Modelling Metabolic Pathways Using Stochastic Logic Programs-Based Ensemble Methods

Huma Lodhi and Stephen Muggleton

Department of Computing, Imperial College,
London SW7 2BZ, UK
{hml, shm}@doc.ic.ac.uk

Abstract. In this paper we present a methodology to estimate rates of enzymatic reactions in metabolic pathways. Our methodology is based on applying stochastic logic learning in ensemble learning. Stochastic logic programs provide an efficient representation for metabolic pathways and ensemble methods give state-of-the-art performance and are useful for drawing biological inferences. We construct ensembles by manipulating the data and driving randomness into a learning algorithm. We applied failure adjusted maximization as a base learning algorithm. The proposed ensemble methods are applied to estimate the rate of reactions in metabolic pathways of Saccharomyces cerevisiae. The results show that our methodology is very useful and it is effective to apply SLPs-based ensembles for complex tasks such as modelling of metabolic pathways.

1 Introduction

Metabolic pathways can be viewed as series of enzyme-catalysed reactions where product of one reaction becomes substrate for the next reaction. These pathways can be branched and interconnected via shared substrates. Quantitative analysis of enzymatic reactions is very important in biomedical applications, biotechnology and drug design. Estimation of reaction rates of enzymes in metabolic pathways is a key problem in quantitative analysis and modelling of metabolism. Behaviour of enzymes in metabolic pathways can be studied using Michaelis-Menten (MM) framework that is very useful in biological kinetics and pharmacokinetics. However information required for MM equation is not easily available and the application of MM equation is not free from problems [1]. In this paper we present a methodology that applies stochastic logic learning in ensemble learning to calculate enzymatic reaction rates.

Ensemble methods are state-of-the-art learning algorithms to solving prediction problems. The underlying aim of ensemble methods is to construct a highly accurate predictor. These methods accomplish this aim by constructing a series of predictors (models). The final model performs the estimation task by aggregating the estimations of individual models. Bagging [2] and boosting [3] are the most popular examples of these methods. Experimental results [4, 5, 6] have demonstrated ensemble methods' ability to generate highly accurate predictors

V. Danos and V. Schachter (Eds.): CMSB 2004, LNBI 3082, pp. 119–133, 2005.

and their surprising contradiction of Ockham's razor that give preference to simple hypotheses over complex ones. In order to understand this phenomenon ensembles have been analysed as they relate to the margin theory [7, 8]. Ensemble methods have also been analysed in terms of bias (measure of the goodness of the average predictor's approximation of the target function) and variance (a measure of the diversity among the base learning algorithm's guess) [5]. Bagging is a variance reduction technique and is very effective for unstable learning methods. It has also been shown that bias and variance can be expressed in terms of margin and margin can be expressed in terms of bias and variance [9]. In this paper we focus on bagging that is particularly useful for unstable predictors and possess characteristics such as parallelization. Parallelization is particularly important for stochastic logic learning where run time can be high depending on the complexity of a problem. These properties make bagging an ideal method to combine with SLPs.

SLPs [10] are generalisations of Hidden Markov Models (HMMs) [11] and Stochastic Context Free Grammars (SCFGs) [12]. They were viewed as a compact approach to representing a probabilistic preference function to provide as a parameter to ILP algorithms. HMMs and SCFGs have been extremely successful in sequence-oriented applications in natural language and bioinformatics. They provide a compact representation of a probability distribution over sequences. This contrasts with Bayesian networks, which represent conditional independences between a set of propositions. It is natural to think of HMMs and SCFGs as representing probabilities over objects in the domain (Halpern's type 1 approach from [13]) and Bayesian networks as representing probabilities over possible worlds (Halpern's type 2 approach). Learning of SLPs can be viewed as learning of parameter estimation or structural learning. In this paper we focus on the parameter estimation task, as we want to study the performance of SLPs-based ensembles to obtain the rate information of reactions in metabolic pathways. In order to learn the parameters over SLPs we employed failure adjusted maximisation (FAM) as a base learner.

The combination of boosting with Inductive Logic Programming (ILP) has been pioneered by Quinlan [14]. Recently Dutra et al. [15] have investigated bagging in Inductive Logic Programming. However ensemble methods have never been applied in conjunction with SLPs. This is the first combination of ensemble learning with SLP learning.

We adapt bagging in SLPs to perform rate estimation task in metabolic pathways. We also present another method, ranbag, to construct an ensemble by driving randomness into a learning algorithm. Bagging and ranbag obtain predictors from FAM. The final estimation is obtained by computing the average of the outputs of all the base predictors. We evaluate SLPs-based bagging and ranbag to modelling metabolic pathway of Saccharomyces cerevisiae. The results show that it is useful to apply ensembles for learning SLPs to modelling metabolic pathways.

The paper is organised as follows. A brief overview of metabolic pathways has been given in Sect. 2. Section 3 explains SLPs and FAM. In Sect. 4 we have described bagging and ranbag. Section 5 explains experimental results.

2 Metabolic Pathways

Genomic data is now being obtained on an industrial scale. Complete drafts of the human genome were published during 2001 [16, 17]. Projects are under way to sequence the genomes of the mouse, rat, zebra fish and puffer fishes *T. nigoviridis* and *Takifugu rubripes*. The focus of genome research is moving to the problem of identifying the biological functions of genes. An application of inductive logic programming to functional genomics is described in [18].

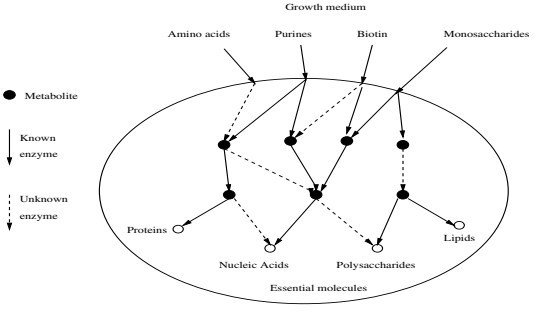

Fig. 1. Illustration of Cell with metabolic network involved in converting growth media into molecules essential for life

Figure 1 illustrates the way in which a network of metabolic reactions within a cell convert a growth medium (input compounds) into molecules, which are essential for life (output compounds). Each intermediate compound (associated with nodes of the graph) is known as a metabolite and each metabolic reaction is mediated by an enzyme (arcs in the graph). Presently not all enzymes involved in metabolic reactions are known. Online databases such as KEGG[1], WIT[2] and BRENDA[3] describe relationships between tens of thousands of enzymes. Measuring the rates of catalysed reactions can identify enzymes. Michaelis-Menten (MM) framework can be used to estimate reaction rates of enzymes. In order to compute these rates, using MM equation, information can be obtained from online database discussed above. These databases may not contain all the required information. Alternatively enzyme kinetics can be studied using learning techniques such as SLPs-based bagging and ranbag. SLPs provide an ideal representation for such data as rates can be viewed as probabilities, which can capture rates as proportions. In this way, the rates of enzymatic reactions can be estimated by computing the parameters over SLPs. In other words SLPs-based ensemble methods provide an efficient way to study enzyme kinetics.

Metabolic pathways have been studied using mathematical models that comprise sets of ordinary differential equations (ODEs). ODEs model the dynamic

[1] http://www.genome.ad.jp/kegg/
[2] http://wit.mcs.anl.gov/WIT2/
[3] http://www.brenda.uni-koeln.de/

proprieties of enzyme-catalysed reactions. There exist simulation tools for ODE cellular modelling. Gepasi [19] and DBsolve [20] are example of such simulation software. In order to handle non-deterministic characteristics of biological system, stochastic techniques such as Next Reaction Method [21] have been developed. In order to apply learning methods such as bagging and ranbag to ODE models or stochastic models, we need base learner that can learn these models. Once a sequence of ODE models or stochastic models is obtained from base learner, bagging-based methods can be applied to generate the final model. The final model that is a combination of individual ODE models or stochastic models can have a higher accuracy than any of the individual base models.

3 Stochastic Logic Programs for Biological Domains

Stochastic logic programs (SLPs) extend standard logic programs in order to represent probabilistic knowledge. SLPs have also been used to provide distribution for sampling the data [22]. In this way the application of SLPs are twofold, they not only provide a way for efficient sampling but also represents complex uncertain knowledge.

Syntax for SLPs: An SLP is a definite labelled logic program. In an SLP all or some of the clauses are associated with probability labels (parameters) and is known as pure SLP or impure SLP respectively. An SLP is said to be normalised if label of the clauses with same predicate symbol in the head sum to one and unnormalised otherwise. Formally an SLP S is a set of labelled definite clauses $p : C$ where $p \in [0, 1]$ is a probability label or parameter and C is a range-restricted definite clause. In this way an SLP provides an efficient representation to model metabolic pathways, where the set of clauses can describe enzymes and probability labels depict rates of reactions in metabolic pathways. Figure 2 shows the syntax of SLPs.

SLP	eg.
Labelled Definite Clause	0.3: like(X,Y) ← pet(Y,X)
Labelled Program (impure)	0.3: proteinfold1(X) ← ..
	0.3: proteinfold2(X) ← ..
Labelled Program (normalised)	0.5: coin(head) ←
	0.5: coin(tail) ←

Fig. 2. Syntax for SLPs

Semantics for SLPs: The semantics for SLPs is illustrated in Fig. 3. SLPs have a distributional semantics, that is one, which assigns a probability distribution to the atoms of each predicate in the Herbrand base of the underlying (unlabelled) logic program. An interpretation M is a model of an SLP S if all the atoms a have a probability assigned by M which is at least the sum of the probabilities of derivations of a with respect to S.

SLP	eg.
Distributional Interpretation	0.3: p(a) .. 0.4: q(a) ..
Distributional Model	0.3: p(a) .. 0.3: q(a) ..
$P \models Q$	0.3: q(a), 1.0: p(X) ← q(X) $\models$ 0.3: p(a)

Fig. 3. Semantics for SLPs

SLP	eg.
SSLD derivation	{0.3: q(a), 1.0: p(X) ← q(X) } refutes goal 0.3: ← p(a)

Fig. 4. Proof for SLPs

Proof for SLPs: Figure 4 shows proofs for SLPs. Probabilities are assigned to atoms according to an SLD-resolution strategy which employs a stochastic selection rule. Derivations can be viewed as Markov chains in which each stochastic selection is made randomly and independently. Thus the probability of deriving any particular atom a is the sum of products of the probability labels on the derivations of a.

Failure Adjusted Maximization (FAM)- An Example of Learning Methods for SLPs

Expectation Maximization (EM) [23] is a well-known maximum likelihood parameter estimation technique. EM is an iterative algorithm that performs the parameter estimation task from incomplete data. Failure adjusted maximization (FAM) [24] uses EM algorithm to compute maximum likelihood estimates for pure, normalised SLPs. In order to apply EM algorithm for parameter estimation task a complete dataset of atoms has its natural representation as incomplete dataset of atoms. A set of atoms yielding the refutation from a complete dataset of derivations makes an incomplete dataset of atoms.

Given a logic program and a set of initial (prior) parameters FAM computes the maximum likelihood estimates in a two step (expectation step and maximization step) iterative learning process. In the expectation step FAM computes the weighted contribution of the clause to deriving a data point and weighted contribution of the clause due to failed derivation. In the maximization step the contribution of the clause is maximised. The value associated with each clause is normalised and becomes an input for the next iteration of FAM. In this way, at each iteration FAM improves the current estimates of the parameters. This process is repeated till convergence. In this paper we applied FAM as a base learner to compute the reaction rate of metabolic pathways.

4 Bagging and Variants

Ensemble methods such as bagging and ranbag work by repeatedly calling a base learner to produce a series of predictors. The final predictor is a combination of individual predictors and generally has a higher accuracy than any of the individual base learners. Although this higher accuracy is due to uncorrelated errors among the base predictors, the following factors also contribute to the success of the ensemble methods. 1) The base learner is too simple (weak) to generate a hypothesis with low error. A predictor that is a combination of these hypotheses can have high accuracy. 2) The base learner is unstable such as decision trees, neural network, an inductive logic programming algorithm and a learning algorithm for SLPs. For an unstable base learner a small change in the learning set significantly affects the generated predictor. 3) The base learning algorithm suffers from some problems (employing search strategies that are not good enough to select a good hypothesis) that can be overcome using a combination of predictors generated by these learning algorithms.

We can view the learning process of bagging and ranbag as comprising two stages. In the first stage base predictors are generated and in the second stage these predictors are combined.

Bootstrap Aggregating (Bagging): We now describe how we have adapted bagging for learning the parameters over SLPs to modelling metabolic pathways. Bagging is based on the idea of resampling and combining. In order to obtain a predictor from the base learner, bagging provides the base learner with bootstrap replicates [25] of the learning set. A bootstrap replicate is constructed by randomly drawing, with replacement, n instances from the learning set of size n. These instances are drawn according to a uniform distribution that is kept on the learning set. The bootstrap replicate may not contain all of the instances from the original learning set and some instances may occur many times. On average bootstrap replicate contain 63.2% of the distinct instances in the learning set.

Require:
Learning Set: $L = \{x_1, , \ldots, x_n\}$ where $x_i \in X$.
A base learner that takes an underlying logic program representing enzymes in a metabolic pathway and a set of prior parameters that gives an initial guess of the rates of enzyme-catalysed reactions.
for $t = 1$ to T do
/* Generate bootstrap sample L^B from a learning set L */
/* Call the base learner with underlying logic program LP and prior parameters P_0. Set the prior parameters according to uniform distribution */
$h_t = BL(L^B, LP, P_0)$
end for
/* The bagged estimation is */
$h_{bag} = \frac{1}{T} \sum_{t=1}^{T} h(\hat{P})_i$

Fig. 5. Bagging for rates estimation in metabolic pathways

Pseudocode for bagging is given in Fig. 5. As input, bagging requires a learning set L of instances of the form $L = \{x_1, \ldots, x_n\}$. The instances are generated independently and identically according to the probability distribution D. In our setting, the learning set contains all the information that is required to measure the reaction rates of enzymes. In order to estimate these rates NMR or mass spectrometric data can be used as learning set. Alternatively an SLP representing metabolic pathway can be used to generate a learning set.

As described, bagging calls learning algorithm BL for T number of times. In order to perform rates estimation a learning algorithm such as FAM is specified. FAM is provided with a bootstrap sample L^B, an underlying logic program and a set of prior parameters (initial guess). Note that the prior parameters are set according to a uniform distribution. For a particular bootstrap sample the estimated parameters (predictions) are denoted by $h(\hat{P})$. This process of drawing a bootstrap sample and obtaining predictions is repeated for T times. The bagged prediction is obtained by computing the average of the outputs of all the base predictors. The bagged estimation for ith parameter is $h_{bag} = 1/T \sum_{t=1}^{T} h(\hat{P})_i$.

Random Prior Aggregating (Ranbag): In order to find out the reaction rates of enzymes in a metabolic pathway we introduce, ranbag, a variant of bagging. Ranbag performs the rates estimation task by driving randomness into FAM. In this ways ranbag is based on the idea of combining a set of diverse predictors. In order to obtain these predictors the prior parameters of FAM are set randomly. The obtained base predictors can be substantially diverse as FAM depends on the selection of prior parameters. Pseudocode for ranbag is given in Fig. 6. As described, ranbag requires a learning set L and a base learner that takes a set of prior parameters and underlying logic program. Let the learning set L represents the data containing the information required to calculate reactions rates of enzymes and the logic program LP is a set of clauses where each clause represents an enzyme. Let the prior parameters provide the initial guess for reaction rates. Ranbag calls learning algorithm BL such as FAM, for T number of iterations. At each iteration FAM is provided with same learning set L, and underlying logic program but prior parameters P_t are set randomly. The estimated parameters (predictions) are denoted by $h(\hat{P})$. This process of

Require:
Learning Set: $L = \{x_1, , \ldots, x_n\}$ where $x_i \in X$.
A base learner that takes an underlying logic program and a set of prior parameters.
for $t = 1$ to T do
/* Set the prior parameters P_t according to random distribution.
/* Call the base learner with underlying logic program LP and prior parameters P_t.
$h_t = BL(L^B, LP, P_t)$
end for
/* The final estimation is */
$h_{ranbag} = \frac{1}{T} \sum_{t=1}^{T} h(\hat{P})_i$

Fig. 6. Ranbag for the estimation of the rates of enzyme-catalysed reactions

setting the prior randomly and obtaining predictors is repeated for T times. The final estimation is obtained by computing the average of the outputs of all the base predictors. The final estimation for ith parameter is $h_{ranbag} = 1/T \sum_{t=1}^{T} h(\hat{P})_i$.

5 Experimental Analysis

In this section, we describe a series of extensive and systematic experiments. We empirically evaluated bagging and ranbag to calculate the rates of enzyme-catalysed reactions in metabolic pathways.

Datasets. We applied bagging and ranbag to modelling aromatics amino acid pathway of Saccharomyces cerevisiae (baker's yeast, brewer's yeast) [26]. We compared modelling performance of bagging with ranbag's performance. Figure 7 shows the aromatic amino acid pathway of yeast. SLPs naturally represent metabolic pathways as they can capture the rate information by way of probabilities (parameters over SLPs). We used the implementation of FAM available at[4]. The metabolic pathways have been represented by an SLP comprising of 21 stochastic clauses. Each clause of SLP provides the probabilistic information about the occurrence and non-occurrence of a reaction. In this way an SLP represents a metabolic pathway and tell the reactions' rates. Furthermore, modelling performance of bagging and ranbag has also been evaluated where a branch has been added in the metabolic networks. A branching metabolic networks is obtained by adding a branch in the same metabolic pathway (shown in Fig. 7). This phenomenon provides us with a new SLP.

As discussed, SLPs provide an efficient way for sampling the data, we generated the data using SLPs for both the chain and branching metabolic pathways. In our experimental setting the two SLPs represent chain and branching scenarios and two datasets are generated using these SLPs. These dataset hereafter are referred to as Chain dataset (non-branching metabolic network) and Branch dataset (branching metabolic network).

Experimental Methodology. Bagging and ranbag obtain base predictors from an SLP learning algorithm, FAM. The coordinates described below can control the performance of FAM and ensembles.

Convergence Criteria: FAM allows specifying the convergence criterion. We set it the log likelihood.

Prior: This corresponds to prior (initial) parameters of FAM. The prior parameters can be set randomly or uniformly. We set the prior parameters of FAM according to a uniform distribution for bagging. The prior has been set according to random distribution for ranbag.

Stopping Criterion: In bagging and ranbag we specify the number of base models T. We set the number of models T to 100.

[4] http://www-users.cs.york.ac.uk/~nicos/sware/

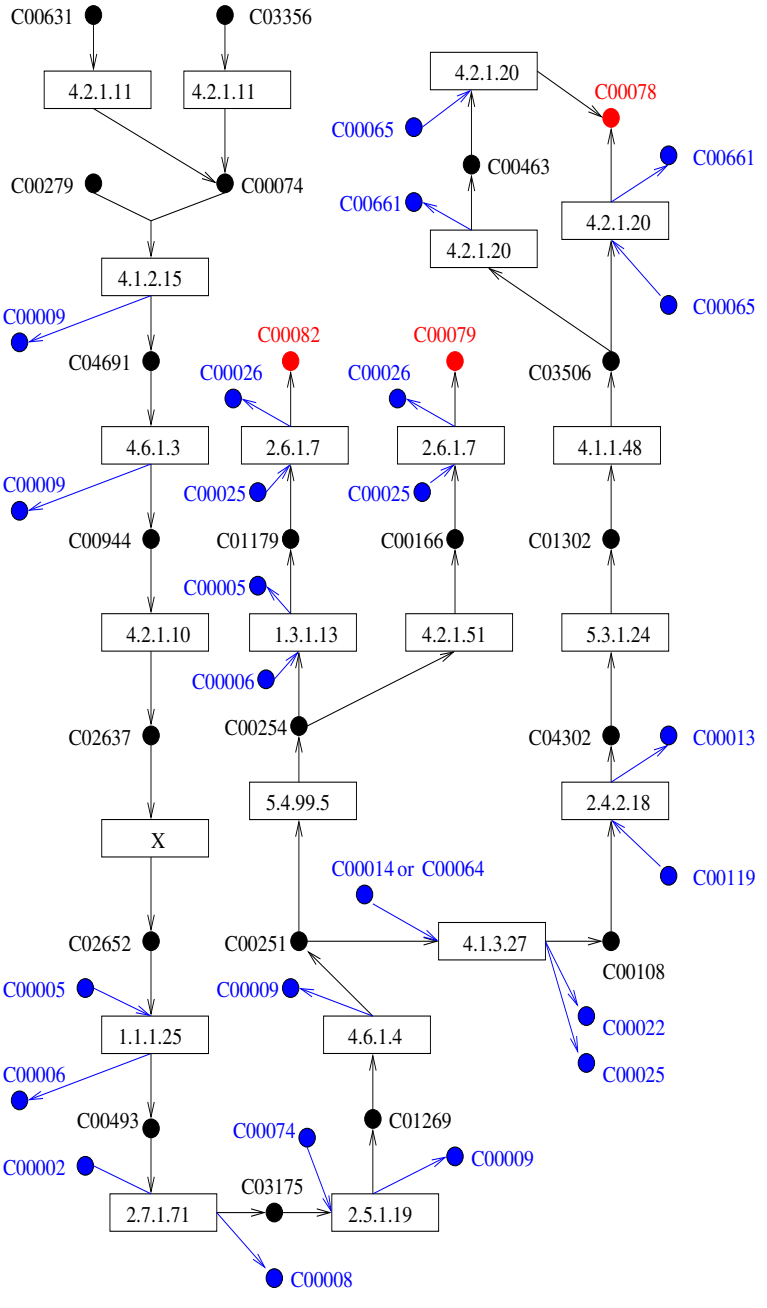

Fig. 7. The aromatic amino acid pathway of yeast. A chemical reaction is represented by a rectangle with its adjacent circles where rectangles represent enzymes and circles represent metabolites. In this figure metabolites are labelled by their KEGG accession numbers and enzymes by the EC number

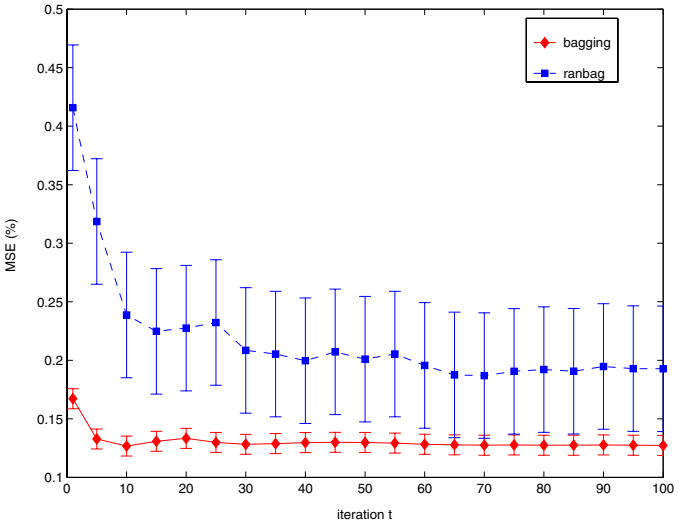

Fig. 8. MSE for non-branching metabolic pathway

As part of our experimental methodology we sampled chain and branch dataset where each dataset comprising of 1000 instances. The experiments were performed 10 times using 10 different sets of chain and branch data.

Evaluation Measure: We used mean squared error (MSE) and Kullback-Leiber (KL) divergence performance measures to estimate the goodness of bagging and ranbag. MSE is given by, $MSE = \frac{\sum_{i=1}^{N}(p_i - \hat{p_i})^2}{N}$, where p_i are the parameters need to be estimated and are termed the true parameters and $\hat{p_i}$ are estimated parameters and N is the total number of parameters. KL divergence to true parameters is given by $KL = \sum_i p_i \log(\frac{p_i}{\hat{p_i}})$

Results. Figure 8 through figure 11 show the results of the experiments. Figure 8 and figure 10 represent the MSE for non-branching and branching metabolic pathway. Figure 9 and figure 11 show the KL divergence to true parameters. The results are averaged over 10 runs of the method. These figures demonstrate how the solution improves with iterations. The performance of bagging and ranbag varies by combining models or predictors. The results show that KL divergence to true parameters and MSE drops to a minimum value by combining a number of diverse models. In other words bagging and ranbag improves the performance of an SLP learning algorithm. The results are described in detail in next paragraphs.

We first consider the non-branching metabolic pathway (chain dataset). The average error for a single model is 0.17% for bagging and 0.29% for ranbag. The results show that both bagging and ranbag improves the performance. Average error for bagging is 0.12% and average error for ranbag is 0.14%. The curves

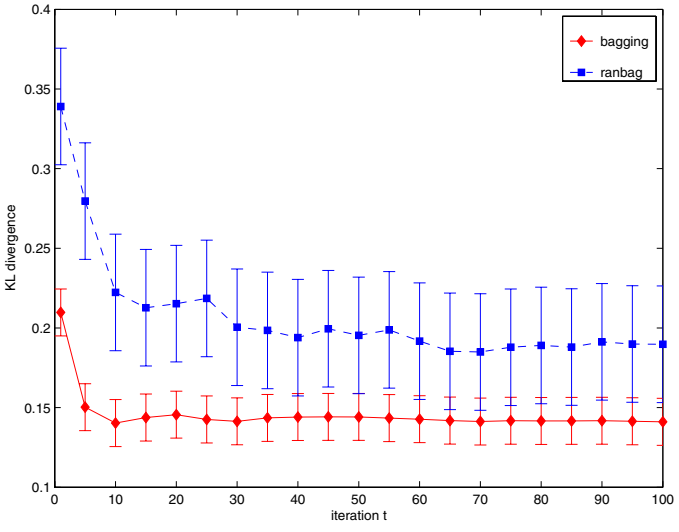

Fig. 9. KL divergence for non-branching metabolic pathway

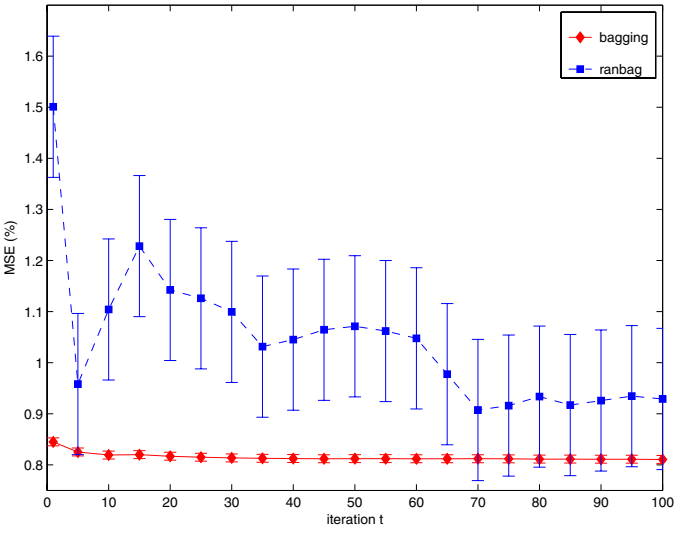

Fig. 10. MSE for branching metabolic pathway

show that both bagging and ranbag achieve substantial improvements. Bagging obtains the improvements within first 15 iterations and ranbag achieves the maximum gain by combining 70 models. The curves also show that after some initial iterations there is no significant improvement in the performance of bagging but ranbag does not show this phenomenon. Figure 9 tells the KL divergence to

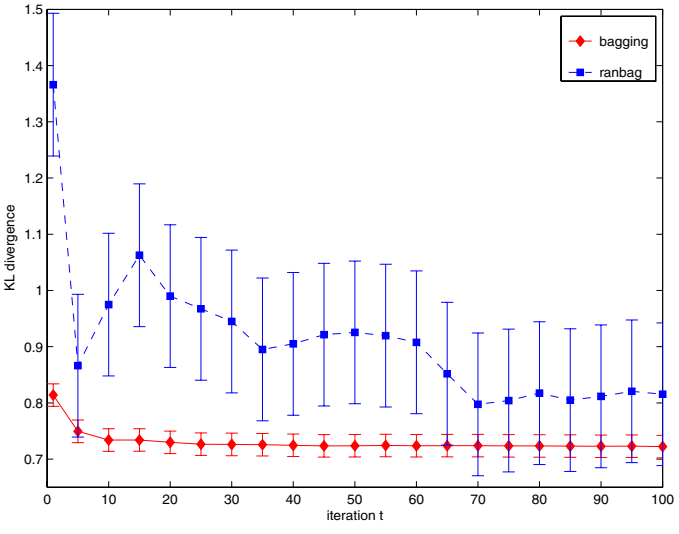

Fig. 11. KL divergence for branching metabolic pathway

true parameters for bagging and ranbag. The curves validate the effectiveness of bagging and ranbag to modelling metabolic pathways.

Our observation is that we can learn substantially diverse models by setting the priors of FAM randomly. Our second observation is that by increasing the number of iteration, error for ranbag and bagging decreases reaching a minimum and becomes stable. In this way, we obtain substantial gain using bagging and ranbag as compare to single model. In terms of KL divergence and MSE bagging shows better performance than ranbag. It is worth noting that average error for individual models for ranbag is considerably higher than the average error for individual models for bagging. It seems that FAM displays a bias in favour of uniform prior. Bagged model has been constructed by setting the prior parameters of FAM according to a uniform distribution whereas prior has been set according to a random distribution for ranbag. Hence, bagging achieves better modelling performance than ranbag due to FAM's bias in favour of uniform prior.

We now describe the results for branching metabolic pathway (branch dataset) for bagging and ranbag. We study the behaviour of bagging and ranbag in conjunction with FAM in a scenario where an alternative path has been added to a pathway. Figure 10 shows MSE for bagging and ranbag. The average MSE for single model for bagging is 0.85% and 1.5% for ranbag. Average error for bagging is 0.81% that is obtained by combining 25 models. Ranbag obtains minimum average MSE of 0.9%. The results show that bagging improves the performance of a single model but the improvement is not substantial. Success of bagging-based methods depends on the fit of the style of the function with the particular data and base learner. It seems that bagging is unable to achieve substantial improvement in performance due to underlying SLP where dataset has also been

generated using SLP. The average MSE for ranbag is substantially better than the average MSE for a single model for ranbag. However average MSE for ranbag is not better than the average MSE for single model for bagging. As discussed in the preceding paragraph FAM's bias in favour of uniform prior seems a cause of the occurrence of this phenomenon. Furthermore, underlying SLP can also account for this phenomenon. Ranbag's performance can be improved by selecting individual models on the basis of some statistical test. Figure 11 represents KL divergence to true parameters for bagging and ranbag. Average KL divergence for single model is 0.81% for bagging and 1.4% for ranbag. Bagging and ranbag substantially minimises KL divergence to true parameters. Average KL divergence for bagging is 0.72% and 0.79% for ranbag.

The results demonstrate the effectiveness of ensemble methods. Sometimes we obtain only a modest gain applying bagging in conjunction with FAM. This happens especially for MSE. The parameter learning process of FAM is influenced by the underlying SLP. It seems that FAM's ability to learn good starting model due to underlying SLP, and FAM's bias in favour of uniform prior account for the modest gain for particular dataset. In summary it is effective and efficient to apply bagging and ranbag to modelling metabolic pathways.

6 Conclusion

Bagging is a useful learning technique especially for unstable predictors. This paper presents a novel study of SLPs-based ensemble methods and addresses an important problem of modelling metabolic pathways. We have focused on the parameter estimation task over SLPs as reaction rates of enzymes in metabolic pathways can be computed by estimating the parameters over SLPs. We have shown how bagging can be adapted to perform maximum likelihood parameter estimation task. We have also shown that an effective ensemble can be constructed by driving randomness into an SLP learning algorithm. The empirical results demonstrate the efficacy of these methods and show that bagging and ranbag obtain substantial gain in performance. In terms of KL divergence and MSE these techniques show sizeable improvements in performance.

Deterministic models (such as ODE models) and stochastic models are popular for cellular modelling. We believe that application of bagging-based methods to ODE models and stochastic models will be effective and efficient. We are looking at the ways to develop base learners to learn ODE models and stochastic models and to apply them in conjunction with bagging-based methods.

SLPs provide an efficient representation for metabolic pathways where each clause of an SLP contains probabilistic information about enzyme-catalysed reactions. One of the important issues to be addressed in our future work is to learn really unknown parameters. In order to capture enzymatic reaction rates each clause of an SLP can be augmented by incorporating temporal information. One way to capture temporal dimension in stochastic logic framework is to add an expression specifying time as an argument in each clause of an SLP.

Acknowledgements

The authors would like to acknowledge the support of the DTI Beacon project "Metalog - Integrated Machine Learning of Metabolic Networks Applied to Predictive Toxicology", Grant Reference QCBB/C/012/00003. Thanks to Nicos Angelopoulos for technical help. Thanks to 2 anonymous reviewers for useful comments.

References

1. Dugglery, R.G., Clarke, R.B.: Experimental design for estimating the parameters of the Michaelis-menten equation from progress curves of enzyme-catalysed reactions. Biochim. Biophys. Acta **1080** (1991) 231–236
2. Breiman, L.: Bagging predictors. Machine Learning **24** (1996) 123–140
3. Schapire, R.E.: A brief introduction to boosting. In: Proceedings of the Sixteenth International Conference on Artificial Intelligence. (1999) 1401–1406
4. Dietterich, T.G.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomisation. Machine Learning **40** (2000) 139–157
5. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithm: bagging, boosting and variants. Machine Learning **36** (1999) 105–142
6. Lodhi, H., Karakoulas, G., Shawe-Taylor, J.: Boosting strategy for classification. Intelligent Data Analysis **6** (2002) 149–174
7. Schapire, R.E., Freund, Y., Barlett, P., Lee, W.S.: Boosting the margin: A new explanation for the effectiveness of voting methods. The Annals of Statistics **5** (1998) 1651–1686
8. Lodhi, H., Karakoulas, G., Shawe-Taylor, J.: Boosting the margin distribution. In Leung, K.S., Chan, L.W., Meng, H., eds.: Proceedings of the Second International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2000), Springer Verlag (2000) 54–59
9. Domingos, P.: A unified bias-variance decomposition for zero-one and squared loss. In: Seventeenth National Conference on Artificial Intelligence, AAAI (2000) 564–569
10. Muggleton, S.H.: Stochastic logic programs. In de Raedt, L., ed.: Advances in Inductive Logic Programming. IOS Press (1996) 254–264
11. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE **77** (1989) 257–286
12. Lari, K., Young, S.J.: The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language* **4** (1990) 35–56
13. Halpern, J.Y.: An analysis of first-order logics of probability. *Artificial Intelligence* **46** (1990) 311–350
14. Quinlan, J.R.: Boosting first-order learning. In Arikawa, S., Sharma, A., eds.: Proceedings of the 7th International Workshop on Algorithmic Learning Theory. Volume 1160 of LNAI., Berlin, Springer (1996) 143–155
15. Dutra, I.C., Page, D., Shavilk, J.: An emperical evaluation of bagging in inductive logic programming. In: Proceedings of the International Conference on Inductive Logic Programming. (2002)
16. Venter, J.C., Adams, M.D., et al, E.W.M.: The sequence of human genome. Science **291** (2001) 1304–1351

17. Consortium, I.H.G.S.: Initial sequencing and analysis of the huma genome. Nature **409** (860–921)
18. Bryant, C.H., Muggleton, S.H., Oliver, S.G., Kell, D.B., Reiser, P., King, R.D.: Combining inductive logic programming, active learning and robotics to discover the function of genes. Electronic Transactions in Artificial Intelligence **6-B1** (2001) 1–36
19. Gepasi, M.P.: A software package for modelling the dynamics, steady states and control of biochemical and other systems. Comput. Appl. Biosci **9** (1993) 563–571
20. Goryanin, I., Hodgman, T.C., Selkov, E.: Mathematical simulation and analysis of cellular metabolism and regulation. Bioinformatics **15** (1999) 749–758
21. Gibson, M.A.: Computational methods for stochastic biological systems. PhD thesis, California Institute of Technology (2000)
22. Muggleton, S.H.: Learning from positive data. Machine Learning (2001)
23. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. J. Royal statistical Society Series B **39** (1977) 1–38
24. Cussens, J.: Parameter estimation in stochastic logic programs. Machine Learning **44** (2001) 245–271
25. Efron, B., Tibshirani, R.: An introduction to bootstrap. Chapman and Hall (1993)
26. Angelopoulos, N., Muggleton, S.: Machine learning metabolic pathway descriptions using a probabilistic relational representation. Electronic Transactions in Artificial Intelligence **6** (2002)