Tunable algorithms for transient follow-up

Tim Staley

TKP Meeting
Manchester, Sept 2014

Aim of this talk

A basic, intuitive understanding of

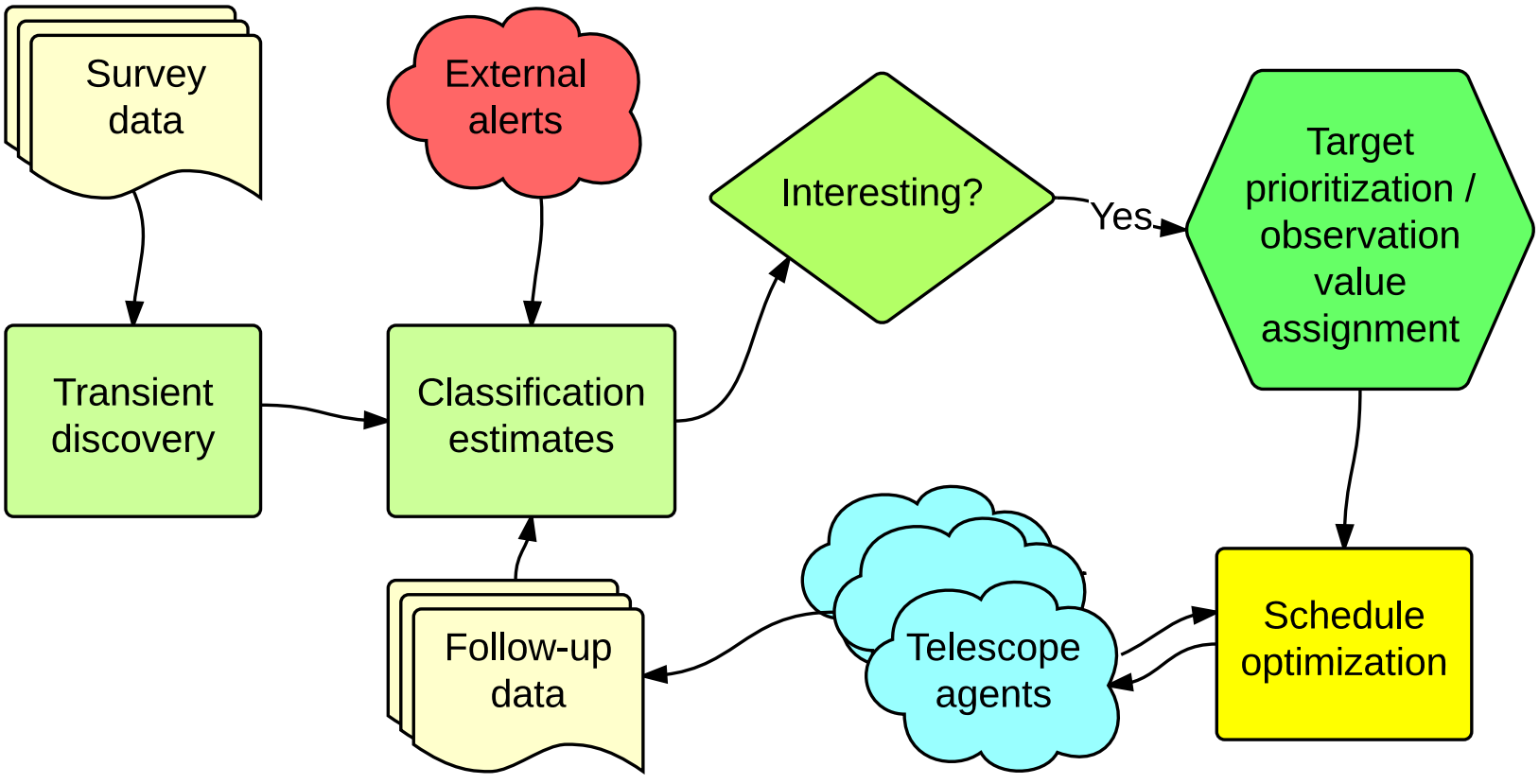
information content

and how this can be used to optimize / automate decision making, a.k.a.

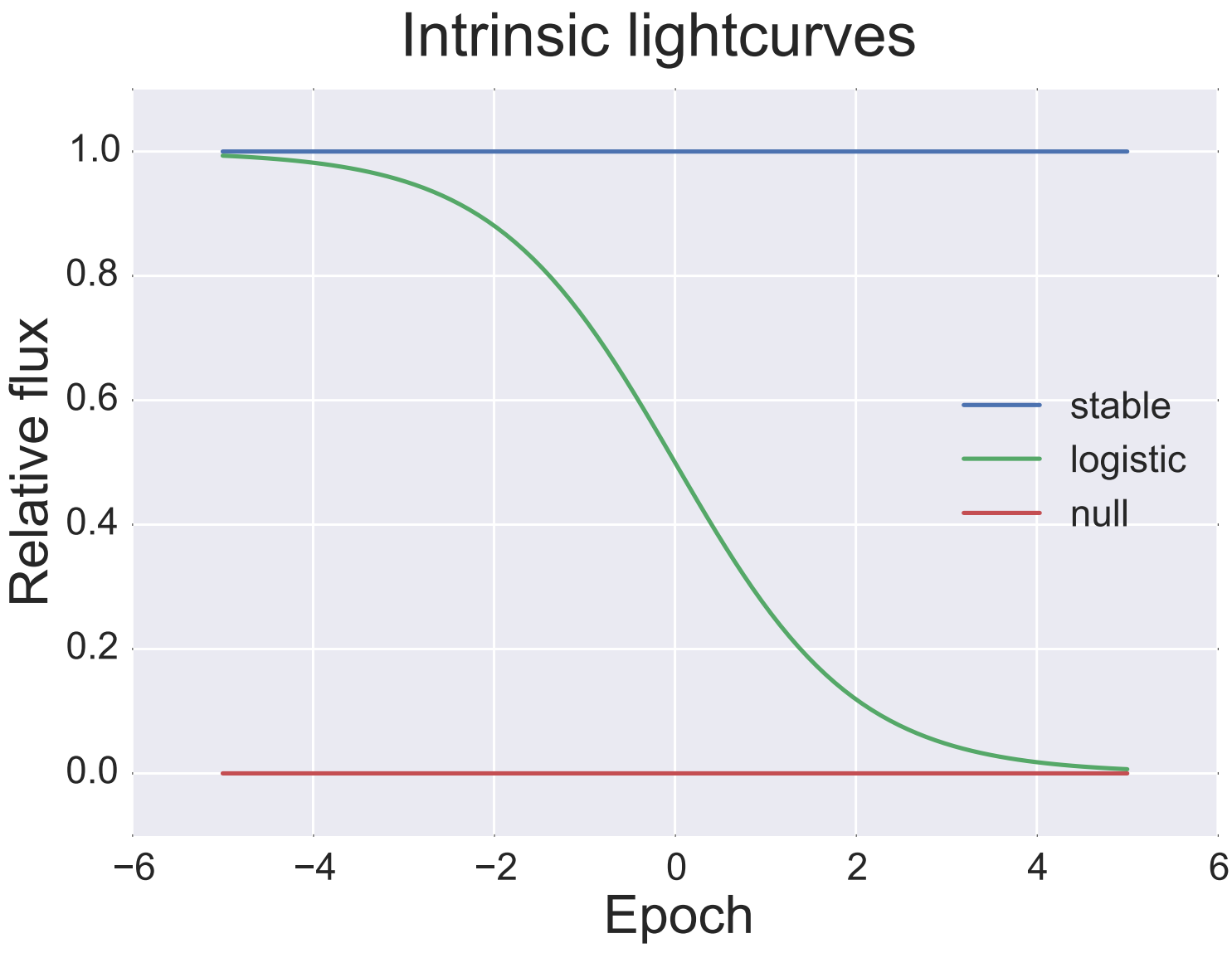
Bayesian decision theory

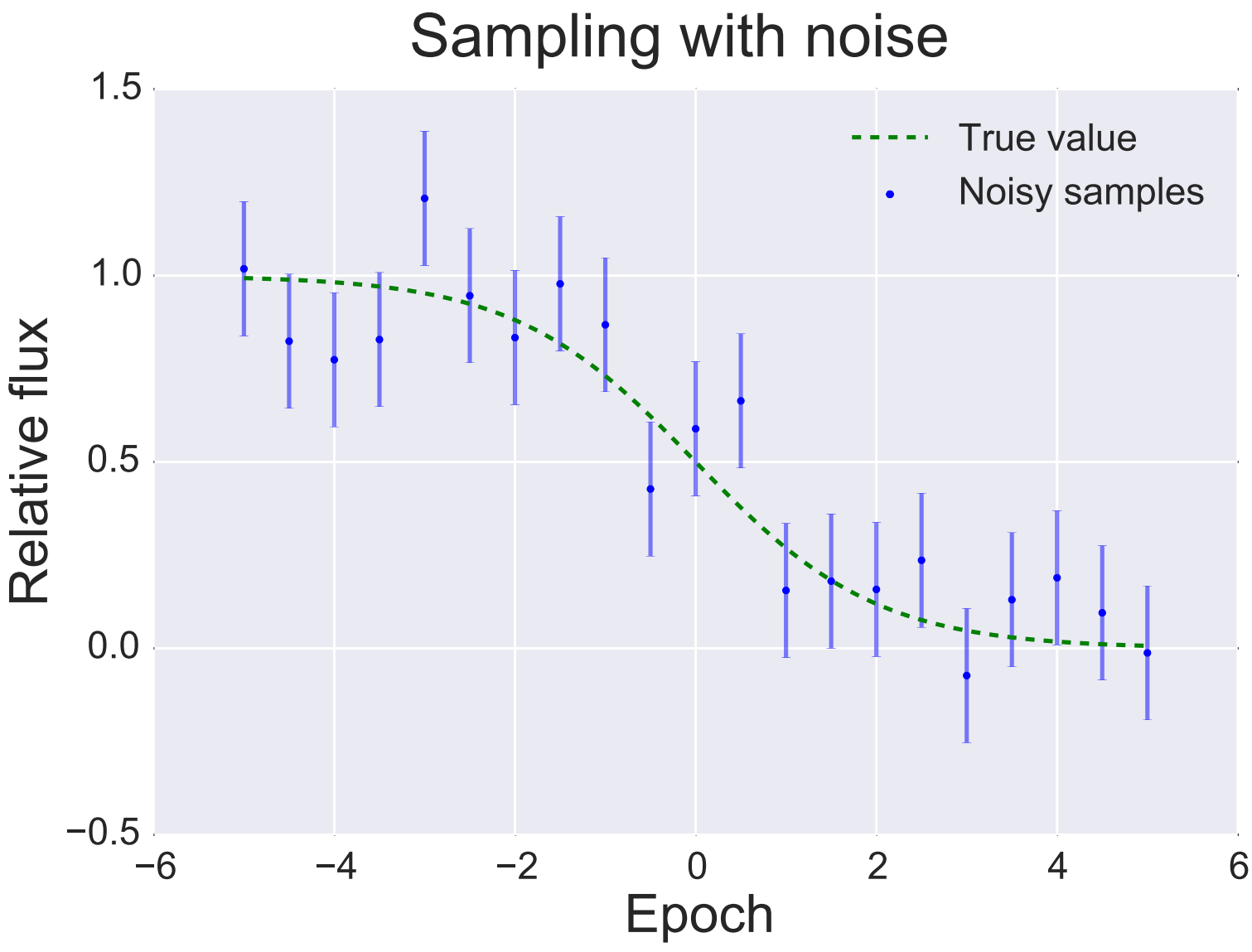
Outline

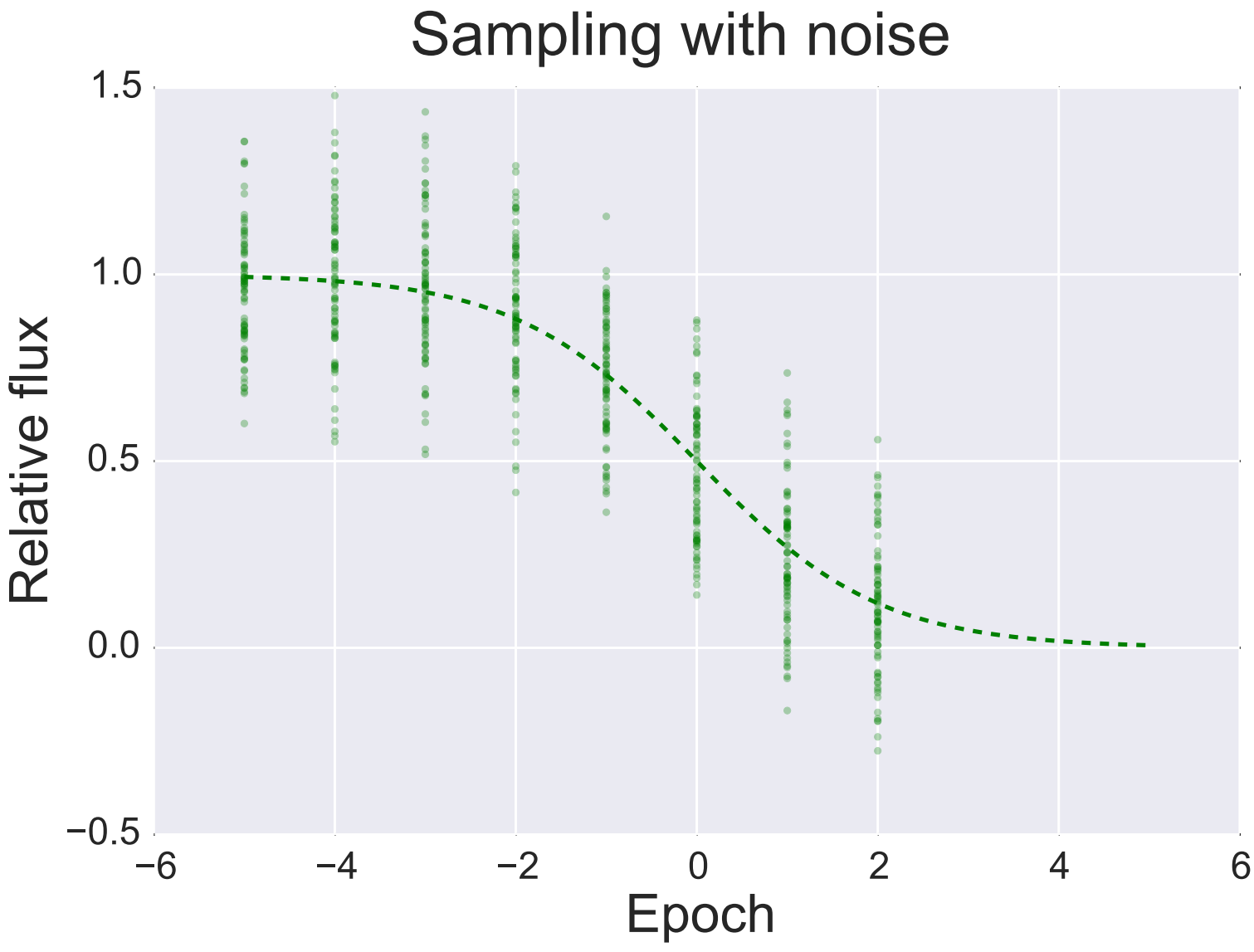
A blueprint for automated follow-up

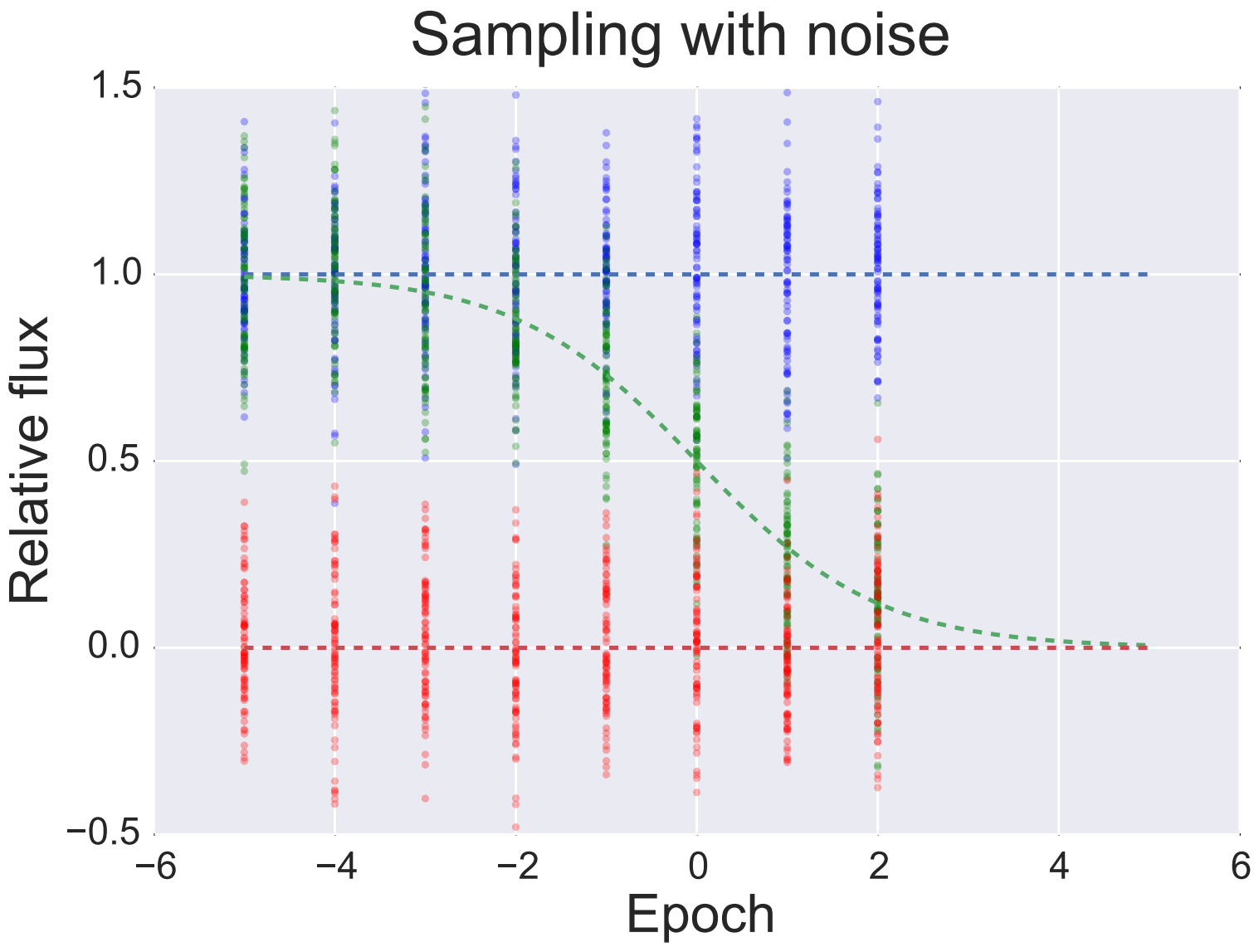


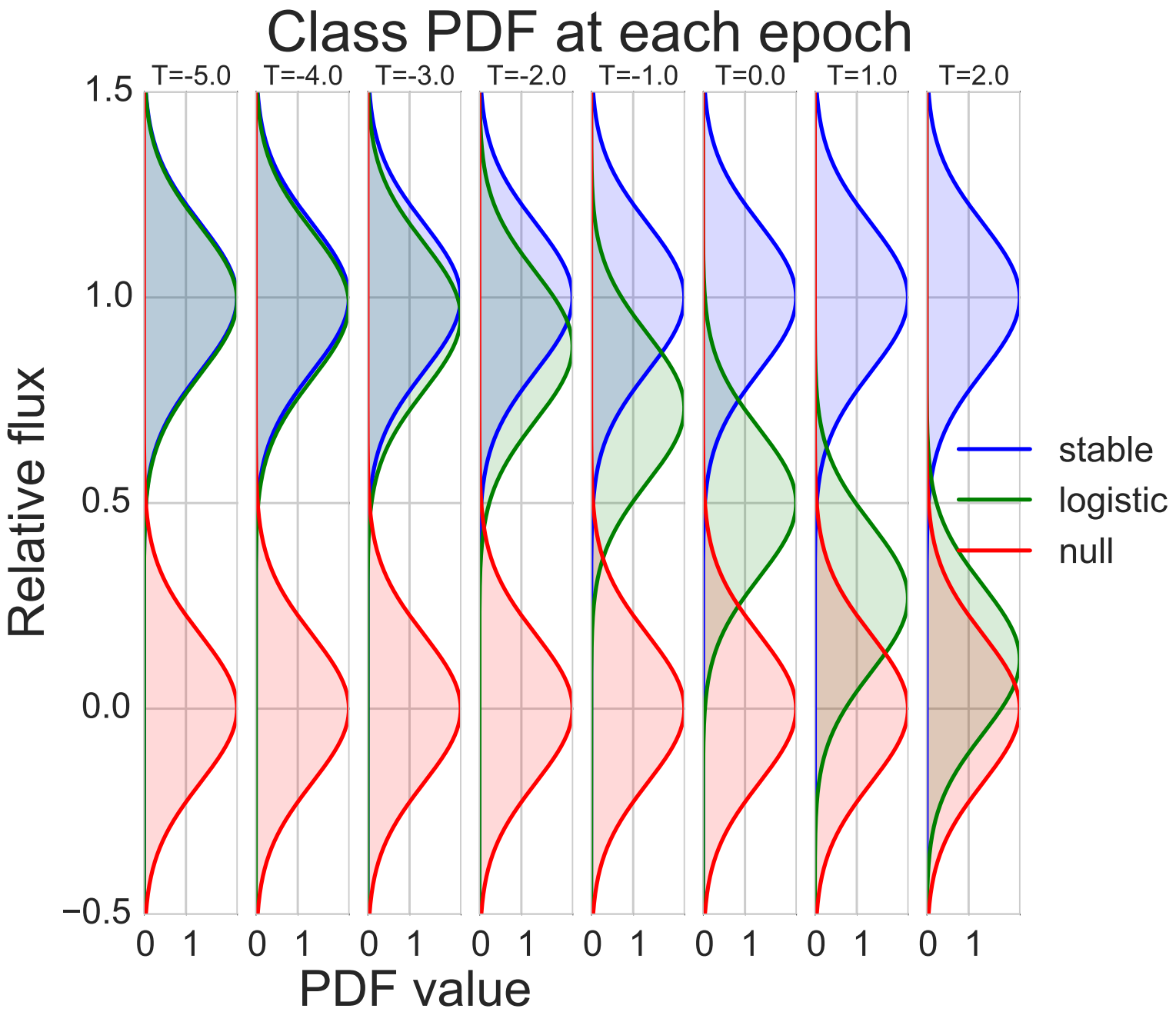
Outline


Intrinsic lightcurves
Relative flux
1.0
0.8
0.6
0.4
0.2
0.0
−6
−4
−2
0
2
4
6
Epoch
stable
logistic
null

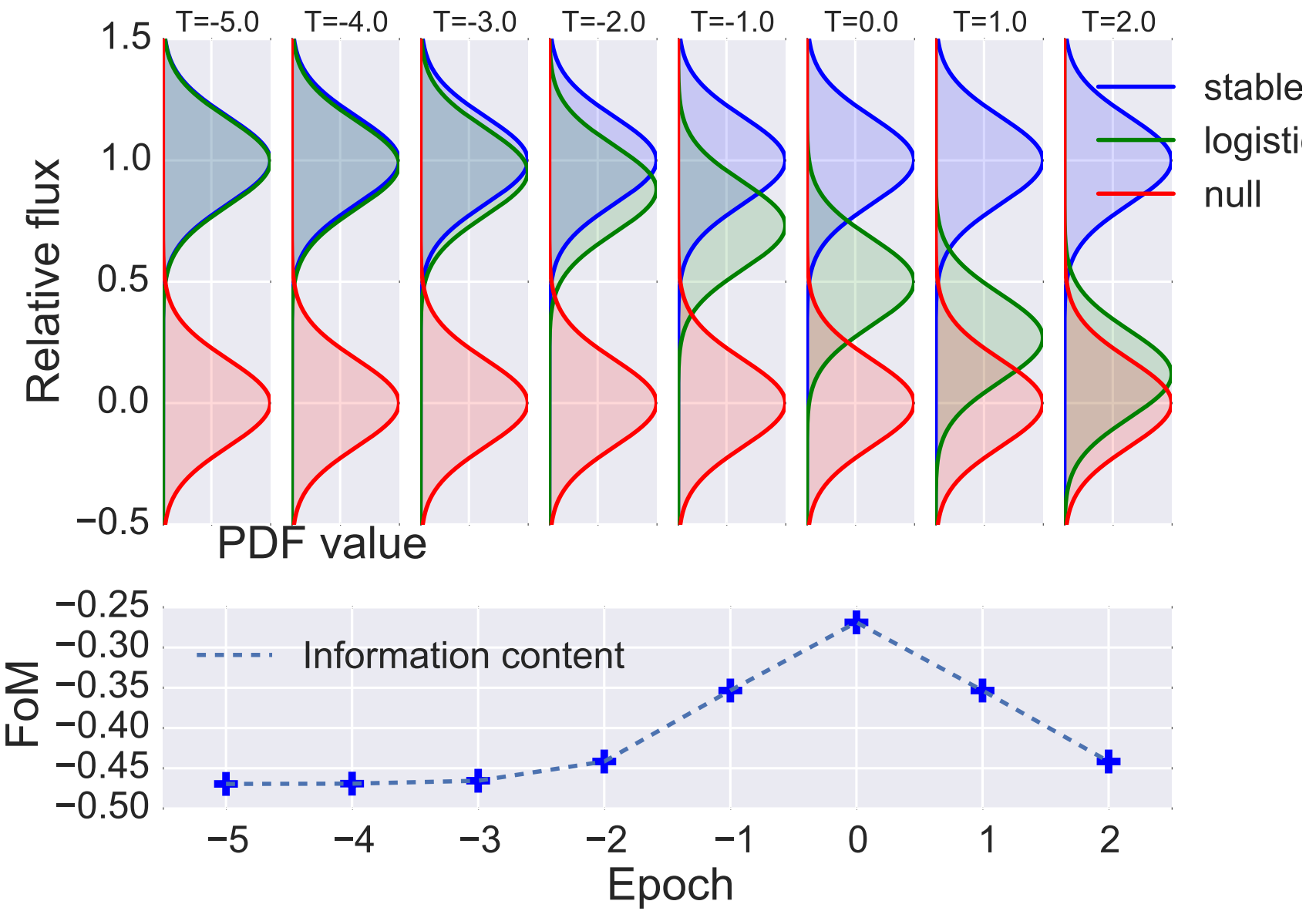

Sampling with noise
True value
Noisy samples
Relative flux
Epoch
1.5
1.0
0.5
0.0
−0.5
−6
−4
−2
0
2
4
6


Sampling with noise
Relative flux
Epoch
1.5
1.0
0.5
0.0
−0.5
−6
−4
−2
0
2
4
6


Sampling with noise
Relative flux
Epoch
1.5
1.0
0.5
0.0
−0.5
−6
−4
−2
0
2
4
6


Class PDF at each epoch
T=-5.0
T=-4.0
T=-3.0
T=-2.0
T=-1.0
T=0.0
T=1.0
T=2.0
1.5
1.0
0.5
0.0
−0.5
Relative flux
0 1
PDF value
stable
logistic
null

Evaluating each epoch



Confusion matrices

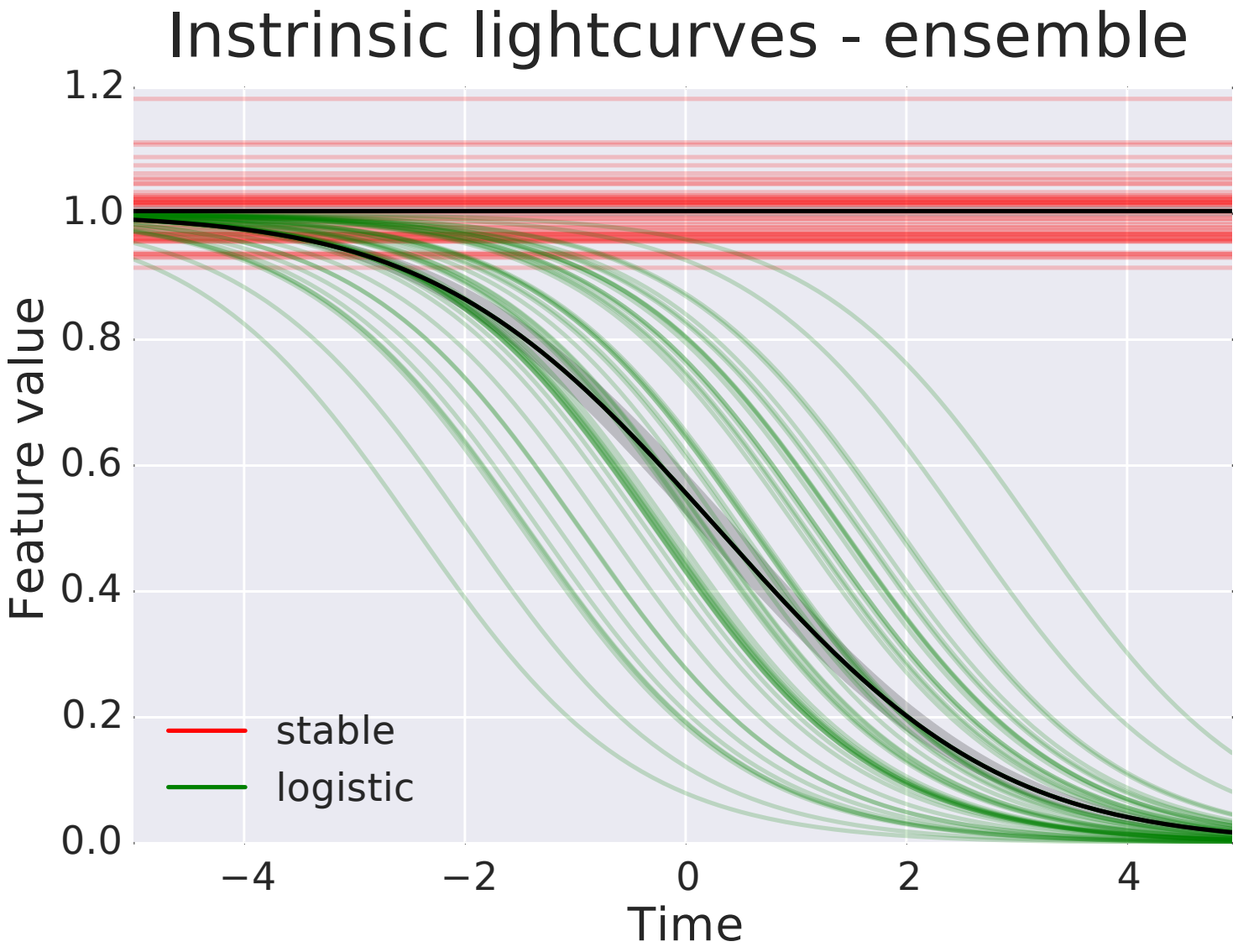
True class	*Labelled*(A)	*Labelled*(B)	*Labelled*(C)
A	$P(\hat{A} \mid A)$	$P(\hat{B} \mid A)$	$P(\hat{C} \mid A)$
B	$P(\hat{A} \mid B)$	$P(\hat{B} \mid B)$	$P(\hat{C} \mid B)$
C	$P(\hat{A} \mid C)$	$P(\hat{B} \mid C)$	$P(\hat{C} \mid C)$

Confusion matrices

True class	*Labelled*(A)	*Labelled*(B)	*Labelled*(C)
A	$P(\hat{A} \mid A)$	$P(\hat{B} \mid A)$	$P(\hat{C} \mid A)$
B	$P(\hat{A} \mid B)$	$P(\hat{B} \mid B)$	$P(\hat{C} \mid B)$
C	$P(\hat{A} \mid C)$	$P(\hat{B} \mid C)$	$P(\hat{C} \mid C)$

Epoch = -2

Label	**logistic**	**stable**	**null**
True class			
logistic	0.387	0.604	0.009
stable	0.302	0.697	0.001
null	0.009	0.003	0.988


Instrinsic lightcurves - ensemble
Feature value
1.2
1.0
0.8
0.6
0.4
0.2
0.0
stable
logistic
−4
−2
0
2
4
Time

Outline

Required knowledge / user-inputs

- Transient rate priors.

Required knowledge / user-inputs

- Transient rate priors.
- Transient lightcurve ensemble models.

Required knowledge / user-inputs

- Transient rate priors.
- Transient lightcurve ensemble models.
- Telescope / noise models.

Required knowledge / user-inputs

- Transient rate priors.
- Transient lightcurve ensemble models.
- Telescope / noise models.
- Follow-up prioritization weightings.

Required software components

- Efficient lightcurve generation library.

Required software components

- Efficient lightcurve generation library.
- MCMC data fitting models and routines.

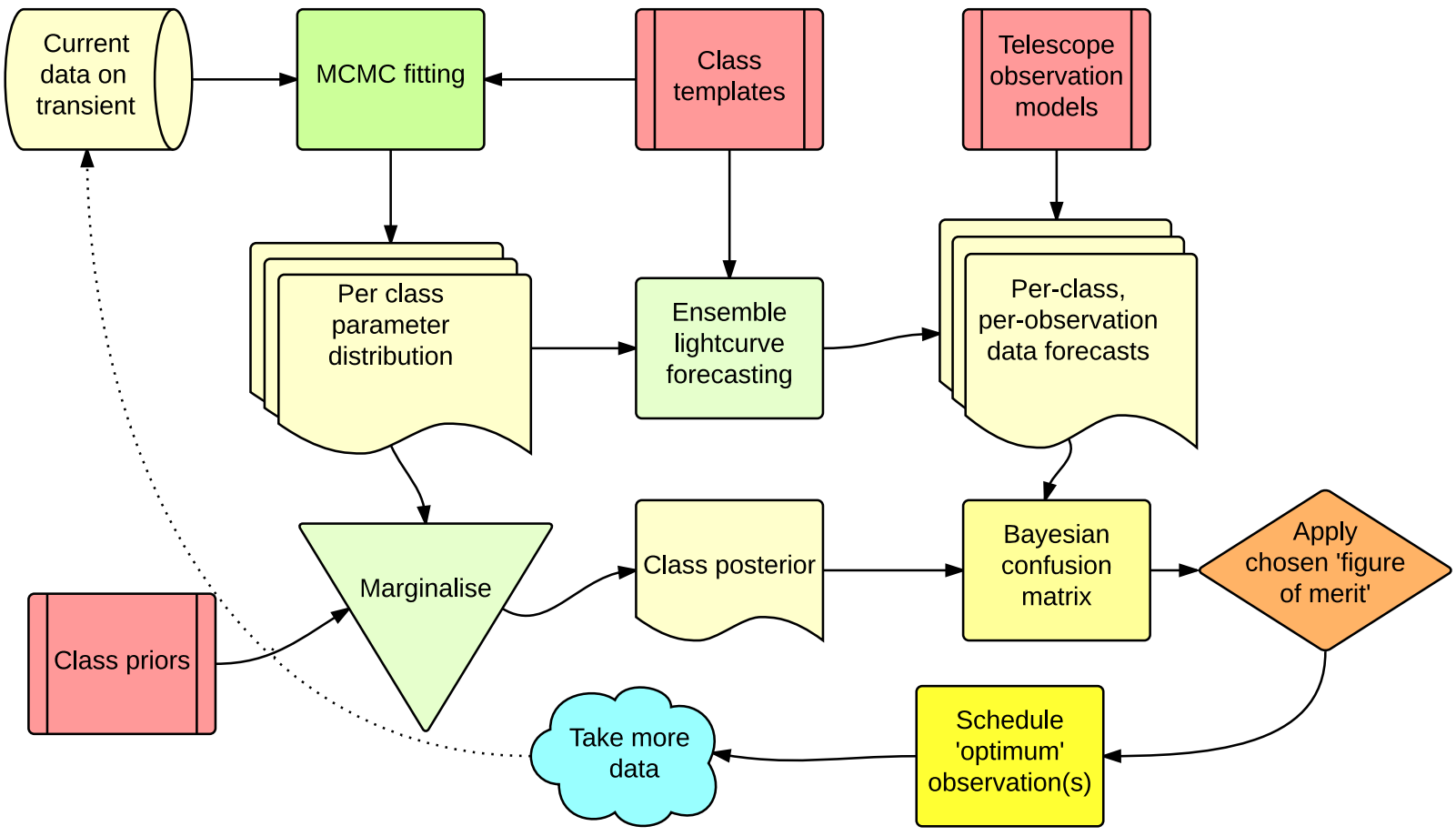
Required software components

- Efficient lightcurve generation library.
- MCMC data fitting models and routines.
- Statistical routines for calculating confusion matrices.

Required software components

- Efficient lightcurve generation library.
- MCMC data fitting models and routines.
- Statistical routines for calculating confusion matrices.
- Observation schedule optimization engine.

Required components



Outline

What's next?

- Finish bolting components together.
- Run simulations, test in more realistic scenarios.
- Interfacing with optimizer / scheduler.

Longer term

- Variational Bayes?
- Gaussian processes?

Outline

Summary

- *Information content* is just a penalty function for scoring predicted observations.
- Using it to decide when to observe is applied *Bayesian decision theory*.
- But doing this for real requires a number of non-trivial software components.
- Nearly ready for testing!