Lorentz Center

International center for scientific workshops

Accelerating the Search for Dark Matter with Machine Learning

from 15 Jan 2018 through 19 Jan 2018

Data-driven constraints to dark matter from dwarf galaxies

Bryan Zaldívar
(Annecy, FR)

Based on work in progress with:
Francesca Calore & Pasquale D. Serpico

Hypothesis

“Fact”:

- There is a non-luminous component of the universe which interacts with us at least through gravitational forces

Assume:

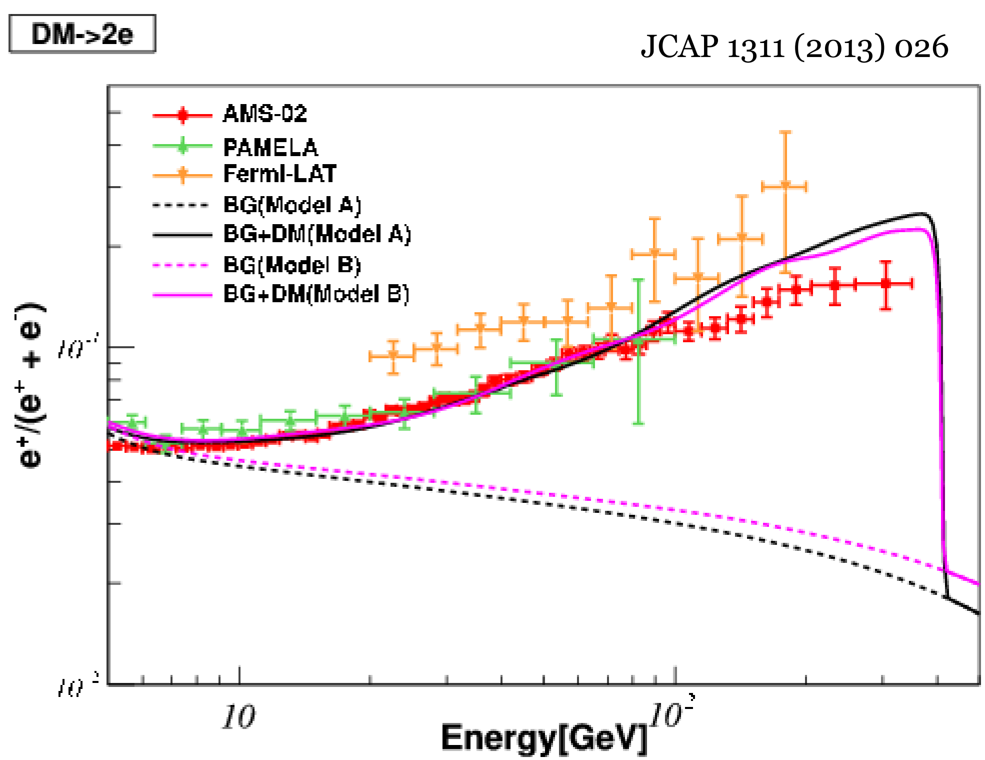
- There may be a contribution to the astrophysical emission coming from (non-gravitational) interactions of dark matter with ordinary matter

(given a physical model)
Dark matter contribution
is fixed

Background contribution
is not fixed

→ **room for machine-learning!**

JCAP 1311 (2013) 026



Aim

<u>To constrain the DM hypothesis</u>

Which data is used:

- photon (gamma-ray) emission from dwarf spheroidal galaxies (dSphs)

Why this is convenient data:

- dSphs are believed to be DM-dominated systems
  (according to gravitational observations)

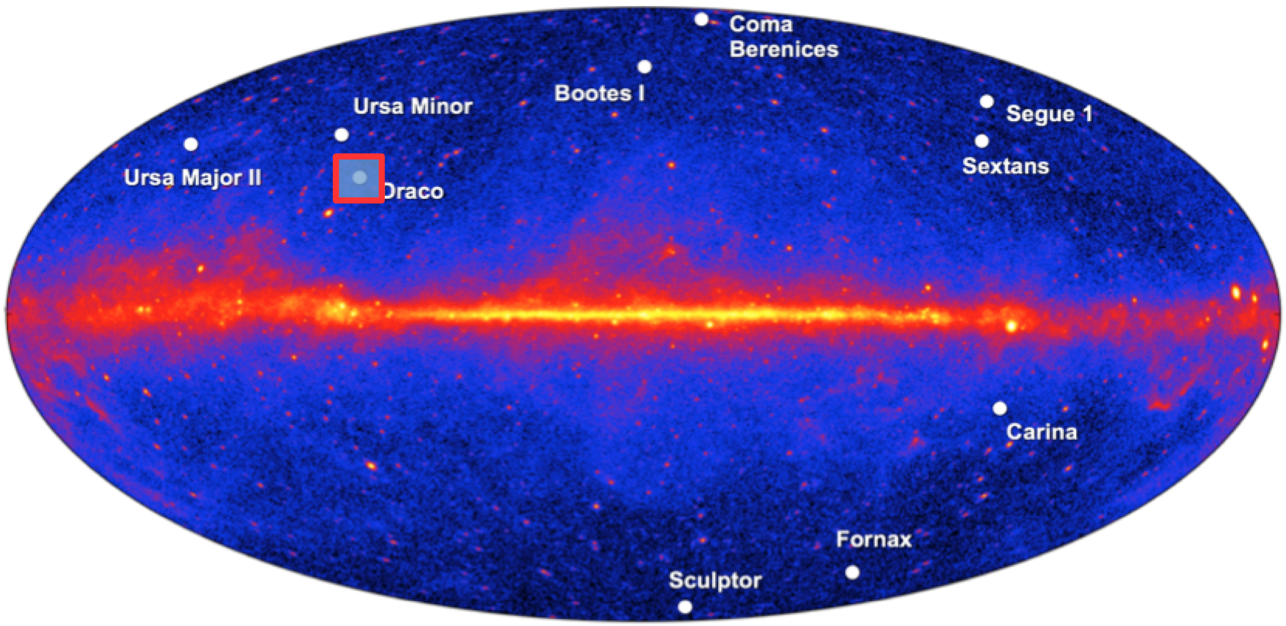
What is needed:

- definition of "control" region
- a method for estimating the background
- a statistical approach

Fermi-LAT's way

(from non-expert opinion)

- *independent* determination of background in a 15°x15° region around each dwarf
- predefined background models (diffuse and isotropic) where *only normalisation is fitted*



Points to improve:

- new (unresolved) spatially-dependent contributions may provide unequal performances in different regions of the sky
- no guarantee that background is consistently determined from one region to another
- Estimation of (theoretical) systematic errors is unclear

A data-driven way

- Be agnostic about a possibly underlying physics as for background is concerned
- *Build a global estimator* based only on data, from reasonably well-defined control regions
- Extrapolation to estimate the background contribution on dwarfs
- Include background uncertainties in the statistical analysis

---

Regression problem

Supervised learning

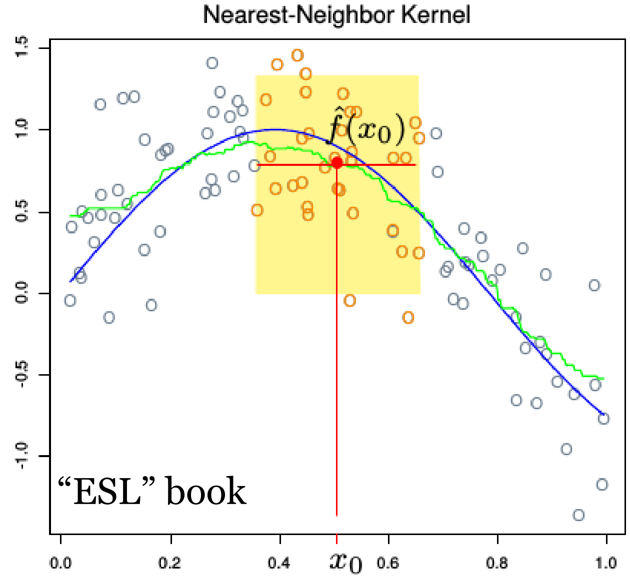
Generating control regions

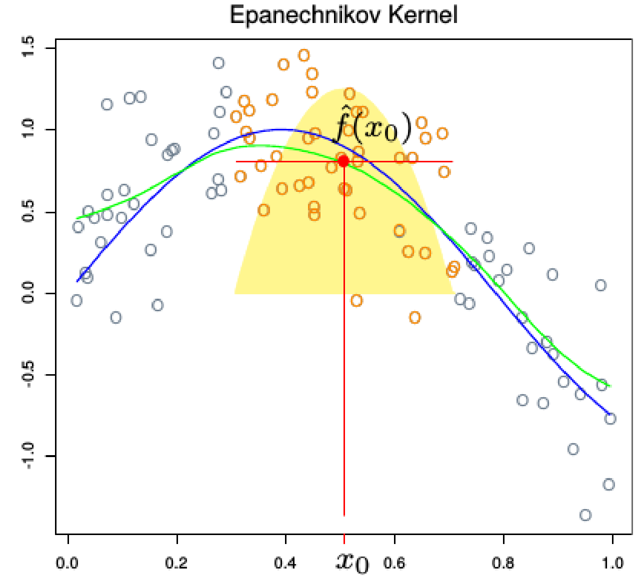
Kernel Density Estimation of dwarfs's spatial distribution

("out-of-the-box" **`scikit-learn`** package)

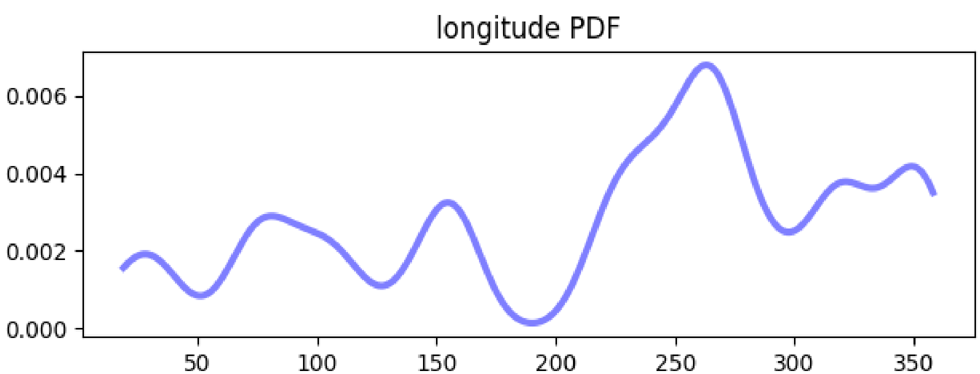
- Gaussian kernel
- optimal smoothing parameters from cross-validation procedure

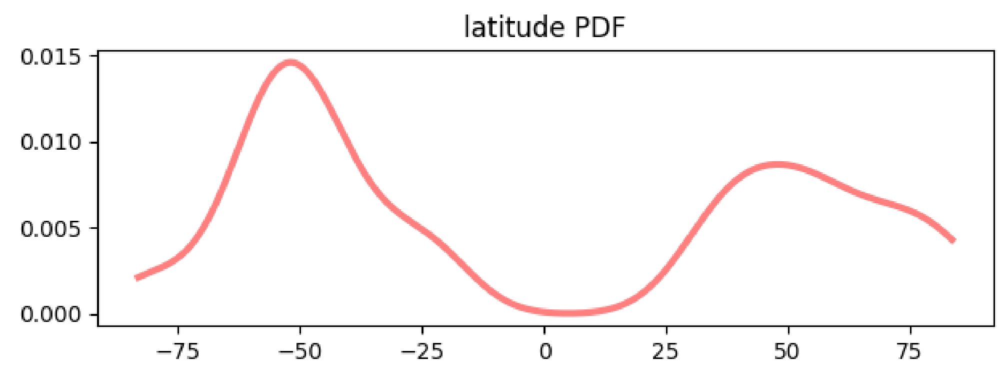
$$\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)}$$





Result:



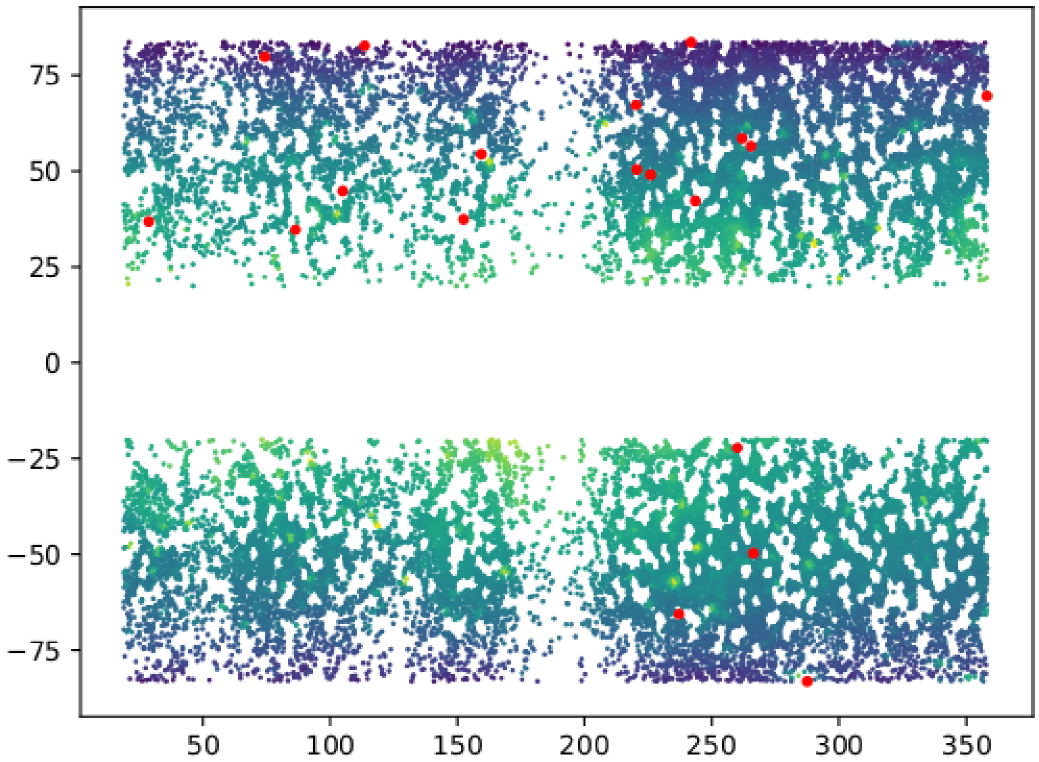


Calore, Serpico, Zaldivar, in preparation

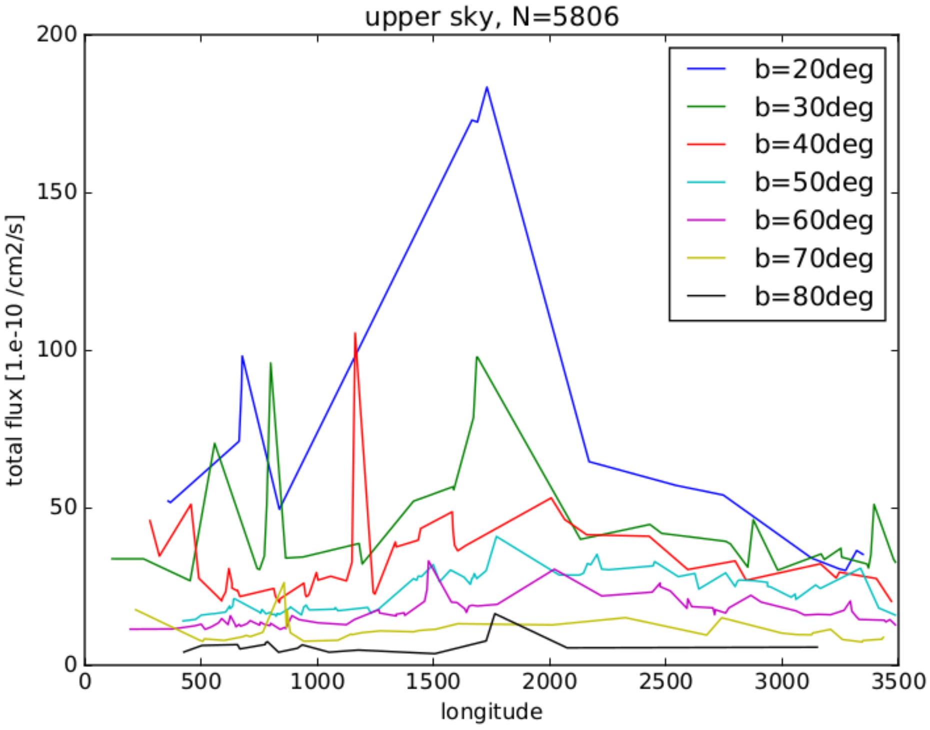
How does data look like?
(control region)

Masking galactic plane, point-like sources and extended sources (Fermi-LAT catalog)

Calore, Serpico, Zaldivar, preliminary



~ 40,000 regions



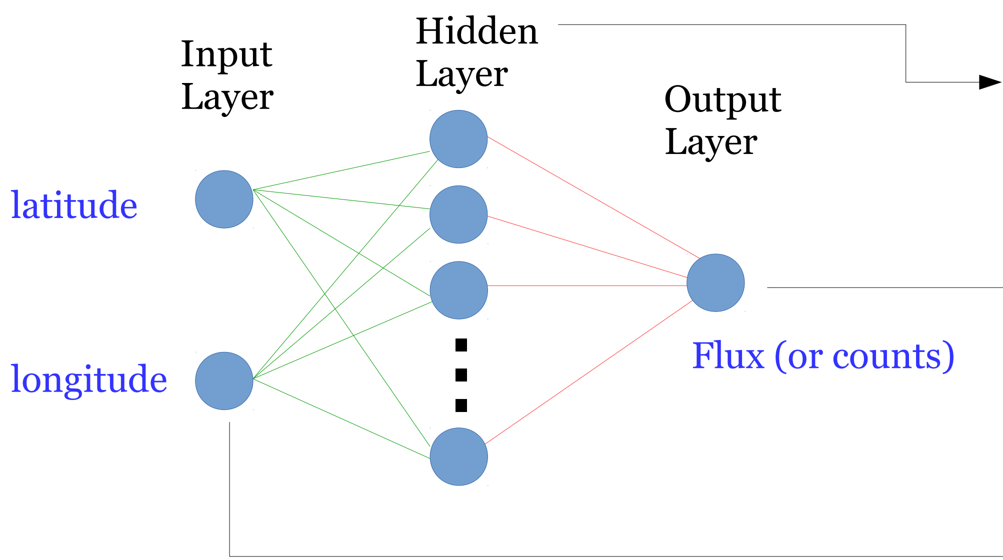
Very noisy!

Feedforward Neural Network try

Universal approximation theorem:

A 1 hidden-layer feedforward NN with (arbitrarily large but) finite number of units can approximate any continuous function. http://neuralnetworksanddeeplearning.com/chap4.html

Implemented from scratch a NN with architecture:



Sigmoid activation function

$$S(x) = \frac{1}{1+e^{-x}}$$

$$Y = S(XW + B)V + C$$

$$R^2 = 1 - \sum_i (y_i - \hat{y}_i)^2 / \sum_i (y_i - \langle y \rangle)^2$$

Result: **Failed**

$$R^2 \lesssim 0.6$$

After many attempts in a very reduced subsample of data (on my laptop)
(no big changes with other activation functions)

General Regression NN

Specht, 1991

keywords: Probabilistic NN, Parzen Window...

Estimate of underlying joint PDF $f(X,Y)$ of data as:

$$\hat{f}(\vec{X},Y) = \frac{1}{(2\pi)^{(p+1)/2}\sigma^p\sigma_Y}\frac{1}{n}\sum_{i=1}^{n}\exp\left[-\frac{(\vec{X}-\vec{X}_i)^T(\vec{X}-\vec{X}_i)}{2\sigma^2}\right]\exp\left[-\frac{(Y-Y_i)^2}{2\sigma_Y^2}\right]$$

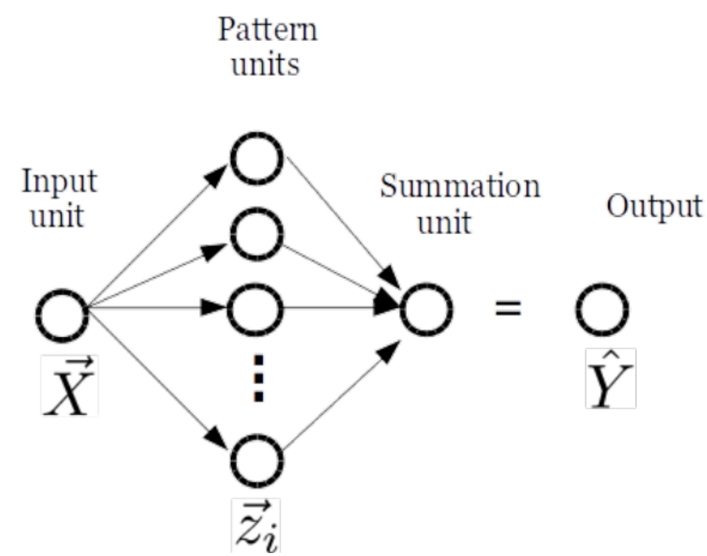
X: input
Y: output

$$\hat{Y}(\vec{X}) = \frac{\sum_{i=1}^{n} Y_i \exp\left[-\frac{D_i^2}{2\sigma^2}\right]}{\sum_{i=1}^{n}\exp\left[-\frac{D_i^2}{2\sigma^2}\right]}, \qquad D_i^2 = (\vec{X}-\vec{X}_i)^T(\vec{X}-\vec{X}_i)$$

Gaussian metric
(but other metrics are equally valid)

Training is a "one passing" procedure

"smoothing parameter" σ to be obtained by optimization



Background prediction at dSphs

Calore, Serpico, Zaldivar, preliminary

dwarf	name	$\log J \pm \Delta_{\log J}$	$\ln(c_{\rm meas})$	$\ln(c_{est})$
1	Boötes I	18.2 ± 0.4	5.209	5.210
2	Canes Venatici I	17.4 ± 0.3	4.787	4.557
3	Canes Venatici II	17.6 ± 0.4	4.248	4.356
4	Carina	17.9 ± 0.1	7.159	7.085
5	Coma Berenices	19.0 ± 0.4	4.220	4.282
6	Draco	18.8 ± 0.1	7.134	7.047
7	Fornax	17.8 ± 0.1	6.223	5.902
8	Hercules	16.9 ± 0.7	7.109	7.209
9	Leo I	17.8 ± 0.2	6.317	6.329
10	Leo II	18.0 ± 0.2	5.501	5.590
11	Leo IV	16.3 ± 1.4	6.114	6.080
12	Leo V	16.4 ± 0.9	6.033	6.404
13	Reticulum II	18.9 ± 0.6	6.229	6.306
14	Sculptor	18.5 ± 0.1	5.460	6.272
15	Segue I	19.4 ± 0.3	6.223	6.334
16	Sextans	17.5 ± 0.2	6.512	6.562
17	Ursa Major I	17.9 ± 0.5	6.146	6.705
18	Ursa Major II	19.4 ± 0.4	6.777	6.723
19	Ursa Minor	18.9 ± 0.2	6.510	6.724

Table 1. The 19 dSphs to be used in the analysis, with measured J factor (and uncertainities, both in log scale) in the 2nd column [Fermi], as well as the measured counts (3rd column) and estimated background counts (last column) in natural log scale.

Statistical Analysis

Let's pretend I am a frequentist for a second...

Model for dwarf *d* and energy bin *e*:

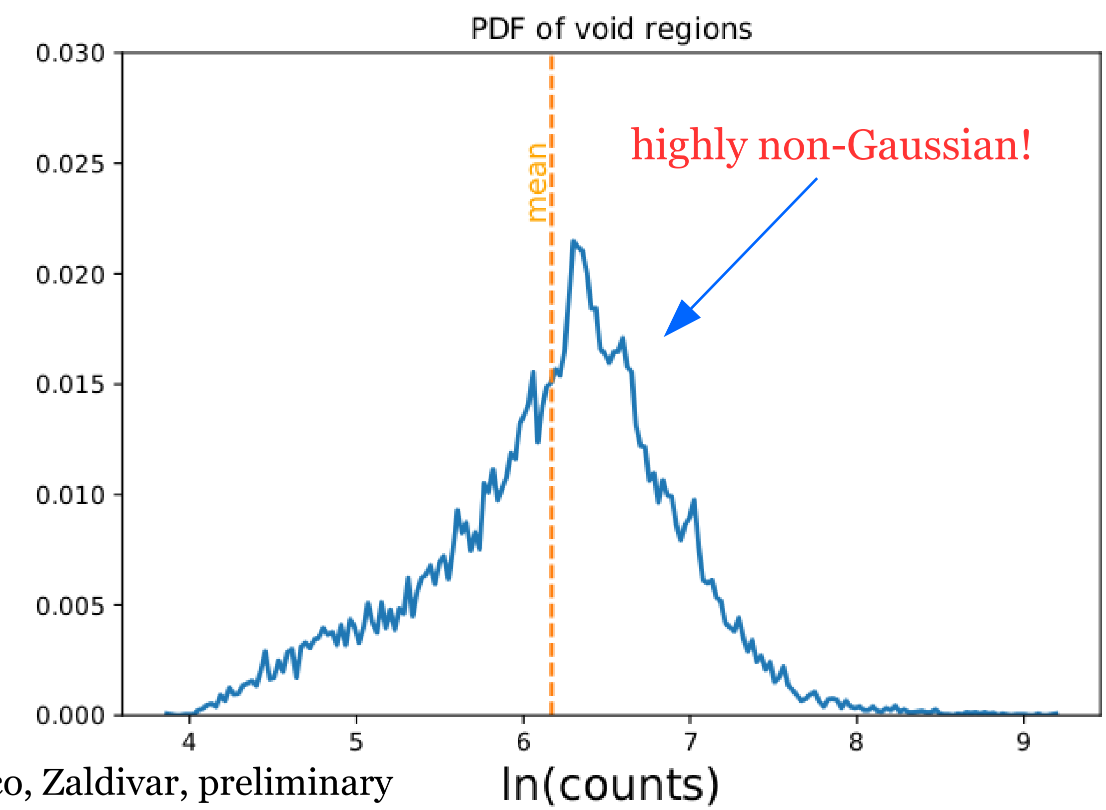
$$\lambda_{d,e} = J_d \langle \sigma v \rangle f_{d,e}(m_{\rm DM}) + b_{d,e}$$

with Likelihood:

Log-normal (as for Fermi)

$$\mathcal{L}_{d,e}(\lambda_{d,e}, J_d, b_{d,e}) = \frac{\lambda_{d,e}^{n_{d,e}} e^{-\lambda_{d,e}}}{n_{d,e}!} \mathcal{N}(\log J_d) \mathcal{B}(b_{d,e})$$

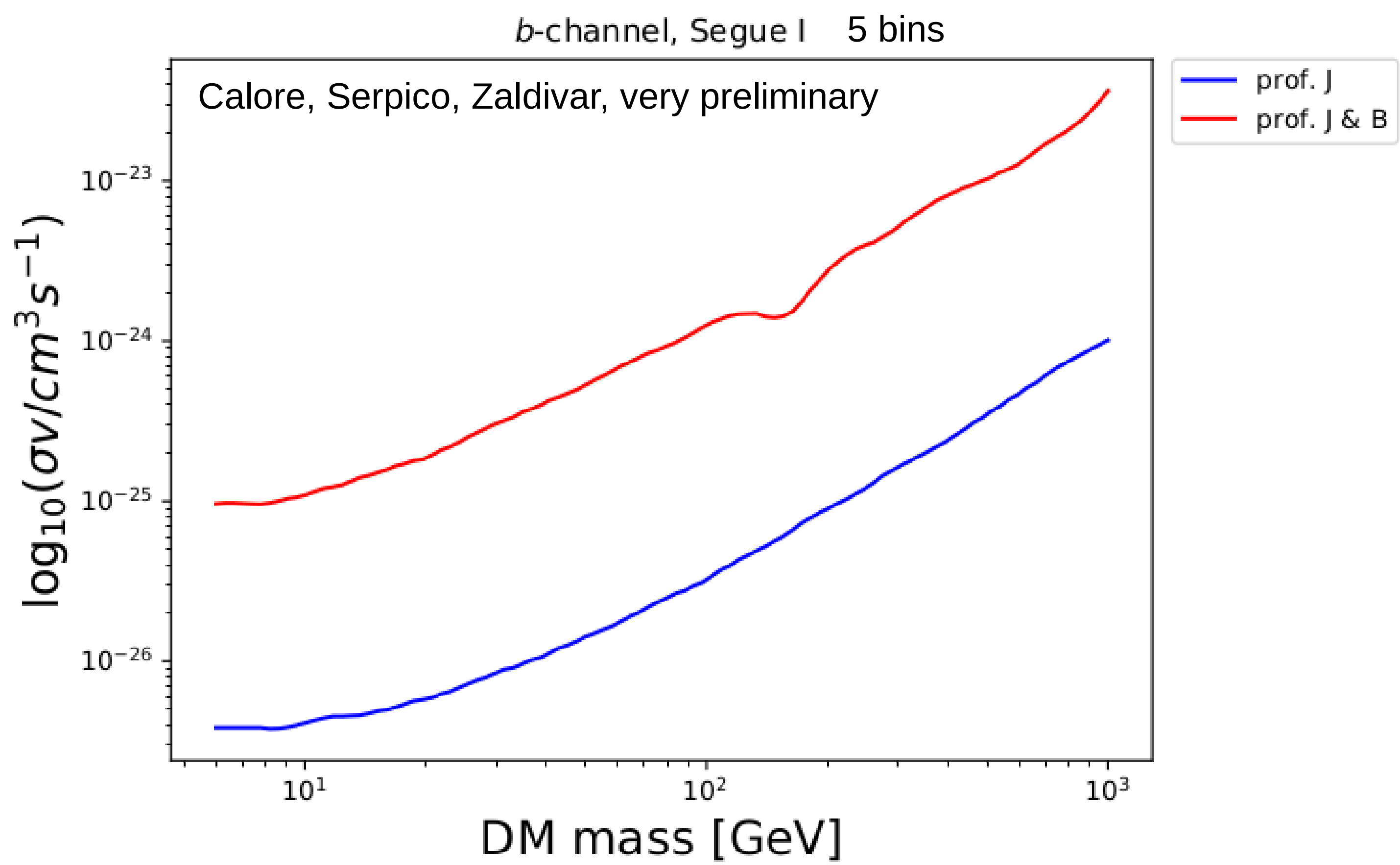
- **step beyond Fermi analysis**
- taken from the mother distribution
  - smoothed
  - re-centred for each dSph



- TS is log-likelihood ratio
- interested in $\langle \sigma v \rangle$ (for fixed mass)
- profiling over J and b

Calore, Serpico, Zaldivar, preliminary

Limits to DM parameter space



Things to play with:

- play with (energy) unbined sample
- dwarf stacking
- etc

Conclusions

- Regression problems are as important as classification for indirect detection
- Old “neural network” provides much (at least) faster estimation
- Background uncertainties are quite relevant for this analysis

Machine learning question

- Are there better methods?

Thanks!