranges over a domain ℵ, and has a probability density function, $P(X)$ is defined as:

$$H(X) = -\sum_{X \in \aleph} P(X) \log P(X). \tag{1}$$

Recently, in Van Uytsel and Compernolle's work [16], the general idea of entropy has been adopted in the specific case that a written sequence W= $\{w_1,w_2,\ldots,w_{k-1,}w_k, w_{k+1},\ldots\}$ is treated as a language model L based composition, having the following estimating formula:

$$\hat{H}_L(X) = -\frac{1}{N}\sum_{W*} Count\left(w_i^n\right) \log_2 p_L\left(w_{i+n} \middle| w_i^{n-1}\right) \tag{2}$$

where the variable X has the form of an n-gram $X = w_i^n \Leftrightarrow \{w_i,w_{i+1,}\ldots,w_{i+n-1}\}$ and $Count(w_i^n)$ is the number of occurrences of w_i^n. The summation runs over all the possible n-length combinations of consecutive w_i (i.e. $W^*=\{\{w_1,w_{2,}\ldots,w_n\},\{w_2,w_3,\ldots,w_{n+1}\},\ldots\}$) and N is the total number of n-grams in the investigated sequence. The second term, $p\left(w_{i+n} \middle| w_i^{n-1}\right)$ in (2), is the conditional probability that relates the n-th element of an n-gram with the preceding n-1 elements. Following the principles of maximum likelihood estimation (MLE), it can be estimated by using the corresponding relative frequencies:

$$\hat{p}\left(w_{i+n}\middle|w_i^{n-1}\right)=\frac{Count\left(w_{i+n}\right)}{Count\left(w_i^{n-1}\right)}. \tag{3}$$

The ***cross-entropy*** between the actual probability distribution *P(X)* (over a random variable *X*) and the probability distribution *Q(X)* estimated from a model is defined as:

$$H(X,Q)=-\sum_{X\in\aleph}P(X)\log Q(X). \tag{4}$$

Two important remarks should be mentioned here. First, the cross entropy of a stochastic process, measured by using a model, is an upper bound on the entropy of the process (i.e. H(X)≤H(X,Q)) [8], [15]). Second, between two given models, the more accurate is the one with the lower cross-entropy [9].

The above entropic estimation together with the general form of (1) and (2), suggesting a direct way to pass from entropy to cross-entropy formulation, are the basis for building our protein similarity measure, described in the sequel.

2.2 The n-gram Based Protein Similarity Measure

Protein sequences from all different organisms can be treated as texts written in a universal language in which the alphabet consists of 20 distinct symbols, the amino-acids. The mapping of a protein sequence to its structure, functional dynamics and biological role then becomes analog to the mapping of words to their semantic meaning in natural languages. Recently, it was suggested that this analogy can be exploited by applying *statistical language modeling* and *text classification techniques* for the advancement of biological sequences understanding (topic on Biological Language Conference, 2003). Scientists within this hybrid research area believe that the identification of Grammar/Syntax rules could reveal entities/relations of high importance for biological and medical sciences.

In the presented approach, we adopted a Markov-chain grammar to build for our protein dataset *2*-gram, *3*-gram and *4*-gram models. To clarify things we chose a hypothetical protein sequence WASQVSENR. In the *2*-gram modeling the available tokens/words were {WA AS SQ QV VS SE EN NR}, while in the *3*-gram representation they were {WAS ASQ SQV QVS VSE SEN ENR}. Based on the frequencies of these tokens/words (estimated by counting) and by forming the appropriate ratios of frequencies, the entropy of an *n*-gram model can be readily estimated using (2). This measure is indicative about how well a specific protein sequence is modeled by the corresponding *n*-gram model. While this measure could be applied to two distinct proteins (and help us to decide about which protein is better represented by the given model), the outcomes cannot be used for a direct comparison of them. Thus, the common information content between two proteins *X* and *Y* is expressed via the formula:

$$E(X,Y)=-\sum_{all\ w_i^n}P_X(w_i^n)\log P_Y\left(w_{i+n}\mid w_i^{n-1}\right) \tag{5}.$$

The first term $P_X\left(w_i^n\right)$ in (5) corresponds to the reference protein sequence *X* (i.e. it results from counting the words of that specific protein). The second term corresponds to the sequence *Y* based on which the model has to be estimated (i.e. it results from counting the tokens of that protein). Variable w_i^n ranges over all the words (that are represented by *n*-grams) of the reference protein sequence.

2.3 Database Searches with the New Similarity Measure

Having introduced the new similarity measure, we proceed here with the description of its use for performing searches within protein databases. The essential point of our approach is that the unknown query-protein (e.g. a newly discovered protein) as well as each protein in a given database (containing annotated proteins with known functionality, structure etc.) are represented via *n*-gram encoding and the above introduced similarity is utilized to compare their representations.

We considered two different ways in which the *n*-gram based similarity is engaged in efficient database searches. The most direct implementation is called hereafter as ***direct method***. A second algorithm, the ***alternating method***, was devised in order to cope with the fact that the proteins to be compared could be of very different length. It is easy to observe the need of having two methods if sequences of very different length are compared. The procedure of experimenting with both methods and contrasting their performances gave the opportunity to check the sensitivity of the proposed measure regarding the length of the sequences.

Direct method. Let S_q be the sequence of a query-protein and $\{S\}=\{S_1, S_2, \ldots S_N\}$ the given protein database. The first step is the computation of ‘perfect’ score (PS) or ‘reference’ score for the query-protein. This is done by computing $E(S_q,S_q)$ using the query-protein both as reference and model sequence (we call here “model” the sequence compared with the query) in equation (5). In the second step, each protein S_i, i=1…N, from the database serves as the model sequence in the computation of a similarity score $E(S_q,S_i)$, with the query-protein serving as reference sequence. In this way, N similarities are computed $E(S_q,S_i)$, i=1,..,N. Finally, these similarities are compared against the perfect score PS by computing the absolute differences $D(S_q,S_i)=|E(S_q,S_i)–PS|$. The ‘discrepancies’ in term of information content between the query-protein and the database-proteins are expressed. By ranking these N measurements, we can easily identify the most similar proteins to the query-protein as those which have been assigned the lowest distance $D(S_q,S_i)$.

Alternating method: The only difference with respect to the direct method is that when comparing the query-protein with those from the database, the role of reference and model protein can be interchanged based on the shortest (the shortest sequence plays the role of reference sequence in (5)). The other steps, perfect-score estimation, ranking and selection, follow as previously.

3 Experiments

3.1 Proteins database

The strategy proposed for measuring protein similarity was presented and validated using a set of 1460 proteins extracted from Astral SCOP 1.67 sequence resource database. From the available/original corpus of data, only those families containing at least 10 protein sequences were included in our new database (this restriction will be appreciated later, since it was dictated by the ***Precision*** measure adopted for evaluation). In this way, 31 different families unequally populated were finally included. We mention that the annotation of our database follows the original annotation, relaying on the biological meaning of similarity concept (and therefore can be considered as providing the 'ground-truth' for the protein classification and the attempted similarity measurements). As a consequence, we expected that all the proteins belonging to the same family would appear as a tight cluster of textual patterns and having a proper similarity measure we could differentiate the existent families.

Our database (of 1460 proteins) was organized in 3 different sets, since the experimental results obtained with the new methods had to be compared with the results obtained with CLUSTALW method that could accept as input, protein sets with up to 500 sequences. The complete protein database (organized in 3 data sets) is available up on request and/or it will be publicly available at the Biopattern website of our laboratory (see acknowledgment section).

3.2 Experimental Results

In order to illustrate the two methods of the proposed strategy, first we followed some classical steps of *Exploratory Data Analysis*. The matrix containing all possible dissimilarity measures $D(S_i,S_j)$, $i,j=1,2,\ldots N$ for the sets 1-3 is depicted in Figures 1-3 respectively. The images are presenting in grey scale the two considered methods corresponding to three different *n*-gram models. In the adopted visualization scheme all the shown matrices (after proper normalization) share a common scale in which the 1/white corresponds to the maximum distance in each matrix. It is worth mentioning here that the 'ideal' spatial outlay is a white matrix with some black, square segments around the diagonal line. From these three figures, it is clearly evident that the *4*-gram based modeling in by both versions of our algorithm has a very good performance when searching within the given database.

Second, in order to provide quantitative measures of performance for the new method, we adopted an index of search accuracy, which is derived from ***Precision*** measure [17]. This index is the ratio computed by dividing the correctly classified number of protein sequences (identified by the algorithm as the 10 most similar ones) with 10 representing the minimal number of sequences within a family. More specifically, each protein in turn was treated as query and we measured the accuracy of the first 10 sequences identified within the set as the most similar to the query-protein. In other words, by taking into consideration the class/family label of each protein, we counted the proteins sharing the same label as the query (i.e. a number from 1 to 10).

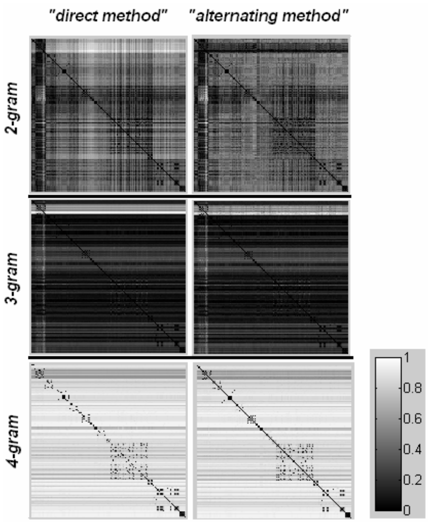

Fig. 1. Visualization of the matrices containing all the possible pairwise dissimilarities for the 497 proteins of Set1, for 2,3,4-gram models.

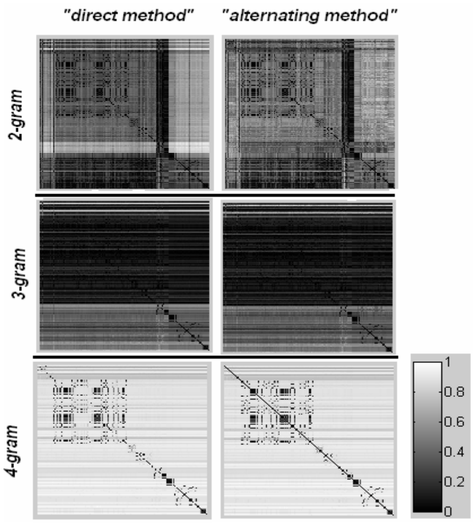

Fig. 2. Visualization of the matrices containing all the possible pairwise dissimilarities for the 497 proteins of Set2, for 2,3,4-gram models.

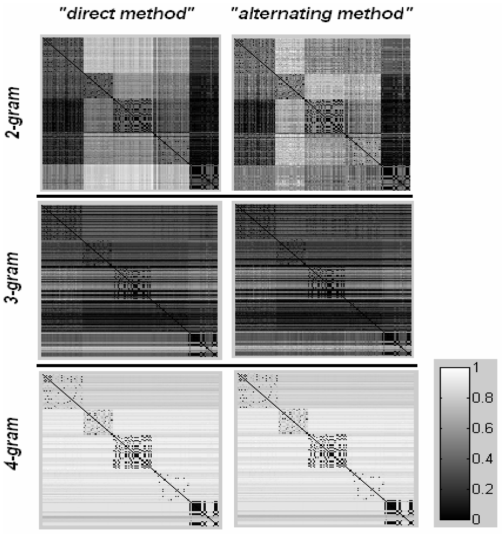

Fig. 1. Visualization of the matrices containing all the possible pairwise dissimilarities for the 466 proteins of Set3, for 2,3,4-gram models.

We repeated the procedure for all the proteins in the individual sets and finally were averaged the estimated parts in order to provide a total ***Precision***-score for each set separately. To help the reader to appreciate the performance of our algorithms, we repeated the same experimental procedure using the similarity scores obtained by applying the CLUSTAL W method to the 3 different protein sets. The available CLUSTAL W tool requires a set of input parameters, and we decided to use the default values: Protein Gap Open Penalty = 10.0, Protein Gap Extension Penalty = 0.2, Protein matrix = Gonnet, provided at the European Molecular Biology Laboratory and European Bioinformatics Institute (EMBL-EBI) web site (http://www.ebi.ac.uk/). In Table 1 are included the precision scores provided by CLUSTAL W and both of our approaches for different *n*-gram models. It is worth mentioning that our algorithmic strategy almost reaches (in the case of 4-gram modeling for the third set) the high performance of CLUSTAL W method. For the sake of completeness, we repeated the ***Precision*** measurements with our method for the overall set of 1460 proteins. The computed values were not significant different from the values corresponding to the three different sets, providing some evidence about the robustness of our method, indicating that its performance scales well with the size of the database.

Table 1. The ***precision*** scores obtained from similarity results given by CLUSTAW tool are in column 'CLUST.W', followed by those obtained using our similarity methods for 2,3,4-gram models for the three data sets.

Set	Clust.W	Direct Method			Alternating Method		
		2-gram	3-gram	4-gram	2-gram	3-gram	4-gram
1	0.872	0.439	0.662	0.830	0.471	0.646	0.823
2	0.921	0.446	0.650	0.874	0.439	0.605	0.860
3	0.932	0.534	0.865	0.931	0.574	0.828	0.919

4 Conclusions

The method we experimented and presented in this paper constitutes a step forward in investigating the engagement of language modelling for characterizing, handling and understanding biological data in the format of sequences. Specifically, we studied the efficiency and effectiveness for searching in protein database of the new measurement method. The experimental results indicate the reliability of our algorithmic strategy for expressing similarity between proteins. Given the conceptual simplicity of the introduced approach, it appears as an appealing alternative to previous well-established techniques.

From the experimens, the *direct method* seems to perform slightly better. If the second method would perform better, we should expect to have significant length differences between sequences classified as similar and belonging to the same family. In the exceptional case when all the compared sequences would have the same length, the *direct method* is equivalent with the *alternating method* and performs very well.

Regarding the order of the employed *n*-gram model, after testing with order of 2,3,4,5 we noticed, as can be seen in Table 1 and Fig.1-3 that the performance of the method increases with the order of the model up to 4. After the order of 5 due to the lack of data, the corresponding maximum likelihood estimates become unreasonable uniform and very low. This sets an upper limit for our model order in the specific database (perhaps slightly higher order model could work in different protein databases).

If we pay more attention to the visual representation of our results (the emerging spots along the main diagonal in the 2D-displays correspond to well-formed groups of proteins, especially in the case of 4-gram modelling), we can consider that the structure revealed by using the new similarity measures bears a biological meaning. More explicitly, we assume that each defined group is indicative for the existence of a family/superfamily of proteins. Despite the fact that this aspect requires a deeper exploration, which is beyond the scope of this paper, it provides a hint that the new measures can be exploited within a proper clustering framework for mining extra information from given biological databases.

The comparison of our similarity scores with those provided by the CLUSTAL W method showed that in terms or performance in retrieval our method approaches the CLUSTAL W one. Considering the algorithmic simplicity and computational efficiency of the new approach, we are justified to suggest it as first choice when search in large databases are required. In terms of time complexity, in absence of a detailed analysis, we are motivated to consider this method as efficient especially when search procedure is running over large sequence databases with long strings of sequences. This motivates us to pursue further on how to achieve even higher performance. At this point, we have to remark that this is only a statistical in nature technique and it could be improved by incorporating biological knowledge such as working with functional groups of amino acids.

ACKNOWLEDGMENT

This work was supported by the EU project Biopattern: Computational Intelligence for biopattern analysis in Support of eHealthcare, Network of Excellence Project No. 508803.

References

1. T. Smith, and M. Watermann, Identification of common molecular subsequences, *J. Mol. Biol.* vol.147, pp.195-197 (1981).

2. A. Krogh, M. Brown, I. Mian, K. Sjolander, and D. Haussler, Hidden Markov models in computational biology: Application to protein modeling, *J. Mol. Biol.*, vol.235, pp.1501-1531 (1994).

3. P. Baldi, Y. Chauvin, T. Hunkapiller, and M.A. McClure, Hidden Markov models of biological primary sequence information, in *Proc. Natl. Acad. Sci. USA*, vol.91(3), pp.1059-1036 (1994).

4. K. Sjolander, K. Karplus, M. Brown, R. Hughey, A. Krogh, I.S. Mian, and D. Haussler, Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology, *J. Bioinformatics*, Vol 12, pp: 327-345 (1996).

5. H. Saigo, J-P. Vert, N. Ueda, and T. Akutsu**,** Protein homology detection using string alignment kernels, *J. Bioinformatics*, vol.20 no.11, pp.1682-1689 (2004).

6. M.K. Ganapathiraju, J. Klein-Seetharaman, N. Balakrishnan and R. Reddy, Characterization of protein secondary structure-application of latent semantic analysis using different vocabulary, *IEEE Signal Processing Magazine*, vol. 21, no.3, pp. 78-87 (2004).

7. N. Krasnogor, and D. A. Pelta**,** Measuring the similarity of protein structures by means of the universal similarity metric, *Bioinformatics Advance Access*, vol. 20, pp. 1015-1021 (2004).

8. C.D. Manning, and H. Schütze, 2000, Foundations of statistical natural language processing, Massachusetts Institute of Technology Press, Cambridge, Massachusetts London, England,pp.554 – 556;557 – 588.

9. D. Jurafsky, and J. Martin, 2000, *Speech and Language Processing*, Prentice Hall, pp. 223-231.

10. M. Ganapathiraju, V. Manoharan, and J. Klein-Seetharaman, Statistical sequence analysis using n-grams, *J. Appl. Bioinformatics*, vol.3 (2), pp.193-200 (2004).

11. S. Erhan, T.Marzolf, and L. Cohen, Amino-acid neighborhood relationships in proteins: breakdown of amino-acid sequences into overlapping doublets, triplets and quadruplets, Int. J. Biomed Comput, vol. 11(1), pp.67-75 (1980).

12. A.Bogan-Marta, N.Laskaris, M.A.Gavrielides, I.Pitas, and K. Lyroudia, A novel efficient protein similaritymeasure based on n-gram modeling, on electronical proceedings of CIMED2005, pp. 122-127.

13. S. Wang, D. Schuurmans, F. Pengun, and Y. Zhao, Semantic N-gram Language Modeling With The Latent Maximum Entropy Principle. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-03)* available at *:* http://citeseer.nj.nec.com/575237.html

14. D. Van Compernolle, Spoken Language Science and Technology, 2003, http://www.esat.kuleuven.ac.be/~compi/pub/spoken_language/TOC.htm

15. P.F. Brown, A. S. Della Pietra, V.J. Della Pietra, L.R. Mercer Robert, and C.L. Jennifer, An estimation of an upper bound for the entropy of English, in *Association for Computational Linguistics,* Yorktown Heights, NY 10598, P.O. Box 704, 1992.

16. D.H. Van Uytsel, and D.Van Compernolle, Entropy-based context selection in variable-length n-gram language models, *IEEE Benelux Signal Proc. Symp.*, pp. 227-230 (1998).

17. R. Baeza-Yates and B. Ribeiro-Neto, in Retrieval Evaluation, *Modern Information Retrieval*, Ed. Addison Wesley, 1999, pp.75-81.

Biological inspired algorithm for Storage Area Networks (ACOSAN)

Anabel Fraga Vázquez[1]
1 Universidad Carlos III de Madrid
Av. Universidad, 30, Leganés, Madrid, SPAIN afraga@inf.uc3m.es

Abstract. The routing algorithms like Storage Area Networks (SAN) algorithms are actually deterministic algorithms, but they may become heuristics or probabilistic just because of applying biological inspired algorithms like Ant Colony Optimization (ACO) of Dorigo. A variant suggested by Navarro and Sinclair in the University of Essex in UK, it is called MACO and it may open new paths for adapting routing algorithms to changes in the environment of any network. A new algorithm is anticipated in this paper to be applied in routing algorithms for SAN Fibre Channel switches, it is called ACOSAN.

1 INTRODUCTION

This paper helps to create new paths for betters routing algorithms based in biological inspired algorithms like ACO (Ant Colony Optimization) of Dorigo [5,18,19]. The base of the paper is to apply these kind of algorithms and variants of that in Fibre Channel's switches (FC), for package routing in an adaptive way. In particular the Multiple Ant Colony Optimization of Navarro and Sinclair and ANTNET of Dorigo are very useful in that matter [11,13,18].

The reminder of this section establishes the history and introduction to the SAN networks, and benefits of this technology. Section 2 explains some key concepts and problems of Fibre Channel networks. Section 3 surveys some related previous work applying ACO to networking problems. Sections 4 describe basis for the algorithms proposed and the algorithm itself. And the final sections cover acknowledgments, future work and conclusions.

1.1. History and definitions

The increasing need year by year to connect disks to computers by SCSI connectors, which is a standard in the eighties in parallel connections, but not so fast as expected because of the problem that parallel connection have in front of serial connections.

Day by day new faster connections are needed and technology does not stop and must not. The Fibre Channel technology is a prove of advance, it starts in the nineties and it has an appreciated speed which is over Gigabits, serial connection, and allows large distances over 10 kilometers.

The external storage is a new discovers, disks not connected anymore point-to-point to servers. Large storage arrays of disks now are the centre of the external storage, it may content five disks or even thousands of disks depending on the size of the company and the data to be storage, terabytes of information are connected by Fibre Channel to servers.

Brocade, an enterprise recognized in the area of Fibre Channel' s switches, is at the moment one of the factories for Fabric topologies in Storage Area Networks (SAN). This company defines SAN as a network for storage and system components, which are all communicated in a Fibre Channel net, used to consolidate and share information, offering high performance links, high availability links, higher speed backups, and support for clustering servers.

1.2. Providers

The main providers of SAN and Fibre Channel spares are placed in Table I. McData is the growing company in the area, followed by Brocade. McData works with IBM and Brocade works with Hewlett Packard (HP). These alliances are strategic for both companies in order to provide a whole package for the customers.

Table 1. Fibre Channel's switches providers

Rank	Provider	Growing rate (%)
1	McData (IBM)	17
2	Brocade (HP)	5
3	Cisco Systems	36
4	CNT/Inrange	8

Cisco is showed as one of the companies with the grater rate of growing but not only for SAN technology, the company is in a privileged position because of the large quantity of switches for any kind of network.

1.3. Structure and topology of SAN networks

There are different topologies for storage networks, the three basics are showed in Figures 1, 2 and 3. Figure 1 shows a point-to-point network with SCSI technology.

Figure 2 shows a Fibre Channel technology based network which covers over ten kilometres. And finally, Figure 3 shows a typical SAN network of Fabric typology.

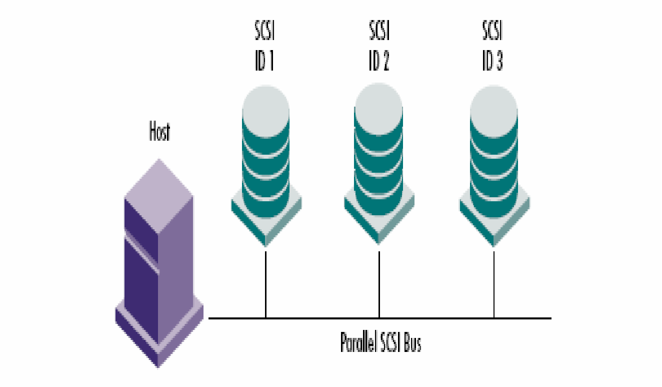

Fig. 1. Basic SCSI technology net between disks and server.

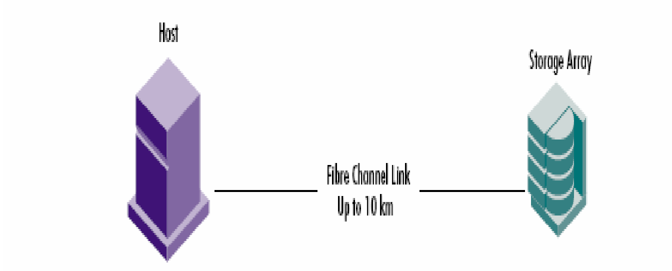

Fig. 2. Fibre Channel net for over 10 kilometres.

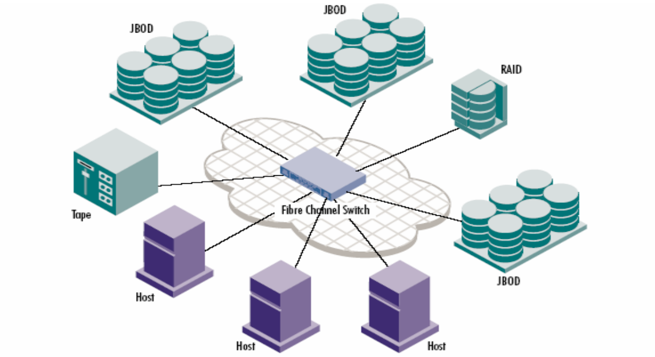

Fig. 3. Typical Fibre Channel network in multi-servers environment.

1.4. Benefits

Storage Area Networks (SAN) guarantee high availability in the nodes interconnected, allows consolidating information in disk arrays able to storage thousands of bits or bytes in a single point of storage. An additional advantage is to reduce network overload originated for automatic backups. Then again, allows to accelerate information access speed and makes certain fault tolerance. In case of a physical disaster, the availability that servers have because of been placed in different locations connected to the storage network makes possible to maintain unaffected the operation of a company.

2. FIBRE CHANNEL

Fibre Channel technology is a generic mechanism that allows high speed of transfers in long distances. It has physical definitions in the physical layer because of the transport protocol, like OSI for TCP/IP, the protocol tolerates TCP/IP and SCSI interfaces.

In general, it is easy to provide routing algorithms for Fibre Channel, but they are not specified or even public in the switches architecture provided. For example José Duato [12] published a BFS algorithm for routing SAN networks with Fibre Channel. But biological inspired algorithms are missing, and they could be an improvement for routing and adapting to networks exclusive of need for redefine algorithms or nodes in a switch.

Fredman, DataCore employee, defines Fibre Channel as a technology based on standards, innovative, functional to replace SCSI connections between disks, backup robots and servers.

Fibre Channel topologies are three:

1. Point-to-point: Based on simple links of connection between disks and servers.
2. Arbitrated Loop (FC-AL): Based on Hubs integrated to the net for routing packages.
3. Fabric: Based on switched network for routing packages.

Fibre Channel uses a communication protocol analogous to OSI with seven layers, similar to TCP/IP. The layers in the protocol are showed in Figure4.

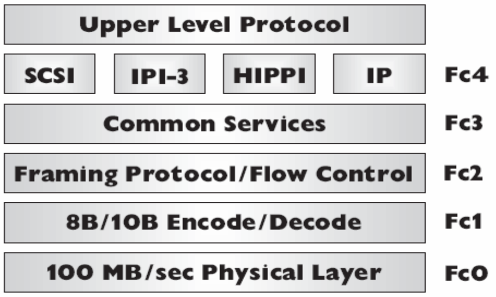

Fig. 4. Fibre Channel’s protocol, similar to OSI model.

3. USES OF ANT COLONY OPTIMIZATION IN ROUTING ALGORITHMS

Marco Dorigo [6,18,19], the father of an special biological inspired algorithms based on studies of ants natural environments and behaviour: Ant Colony Optimization (ACO) and ANTNET. In general, ants leave pheromone trail in the way to the nest and the food location. So movements are based in quantities of pheromone in the paths to follow. As much pheromone located in one path then higher will be the probability to go in the course of that way. There are always of course ants that not follow the most probable path for obtaining new sources of provisions.

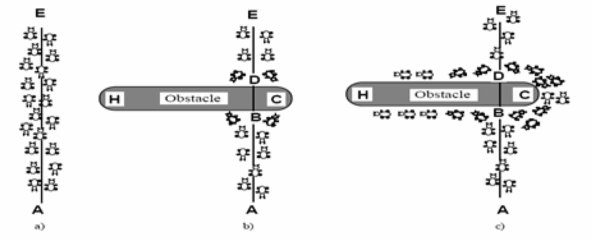

Fig. 5. Ants travelling from nest to food and suddenly an obstacle appears (Taken from Dorigo's paper explanation of ACO algorithms).

As a general rule, an heuristic function is involved in the movement of the ants in the natural environment where they are located. If in the way between nest and provisions suddenly appears an obstacle as showed in Figure 5 [13], then ants will generate a new trail and the shortest path will be used naturally because the pheromone trail is strongest in the short path for moving from nest to food and so on. This phenomenon occurs because more ants travel for the short path than for the long one. In case of two paths or more available and two shorts paths are accessible then a balance of ants traveling on them will be observed (Figure 6) [13].

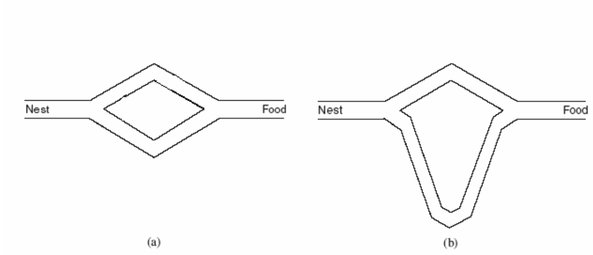

Fig. 6. More than one optimum path from nest to food.

The load balance of ants is showed in Figure 7 [13]. The first chart shows the case "a" of Figure 7 and the second chart shows the case "b" of that figure.
Dorigo [6] established an algorithm for pheromone evaporation, it occurs naturally, if not the search for new food would not be possible.

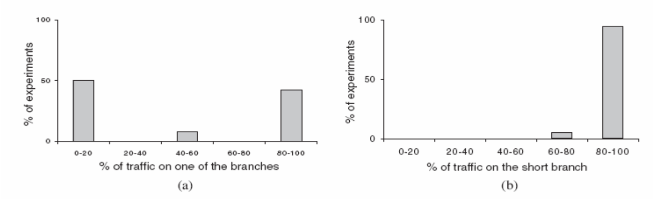

Fig. 7. Load balance of ants with more than one path possible [6,18,19].

4. ROUTING ALGORITHM

This section shows the basis and the algorithm itself.

4.1. Basis

MACO is an algorithm based in the ACO algorithm; it was originated in Essex University in UK. It was proposed for load balancing and routing in networks. It uses different ant colonies with different pheromones types, so ants of one colony will be repelled by others' pheromones and attracted by owns pheromones. It is an innovative approach and useful.

It has three kinds of algorithms. The first, it is based in local update of pheromone values. The second, it is based in update of pheromone values depending on distance. And the third, based in a global update of pheromone values depending on quantity of ants going on one path [13].

By this means, avoiding the problem of down nodes or faults in the network are interesting for adaptive environment algorithms.

4.2. Algorithm for storage area networks

Thanks to the MACO algorithm [13] and ANTNET [18] as a basic idea, is possible to postulate an algorithm called ACOSAN for Fibre Channel's switches as follows:

1. It is used more than one ant colony in order to produce different kinds of pheromones with the intention to find different optimal paths of routing for load balance purposes. A routing table will be filled with the information of routing between nodes of the net in the switch.
2. There must exist a first full filled routing table, it must be initialized using a deterministic algorithm, a manual default configuration or an existing table with default values of pheromone. It must be done in order to avoid the initial delay for generating the first version of that table.
3. Cycles of ant colonies will circumnavigate the network of switches and routing tables will be updated with new values of pheromones. It will allow maintaining distributed tables all the time.

4. The repelling mechanism will help ants to go over different routes, so load balancing routes could be generated.
5. In each cycle ants moves from every start and end possible at each point. The selection of links is stochastic, it depends of level of repelling and attraction of pheromones by ants, it is defined in the probabilistic function which is explained in the mathematical bases of the algorithm.
6. Once a loop is detected and no new version of the table is generated after a number finite of cycles, the ants are destroyed.
7. In the moment all ants are destroyed then a new cycle is placed after a variable of time defined in the algorithm in order to find better paths of routing in case of environmental changes in the links.
8. Ants must update pheromone trails in each cycle in local (on every switch). And trail must be updated as well after a cycle ends depending on global distances of paths and pheromone amounts in each path.
9. The pheromones are evaporated in each cycle as in the real life, if not new paths would not be found.
10. The best path in each cycle is stored for generate a final routing table of the cycle.

This algorithm can be applied in generic networks on top. But Fibre Channel is the primary target.

4.3. Mathematical bases for routing algorithm

The mathematical bases for the algorithm proposed are clearly established in Dorigo's papers [3,4,5,6] and Navarro and Sinclair's paper [13] and they are applied to this algorithm also, as shown in principle 1 to 7.

The algorithm will be able to control attraction of pheromones using formula (1), where is the weight of attraction for a link k by ant j, and will be the sum of pheromone over all links available, the ant must select one, where is the set of all links available for ant j depending on the position in the net.

$$\alpha_{kj} = \frac{p_{kj}}{\sum_{i \in A_j} \left(P_{ij} \right)} \qquad (1)$$

The repelling formula for pheromones by each ant will be updated globally conditional on the amount of ants in each link (3), it is the best result as suggested by Navarro and Sinclair [13], where (2) represents the use of the link k, by total sum of ants of each types crossing the link k. But, is evaporated as pheromone at the end of each cycle represents the constant of evaporation.

$$u_k^{t+1} = \rho \bullet u_k^{t} \left(\forall k, t = \sum_{i=1}^{T} S_i \right) \qquad (2)$$

$$\beta_{kj} = \frac{u_k}{\sum_{i \in A_j} (u_i)} \quad (3)$$

The probability function for movement of an ant j for taking link k will be (4) represented by where is an important constant defining the weight that repelling will have for a different kind of pheromone found in the journey. Dividing by is sure that probability will be improved as much as attraction is greater, and diminish as much as repelling grows.

$$\gamma_{kj} = \frac{\alpha_{kj} / \beta_{kj}^{\varepsilon}}{\sum_{i \in A_j} \left(\alpha_{ij} / \beta_{ij}^{\varepsilon} \right)} \quad (4)$$

The pheromones` update rule is showed in equation (5) without evaporation, used by Coloni [3,4,5] where Q is the amount of pheromone placed by an ant when passing thought a link, and evaporation reduce in the amount of pheromone placed for each link, it is denominated evaporation factor (6), where T represents each cycle of the algorithm.

$$p_{kj}^{t+1} = p_{kj}^{t} + Q \quad (5)$$

$$p_{kj}^{t+1} = \rho \bullet p_{kj}^{t} \left(\forall j, \forall k, t = \sum_{i=1}^{T} S_i \right) \quad (6)$$

If global update is considered by distance and not by amount of ants in each link, then equation for pheromone update will be (7) and factor would not be used, where will be the distance of link k, in route followed by an ant j.

$$p_{kj}^{t+1} = p_{kj}^{t} + Q / L_j^T \left(\forall j, \forall k \in R_j^T, t = \sum_{i=1}^{T} S_i \right) \quad (7)$$

4.4. Probabilistic vs. Deterministic

Analyzing deterministic and probabilistic algorithms in references [13], as shown in Figure 8, in general a heuristic algorithm will converge as a deterministic in similar solutions, but a gap is in the beginning of the algorithms. That is the reason to use a deterministic initial routing table for avoiding this gap at the beginning.

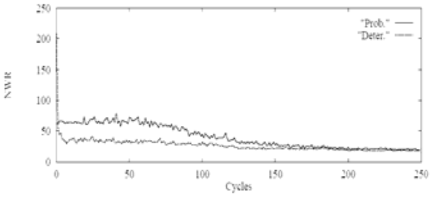

Fig. 8. Probabilistic/deterministic algorithms' general performance [13].

5. CONCLUTIONS AND FUTURE WORKS

An algorithm for routing in adaptive environments is shown in the paper using biological inspired algorithms like Ant Colony Optimization. It will help to detect changes in the network and overload in some nodes. In general, evolving algorithms are useful to solve these problems.

The need of an ACO algorithm for networks was touched by Dorigo [18,19] in the ANTNET seminal work in 1997, but applying those techniques to Fibre Channel networks for SAN are novel, it could be a preliminary approach for adapting such kind of algorithms into this technology.

ACO, MACO and new algorithms as shown in the paper are able to adapt to the environment without need of human interaction in case of failure.

Ants in the network are not a problem for high speed networks like Fibre Channel switches, if the problem of changing environments is solved by this kind of algorithms [9].

Mong's work refers to the use of ACO algorithm in routing switches instead of OSPF or RIP algorithms. Empirical results using ACO in routing and load balance are encouraging [11].

Formal studies like Gutjahr's studies [9] shown that convergence of ACO based algorithms tend to one, but this study was done for a typical NP problem: travelling salesman problem.

A future work to this paper is to apply the algorithm in a fabric switched network [16], but it is not an easy task. Some simulations were done by Navarro and Sinclair [13] in generic networks, but could be interested to apply this modified algorithm based in ACO and MACO in storage area networks.

The repelling factor is the key for disjunctive paths if it is settled properly. If the value of are higher then disjunctive paths will be obtain in the final routing tables.

6. ACKNOWLEDGMENTS

Thanks to anonymous people who reviewed this paper and helped with valuable comments to formulate it.

7. REFERENCES

1. C, Blum. *Beam-ACO hybridizing ant colony optimization with beam search: an application to open shop scheduling*. Computers & OR 32: 1565-1591 (Belgium, 2005).
2. *Brocade web page*. (October, 2005) http://www.brocade.com
3. A. Coloni, M.Y. Dorigo, V. Maniezzo. *An investigation of some properties of an ant algorithm.* (Proc. Parallel Problem Solving from Nature Conference PPSN`92, Brussels, Belgium, 1992) pp. 509-520.

4. A. Coloni, M. Dorigo, V. Maniezzo. *Distributed optimization by ant colonies*. (Proc. First European Conf. On Artificial Life (ECAL`91), París, France, 1991) pp. 134-142.
5. M. Dorigo, V. Maniezzo, A. Coloni. *The ant system: a cooperative learning approach to the travelling salesman problem.* IEEE Trans. Evolutionary Computation, 1(1):1-13, (1996).
6. M. Dorigo, V. Maniezzo, A. Colorni. *The Ant System: Optimization by a colony of cooperating agents*. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1):29-41. (1996).
7. *Fibre Channel Industry Association (FCIA)* (October, 2005) http://www.fibrechannel.org
8. M. Fredman. *An introduction to SAN Capacity Planning*. Datacore Software Company. http://www.demandtech.com/Resources/Papers/Intro%20to%20SAN%20capacity%20planning.pdf (March, 2006)
9. W. Gutjahr. *A generalized convergence result for the Graph-Based Ant System Metaheuristic*. Future Generation Computer Systems. Vienna. (2000).
10. *McDATA page*. (October, 2005) http://www.mcdata.com
11. K. Mong, W. Hong. *Ant Colony Optimization for Routing and Load-balancing: Survey and New Directions*. IEEE Transactions on systems, man, and cybernetics – Part A: Systems and Humans, (33):5. (September, 2003).
12. X. Molero, F. Silla, V. Santonia, J. Duato. *Modeling and Evaluation of Fibre Channel Storage Area Networks.* http://csce.uark.edu/~aapon/courses/ioparallel/presentations/31.ppt España. (2005).
13. G. Navarro, M. Sinclair. *Ant Colony Optimisation for Virtual-Wavelength-Path Routing and Wavelength Allocation. (MACO)* Univ. Of Essex. UK. (NASA). Proceedings of the Congress on Evolutionary Computation. (1999).
14. M. Reimann., M. Laumanns. *A hybrid ACO algorithm for the Capacitated Minimum Spanning Tree Problem.* ECAI2004. Workshop HYBRID METAHEURISTICS (HM 2004). (2004).
15. R. Schoonderwoerd, O. Holland, J. Bruten. *Ant-like agents for load balancing in telecommunication networks.* In: AGENTS '97: Proceedings of the first international conference on Autonomous agents, New York, NY, USA, ACM Press (1997). pp. 209-216.
16. *Storage Networking Industry Association web page (SNIA)* (October, 2005) http://www.snia.org
17. T. Skie, O. Lysne, J. Flich, P. López, A. Robles, J. Duato. *Lash-Tor: A generic transition-oriented routing algorithm*. ICPADS 2004. (2004) pp. 595-604.
18. G. Di Caro, M. Dorigo. *Two Ant Colony Algorithms for Best-Effort Routing in Datagram Networks.* Proceedings of PDCS'98 - 10th International Conference on Parallel and Distributed Computing and Systems. (Las Vegas, Nevada, October 28-31, 1998).
19. M. Dorigo, T. Stuetzle. *Ant Colony Optimization.* MIT Press. ISBN 0-262-04219-3. (2004).