Machine Learning Metabolic Pathway descriptions using a Probabilistic Relational Representation

Nicos Angelopoulos and Stephen Muggleton

{nicos,shm}@doc.ic.ac.uk.

Imperial College, London.

structure

Structure of the talk

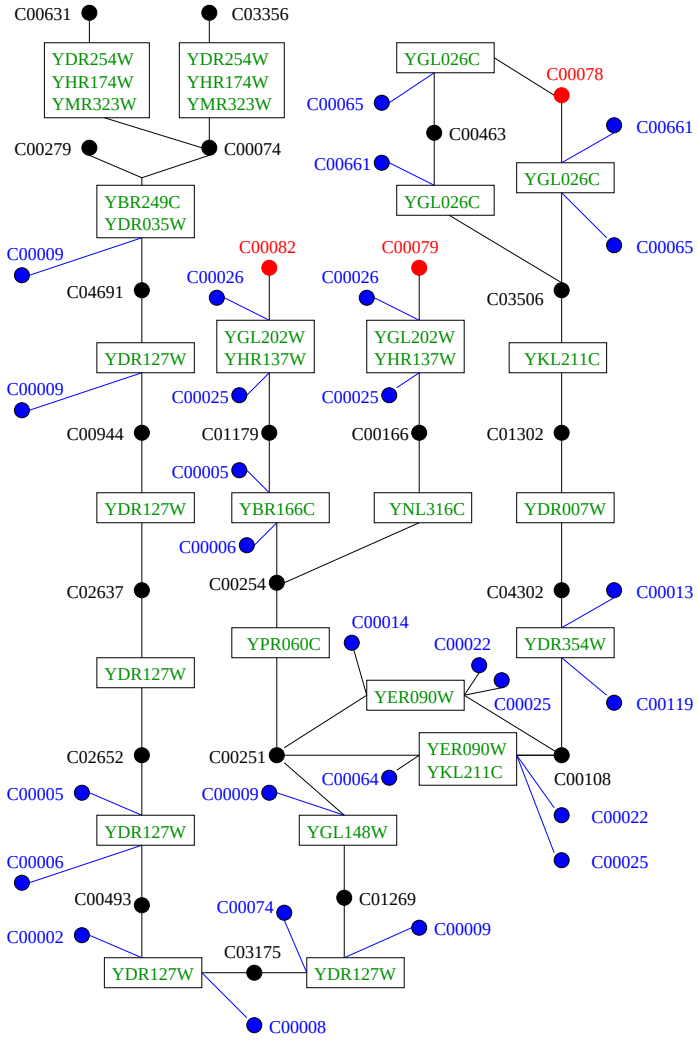
- motivation
  (pathways, learning, relational, probabilistic)
- Stochastic Logic Programs
- parameter estimation with FAM
- experiments with chain probabilistic pathway
- experiments with branching probabilistic pathway

pathways

Metabolic pathways

- represent biochemical reactions in the cell of organisms
- are publicly available in databases such as KEGG
- are cross-referenced with other data, such as gene sequences
- there are relationships across species due to evolution

aromatic amino acid



machine learning

Public databases are, almost by definition, incomplete and containing incorrect information.
Amongst other reasons incompleteness is due to:

- unknown enzymes
- lack of interest/resources for documenting secondary pathways

Machine learning can use observational data to

- revise
- augment
- verify

metabolic pathway descriptions. Of particular interest is the use of cross-species information

relational

Relational representations can express background knowledge at various levels of biological detail. The ability to incorporate existing knowledge enhances ability to learn.

For instance in metabolic pathways, additional knowledge might include

- physical properties of substrates and products for individual reactions
- the existence of required co-factors and absence of blocking inhibitors
- the availability of similar pathway in other cells

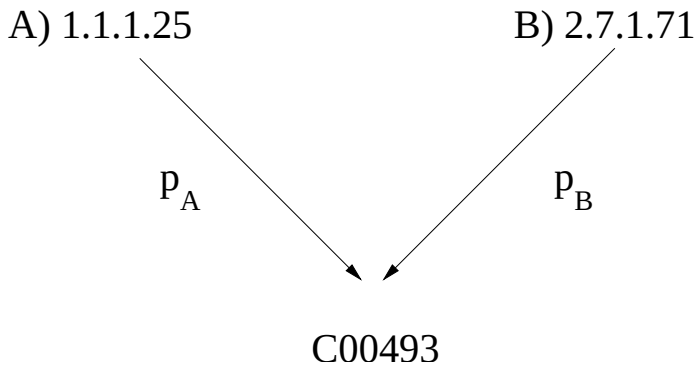
probabilistic

Various forms of uncertainty arise when modelling biological systems.
Two main sources are:

- competing biological processes
- lack of detail in the model

We consider two scenarios of extending metabolic pathways in these directions.

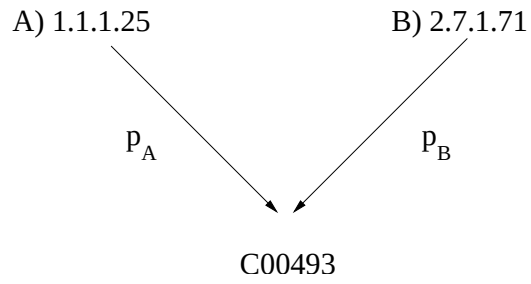
rates as probabilities



Pathways do not take into account the rates with which enzymes consume their substrates to produce metabolites. In the case of alternative production paths for a single metabolite it is impossible to distinguish the contribution of each path.

One way to model the difference in rates is by way of probabilities which captures the rates as proportions.

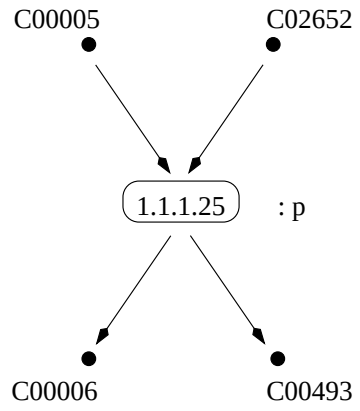
rates for probabilistic ML



Rate constants can be used in conjuction with Michaelis-Menten equation to derive this probabilities. However databases such as Brenda record very few rate constants.

ML techniques can be used to extrapolate these from experimental data.

lack of detail as probabilities



Due to a number of factors, such as physical chemistry, temperature, intracellular distance etc., reactions may not happen even if substrates are present.

Lack of detail in the model can then be modelled as probability on the event of the reaction happening.

rest of talk

- modelling lack-of-detail in SLPs
- parameter estimation with FAM
- experiments with chain probabilistic pathway
- experiments with branching probabilistic pathway
- conclusions

SLPs

A stochastic logic program, is a parameterised logic program. Each clause, of a probabilistic predicate, has attached to it a *parameter* (or *label*).

Example program

```
1/2 :: nat( 0 ).
1/2 :: nat( s(X) ) :- nat( X ).
```

It is *normalised* if the sum of the parameters for the clauses of each probabilistic predicate is equal to 1.

An SLP is *pure* if all its predicates are parametrised.

FAM, Cussens (2001)

Parameter estimation: estimate tuple of parameters $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ when given frequency of n observations $y = (y_1 - N_1, y_2 - N_2, \ldots, y_n - N_n)$ which are assumed to have been generated from $\mathcal{S}$ according to unknown distribution $p_{(\lambda,\mathcal{S},G)}$.

Failure Adjusted Maximisation is an EM algorithm, where adjustment is expressed in terms of failure observations. The expected frequency of a clause is:

$$\psi_\lambda[\nu_i \mid y] = \sum_{k=1}^{T} N_k \psi_\lambda[\nu_i \mid y_k] + N(Z_\lambda^{-1} - 1)\psi_\lambda[\nu_i \mid fail] \quad (1)$$

fam algorithm

1. Set $h = 0$ and $\lambda^{(0)}$ to some estimates such that $Z_{\lambda^{(0)}} > 0$
2. For parameterised clause C_i compute $\psi_{\lambda^{(h)}}[\nu_i \mid y]$ using (Eq. 1).
3. Let $S_i^{(h)}$ be the sum of $\psi_{\lambda^{(h)}}[\nu_{i'} \mid y]$ for all $C_{i'}$ of the same predicate as C_i.
4. If $S_i^{(h)} = 0$ then $l_i^{(h+1)} = l_i^{(h)}$ otherwise

$$l_i^{(h_1)} = \frac{\psi_{\lambda^{(h)}}[\nu_i \mid y]}{S_i(h)}$$

5. Increment h and go to 2 unless $\lambda^{(h+1)}$ has converged.

implementation

SLP clauses are transformed so that,

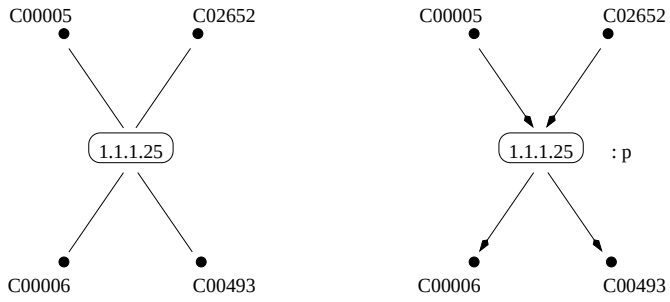
- identification is added to each clause
- probability of a derivation is returned
- the path of a derivation as a list of ids, is returned
- would-be failures simply set a flag and succeed
- curtail infinite or very long computations, by approximating their probability to zero

FAM on singular SLPs

Although FAM has been introduced for pure SLPs we applied it to a slightly more general class. Singular SLPs allow for impure/mixed SLPs in as far as that all derivations of a specific goal map to distinct stochastic path.

A stochastic path is a sequence of the used probabilistic clauses.

probabilistic pathways



Semantics of the attached probability are :
"Given the inputs are present, the reaction will happen with probability p."

Probability is attached to the reaction not to the enzyme.

assumptions

We have made two major simplifying assumptions

- reactions deplete their inputs
- each reaction is only considered, at most, once

simulation

We run simulated experiments in order to

- obtain estimates on required learning data-size
- observe behaviour of FAM

PE scenario

Our experiments observe the following pattern :

- an SLP with n true parameters $\lambda = \langle \lambda_1, \lambda_2, \ldots, \lambda_n \rangle$ is used to sample T samples
- sampling goal is $can\_produce(+Substrates, -Metabolites)$
- parameters replaced by uniformly distributed ones
- use FAM to obtain $\bar{\lambda} = \langle \bar{\lambda}_1, \bar{\lambda}_2, \ldots, \bar{\lambda}_n \rangle$

chain pathway

We have added direction and mock probabilities to the aromatic amino acid pathway and run the following two sets of experiments.

x	t	I_x^t	$S_x^{l^t}$	$S_x^{u^t}$	$S_x^{i^t}$
a	1	10	100	1000	100
	2	20	110	1010	100
b	1	5	100	3300	400
	2	10	110	3310	400

measures

FAM to observe two values

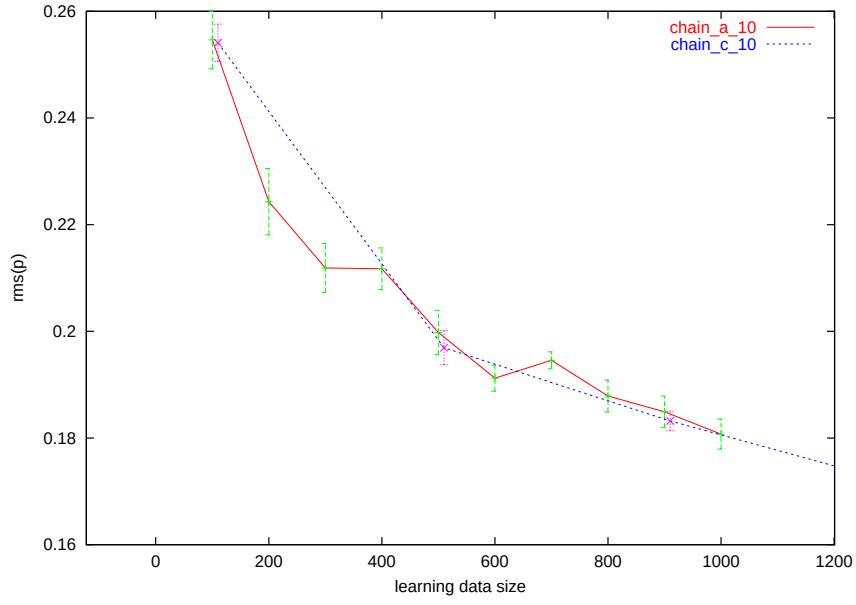
- accuracy, root mean square for parameters

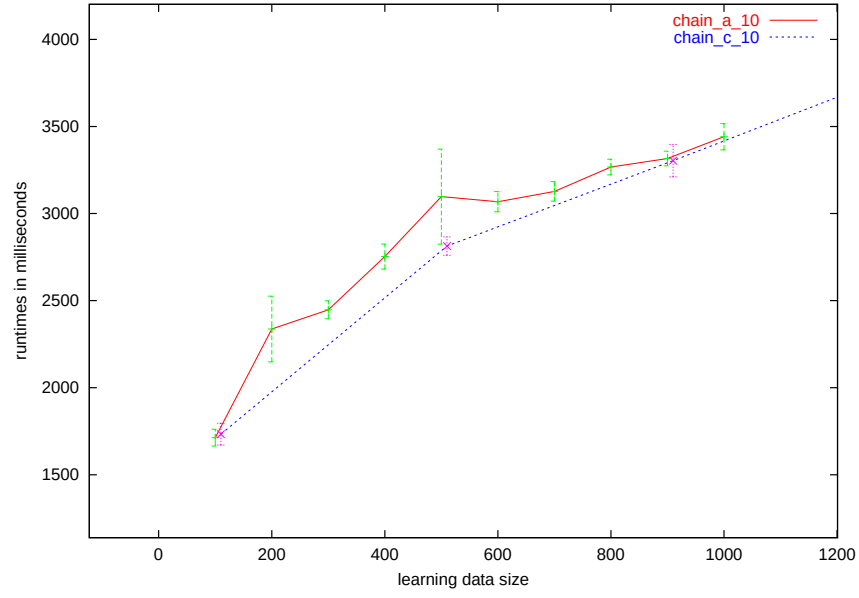
$$R_x^{t_i} = \sqrt{\left(\frac{\sum_{j=1}^{N} (p_j - \bar{p}_{(x,t,i,j)})^2}{n} \right)}$$

  and taking mean and sdv over t

- raw execution times for runtimes

chain plots


chain_a_10
chain_c_10
rms(p)
learning data size
0.26
0.24
0.22
0.2
0.18
0.16
0
200
400
600
800
1000
1200

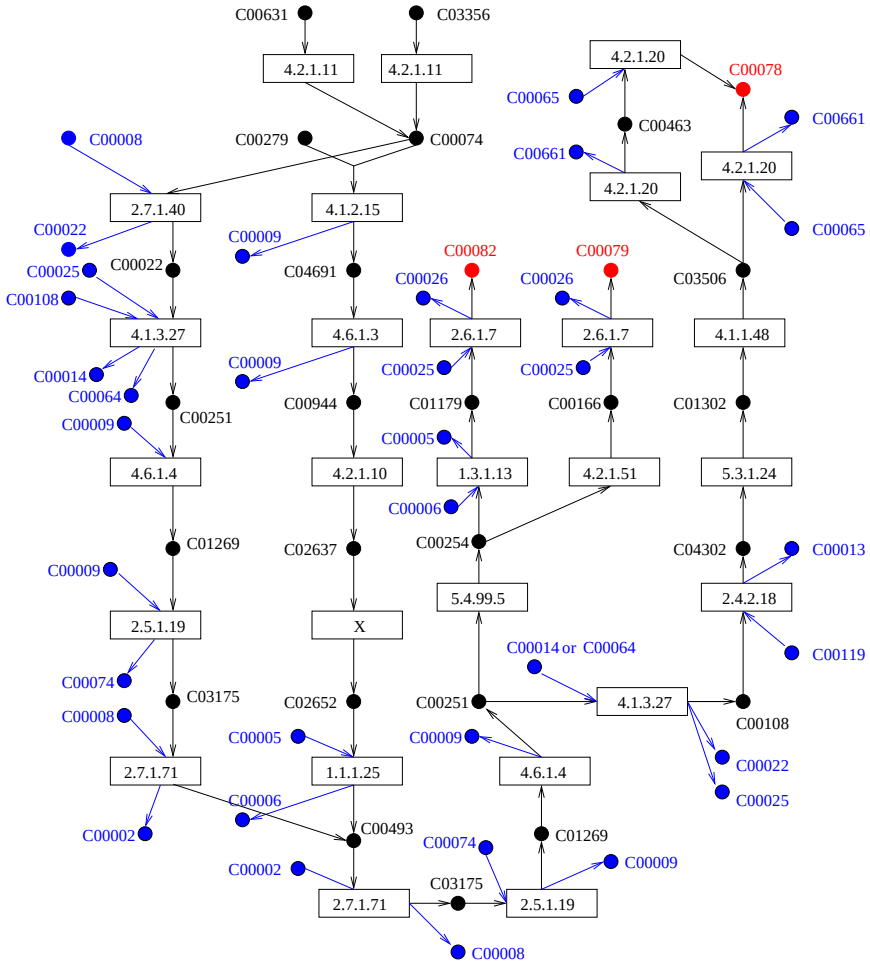

chain_a_10
chain_c_10
runtimes in milliseconds
learning data size
4000
3500
3000
2500
2000
1500
0
200
400
600
800
1000
1200

branching

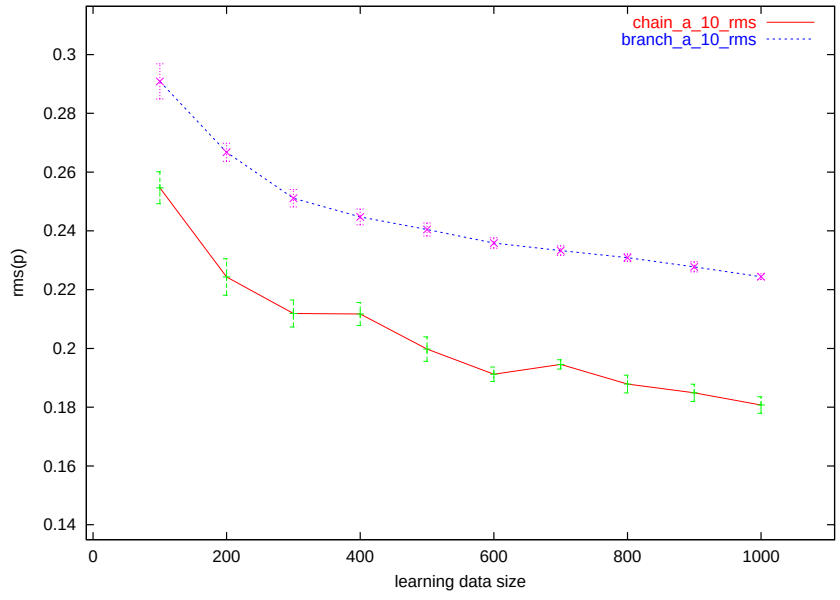
To compare the effect of secondary paths we, artificially, extended the pathway with an alternative path of length five, near the top of the graph.

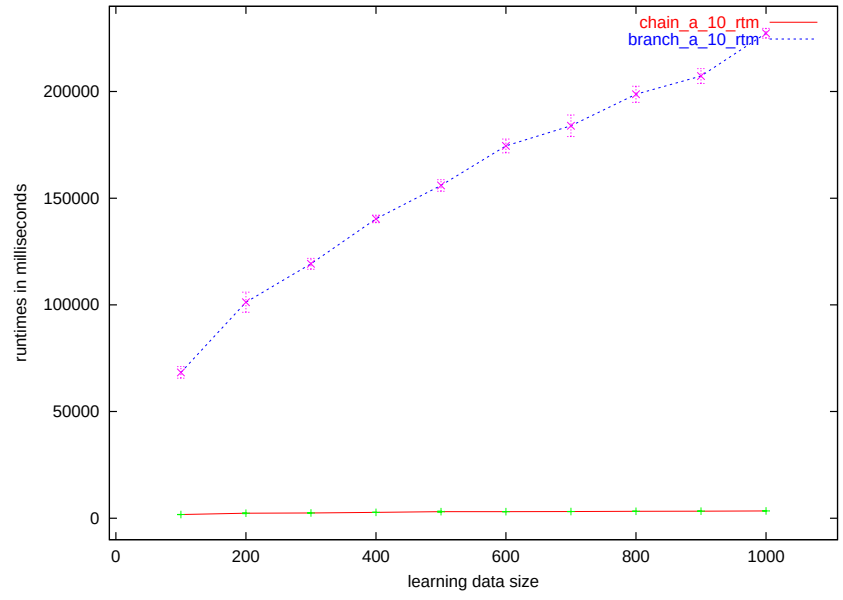
The secondary path only *fires* when there is a failure in the primary path.

artificial pathway



comparative plots





FAM future work

Improve efficiency by :

- storing expressions ($\sum_r \psi_\lambda(r)\nu_i(r)$) rather than
  (re-)doing the proofs at each iteration.
- simplification of such expressions (and their equivalence to graph reduction).

Extend algorithm to cover impure SLPs.

bottom line

Currently we have run FAM to get initial estimates on the data size required for learning actual parameters.

Machine learning tasks on probabilistic pathways :

- pathway completion
- pathway verification
- reaction rate estimation (different representation)