Inductive Logic Programming
Lecture 2.3
Stochastic Logic Programs and Bayesian Meta-Interpretive Learning

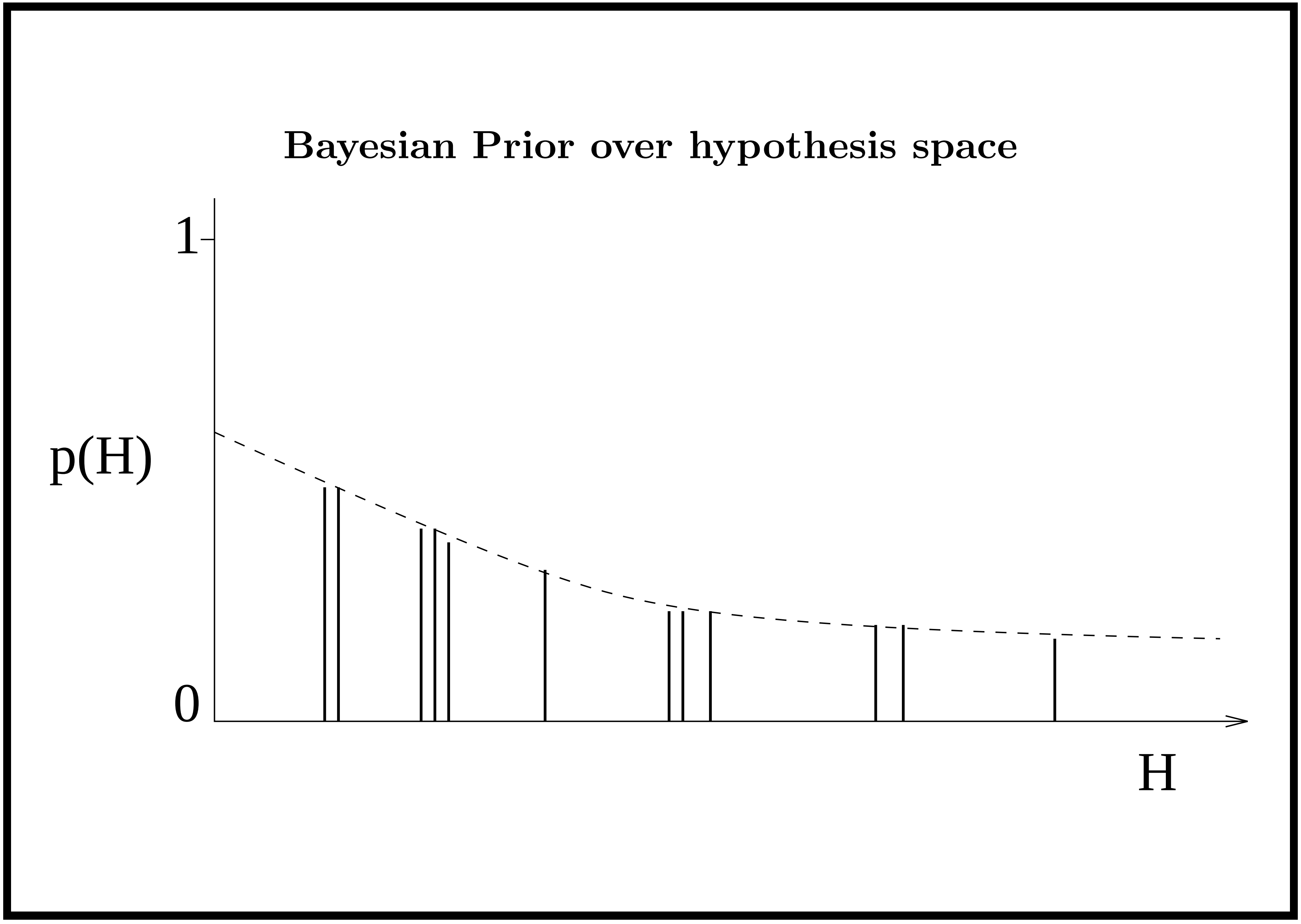
Stephen Muggleton
Department of Computing
Imperial College, London

5th July, 2018

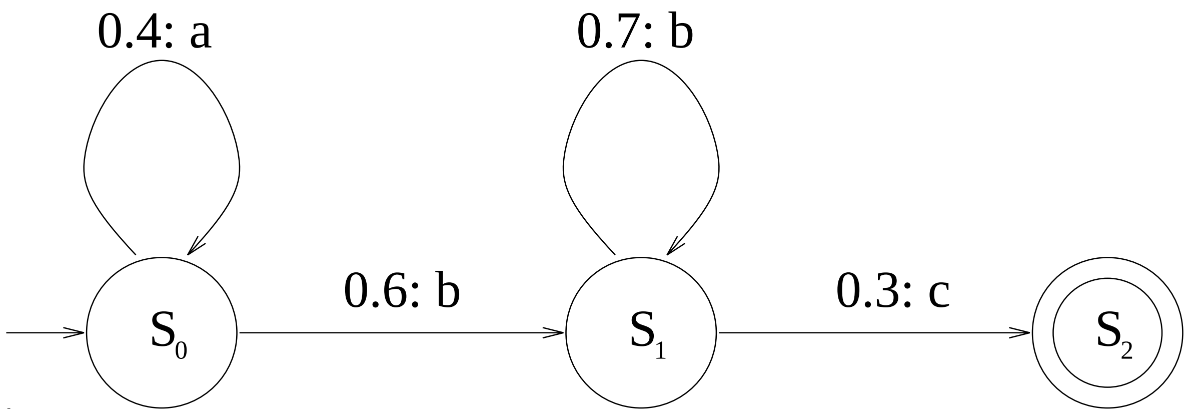
Paper for this lecture

Paper07: S.H. Muggleton. Stochastic logic programs. In L. de Raedt, editor, Advances in Inductive Logic Programming, pages 254-264. IOS Press, 1996.

Paper08: S.H. Muggleton, D. Lin, J. Chen, and A. Tamaddoni-Nezhad. Metabayes: Bayesian meta-interpretative learning using higher-order stochastic refinement. In Gerson Zaverucha, Vitor Santos Costa, and Aline Marins Paes, editors, Proceedings of the 23rd International Conference on Inductive Logic Programming (ILP 2013), pages 1-17, Berlin, 2014. Springer-Verlag. LNAI 8812.


Bayesian Prior over hypothesis space
1
p(H)
0
H

Stochastic automaton



$$Pr(abbc) = (0.4)(0.6)(0.7)(0.3) \simeq 0.05$$

$$\sum_{s \in L} Pr(s) = 1$$

Stochastic grammar

$$0.4 : S_0 \rightarrow aS_0$$
$$0.6 : S_0 \rightarrow bS_1$$

$$0.7 : S_1 \rightarrow bS_1$$
$$0.3 : S_1 \rightarrow cS_2$$

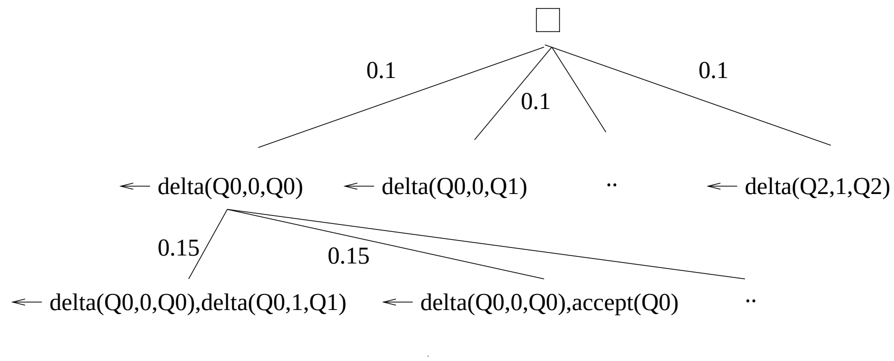
$$1.0 : S_2 \rightarrow \lambda$$

Stochastic logic program (SLP)

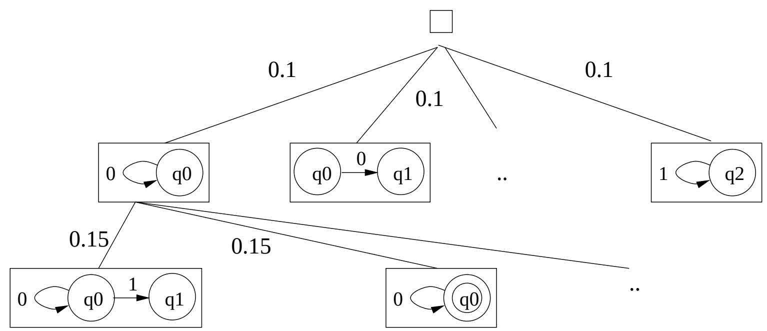
```
0.4: s0([a|X],Y) :- s0(X,Y).
0.6: s0([b|X],Y) :- s1(X,Y).

0.7: s1([b|X],Y) :- s1(X,Y).
0.3: s1([c|X],Y) :- s2(X,Y).

1.0: s2([],[]).
```

Stochastic refinement tree

Clauses



Finite State Acceptors (FSAs)



MetaBayes Refinement framework

Setting	$B, \neg E \models \neg H$
Meta-rule	$\exists \mathcal{S} \forall \mathcal{T} P(s_1, .., s_m) \leftarrow .., Q_i(t_1, .., t_n), ..$
Stochastic refinement	$\sigma^*(C) = \{\langle D_i, p_i \rangle \| D_i \in \rho^*(C),\ p_i \in [0,1]$ and $\sum p_i = 1$ for $1 \leq i \leq \|\rho^*(C)\|\}$.
Prior	$Pr(H\|B) = \sum_{\langle H,p \rangle \in \sigma^*(\neg B)} p$ and $Pr(H) = Pr(H\|\emptyset)$
Likelihood	$Pr(E\|B,H) = \begin{cases} 1 & \text{if } B, H \models E \\ 0 & \text{otherwise} \end{cases}$.
Posterior	$Pr(H\|B,E) = \frac{Pr(H\|B)Pr(E\|B,H)}{c}$

Generalised Meta-interpreter

```
prove([],Prog,Prog).
prove([Atom|As],Prog1,Prog2) :-
        metarule(RuleName,HO_Sub,(Atom :- Body),OrderTest),
        OrderTest,
        abduce(metasub(RuleName,HO_Sub),Prog1,Prog3),
        prove(Body,Prog3,Prog4),
        prove(As,Prog4,Prog2).
```

Meta-rules

FSA

```
metarule(acceptor ,[Q],([Q,[],[]]  :-  []), (nonterm(Q))).
metarule(delta ,[P,C,Q],([P,[C|X],Y]  :-  [[Q,X,Y]]),
        (nonterm(Q),nonterm(P))).
```

Dyadic

```
metarule(instance ,[P,X,Y],([P,X,Y]  :-  []),(pred(P))).
metarule(base ,[P,Q],([P,X,Y]  :-  [[Q,X,Y]]),
        (pred_above(P,Q), obj_above(X,Y))).
metarule(tailrec ,[P,Q],([P,X,Y]  :-  [[Q,X,Z], [P,Z,Y]]),
        (pred_above(P,Q), obj_above(X,Z), obj_above(Z,Y))).
metarule(chain ,[P,Q,R],([P,X,Y]  :-  [[Q,X,Z], [R,Z,Y]]),
        (obj_above(X,Z), obj_above(Z,Y))).
```

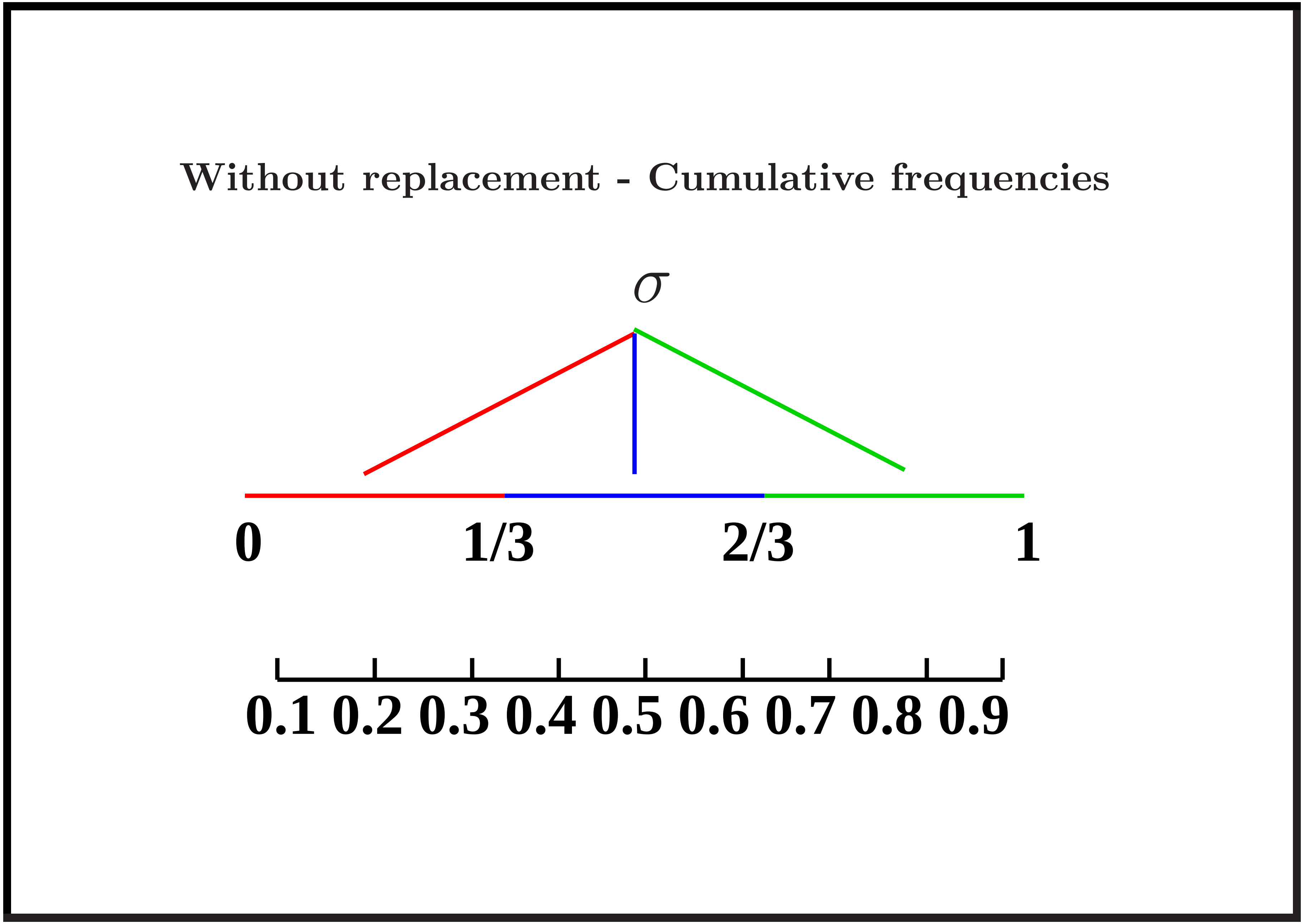
Implementation

Three variants of generalised meta-interpreter with uniform distribution stochastic refinement of meta-rules.

MetaBayes$_{SR}$ Model averaging prediction based on sampling hypotheses **with** replacement.

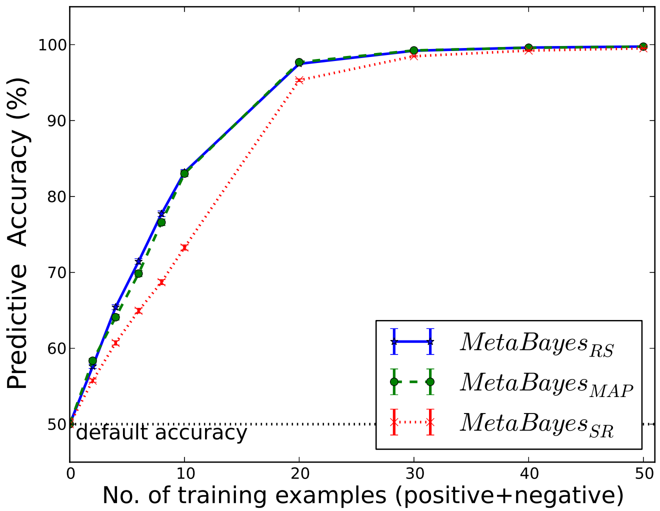
MetaBayes$_{RS}$ Model averaging prediction based on sampling hypotheses **without** replacement.

MetaBayes$_{MAP}$ Prediction based on leftmost maximum posterior hypothesis.

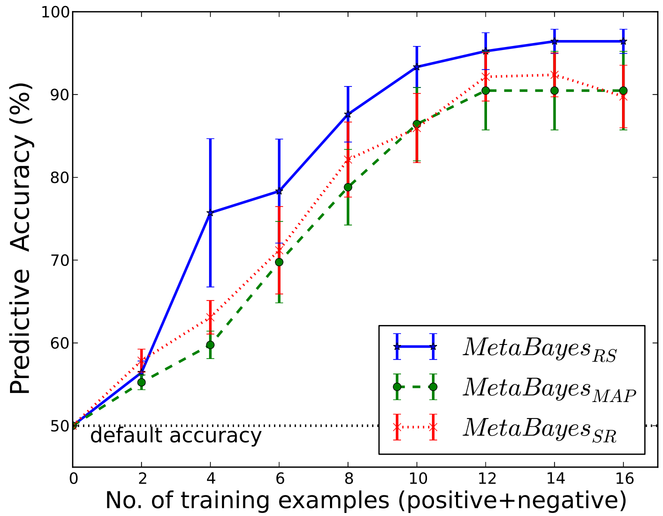


Without replacement - Cumulative frequencies
σ
0
1/3
2/3
1
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9

Experiments



Regular



Family

Related work

Bayes's prediciton optimal Bernardo & Smith (1994), Buntine (1990)

Error bounds Haussler, Kearns, Shapire (1994)

Ensemble methods Freund & Shapire (1997), Zhu and Zou (2009)

SRL and PILP Getoor (2005) and De Raedt et al (2008)

Summary and limitations

- SLPs generalise stochastic grammars
- Bayesian prior implemented as meta-interpreter over higher-order SLP
- Sampling allows approximation of Bayes' predictor
- Outperforms MAP
- Speed/accuracy tradeoff
- Noise and active learning further work