Robust Automatic Classification of variable stars:

Dealing with incomplete time series and discovering deep patterns for light-curve representation

Karim Pichara
Computer Science Department PUC

Nicolás Castro, DCC UC - Andrés Riveros, DCC UC - *Pavlos Protopapas, IACS*

People developing this project

Nicolás Castro
Master Student

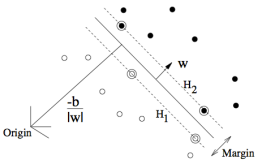
Andrés Riveros
Master Student

Pavlos Protopapas
IACS, Harvard

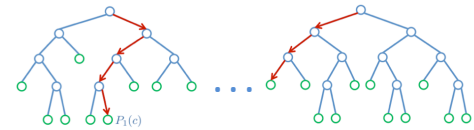
Karim Pichara
DCC, PUC

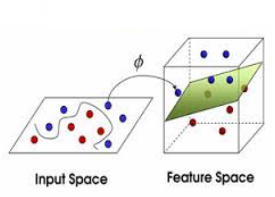
Automatic classification

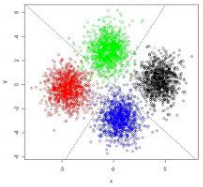
Support Vector Machines

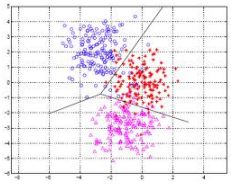


Random Forest





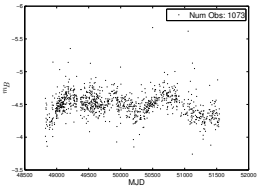


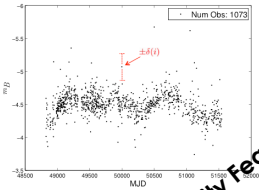


Lightcurve Representation

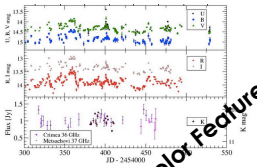
Numerical descriptors of Lightcurves

Lightcurve

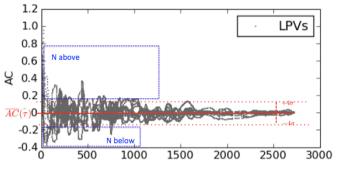




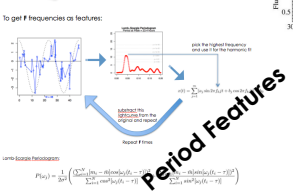
Variability Features



Color Features



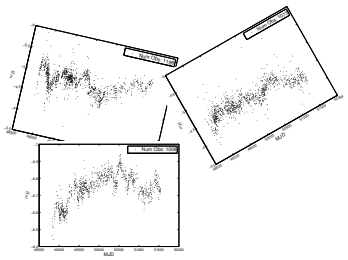
Auto-Correlation Features



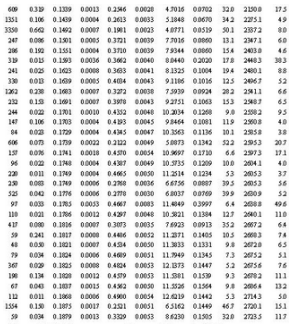
Period Features

Lightcurve Representation

lightcurves

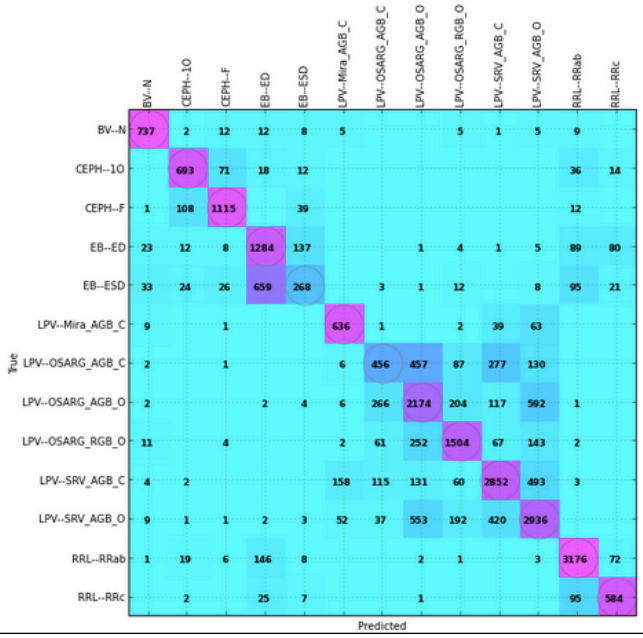


Dataset of Features

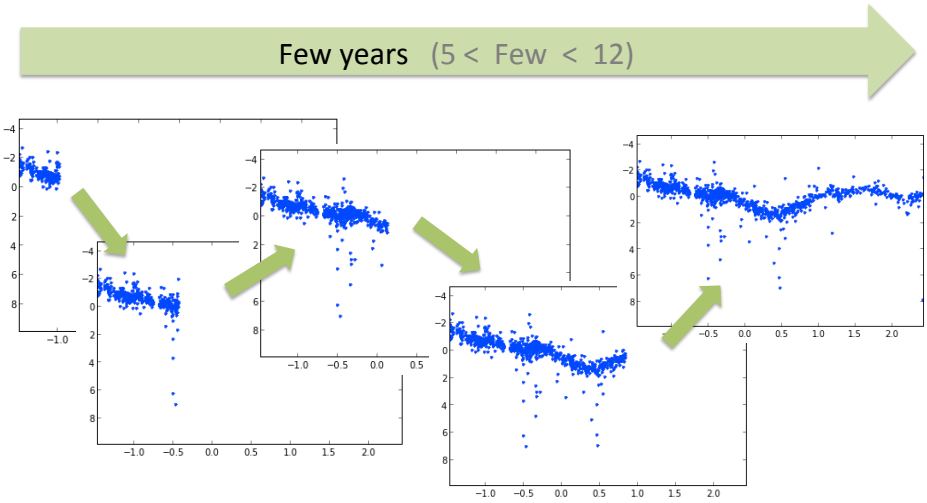


609	0.319	0.1339	0.0013	0.2346	0.0028	4.7016	0.0702	32.0	2159.0	17.5
1351	0.106	0.1439	0.0004	0.2613	0.0033	5.1948	0.0670	34.2	2275.1	4.9
3350	0.662	0.1492	0.0007	0.1981	0.0023	4.0771	0.0519	30.1	2337.2	8.0
247	0.086	0.1301	0.0005	0.3721	0.0039	7.7016	0.0860	13.1	2347.1	6.0
286	0.192	0.1551	0.0004	0.3710	0.0039	7.9344	0.0860	15.4	2403.0	4.6
319	0.015	0.1393	0.0036	0.3662	0.0040	8.0440	0.2020	17.8	2448.3	39.3
241	0.025	0.1623	0.0008	0.3633	0.0041	8.1525	0.1004	19.4	2400.1	8.8
330	0.013	0.1639	0.0005	0.4104	0.0043	9.1186	0.1016	12.5	2496.7	5.2
1262	0.238	0.1683	0.0007	0.3272	0.0038	7.3939	0.0924	28.2	2361.1	6.6
232	0.153	0.1691	0.0007	0.3978	0.0043	9.2751	0.1063	15.3	2360.7	6.5
244	0.022	0.1701	0.0010	0.4332	0.0048	10.2034	0.1268	9.8	2559.2	9.5
147	0.106	0.1703	0.0004	0.4305	0.0045	9.9464	0.1001	11.9	2393.8	4.0
84	0.023	0.1729	0.0004	0.4345	0.0047	10.3563	0.1136	10.1	2385.8	3.8
606	0.073	0.1759	0.0022	0.2122	0.0049	5.0873	0.1342	52.2	2395.3	20.7
137	0.076	0.1741	0.0010	0.4370	0.0034	10.9697	0.1710	6.6	2397.3	17.1
96	0.022	0.1749	0.0004	0.4307	0.0049	10.3735	0.1209	10.0	2404.1	4.0
220	0.011	0.1749	0.0004	0.4465	0.0050	11.2514	0.1234	5.3	2405.3	3.7
230	0.003	0.1749	0.0006	0.2788	0.0036	6.6756	0.0897	39.5	2405.3	5.6
525	0.042	0.1776	0.0006	0.2778	0.0030	6.8037	0.0769	39.9	2650.9	5.2
97	0.033	0.1765	0.0013	0.4467	0.0083	11.4849	0.3597	6.4	2658.8	49.6
110	0.021	0.1786	0.0012	0.4297	0.0048	10.3821	0.1384	12.7	2640.1	11.0
417	0.000	0.1816	0.0007	0.3073	0.0035	7.6923	0.0913	35.2	2697.2	6.4
59	0.241	0.1817	0.0008	0.4486	0.0052	11.2371	0.1405	10.5	2693.3	7.4
40	0.030	0.1821	0.0007	0.4334	0.0050	11.3833	0.1331	9.8	2672.0	6.5
79	0.034	0.1824	0.0006	0.4689	0.0051	11.7949	0.1345	7.3	2675.2	5.1
167	0.029	0.1825	0.0008	0.4424	0.0053	12.1373	0.1447	5.2	2675.6	7.6
198	0.134	0.1828	0.0012	0.4379	0.0053	11.3981	0.1539	9.3	2679.2	11.1
67	0.043	0.1837	0.0013	0.4362	0.0050	11.3526	0.1564	9.8	2686.4	13.2
112	0.011	0.1868	0.0006	0.4960	0.0054	12.6219	0.1442	5.3	2714.3	5.0
1554	0.130	0.1875	0.0017	0.2321	0.0051	6.3162	0.1449	46.7	2720.1	15.1
59	0.034	0.1899	0.0013	0.3329	0.0033	8.6230	0.1505	32.0	2723.5	11.7

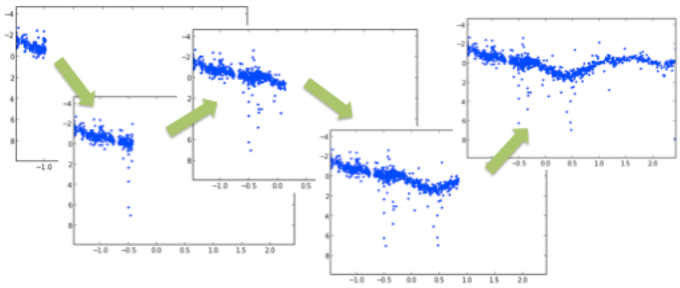
Automatic classification: Results so far



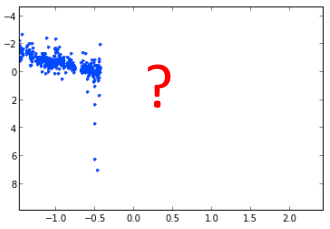
Motivation: Observational projects take several years to finish the lightcurves



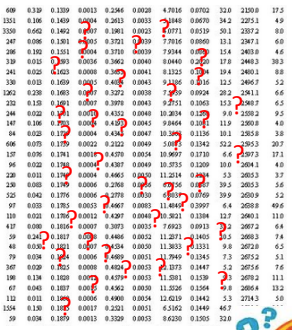
May be we can do something in the meantime?



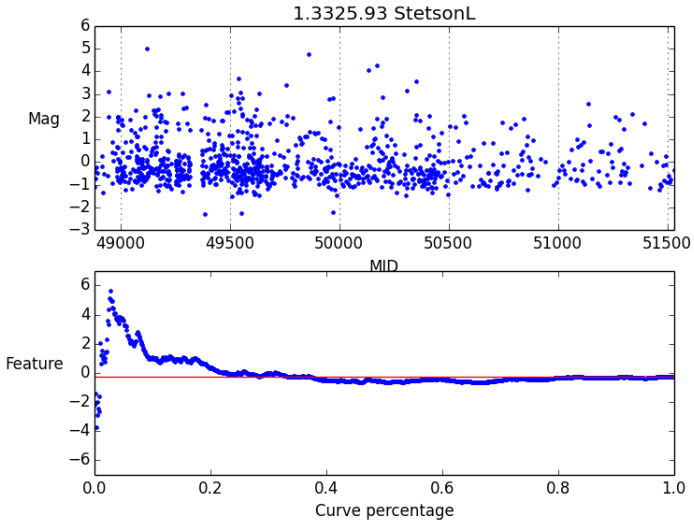
Directly calculating the features from incomplete lightcurves seems not right



Dataset of Features

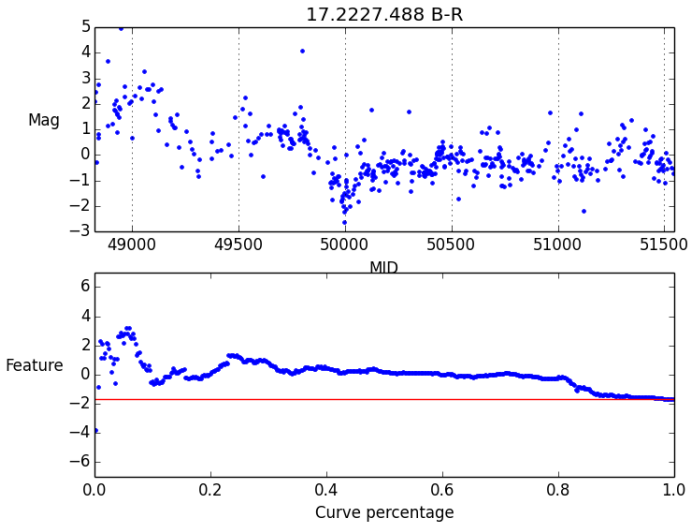


Feature convergence



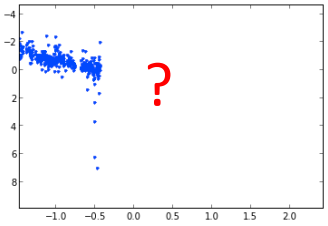
12

Feature convergence

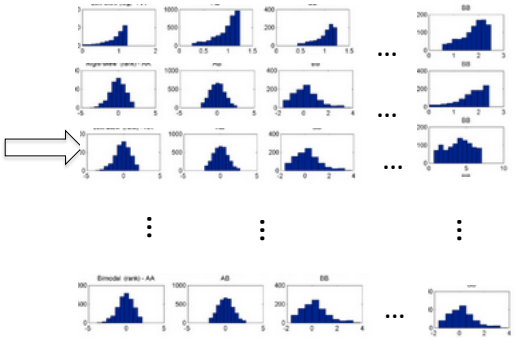


13

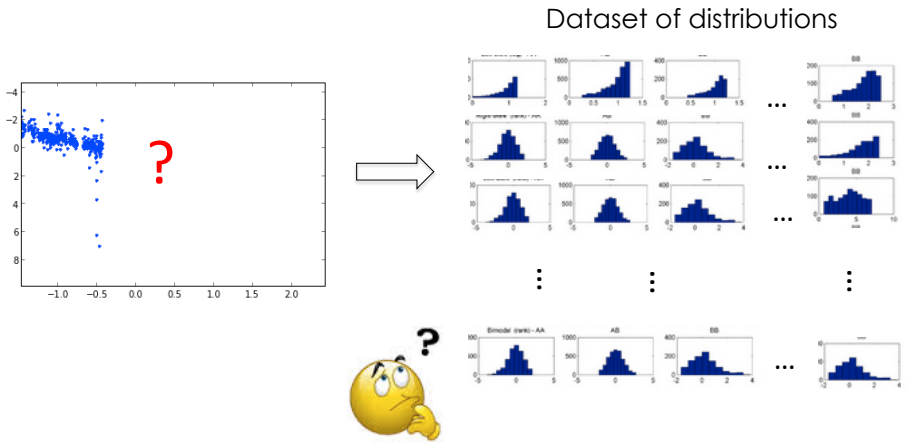
Idea: lets calculate distributions of features in order to model uncertainty



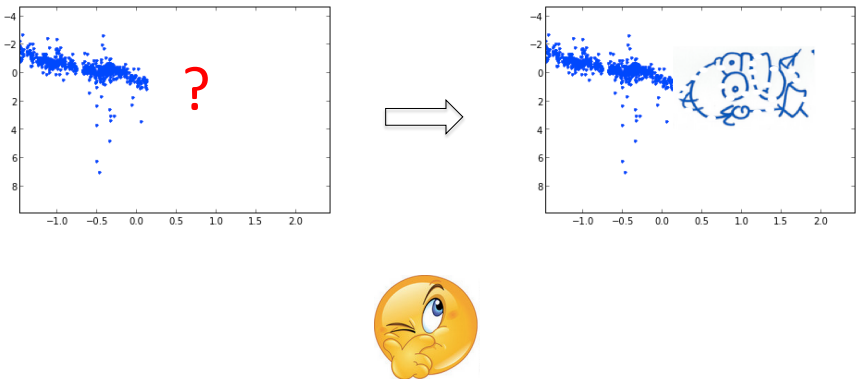
Dataset of distributions



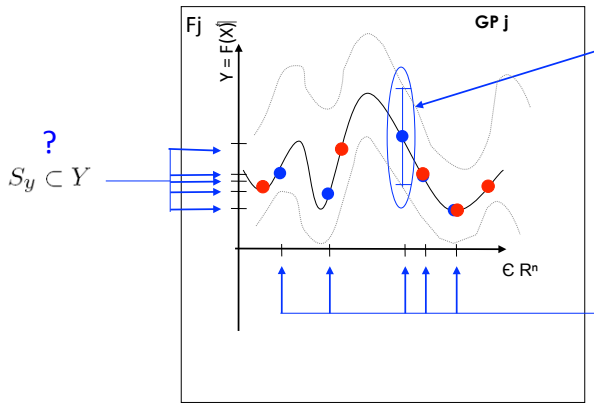
How we can obtain distributions of features?



In general we have an idea of what follows in the incomplete lightcurves



Gaussian Processes



Normal Distribution for predictions

?

$S_y \subset Y$

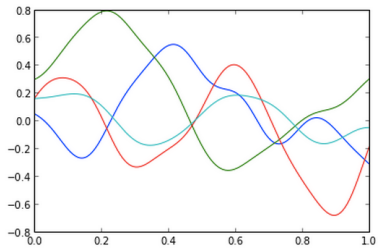
$S_x \in X$

Posterior mean and covariance matrix for Sy given observations →

$$\mu_{S_y|S_x,A_x,A_y} = \mu_{S_y} + \Sigma_{S_xA_x}\Sigma^{-1}_{A_xA_x}(A_y - \mu_{A_y})$$

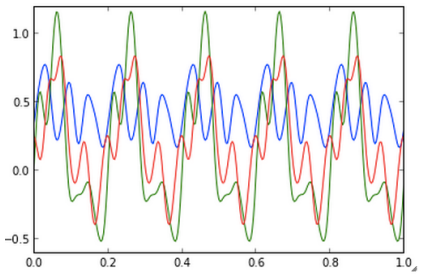
$$\sigma^2_{S_y|S_x,A_x,A_y} = K(S_x,S_x) - \Sigma_{S_xA_x}\Sigma^{-1}_{A_xA_x}\Sigma^{T}_{S_xA_x}$$

Gaussian Processes



$$k_{\text{SE}}(x,x') = \sigma^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right)$$

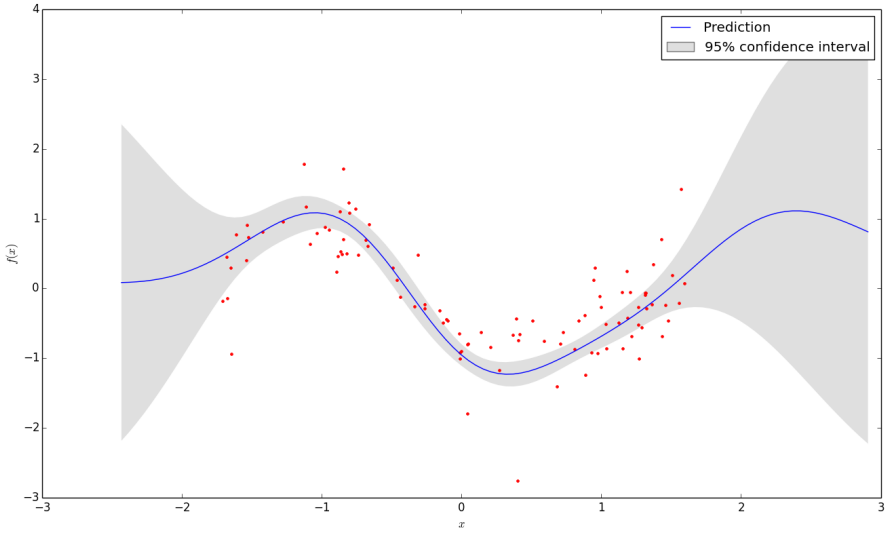
$$k_{\text{Per}}(x,x') = \sigma^2 \exp\left(-\frac{2\sin^2(\pi|x-x'|/p)}{\ell^2}\right)$$



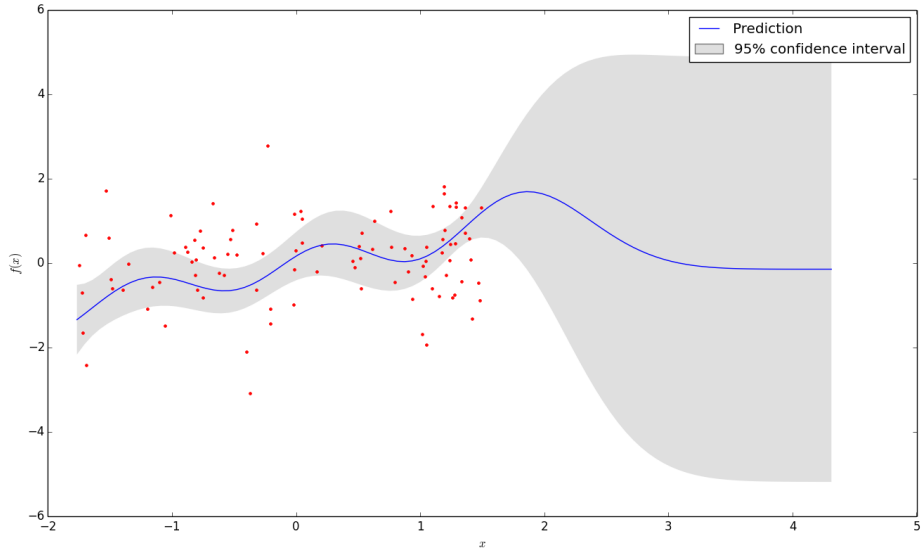
$$\mu_{S_y|S_x,A_x,A_y} = \mu_{S_y} + \Sigma_{S_xA_x}\Sigma^{-1}_{A_xA_x}(A_y - \mu_{A_y})$$

$$\sigma^2_{S_y|S_x,A_x,A_y} = K(S_x,S_x) - \Sigma_{S_xA_x}\Sigma^{-1}_{A_xA_x}\Sigma^{T}_{S_xA_x}$$

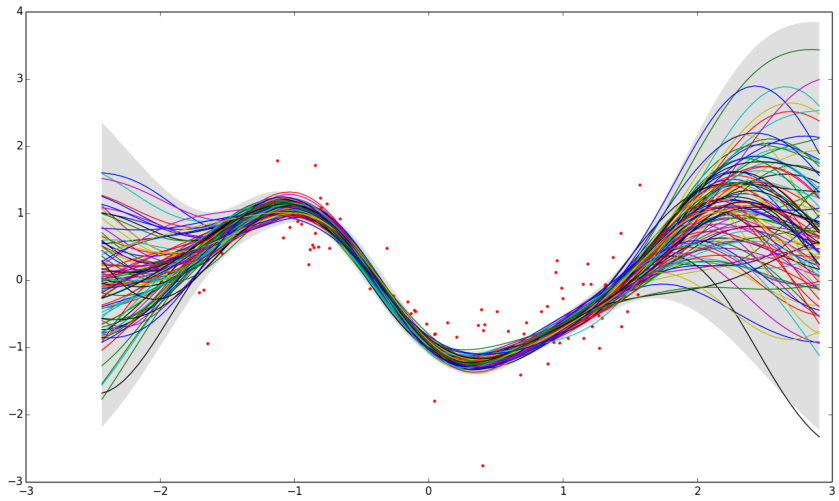
Gaussian Processes



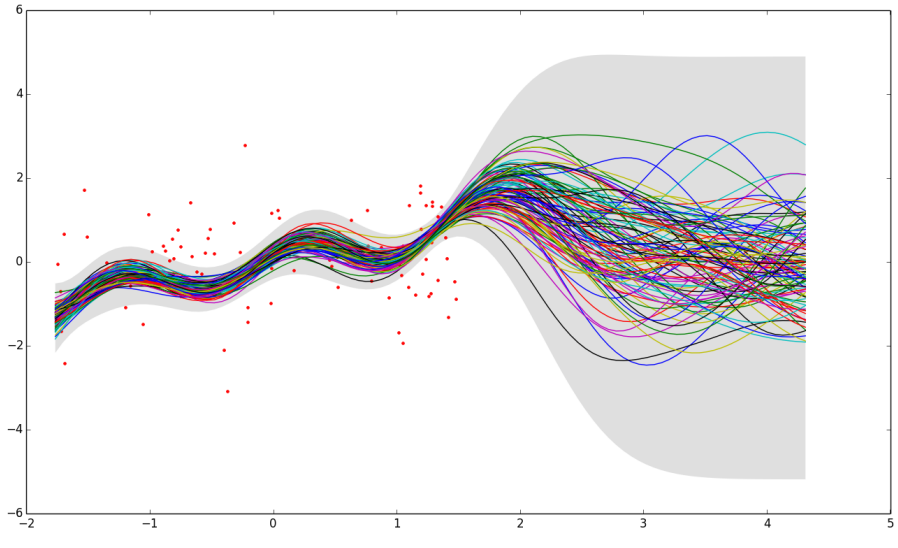
Gaussian Processes



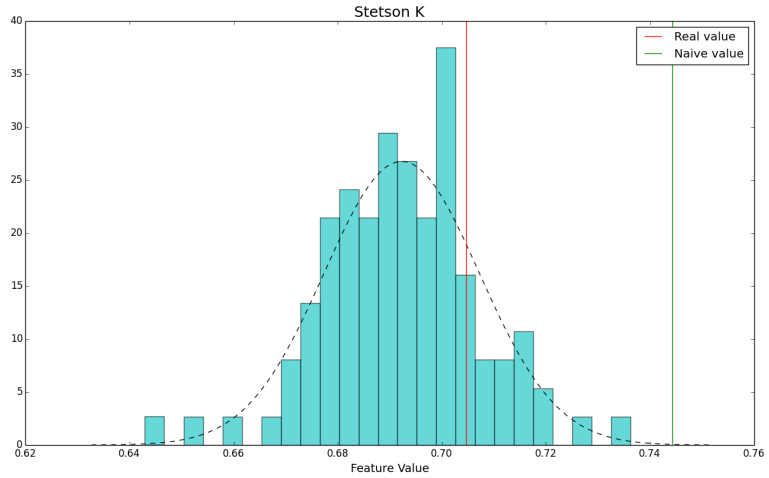
With Gaussian Processes we can take samples from lightcurves



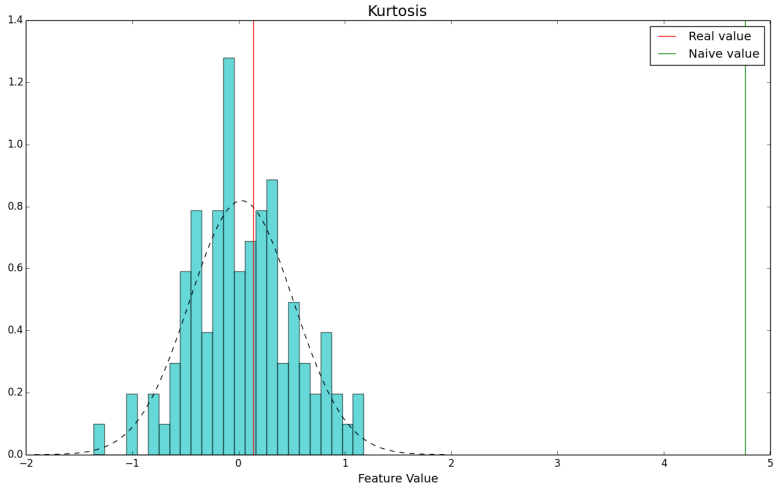
With Gaussian Processes we can take samples from lightcurves



Having samples from lightcurves we can obtain samples from features

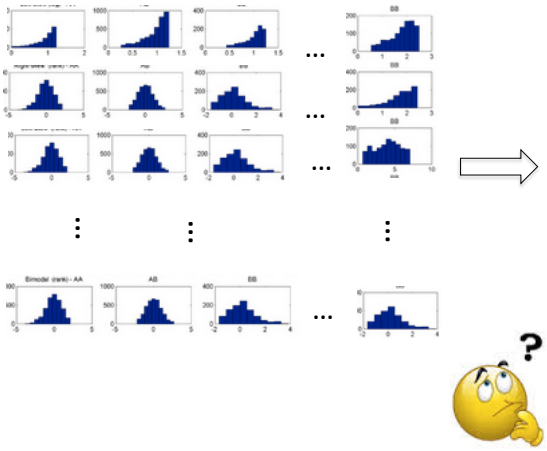


Having samples from lightcurves we can obtain samples from features

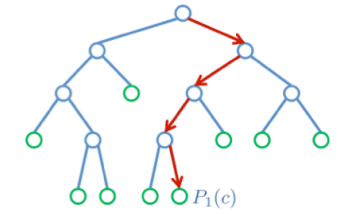


Once we have the features as distributions, how we can train a classifier?

Dataset of distributions

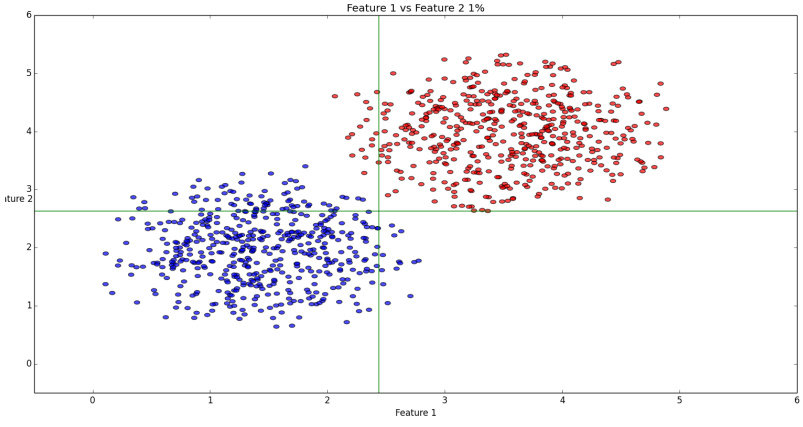


Model

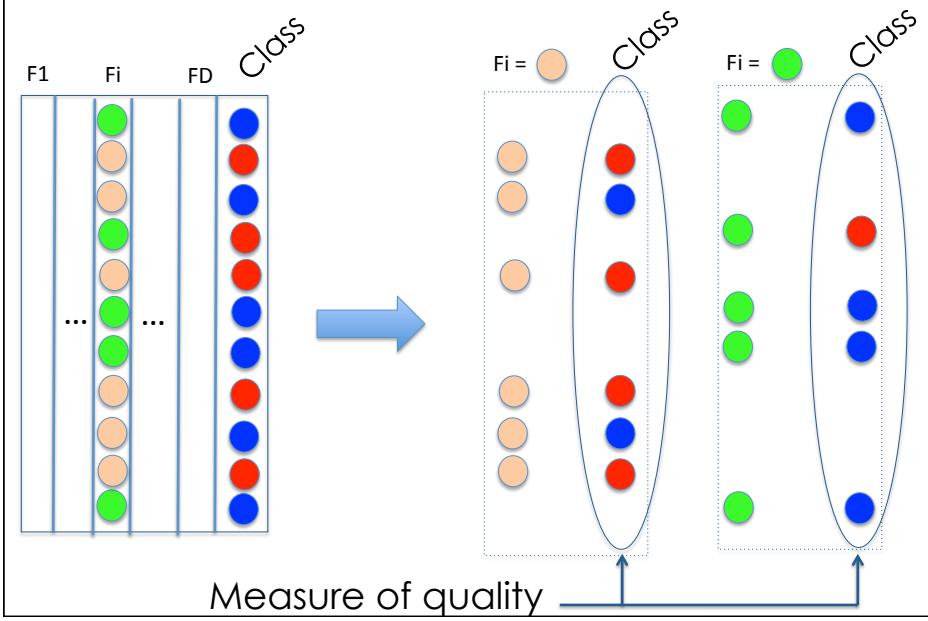


Plain decision trees:

Search for splits in the variables that separate the data in different classes



Information Gain



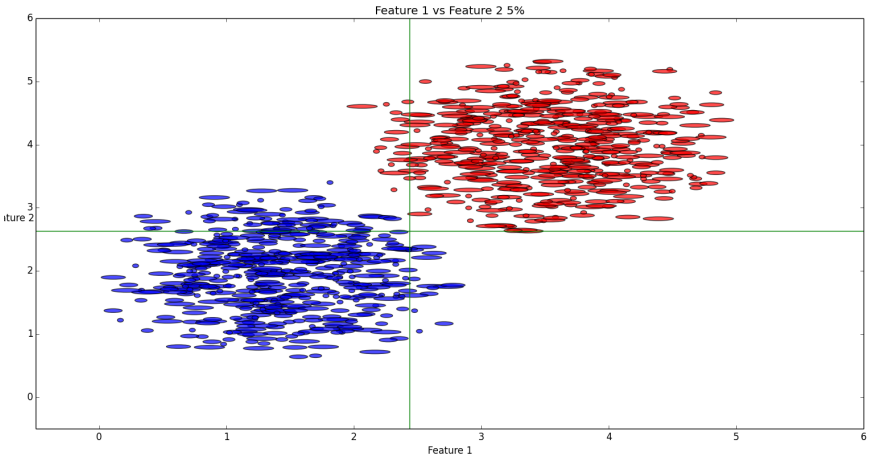
Measuring the quality of a split

Shannon's Entropy $H(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$

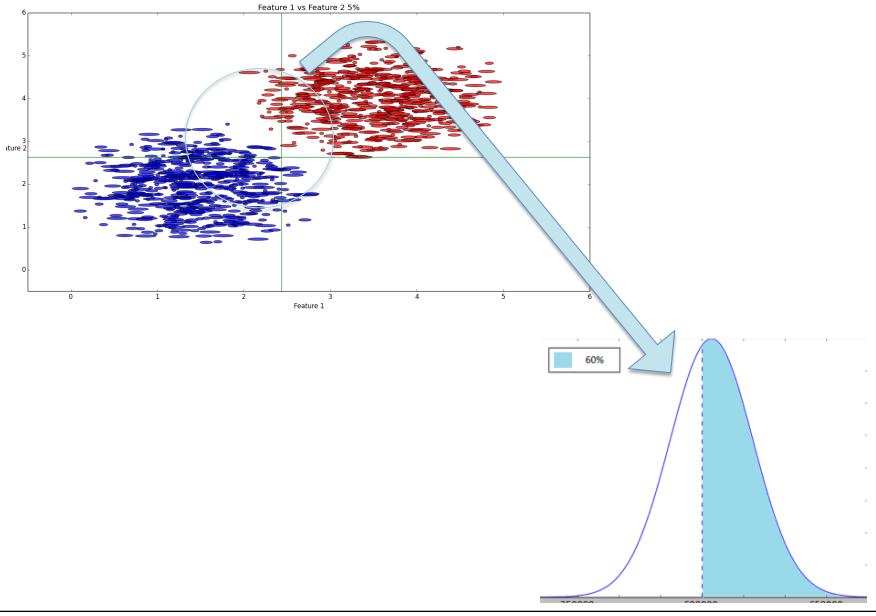
where pi is the probability that an arbitrary tuple in D belongs to class i. It is estimated as:

$$p_i = \frac{|C_{i,D}|}{|D|} \quad , \quad |C_{i,D}| = \text{Number of samples of class I in D}$$

Modified decision tree:



Probabilistic cuts:



Expected entropy

$$S(D) = -\sum_{i=1}^{m} p_i \log_2 p_i$$

The way pi is estimated changes

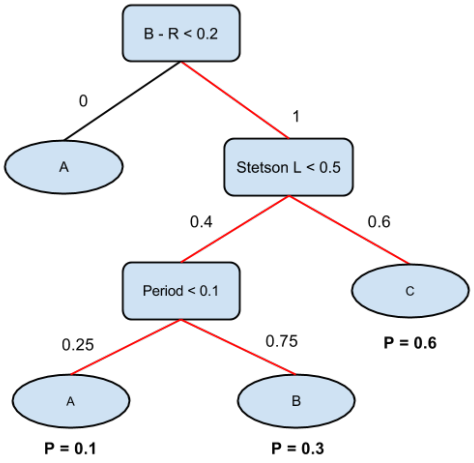
$$p_i = \frac{|C_{i,d}|}{D} \quad , \quad |C_{i,d}| = \sum_{j \in i} \mathbf{w} \cdot \int_A (\mathbf{x}_j)$$

$$|D| = \sum_{\forall j} \int_A (\mathbf{x}_k)$$

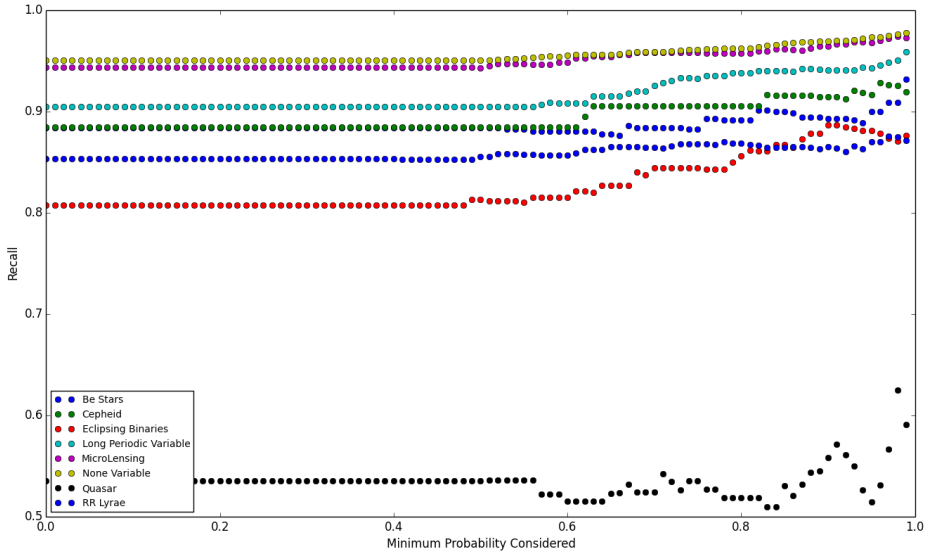
We weight the importance of each tuple by multiplying it by the probability that it will get to that given node.

After learning: Probabilistic classification

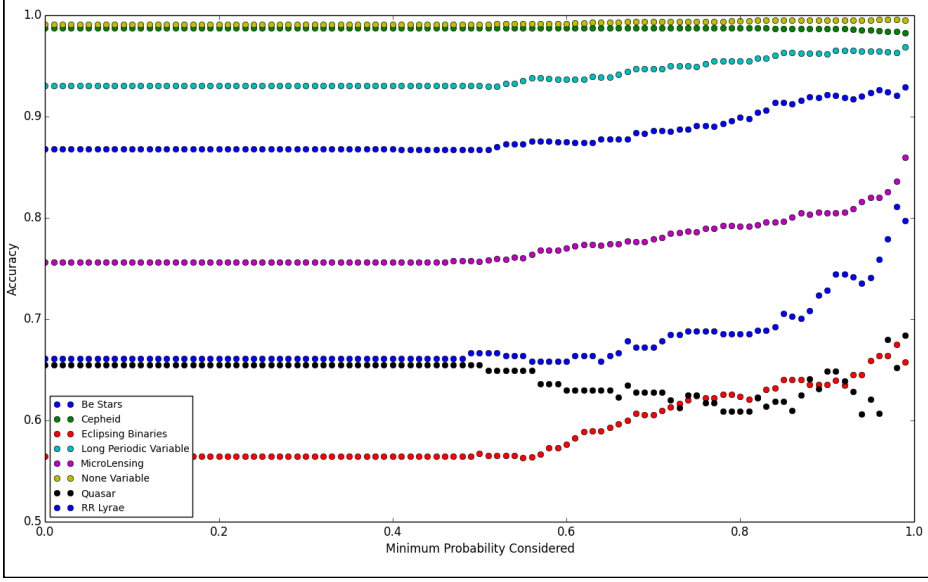
- A sample can go through many paths when being classified.
- The model chooses the class with highest probability.



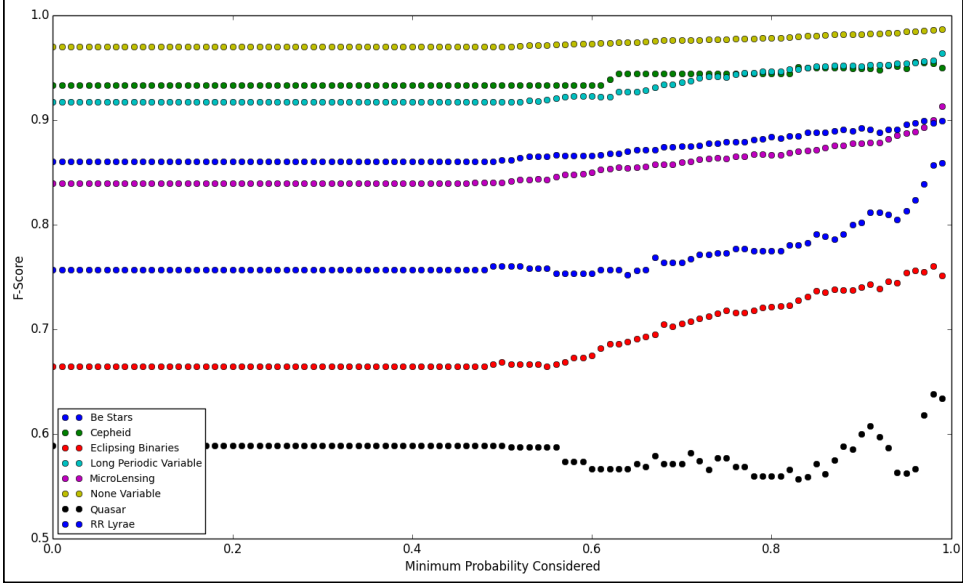
Classification accuracy so far



Classification accuracy so far



Classification accuracy so far



Conclusions and Future Work

- We can perform automatic classification with "incomplete" lightcurves handling uncertain features
- For some classes, we can have good accuracies before completing the observational time
- We are working to increase our list of features with uncertainty

Conclusions and Future Work

- We expect to improve the fit of the Gaussian Processes in order to get better accuracies
- We will extend the uncertain decision tree to a uncertain Random Forest
- We will model the uncertainty of features with more suitable distributions

Conclusions and Future Work

- Still a lot to do with our models

Thank you!!!!

Nicolás Castro
Master Student

Andrés Riveros
Master Student

Pavlos Protopapas

Karim Pichara