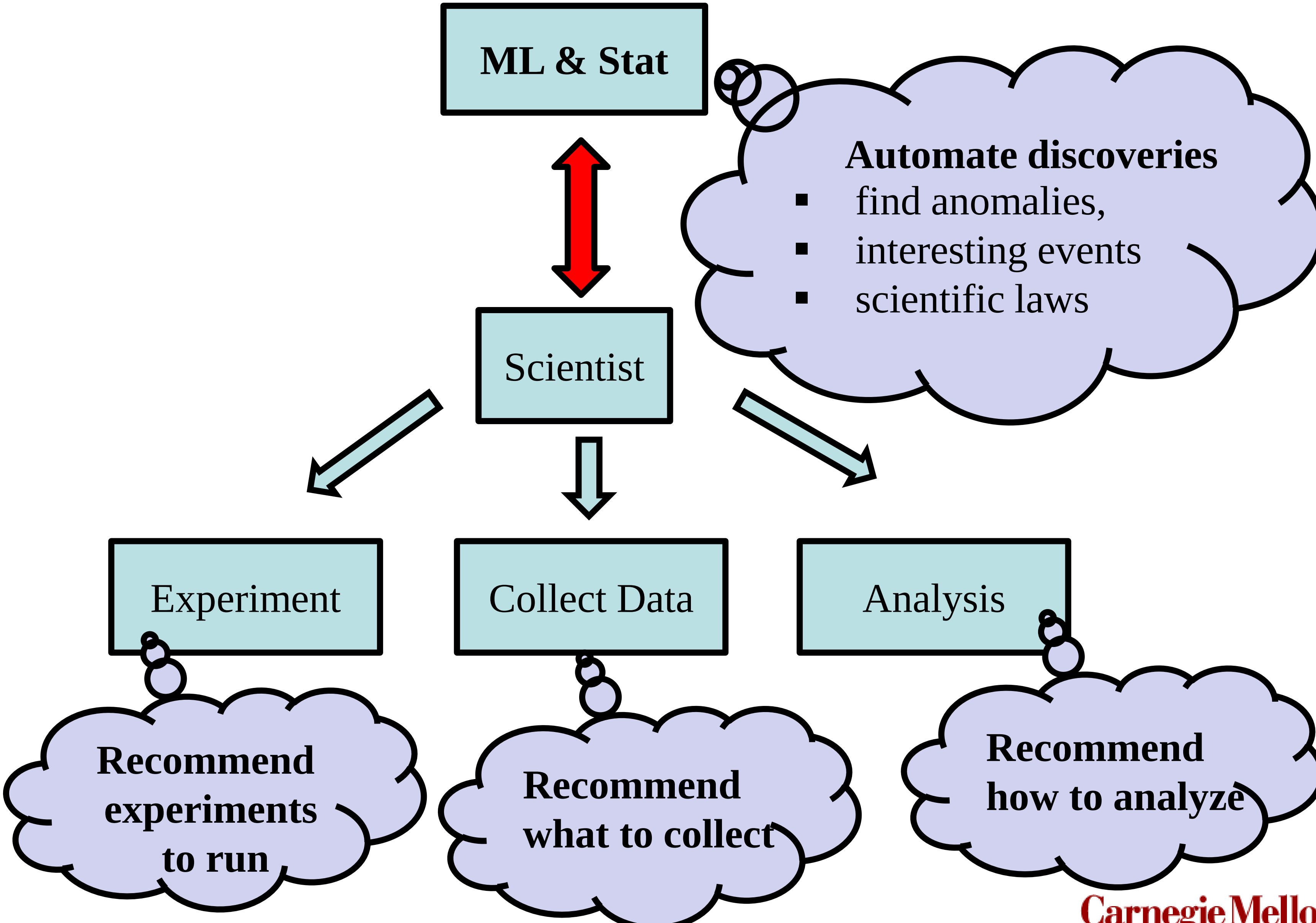
Distribution Regression and its Applications

Barnabás Póczos

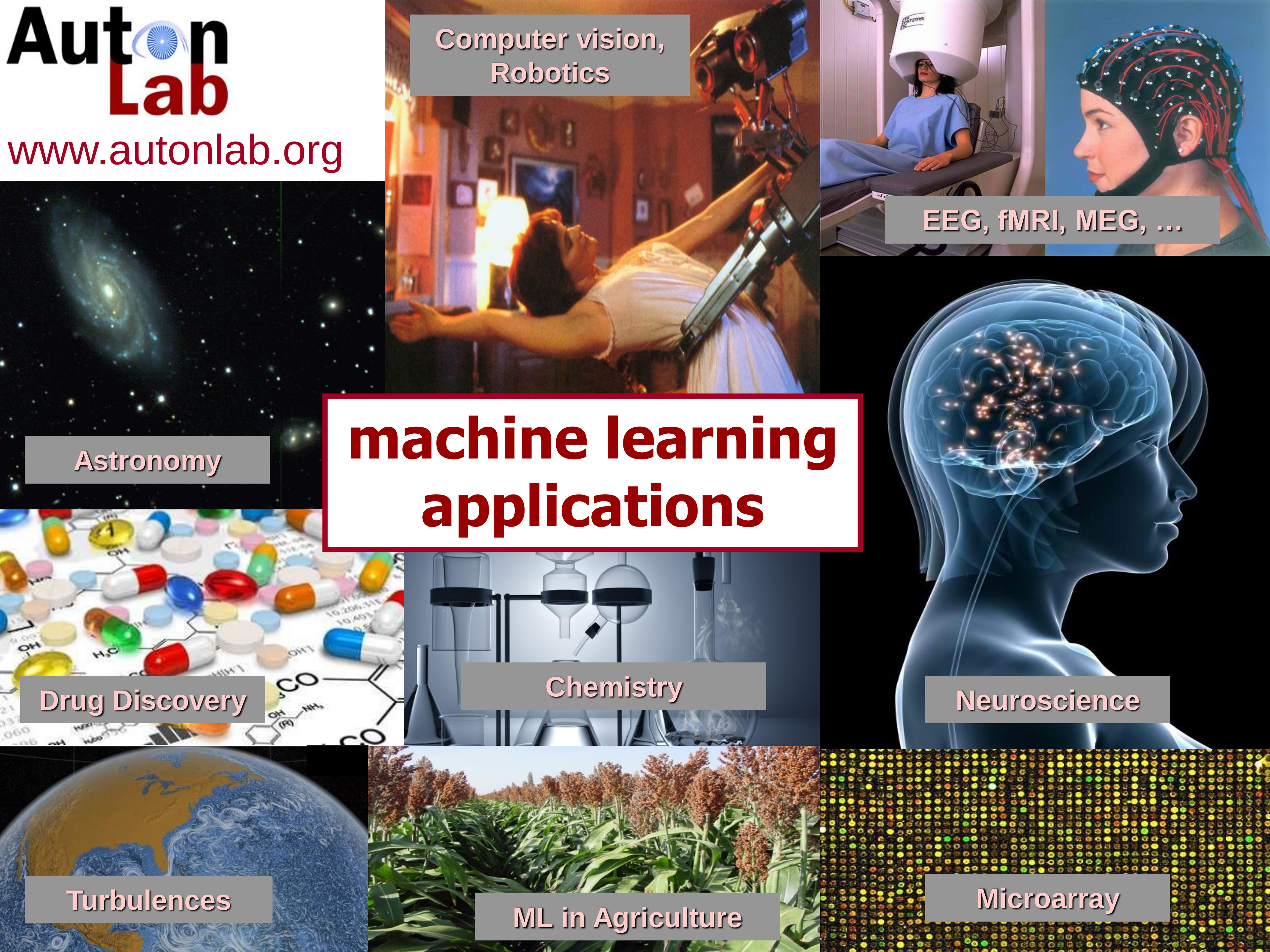
Carnegie Mellon University

Oct 12, 2017

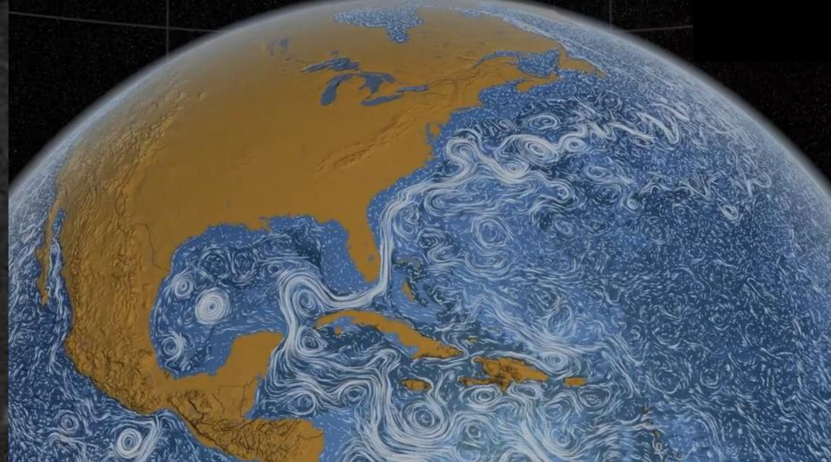
Goal: Create a Scientific Assistant



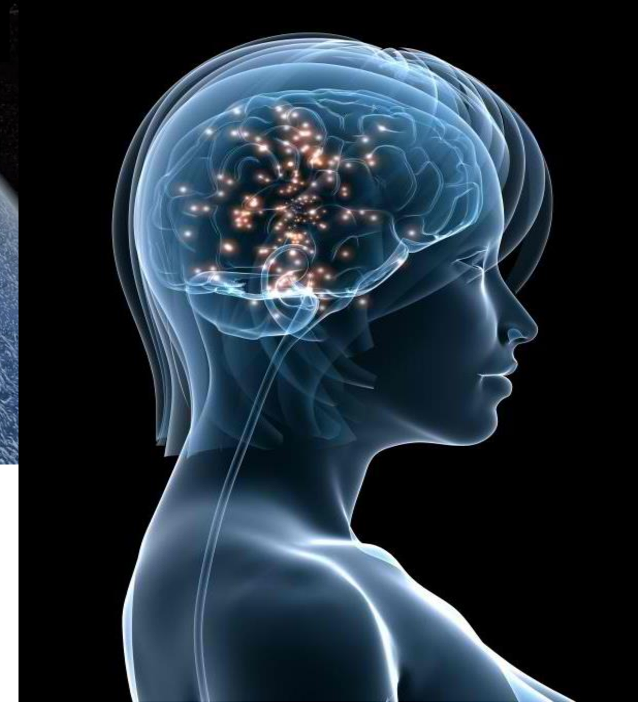
Carnegie Mellon

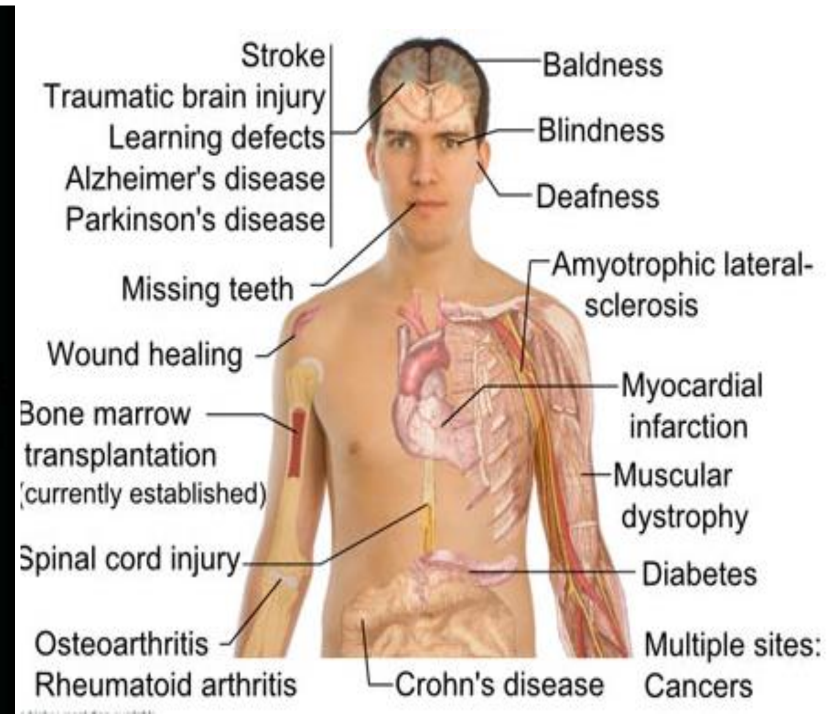

Auton Lab
www.autonlab.org
Computer vision, Robotics
EEG, fMRI, MEG, …
Astronomy
machine learning applications
Drug Discovery
Chemistry
Neuroscience
Turbulences
ML in Agriculture
Microarray

Why are we all here?

Curious




Stroke
Traumatic brain injury
Learning defects
Alzheimer's disease
Parkinson's disease
Missing teeth
Wound healing
Bone marrow transplantation (currently established)
Spinal cord injury
Osteoarthritis
Rheumatoid arthritis
Crohn's disease
Baldness
Blindness
Deafness
Amyotrophic lateral-sclerosis
Myocardial infarction
Muscular dystrophy
Diabetes
Multiple sites: Cancers

**To solve these problems,
our main tool is always the same**

Collect data & learn from data

The world is very complicated...

We have to understand complex relationships across the data.

Basic questions about the data

- ❑ *How random is the data?*
  - How large is its entropy?
- ❑ *How large is the dependence among the instances?* Which variables are dependent, which ones are independent?
  - How large is their mutual information?
- ❑ *How different are the distributions of the instances?*
  - How large is the divergence between the distributions?

Difficult & Important

⇒ We need Entropy, Dependence, and Divergence estimators to do machine learning

Entropy, Mutual Information, Divergence

C. Shannon

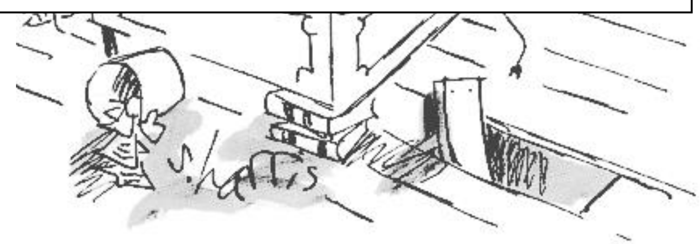
$$H = -\int p \log p$$

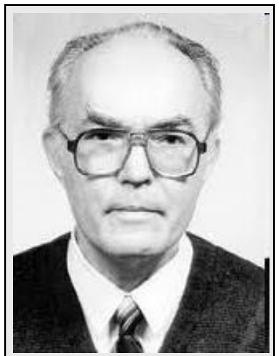
$$KL(p||q) \doteq \int p \log \frac{p}{q}$$

$$I = KL(p|| \prod p_i)$$

Fernandes & Gloor: Mutual information is critically dependent on prior assumptions: **would the correct estimate of mutual information please identify itself?**

BIOINFORMATICS Vol. 26 no. 9 2010, pages 1135–1139





I. Csiszár

$$D_f(p||q) \doteq \int f\left(\frac{p(x)}{q(x)}\right) q(x)\, d\mu(x)$$

MI = Divergence between $p(x_1, \ldots, x_d)$ and $\prod_{i=1}^{d} p_i(x_i)$

Developing efficient estimators for mutual information and related quantities is highly important in many applications.

- “**Mutual information**” query produces 325,000 hits on Google Scholar, and the first 10 papers have more than 30,065 citations.

- Most of these papers are application papers, e.g. in feature selection, computer vision, medical image processing, image alignment, and data fusion. As we find better estimators, such applications can simply use them .

- “**Big Data**” search on Google Scholar produces 181,000 hits, and the first 10 hits have 12,872 citations.

- Similarly, the “**Deep Learning**” search produces 106,000 hits, and the first 10 papers have 8,485 citations (as of May 28, 2017).

How should we estimate them?

Using $X_{1:n} \doteq (X_1, \ldots, X_n)$ i.i.d. sample $\sim f$

Estimate Rényi entropy $R_\alpha = \frac{1}{1-\alpha} \log \int f^\alpha(\mathrm{x}) \mathrm{dx}$

Naïve plug-in approach using density estimation

- histogram
- kernel density estimation
- k-nearest neighbors [D. Loftsgaarden & C. Quesenberry. 1965.]

Density: nuisance parameter
Density estimation: difficult, **curse of dimensionality!**

How can we estimate them directly,
without estimating the density?

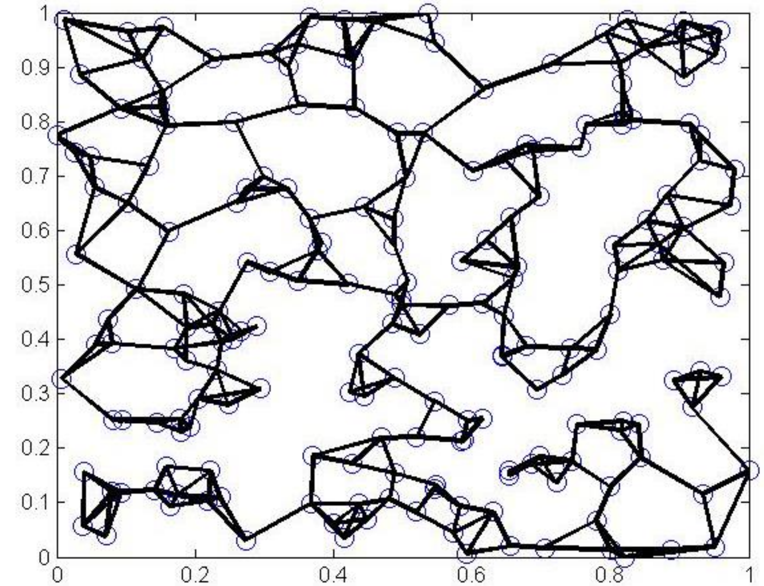
ENTROPY ESTIMATION

without density estimation

Using $X_{1:n} \doteq (X_1, \ldots, X_n)$ i.i.d. sample $\sim f$

Estimate Rényi entropy $R_\alpha = \frac{1}{1-\alpha} \log \int f^\alpha(\mathbf{x}) d\mathbf{x}$

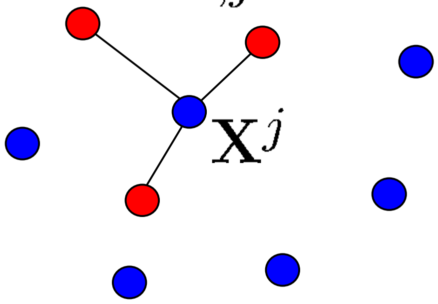
Rényi-α entropy estimators using kNN graphs



$\mathbf{X}^1, \ldots, \mathbf{X}^n \sim f$ i.i.d. samples in $\mathbb{R}^d$

Let $p \doteq d - d\alpha$, k fixed.

Let $\mathcal{N}_{k,j}$ be the set of the k nearest neighbours of $\mathbf{X}^j$ in $\{\mathbf{X}^1, \ldots, \mathbf{X}^n\}$



$k = 3$

Calculate: $L_n = \sum_{j=1}^{n} \sum_{\mathbf{V} \in \mathcal{N}_{k,j}} \|\mathbf{V} - \mathbf{X}^j\|^p$

$$H_n(\mathbf{X}^{1:n}) \doteq \frac{1}{1-\alpha} \log\left(\frac{L_n}{\beta_{d,p,k} n^{\alpha}}\right)$$

Distances / Divegences between Distributions

Euclidean: $D(p,q) = (\int (p(x) - q(x))^2 dx)^{1/2}$

Kullback-Leibler: $D(p,q) = KL(p,q) = \int p(x) \log \frac{p(x)}{q(x)} dx$

Renyi: $D(p,q) = R_\alpha(p\|q) = \frac{1}{\alpha-1} \log \int p^\alpha q^{1-\alpha}$

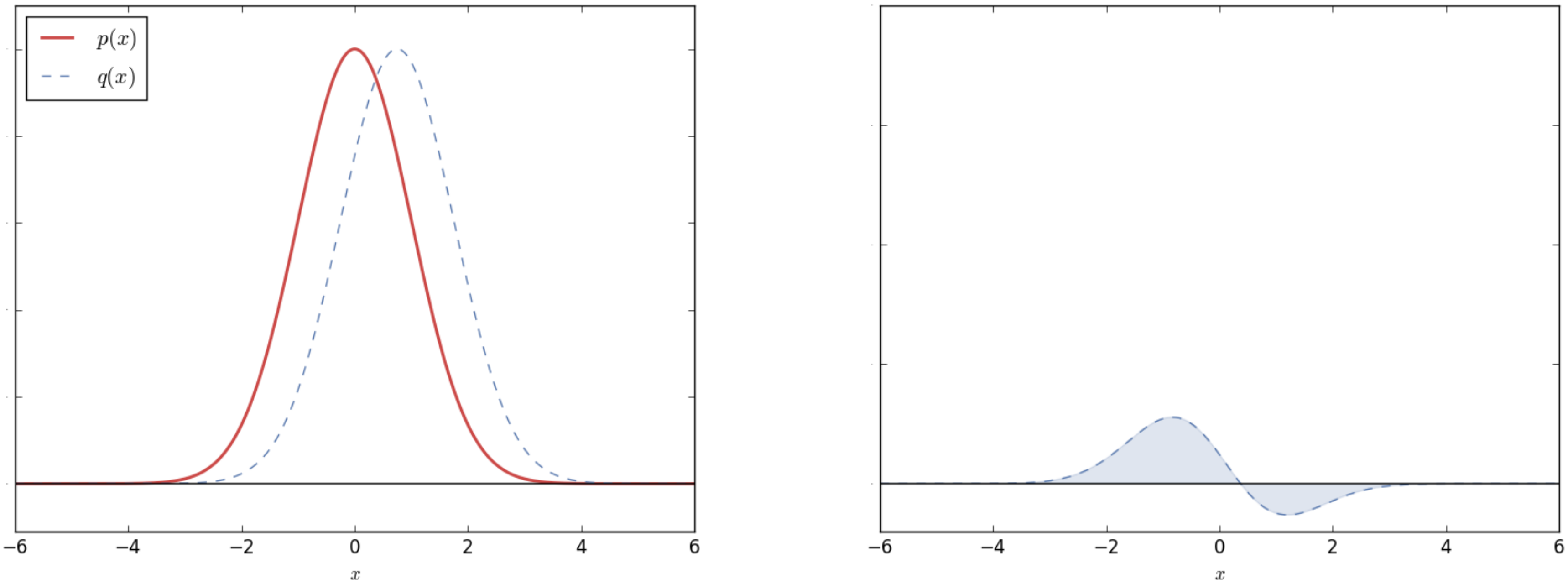
RÉNYI DIVERGENCE ESTIMATION

without density estimation

Using $X_{1:n} = \{X_1, \ldots, X_n\} \sim p \quad Y_{1:m} = \{Y_1, \ldots, Y_m\} \sim q$

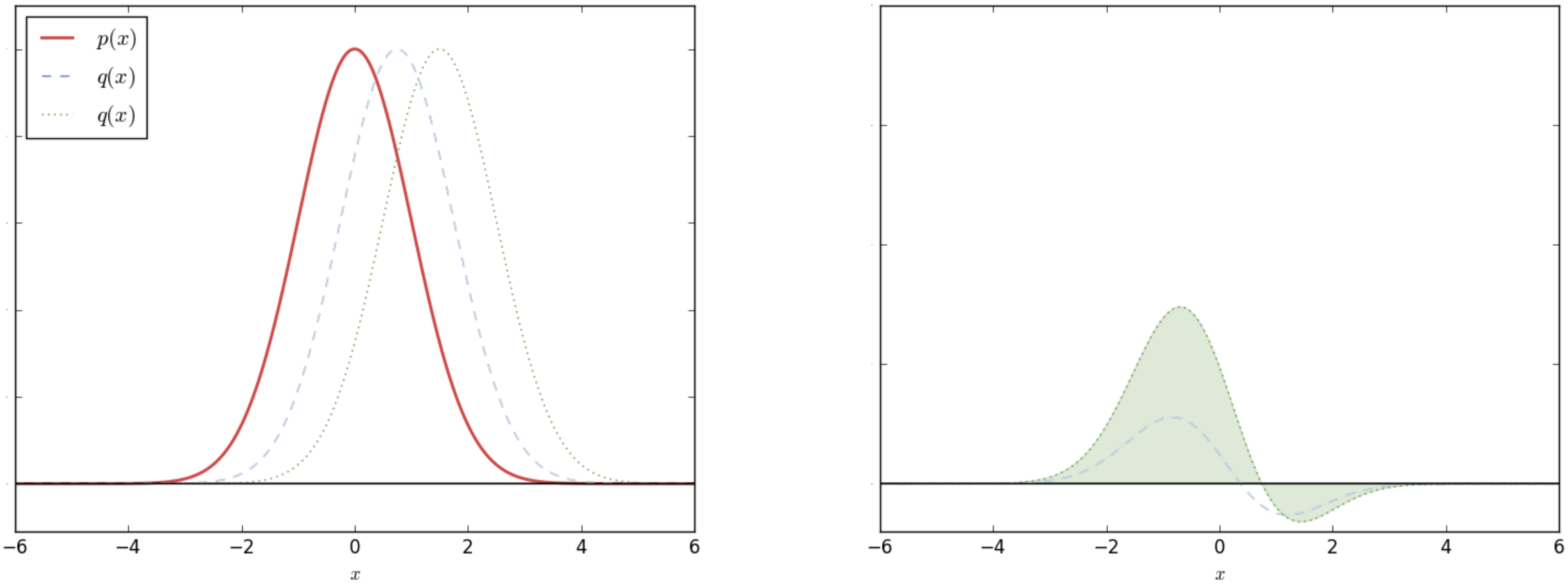
Estimate divergence $R_\alpha(p\|q) \doteq \frac{1}{\alpha - 1} \log \int p^\alpha q^{1-\alpha}$

KL Divergence



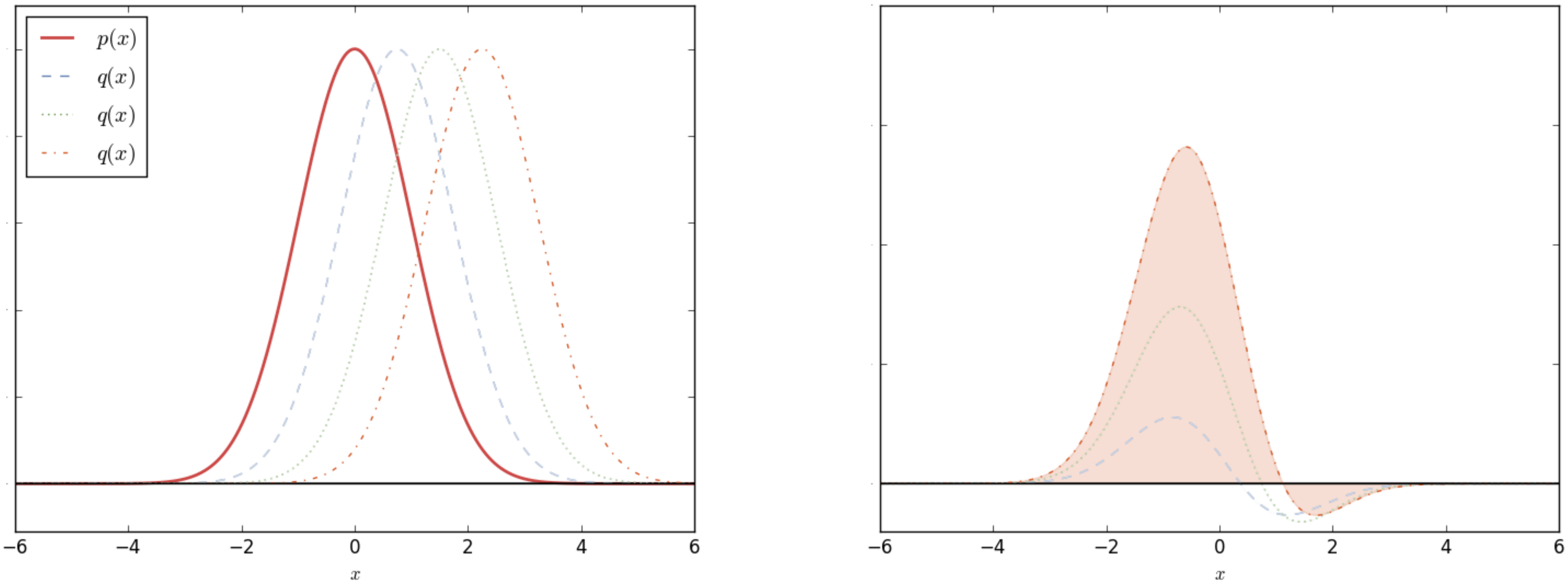
$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \ln\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x$$

KL Divergence



$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \ln\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x$$

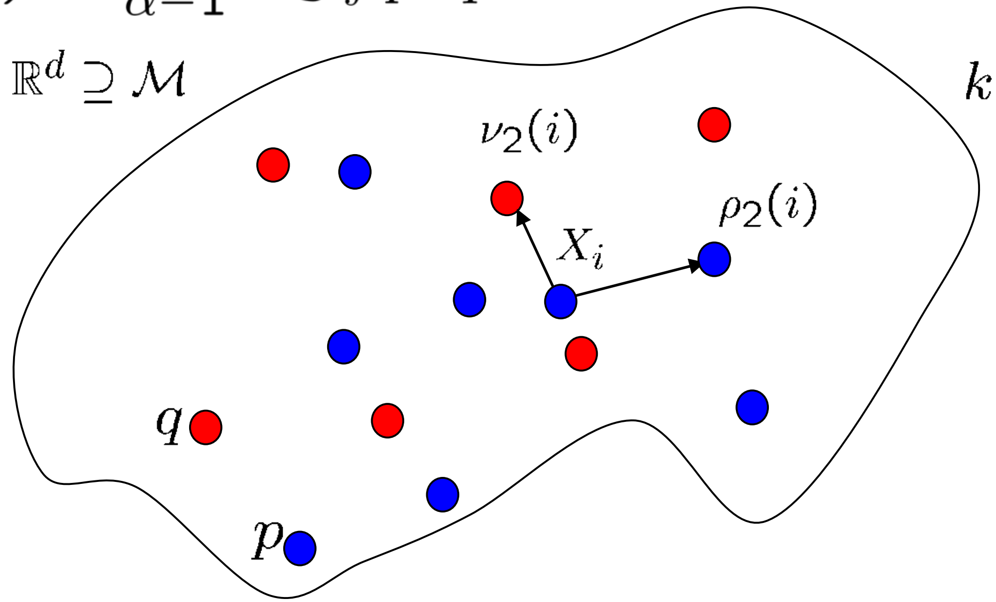
KL Divergence



$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \ln\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x$$

The Estimator

Renyi: $R_\alpha(p\|q) = \frac{1}{\alpha-1} \log \int p^\alpha q^{1-\alpha}$



$k = 2.$

$k \geq 1$, fixed.

$\rho_k(i)$: the distance of the k-th nearest neighbor of X_i in $X_{1:n}$

$\nu_k(i)$: the distance of the k-th nearest neighbor of X_i in $Y_{1:m}$

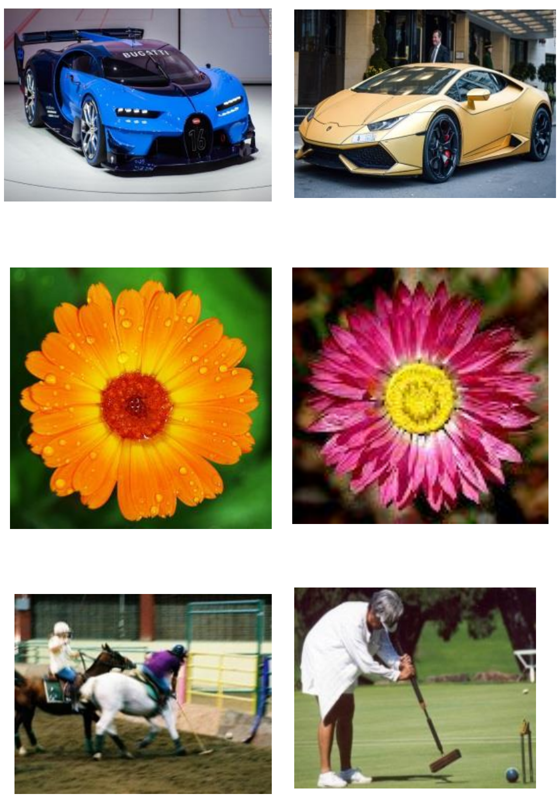
$$D_\alpha(p\|q) \doteq \int p^\alpha q^{1-\alpha}$$

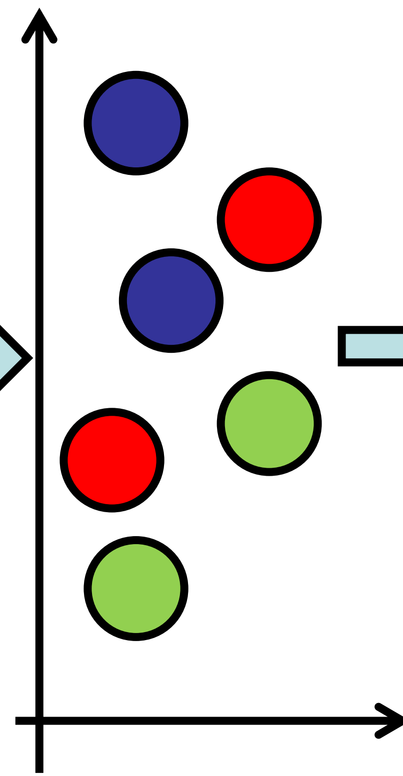
$$\widehat{D}_\alpha(X_{1:n}\|Y_{1:m}) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{(n-1)\rho_k^d(i)}{m\nu_k^d(i)}\right)^{1-\alpha} \frac{\Gamma(k)^2}{\Gamma(k-\alpha+1)\Gamma(k+\alpha-1)}$$

Machine Learning on Complex Objects

Traditional Machine Learning

Observations → Feature vectors → Training data of feature vectors



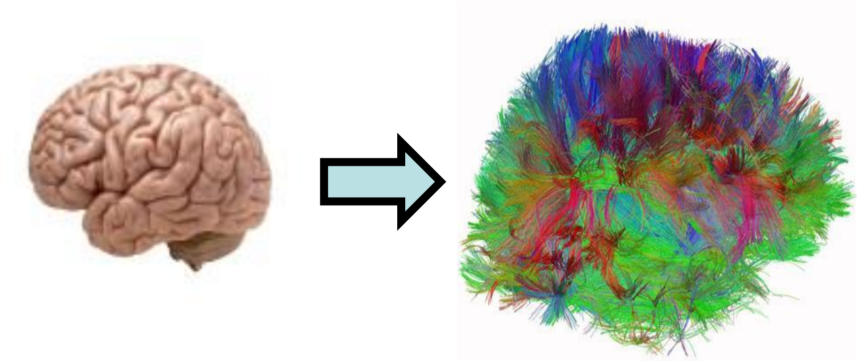


ML algorithm: classification, regression, clustering, etc

Complex Data is Everywhere

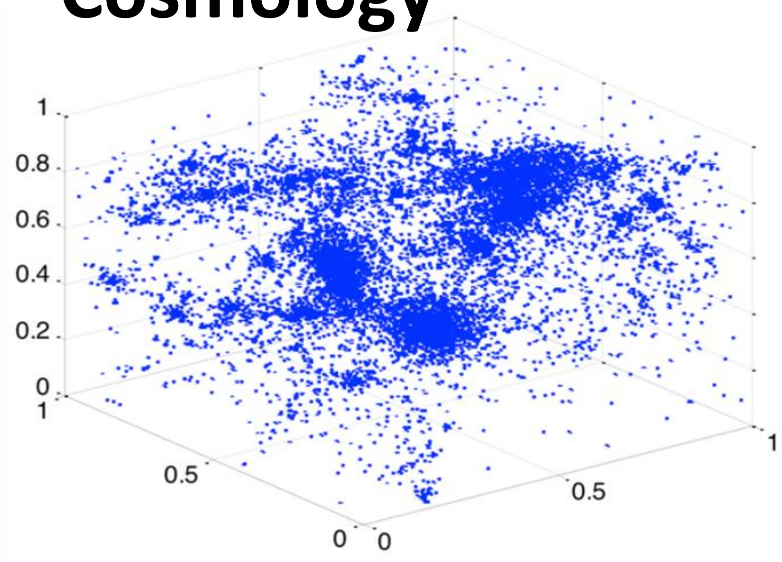
Finance

Neuroscience



Diffusion Weighted Imaging

Cosmology



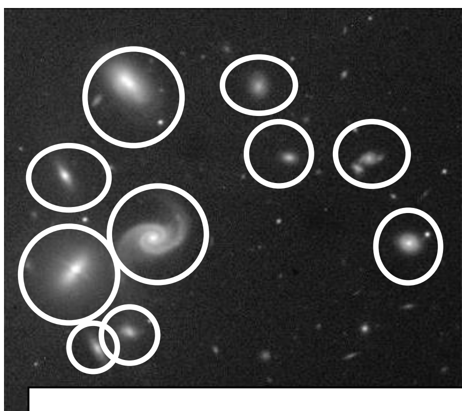
Images

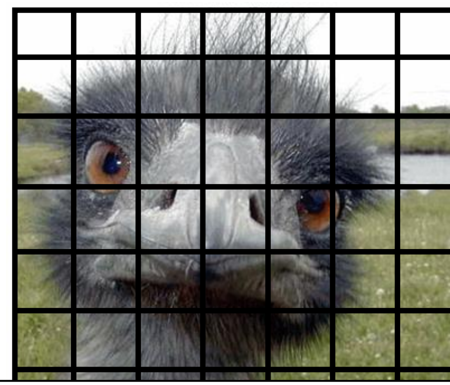
Generalize ML to sets and distributions

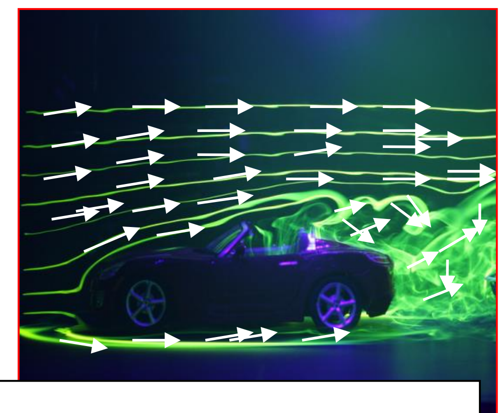
Most machine learning algorithms operate on **vectorial objects**.

The world is **complicated.** Often

- hand crafted vectorial features are not good enough
- natural to work with complex inputs directly (**sets** or **distributions**...)







Classify galaxy clusters

- ❑ Each **galaxy** can be represented by a **feature vector**
- ❑ Each **cluster** can be represented by a **set** of these vectors
- ❑ We can't concatenate the feature vectors into a huge vector

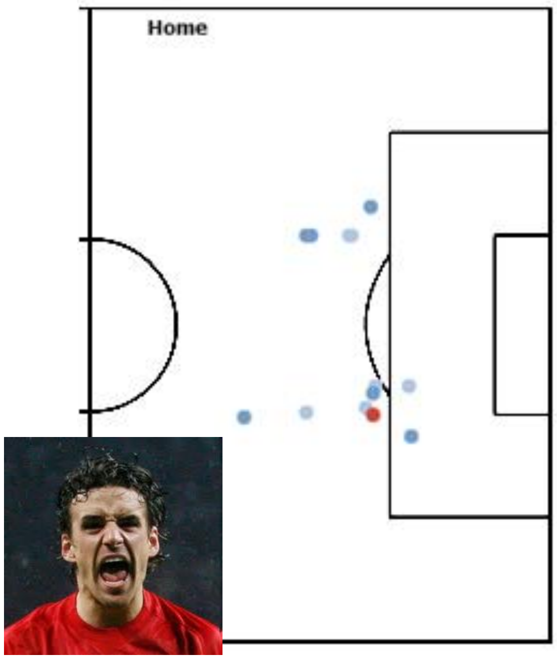
Deal

on

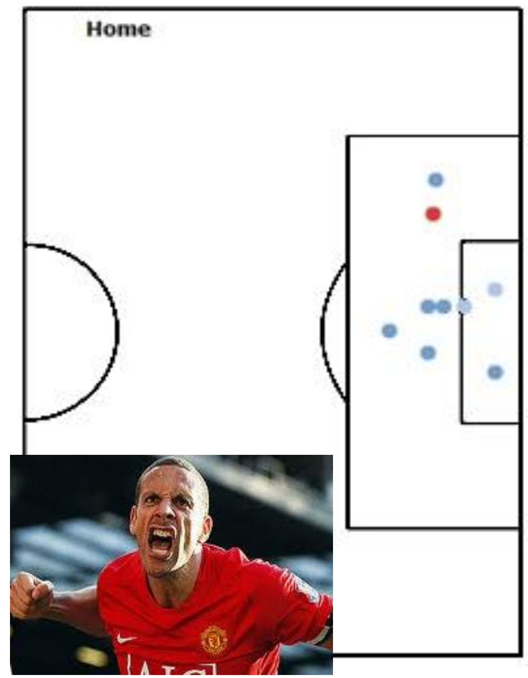
❑ do ***ML on these unknown distributions*** represented by sets

Distributional Data

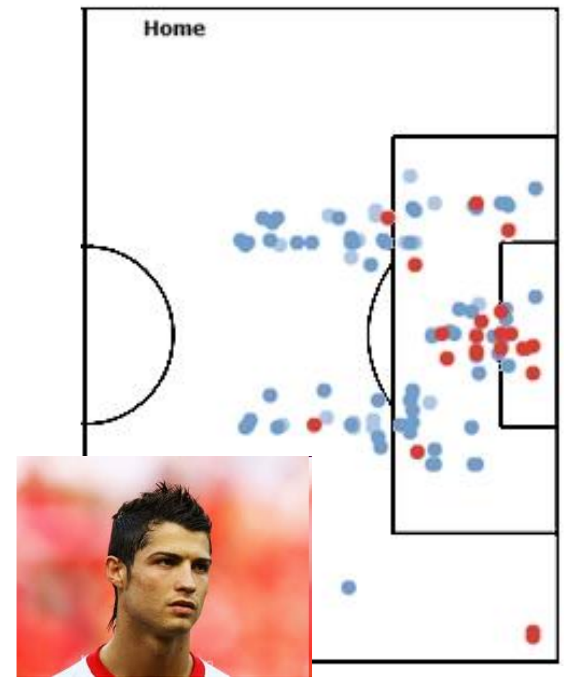
Manchester United 07/08



Owen Hargreaves



Rio Ferdinand



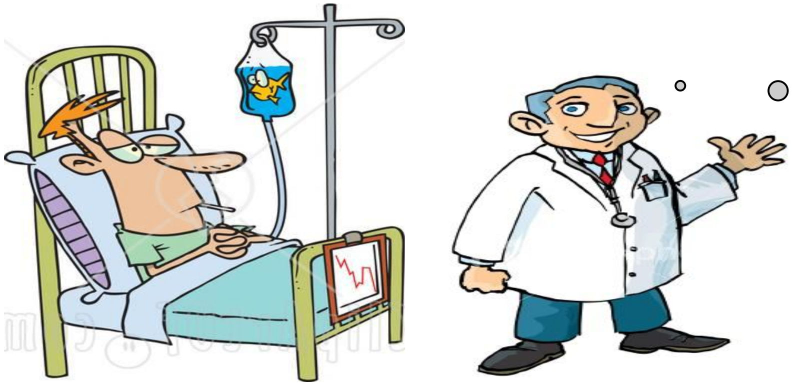
Cristiano Ronaldo

Shot Type

- Goals
- Shots on Goal
- Shots

www.juhokim.com/projects.php

ML on Distributions



healthy or sick?

Standard machine learning

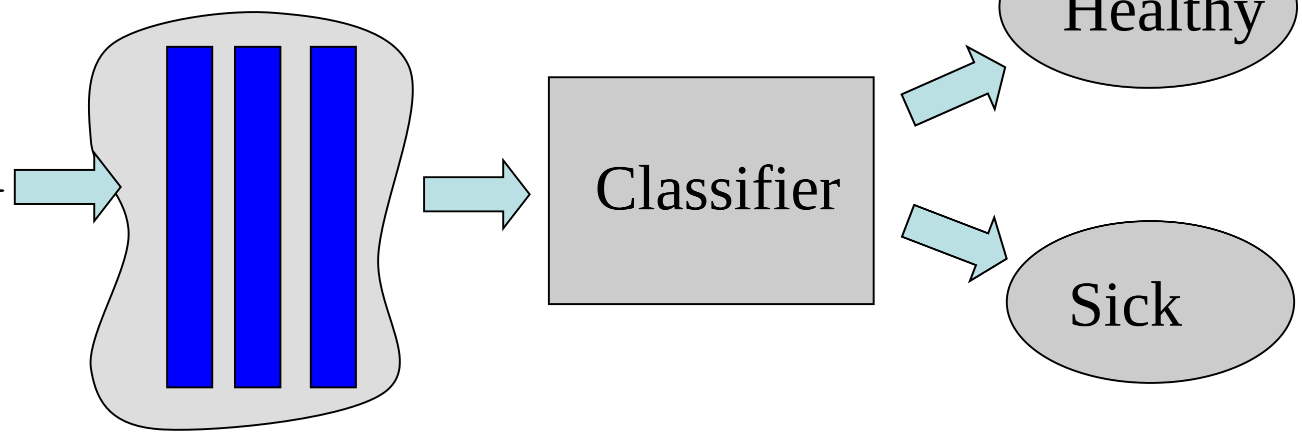
ML on sets/distributions

Medical tests:

blood pressure,
heart rate,
temperature,
blood sample
…

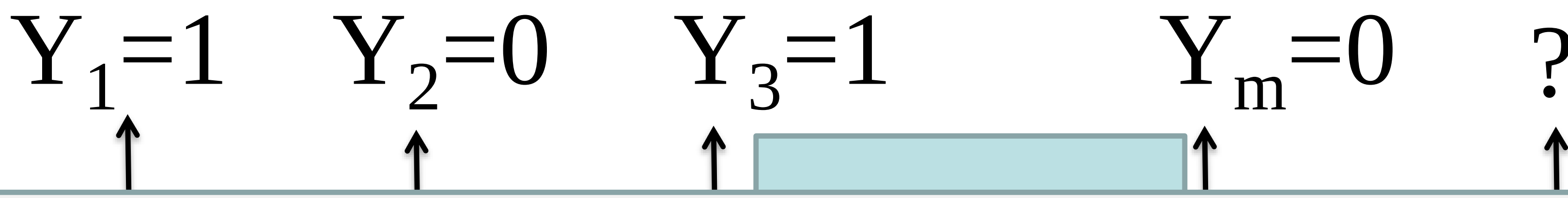
Feature vector

Set of feature vectors



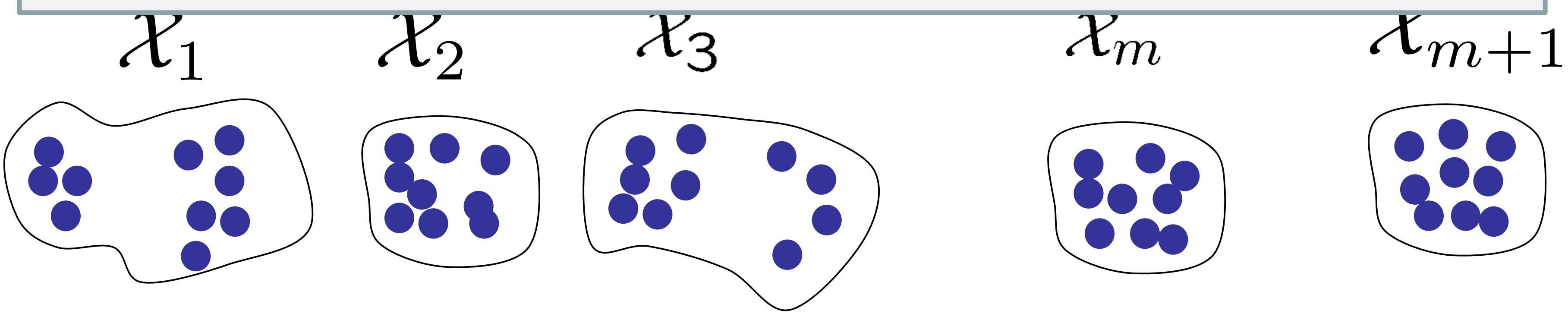
What happens if we repeat the medical tests?

Distribution Regression / Classification

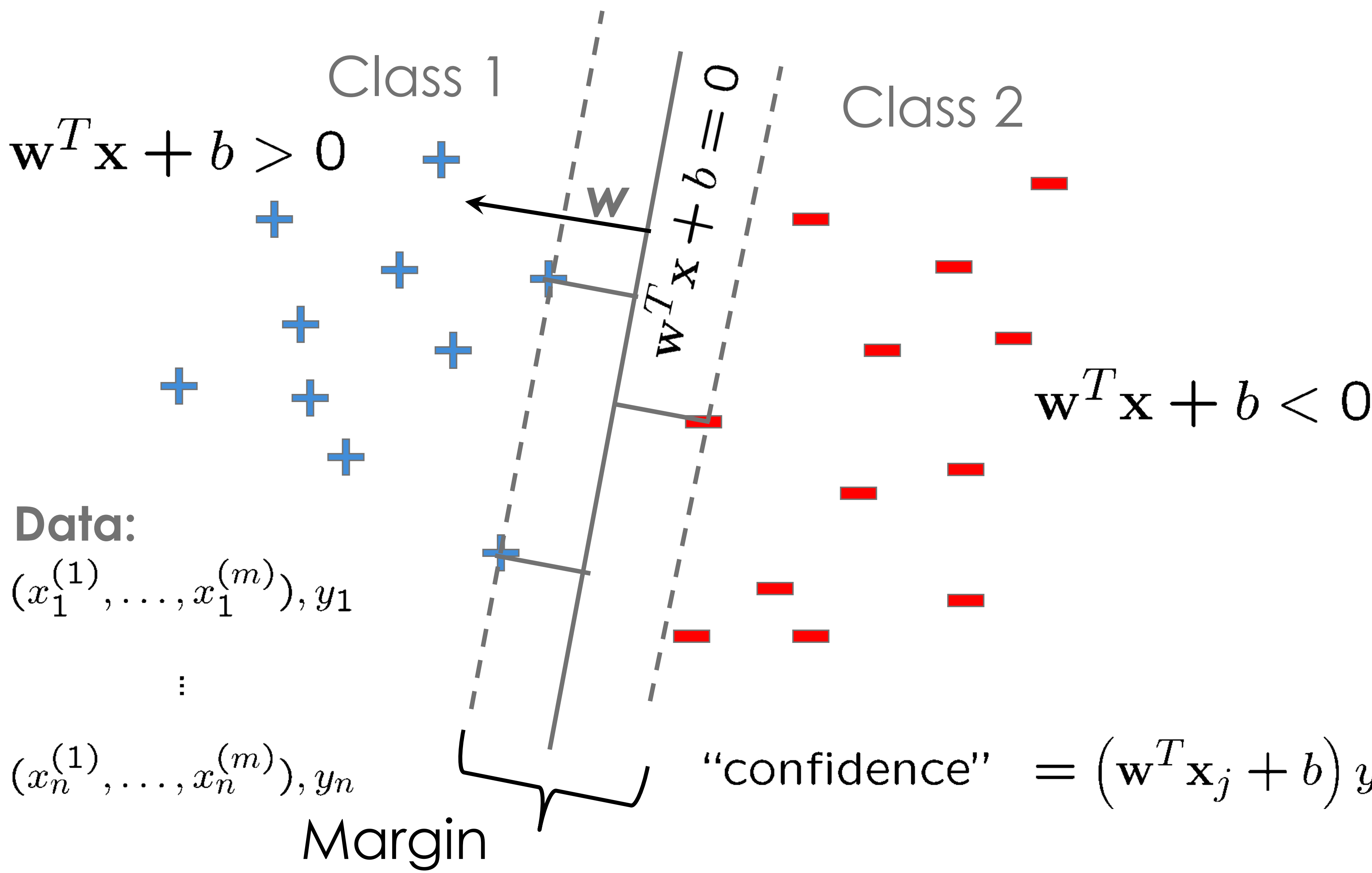


Differences compared to standard methods on vectors

- The inputs are distributions, density functions (not vectors)
- We don't know these distributions, only sample sets are available (error in variables model)



Support Vector Machines



Data:

$(x_1^{(1)}, \ldots, x_1^{(m)}), y_1$

$\vdots$

$(x_n^{(1)}, \ldots, x_n^{(m)}), y_n$

"confidence" $= \left(\mathbf{w}^T\mathbf{x}_j + b\right) y_j$

The Primal Hard SVM

- Given $D = \{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$ training data set.
- Assume that D is **linearly separable**.

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w} \in \mathbb{R}^m} \frac{1}{2} ||\mathbf{w}||^2$$

$$\text{subject to } y_i \langle \mathbf{x}_i, \mathbf{w} \rangle \geq 1, \ \forall i = 1, \dots, n$$

Prediction: $f_{\hat{\mathbf{w}}}(\mathbf{x}) = \text{sign}(\langle \hat{\mathbf{w}}, \mathbf{x} \rangle)$

This is a QP problem (m-dimensional)
(Quadratic cost function, linear constraints)

The Dual Hard SVM

$$\boldsymbol{Y} \doteq diag(y_1, \ldots, y_n),\ y_i \in \{-1, 1\}^n$$

$\boldsymbol{K} \in \mathbb{R}^{n\times n} \doteq \{K_{ij}\}_{i,j}^{n,n}$, where $K_{ij} \doteq \langle \mathbf{x}_i, \mathbf{x}_j \rangle$ Gram matrix.

$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \boldsymbol{\alpha}^T \mathbf{1}_n - \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Y}\boldsymbol{K}\boldsymbol{Y}\boldsymbol{\alpha}$$

subject to $\alpha_i \geq 0,\ \forall i = 1, \ldots, n$

Quadratic Programming (n-dimensional)

Lemma $\hat{\mathbf{w}} = \sum_{i=1}^{n} \hat{\alpha}_i y_i \mathbf{x}_i$

Prediction: $f_{\hat{\mathbf{w}}}(x) = \text{sign}(\langle \hat{\mathbf{w}}, \mathbf{x} \rangle) = \text{sign}(\sum_{i=1}^{n} \hat{\alpha}_i y_i \underbrace{\langle \mathbf{x}_i, \mathbf{x} \rangle}_{k(\mathbf{x}_i, \mathbf{x})})$

Distribution Classification

We have T sample sets, $(\mathbf{X}_1, \ldots, \mathbf{X}_T)$. **[Training data]**
$\{X_{t,1}, \ldots, X_{t,m_t}\} = \mathbf{X_t} \sim p_t$. $\mathbf{X_t}$ has class $Y_t \in \{-1, +1\}$.

What is the class label Y of $\mathbf{X} = \{X_1, \ldots, X_m\} \sim p$?

Solution: *Use RKHS based SVM!*

Calculate the Gram matrix $K_{ij} \doteq \langle \phi(p_i), \phi(p_j) \rangle_{\mathcal{K}} = K(p_i, p_j)$

Dual form of SVM:

$$\widehat{\alpha} = \arg\max_{\alpha \in \mathbb{R}^T} \sum_{i=1}^{T} \alpha_i - \frac{1}{2} \sum_{i,j}^{T} \alpha_i \alpha_j y_i y_j K_{ij},$$

subject to $\sum_i \alpha_i y_i = 0$, $0 \leq \alpha_i \leq C$.

$$Y = \text{sign}\left(\sum_{i=1}^{T} \widehat{\alpha}_i y_i K(p_i, p)\right) \in \{-1, +1\}$$

Problems: We do not know p_i, p, $K(p_i, p_j)$, or $K(p_i, p)$...

Kernel Estimation

Linear kernel: $K(p,q) = \int pq$

Polynomial kernel: $K(p,q) = (\int pq + c)^s$

Gaussian kernel: $K(p,q) = \exp(-\frac{1}{2\sigma^2}(\int (p-q)^2).$

We only need to estimate $\int p^\alpha q^\beta$ terms.

We already know how!

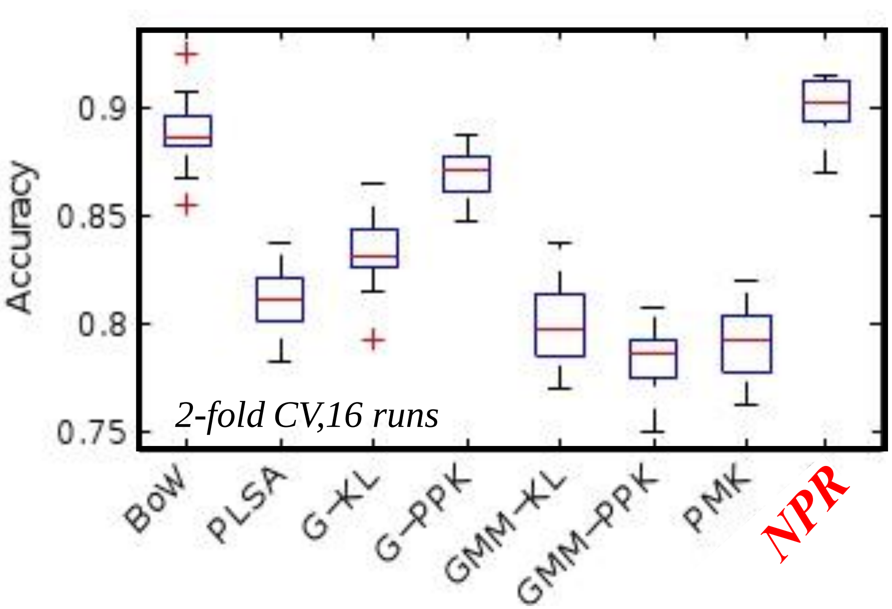
We can also try to use other $\mu(p,q)$ divergences, e.g. Rényi ...

The $\{\widehat{K}_{i,j}\}_{ij}$ Gram matrix might not be PSD!

Solution: make it symmetric, and project it to the cone of PSD matrices

Object Classification
ETH-80 [Leibe and Schiele, 2003]

8 categories, 400 images, each image is represented by 576 18 dim points



- BoW: **88.9%**
- NPR: **90.1%**

Póczos, Xiong, Sutherland, & Schneider, *CVPR 2012*

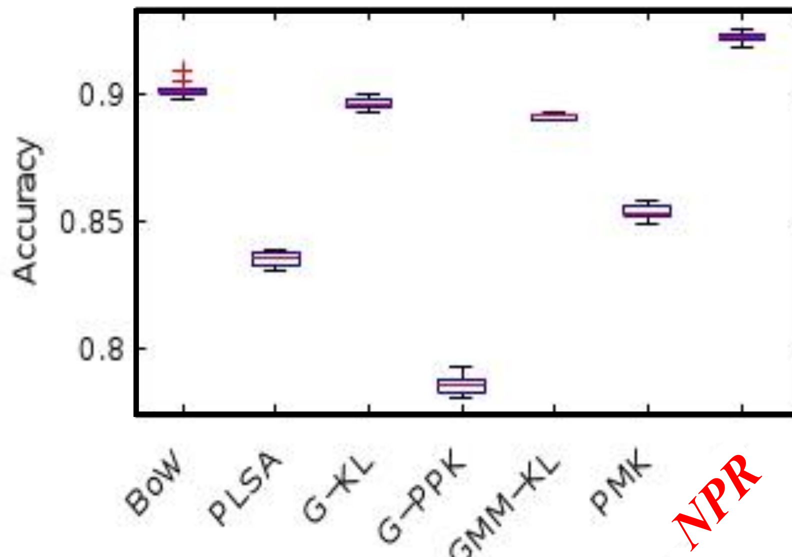
Outdoor Scenes Classification
[Oliva and Torralba, 2001]

coast *forest* *highway* *city*

mountain *country* *street* *tall building*

8 categories, 2688 images,
each represented by 1815 53 dim points.

- Best published: **91.57%**
  (Qin and Yung, ICMV 2010)
- NPR: **92.3%**



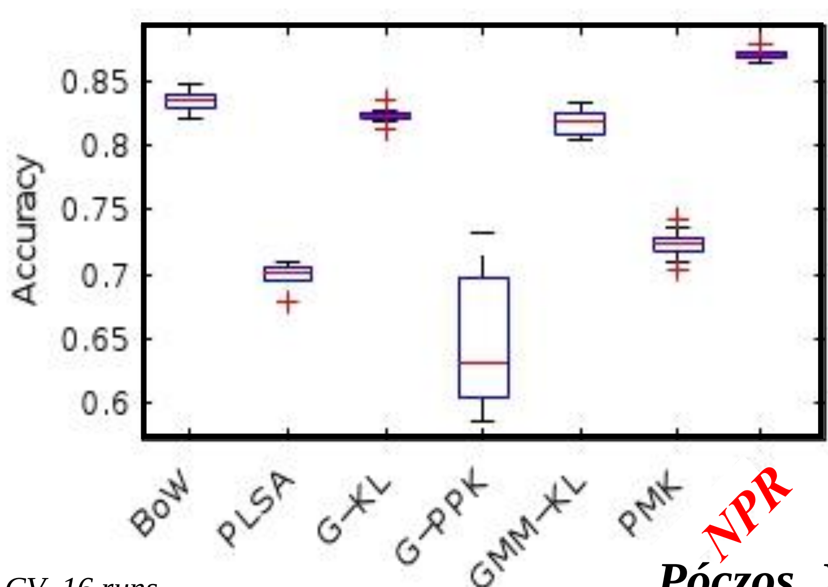
10 fold CV, 16 runs

Póczos, Xiong, Sutherland, & Schneider, *CVPR 2012*

Sport Events Classification
[Li and Fei Fei, 2007]

badminton *bocce* *croquet* *polo* *climbing* *rowing* *sailing* *snowboard*

8 categories, 1040 images, each represented by 295 to 1542 57 dim points.



- ❑ Best published: **86.7%**
  (Zhang et al, CVPR 2011)
- ❑ NPR: **87.1%**

2 fold CV, 16 runs

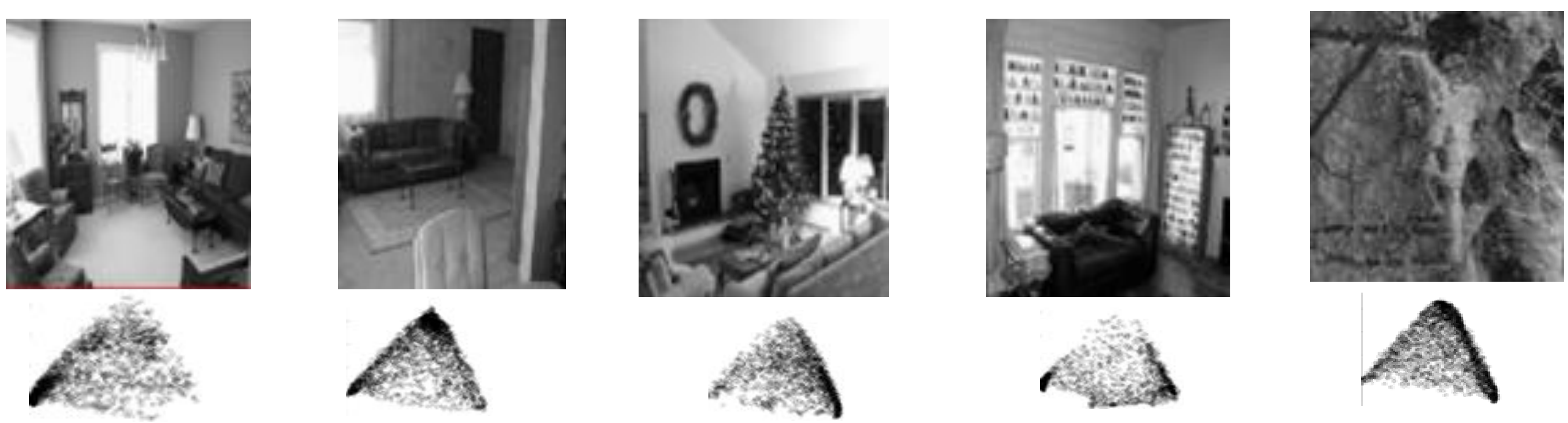
***Póczos, Xiong, Sutherland, & Schneider**, CVPR 2012*

Detecting Anomalous Images

B. Póczos, L. Xiong & J. Schneider, UAI, 2011.

50 highway images

5 anomalies

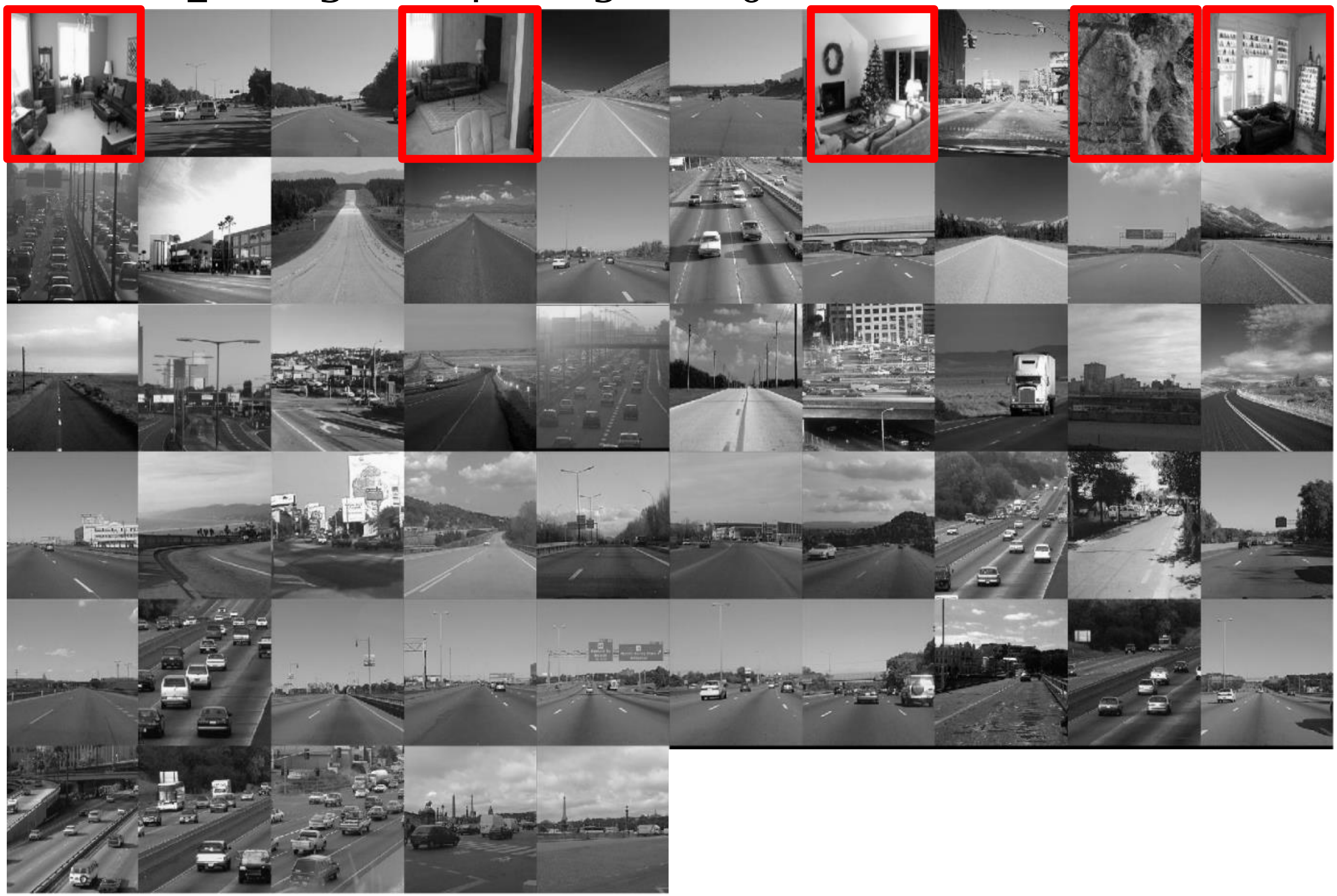


2-dimensional sample set representation of images (128 dim SIFT ⇒ 2 dim)

Anomaly score: divergences between the distributions of these sample sets

Detecting Anomalous Images

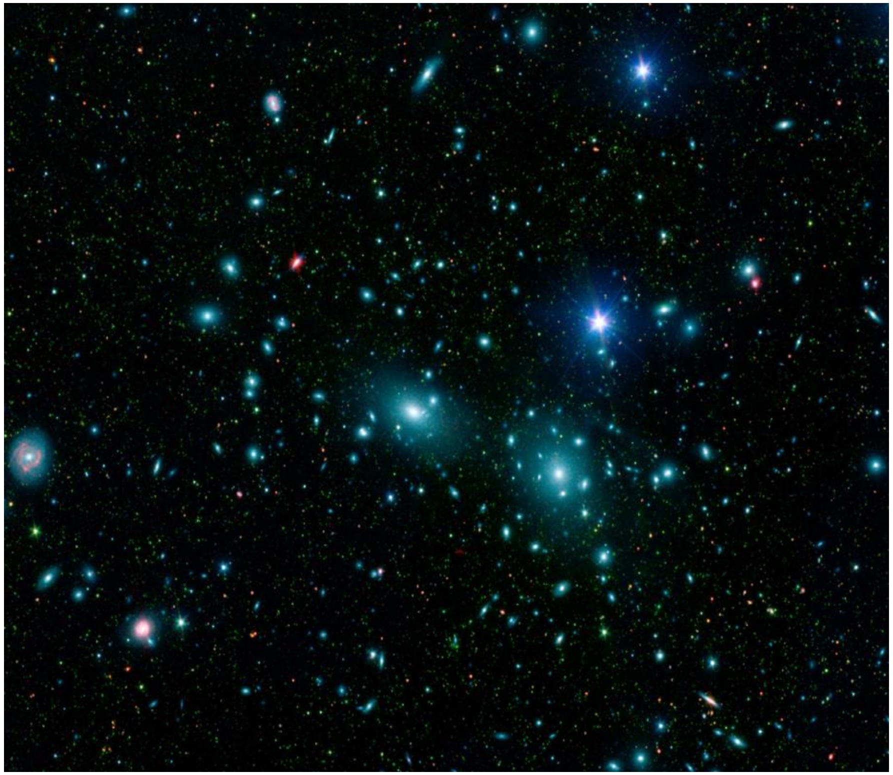
1 2 3 4 5 6 7 8 9 10



51 52 53 54 55

Cosmology Applications

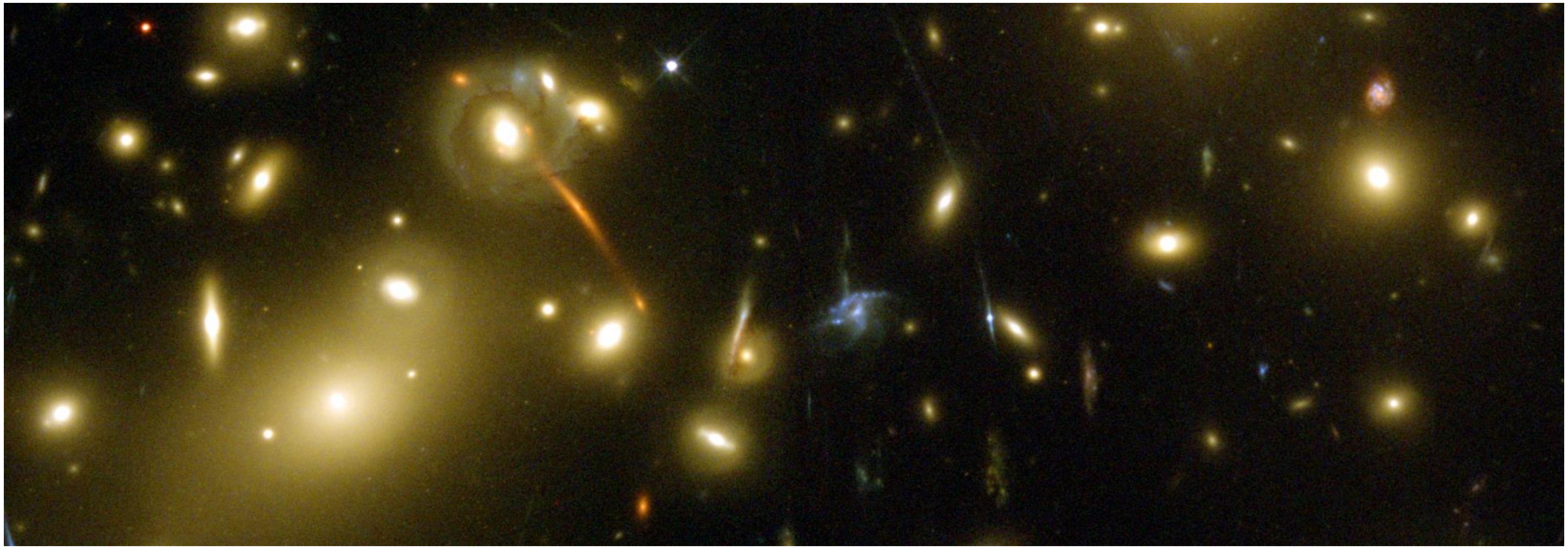
Scientific Applications



- Find new "scientific laws" / do better prediction
  (e.g. in estimating the mass of galaxy clusters)
- Find interesting/anomalous objects in the sky
- Recommend experiments to find the parameters of Universe

Image credit: nasa.gov, Hubble Space Telescope

Find new scientific laws in physics



Goal: Estimate dynamical mass of galaxy clusters.

Importance: Galaxy clusters are being the largest gravitationally bound systems in the Universe. Dynamical mass measurements are important to understand the behavior of dark matter and normal matter.

Difficulty: We can only measure the velocity of galaxies not the mass of their cluster. Physicists estimate dynamical cluster mass from single velocity dispersion.

Our method: Estimate the cluster mass from the whole distribution of velocities rather than just a simple velocity distribution.

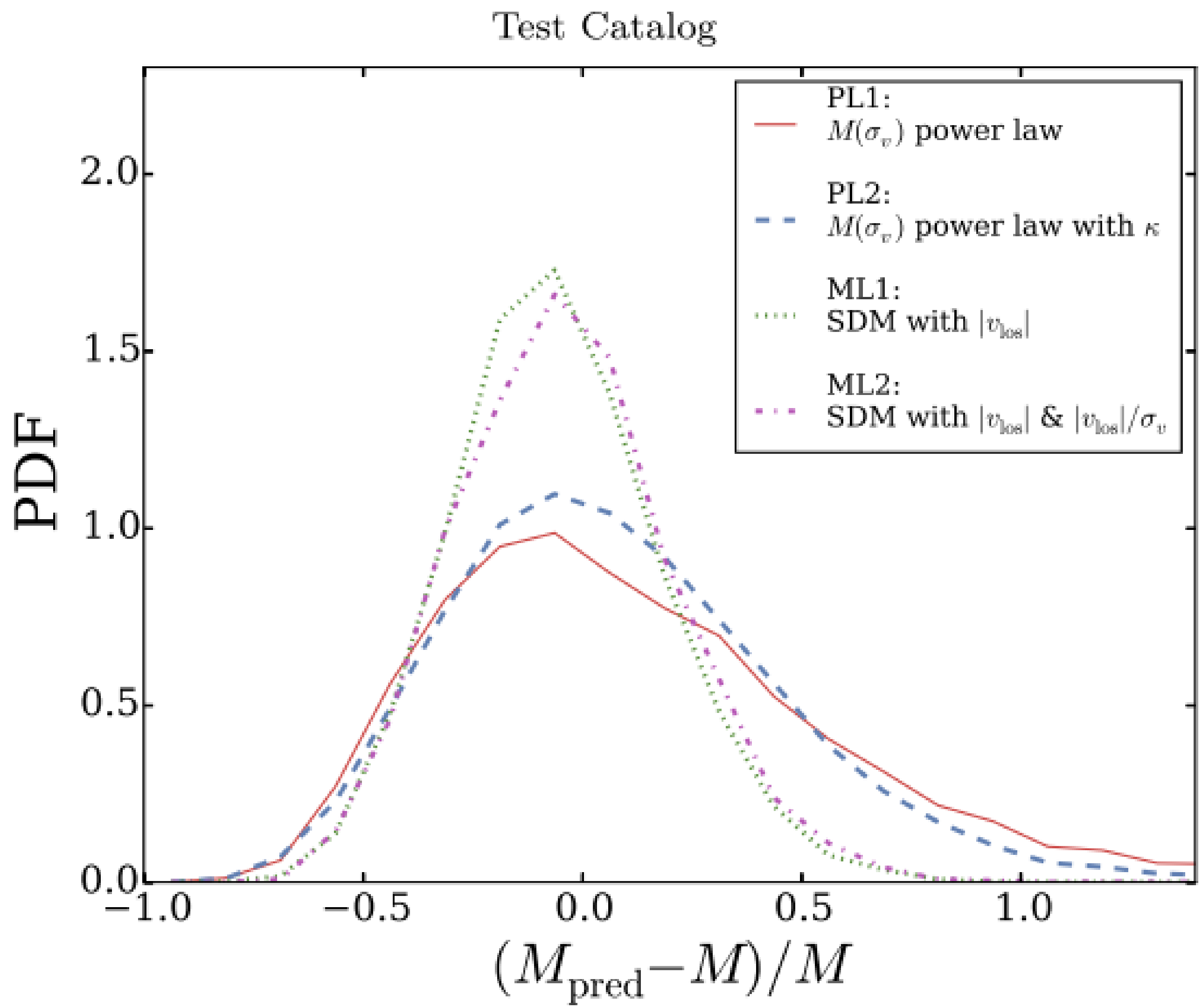
Support Distribution Machines (SDM) Regressor

From a distribution, predict a scalar.

galaxy properties:
line of sight velocity,
plane of sky position

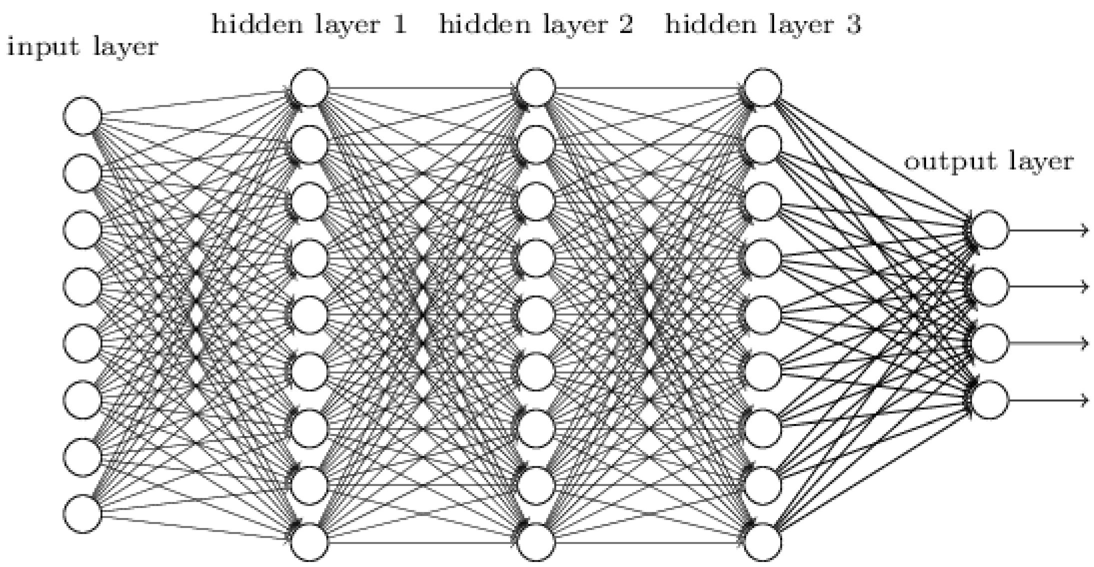
cluster
log(mass)

Estimate dynamical mass of galaxy clusters

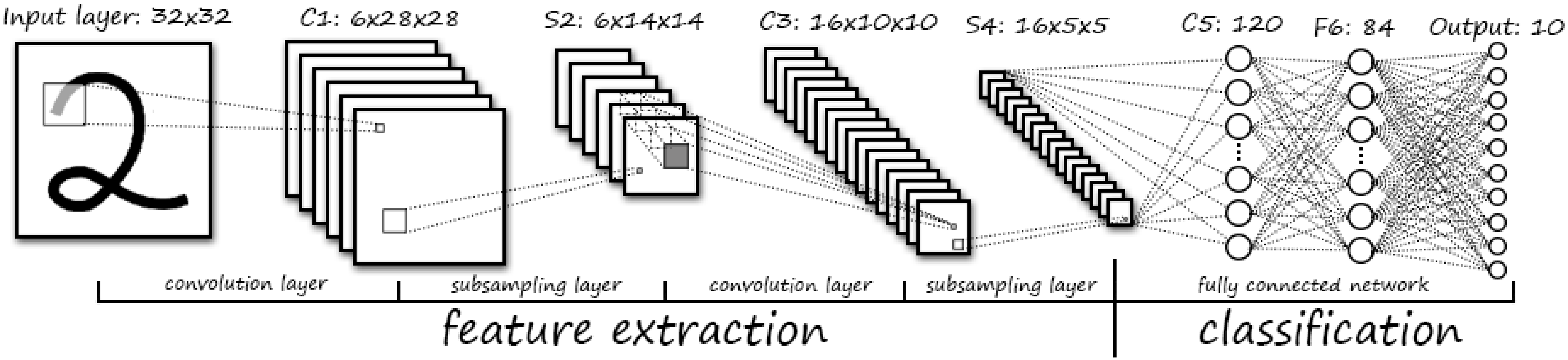


Michelle Ntampaka et al, A Machine Learning Approach for Dynamical Mass Measurements of Galaxy Clusters, APJ 2015

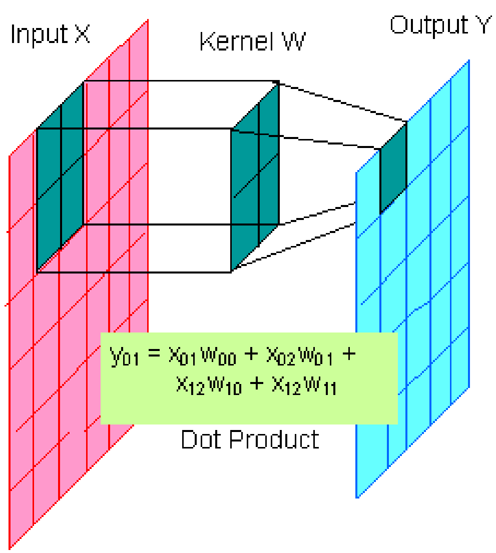
Neural Networks

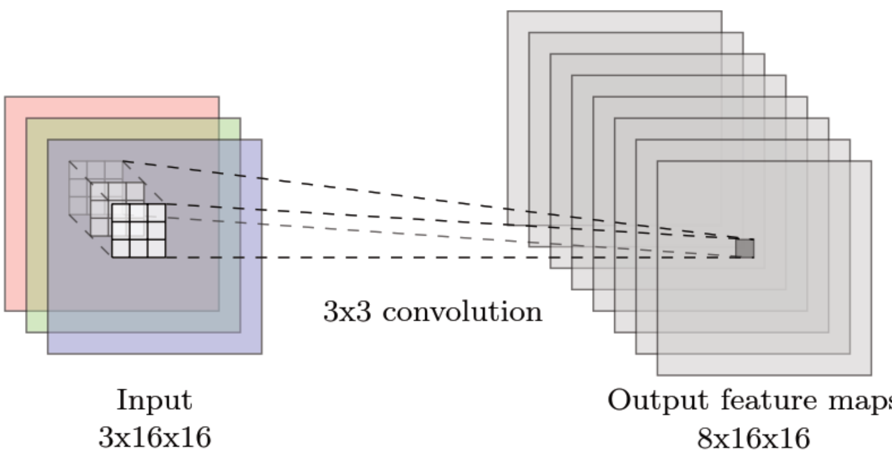


Convolutional Neural Networks

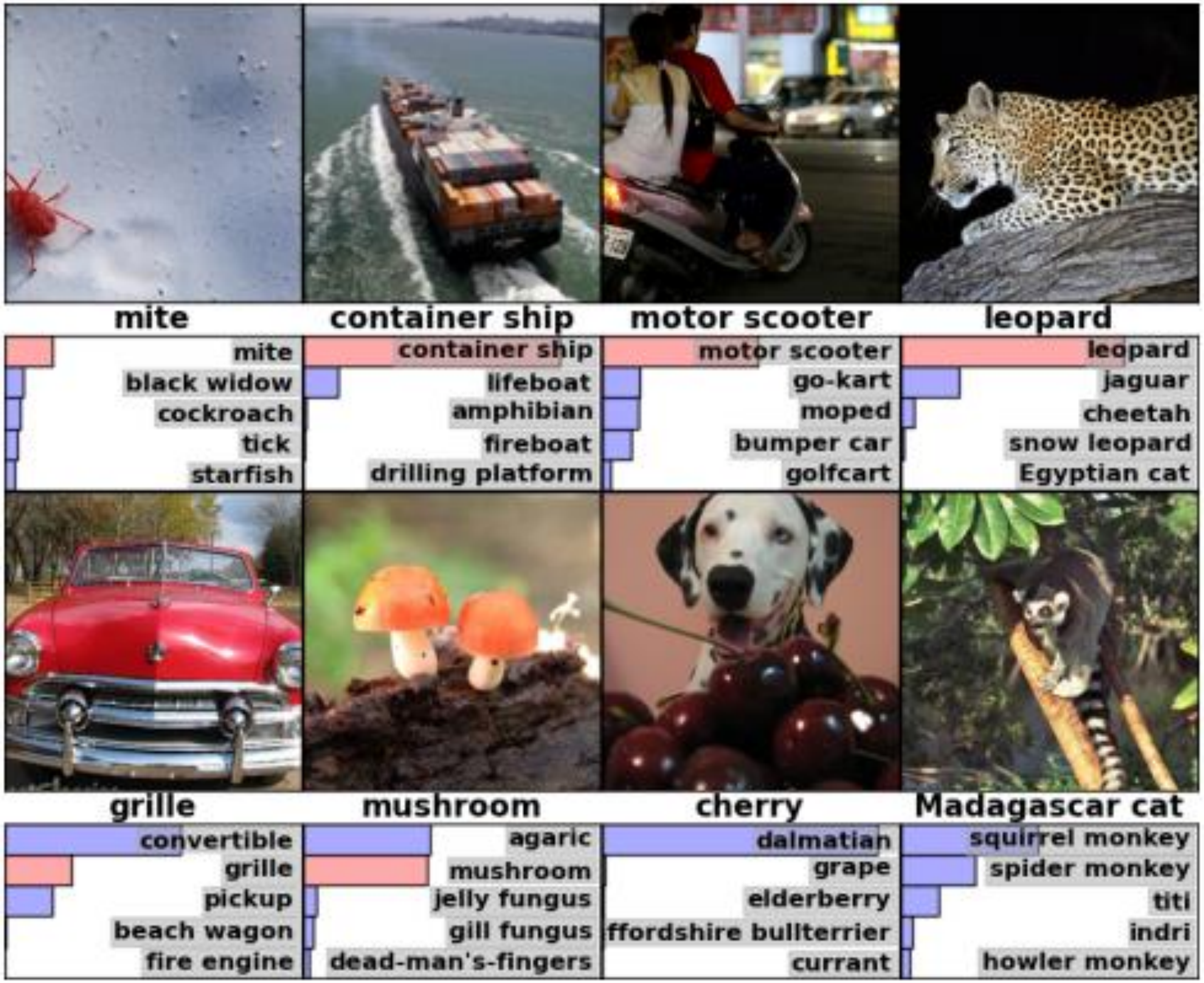


(LeNet)





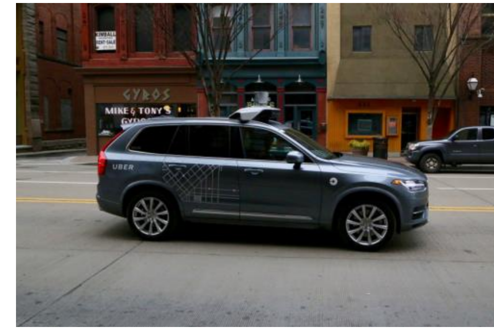
Imagenet Challenge



Carnegie Mellon

Self-driving Cars

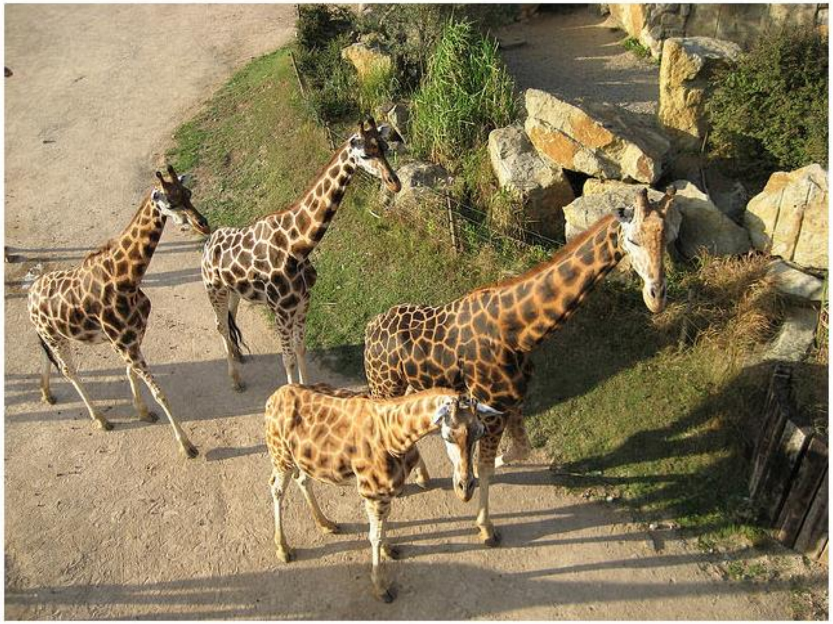
Credit: Kaiming He (https://youtu.be/WZmSMkK9VuA)



Caption Generation

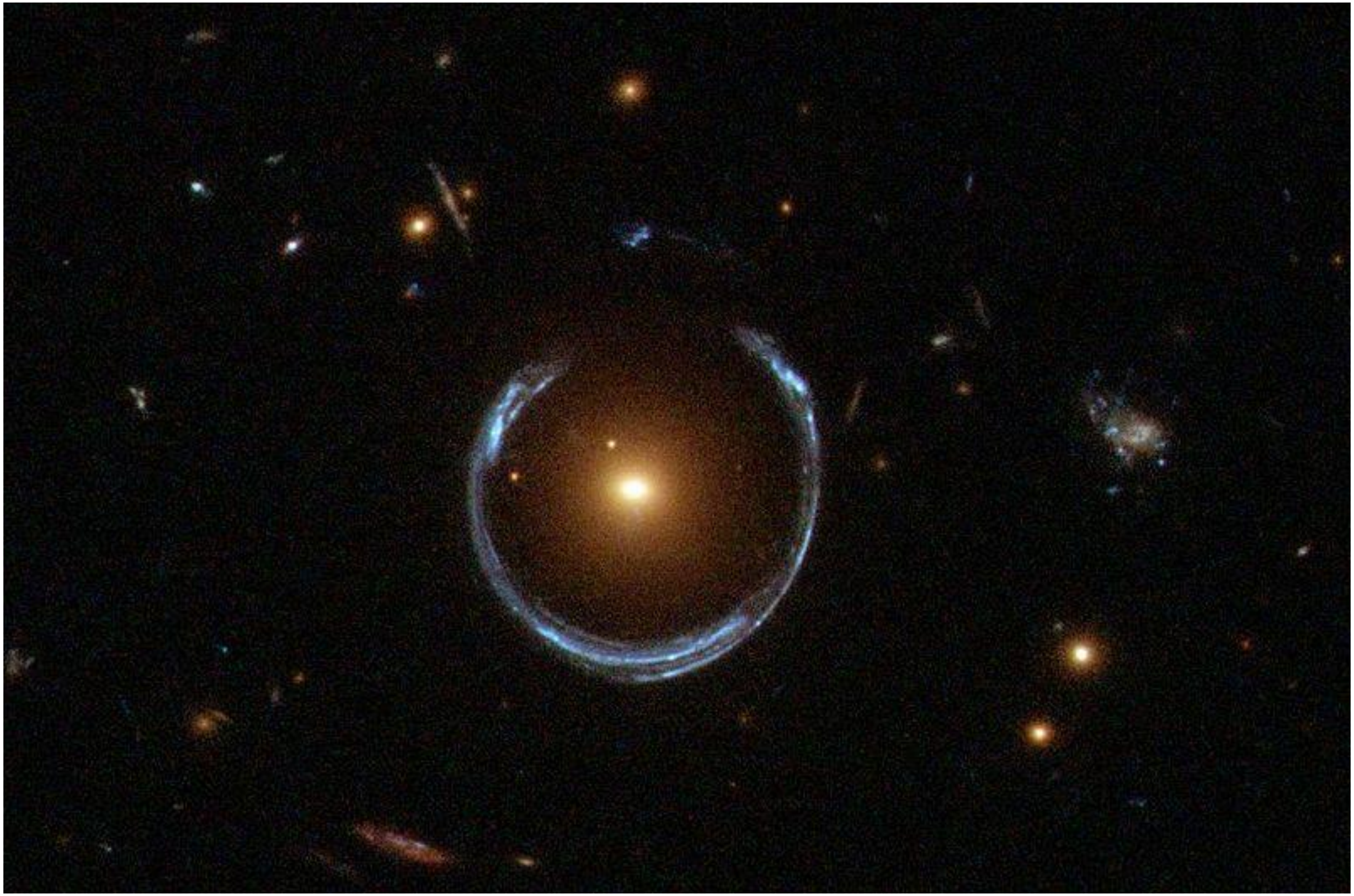
A person skiing down a snow covered slope.

A group of giraffe standing next to each other.



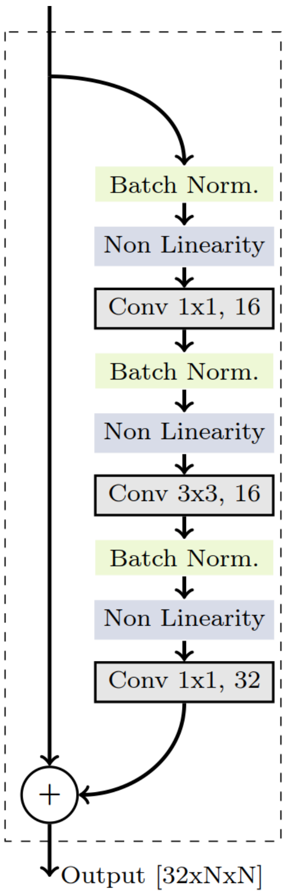
Carnegie Mellon

Weak Lensing Challenge

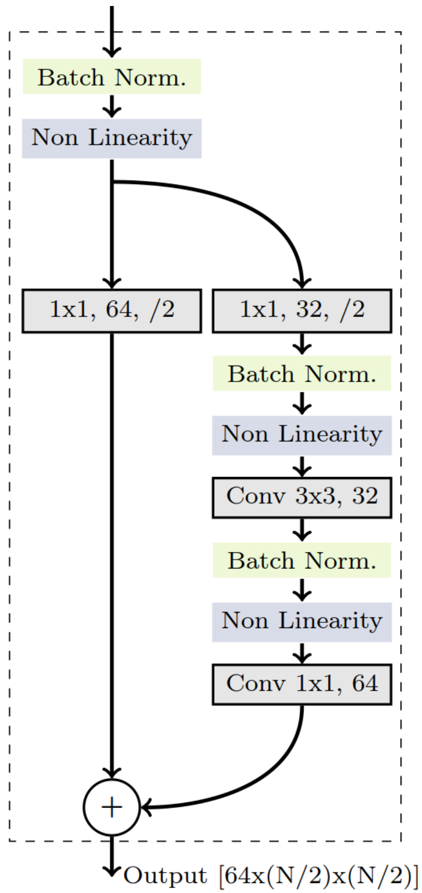


CMU DeepLens

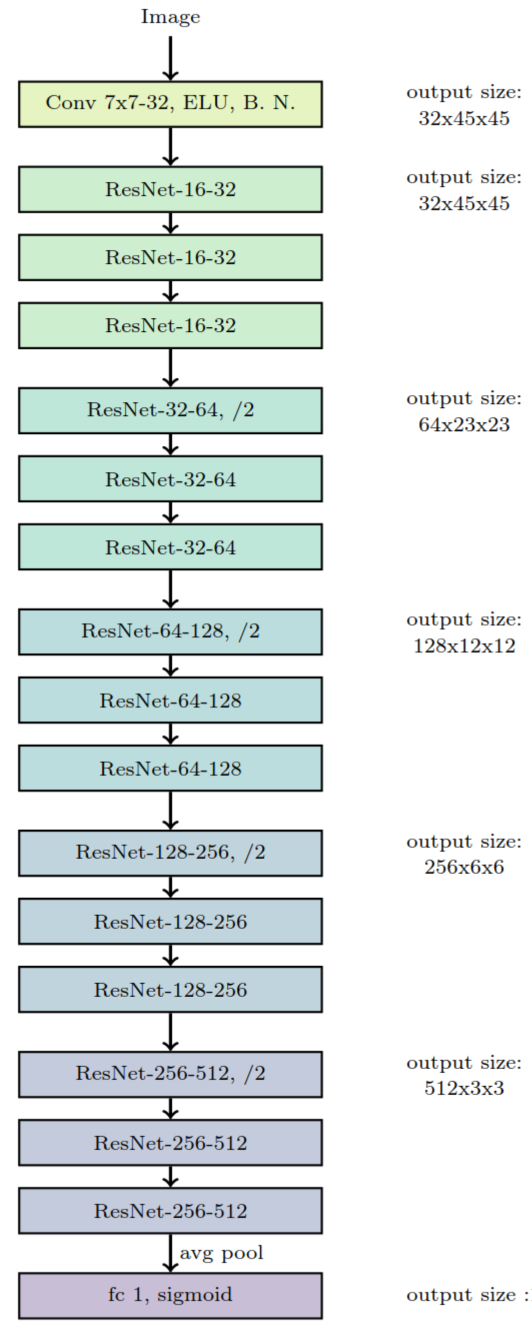
CMU DeepLens: Deep Learning For Automatic Image-based Galaxy-Galaxy Strong Lens Finding



(a) ResNet-16-32



(b) ResNet-32-64, /2



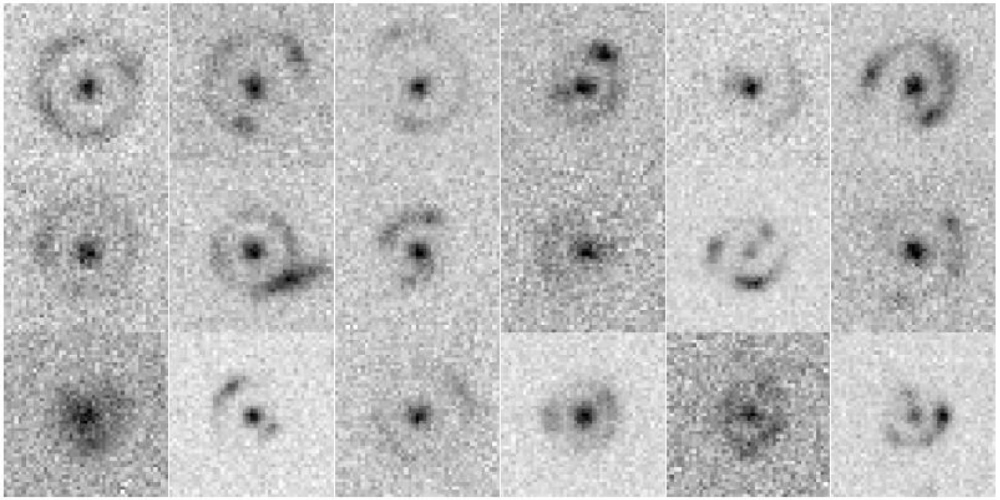
Left (a): ResNet-16-32 unit, preserving the size and depth of the input.
Right (b): ResNet-32-64,/2 unit simultaneously increasing the depth of the output (from 32 channels to 64) and downsampling by a factor 2 its resolution

Results

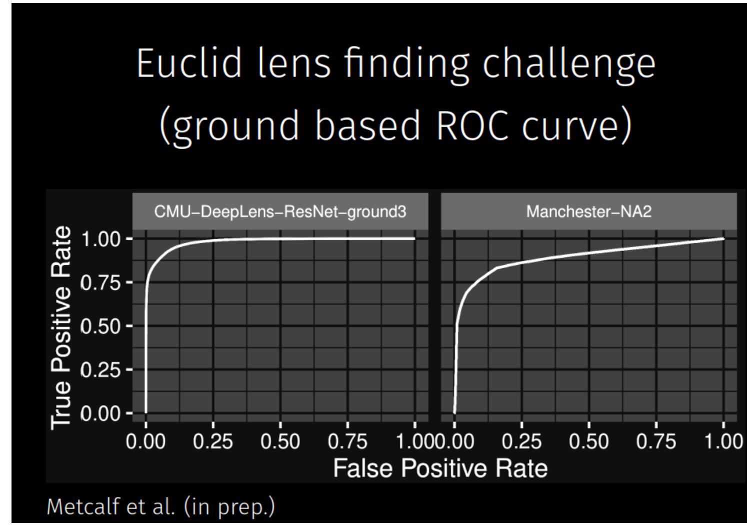
Name	type	AUROC	TPR_0	TPR_{10}	short description
CMU-DeepLens-ResNet-ground3	Ground-Based	0.98	0.09	0.45	CNN
CMU-DeepLens-Resnet-Voting	Ground-Based	0.98	0.02	0.10	CNN
LASTRO EPFL	Ground-Based	0.97	0.07	0.11	CNN
CAS Swinburne Melb	Ground-Based	0.96	0.02	0.08	CNN
AstrOmatic	Ground-Based	0.96	0.00	0.01	CNN
Manchester SVM	Ground-Based	0.93	0.22	0.35	SVM / Gabor
Manchester-NA2	Ground-Based	0.89	0.00	0.01	Human Inspection
ALL-star	Ground-Based	0.84	0.01	0.02	edges/gradiants and Logistic Reg.
CAST	Ground-Based	0.83	0.00	0.00	CNN / SVM
YattaLensLite	Ground-Based	0.82	0.00	0.00	SExtractor
LASTRO EPFL	Space-Based	0.93	0.00	0.08	CNN
CMU-DeepLens-ResNet	Space-Based	0.92	0.22	0.29	CNN
GAMOCLASS	Space-Based	0.92	0.07	0.36	CNN
CMU-DeepLens-Resnet-Voting	Space-Based	0.91	0.00	0.01	CNN
AstrOmatic	Space-Based	0.91	0.00	0.01	CNN
CMU-DeepLens-ResNet-aug	Space-Based	0.91	0.00	0.00	CNN
Kapteyn Resnet	Space-Based	0.82	0.00	0.00	CNN
CAST	Space-Based	0.81	0.07	0.12	CNN
Manchester1	Space-Based	0.81	0.01	0.17	Human Inspection
Manchester SVM	Space-Based	0.81	0.03	0.08	SVM / Gabor
NeuralNet2	Space-Based	0.76	0.00	0.00	CNN / wavelets
YattaLensLite	Space-Based	0.76	0.00	0.00	Arcs / SExtractor
All-now	Space-Based	0.73	0.05	0.07	edges/gradiants and Logistic Reg.
GAHEC IRAP	Space-Based	0.66	0.00	0.01	arc finder

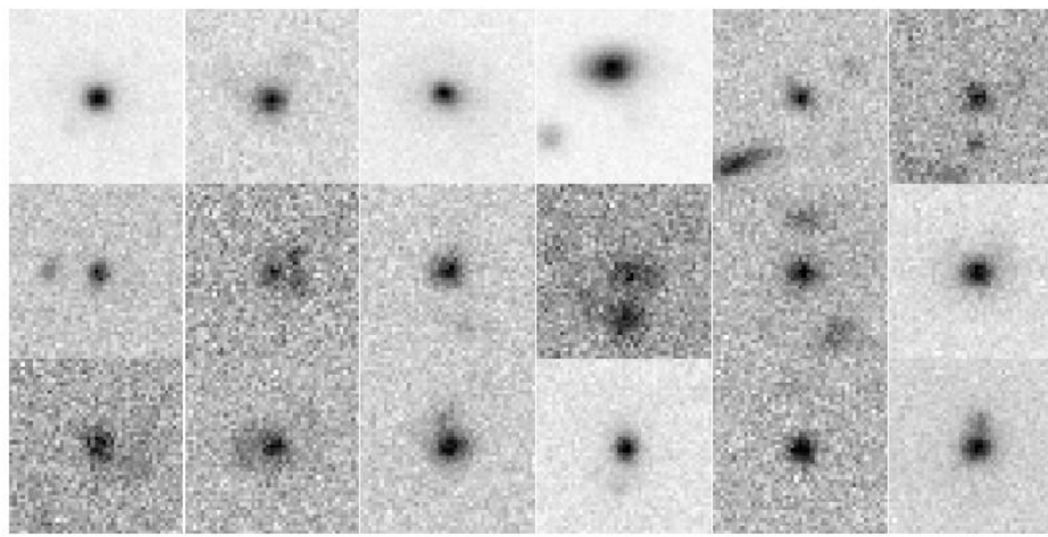
3. The AUROC, TPR_0 and TPR_{10} for the entries in order of AUROC.

Results



(a) True positives single images



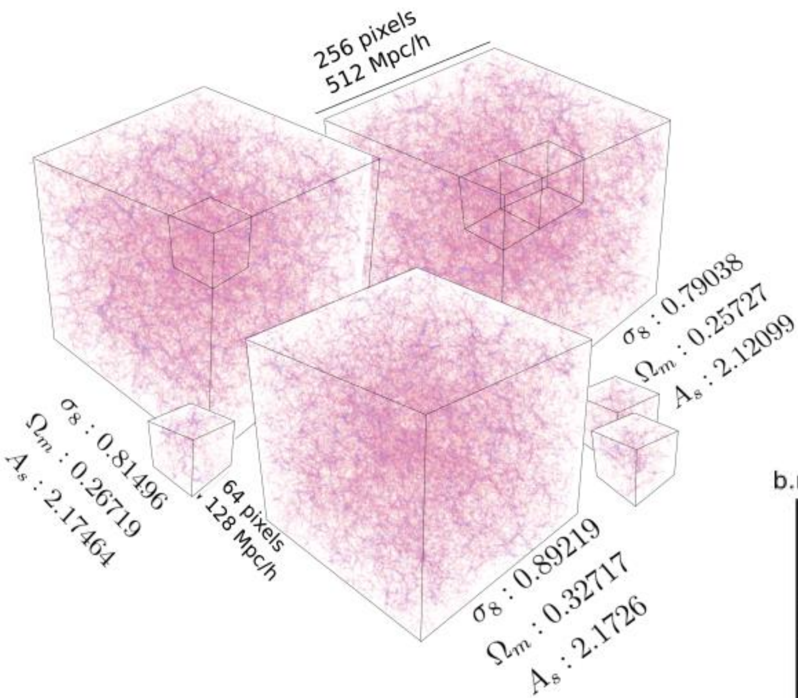


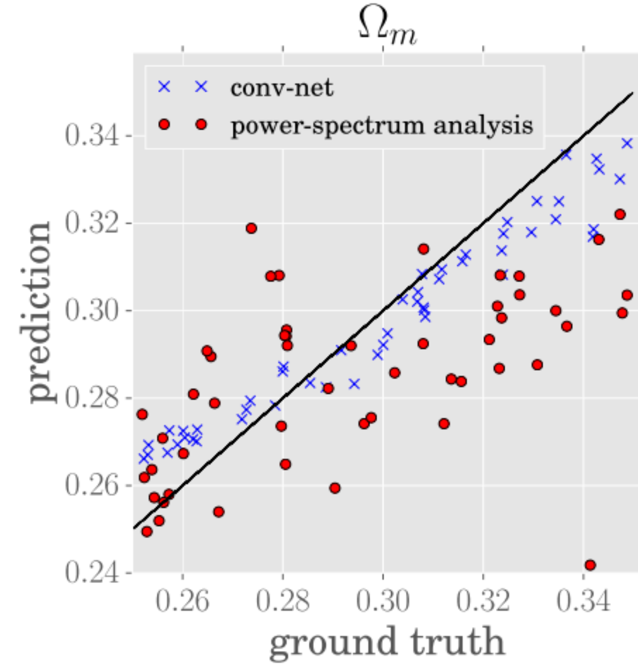
(b) False positives single images

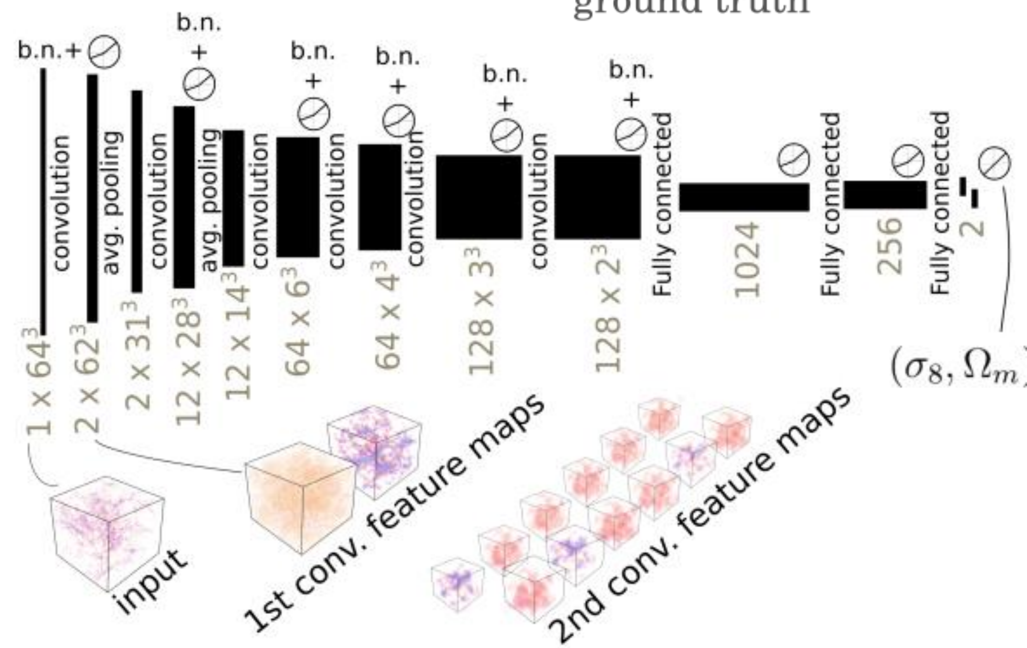
Carnegie Mellon

Find the parameters of Universe

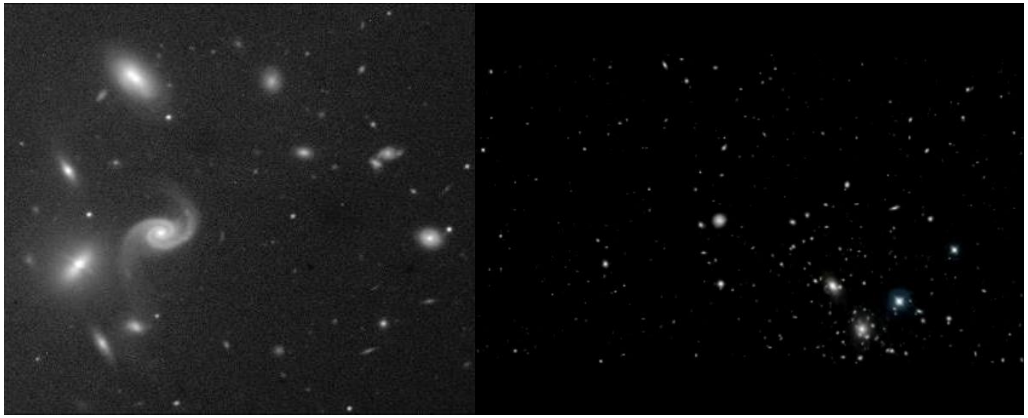
Given a distribution of particles, our goal is to predict the parameters of the simulated universe





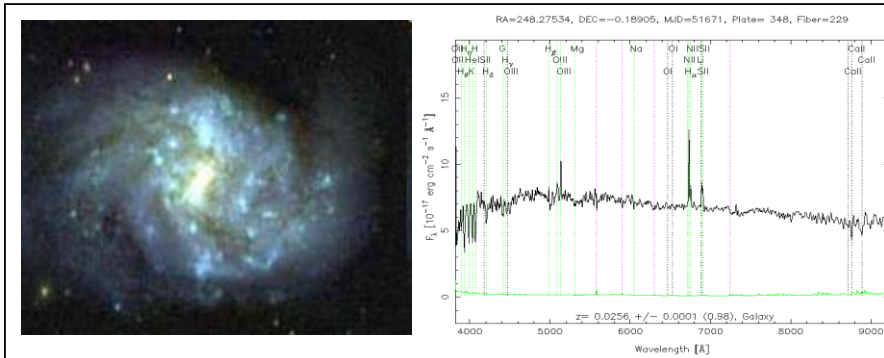


Find interesting Galaxy Clusters

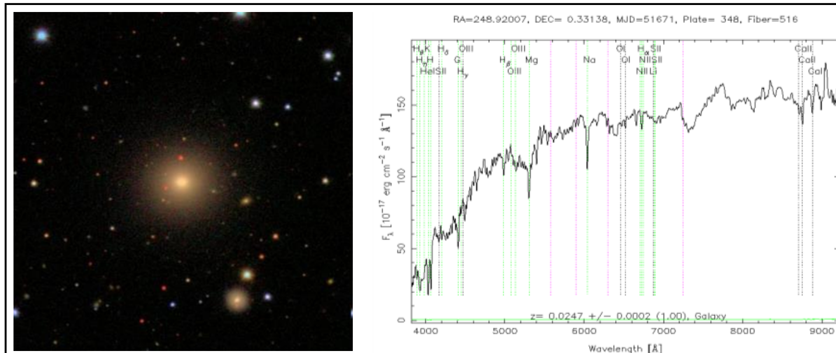


Sloan Digital Sky Survey (SDSS)

- ❑ continuum spectrum
- ❑ 505 galaxy clusters (10-50 galaxies in each)
- ❑ 7530 galaxies



Blue galaxy



Red galaxy

What are the most anomalous galaxy clusters?

The most anomalous galaxy cluster contains mostly

- ❑ star forming blue galaxies
- ❑ irregular galaxies

B. Póczos, L. Xiong & J. Schneider, UAI, 2011.

Credits: ESA, NASA

Carnegie Mellon

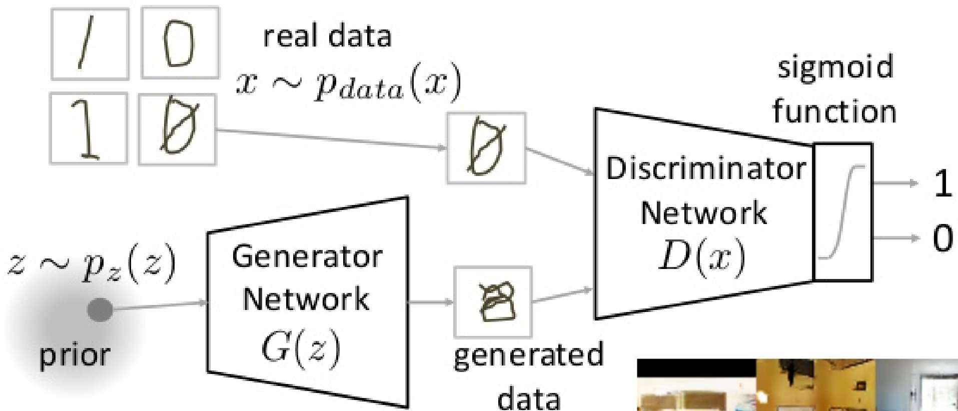
Generative Neural Networks

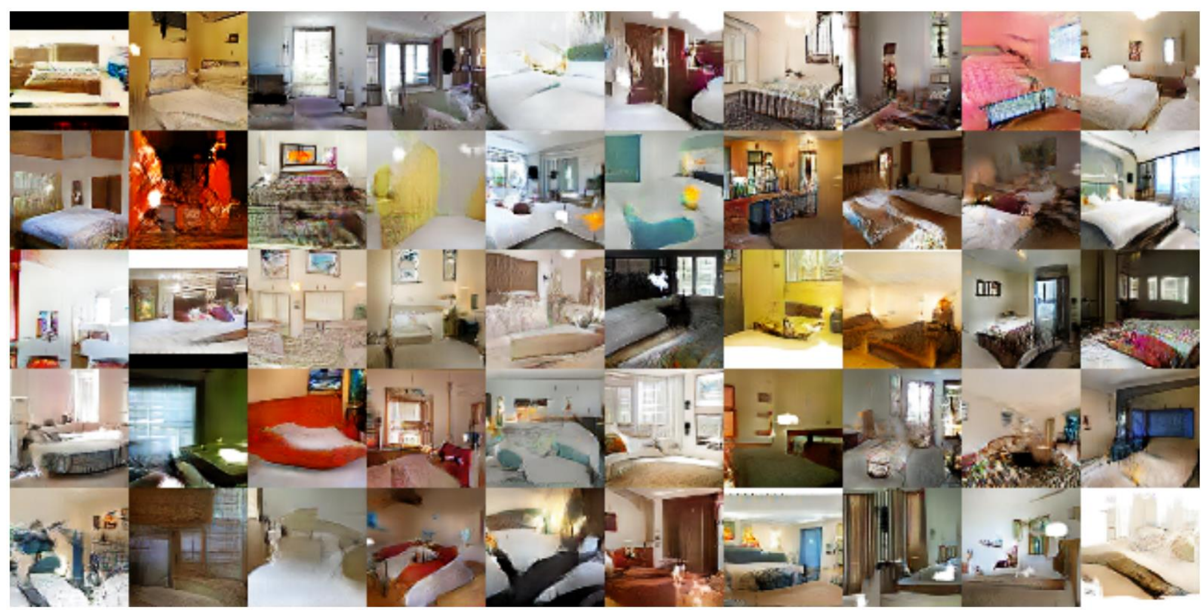
Generating Realistic Galaxy Images

Generative Adversarial Networks

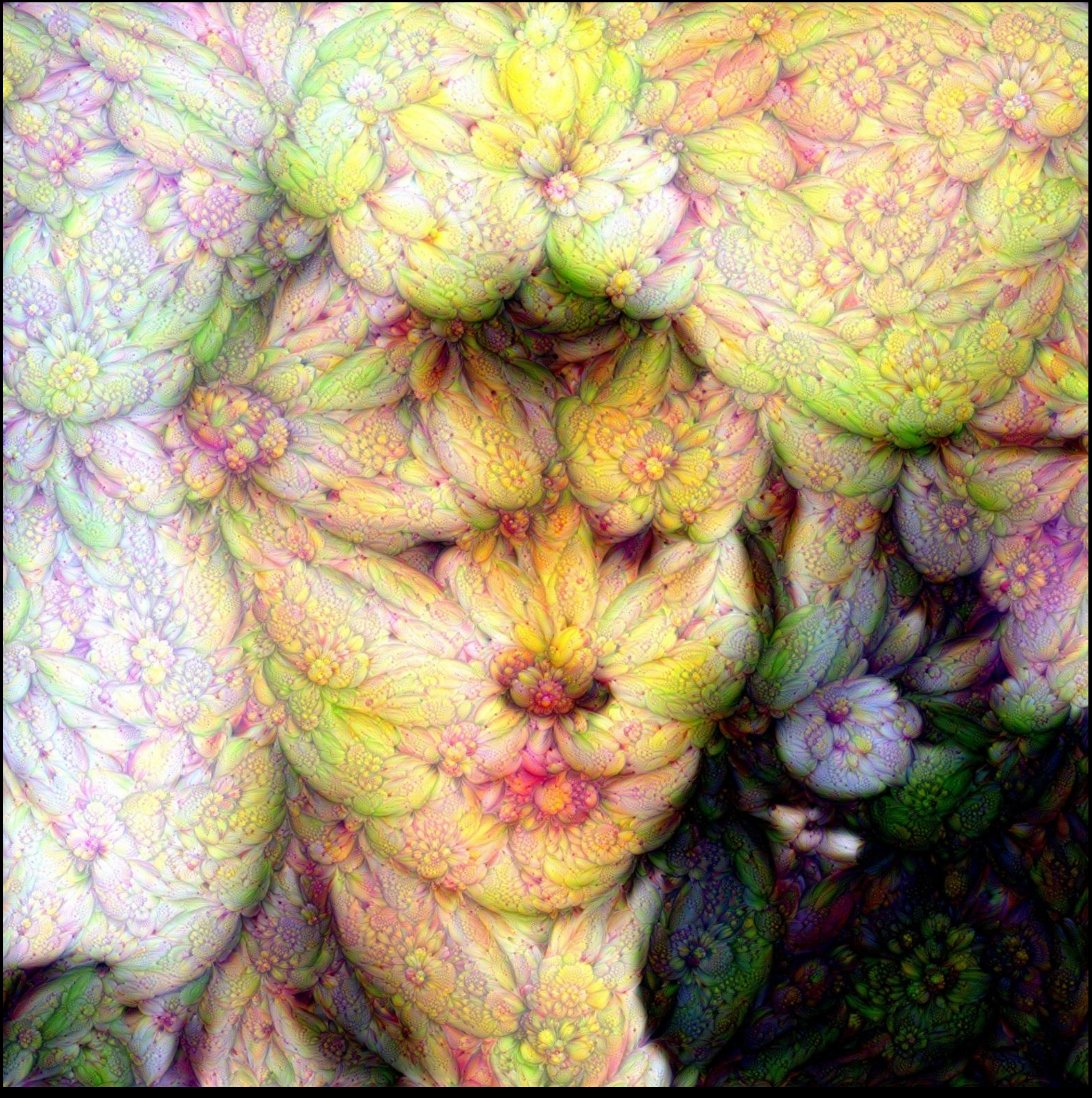
$$\min_G \max_D V(D,G)$$

$$V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$





Generative Neural Networks For Art



Generating Realistic Galaxy Images

nature International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video | For Au

Archive > Volume 542 > Issue 7639 > News > Article

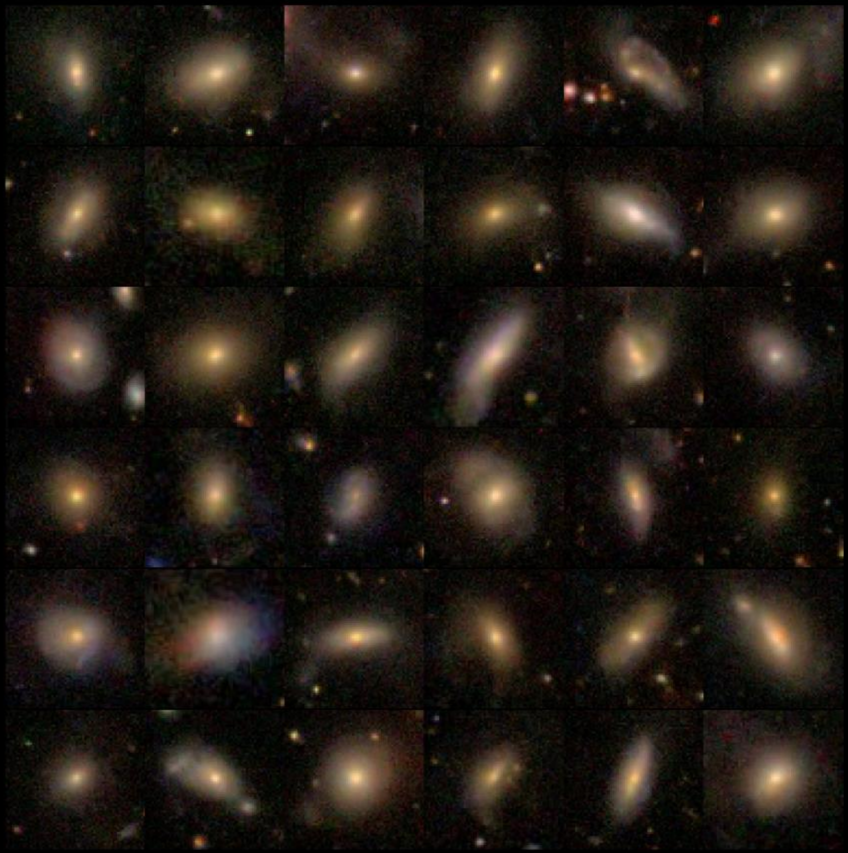
NATURE | NEWS

Astronomers explore uses for AI-generated images

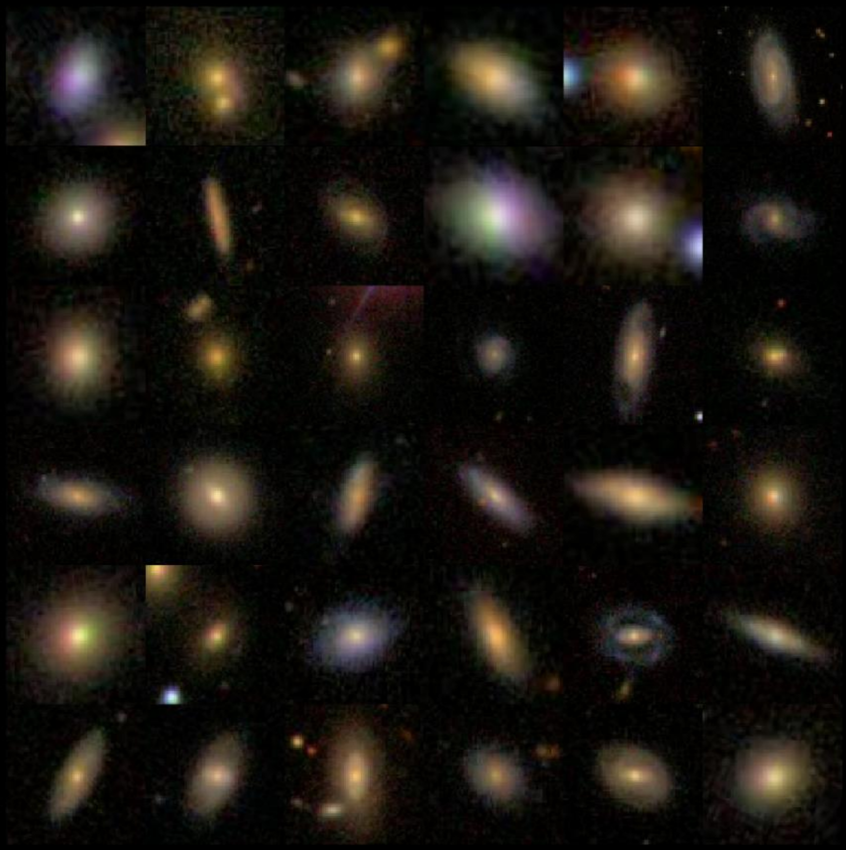
Neural networks produce pictures to train image-recognition programs and scientific software.

S. Ravanbakhsh, AAAI 2017

visual Turing test

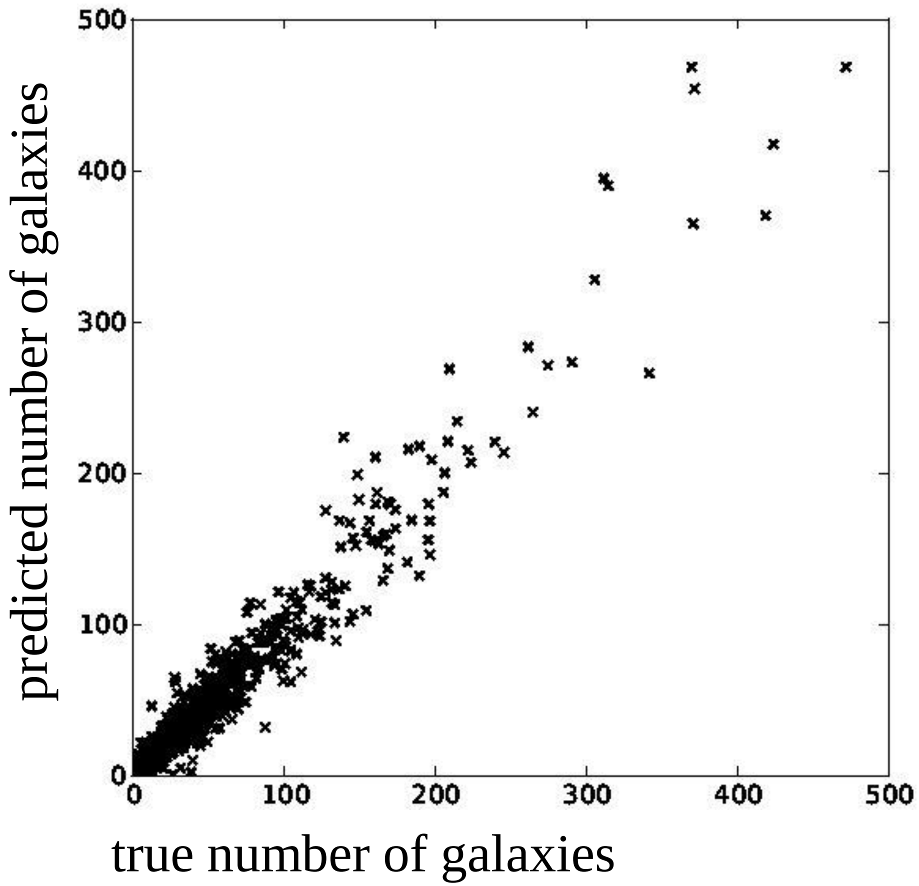


Mock - PixelCNN



Real - SDSS

Learning Relationships from Simulations



[Xiaoying Xu, 2012]

Goal: predict the number of galaxies in a halo from a half dozen dark matter halo parameters

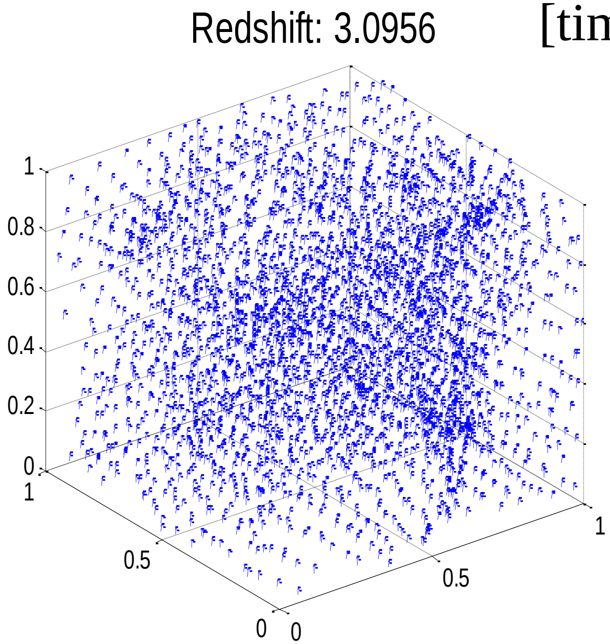
(#particles in a halo, velocity dispersion, max circular velocity, half mass radius,…)

data: Millenium simulation 395,832 halos

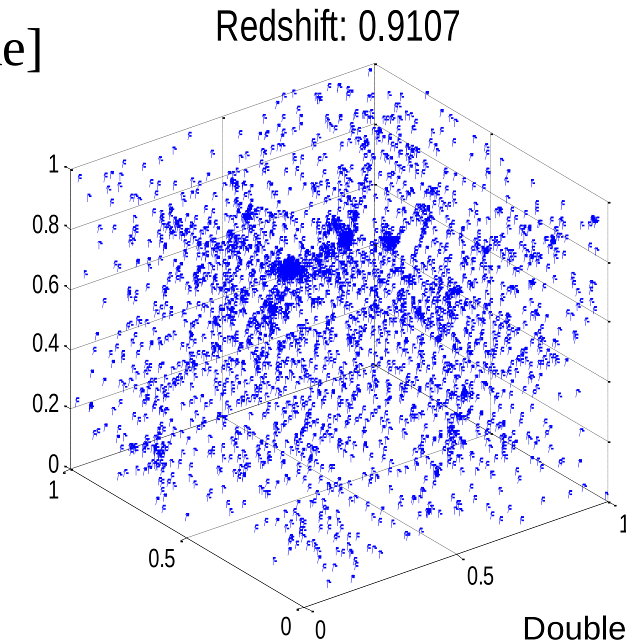
method: support vector regression

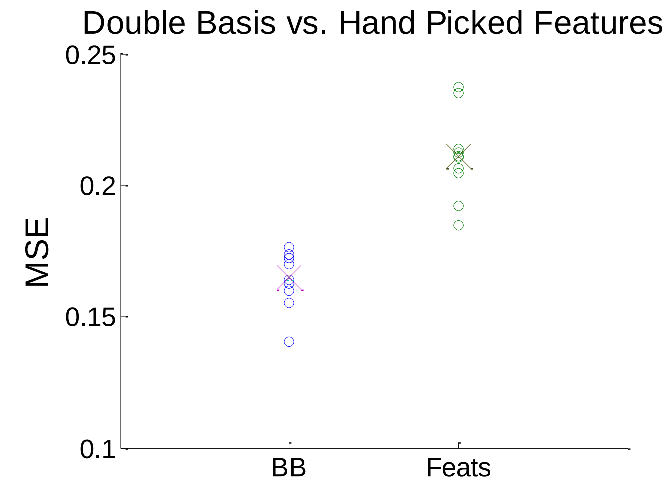
Learning Relationships from Simulations

Given a distribution of particles, our goal is to predict the redshift value that the particles were observed in.

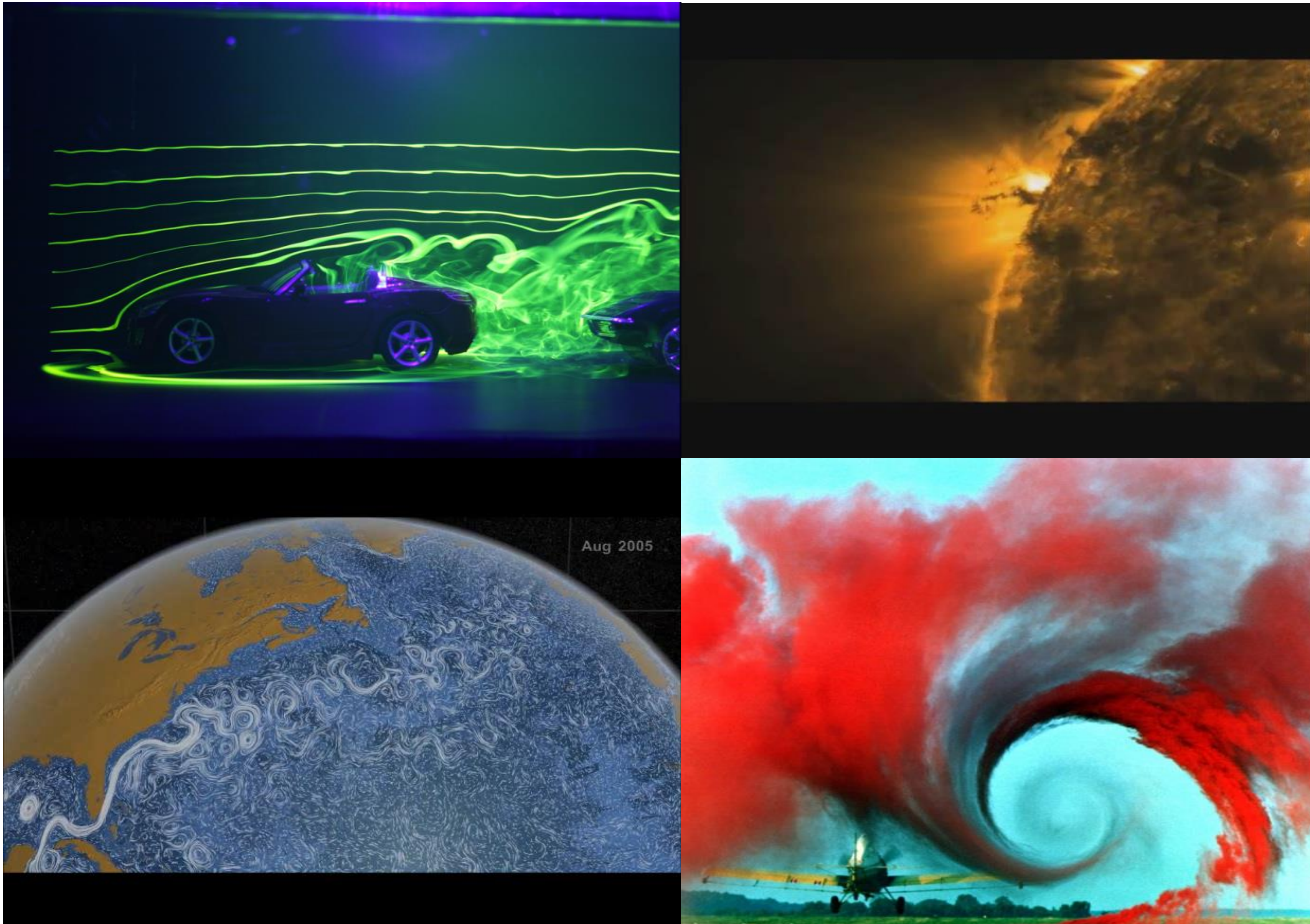


[time]





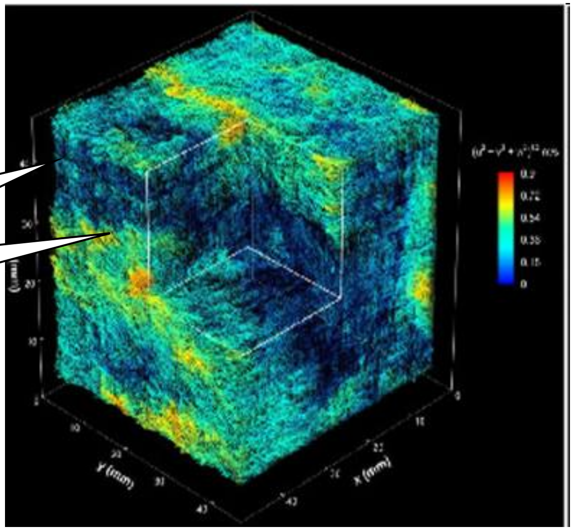
ML to Help Understanding Turbulences



Turbulence Data Classification

Simulated fluid flow through time

(JHU Turbulence Research Group, Alex Szalay)



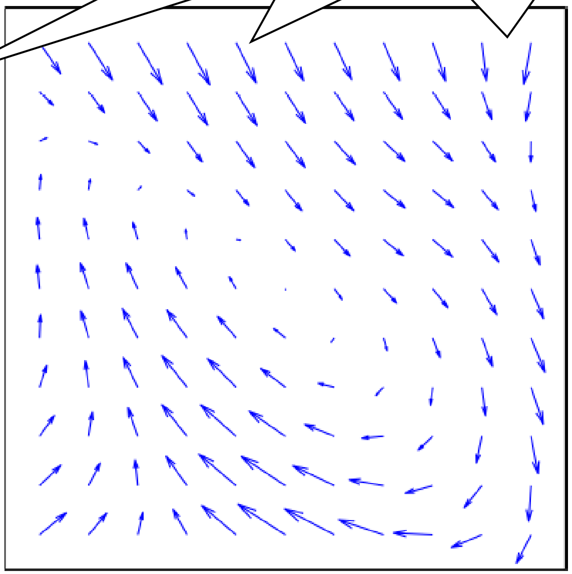
Goal: *find interesting events, p… …terns … …ena*

find vortices

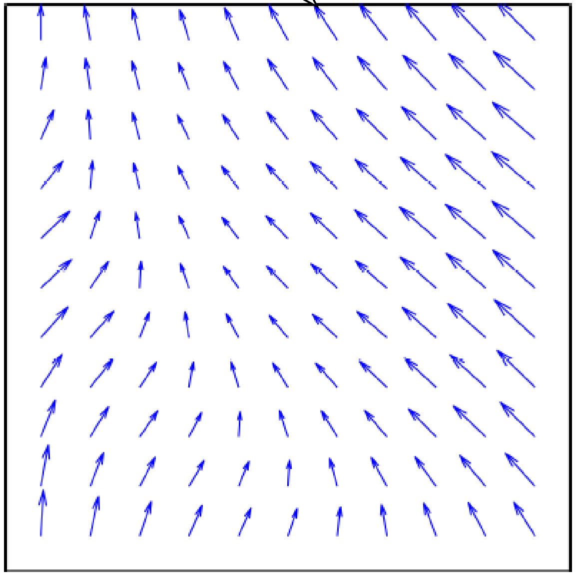
Something interesting happened?

Results: Leave one out cross-validation : 97%

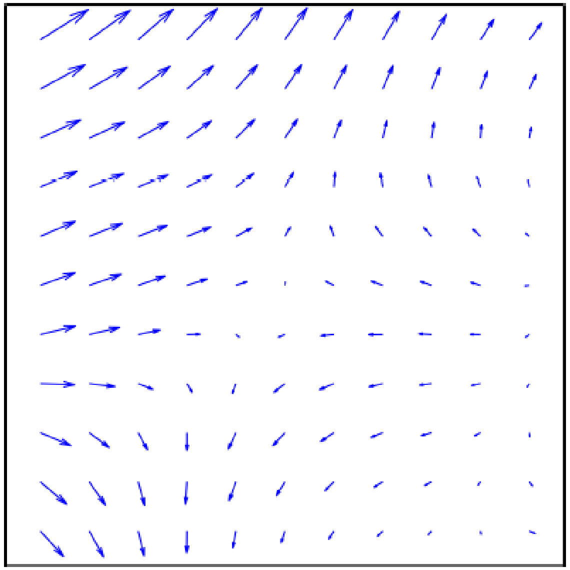
…ions



Positive (vortex)



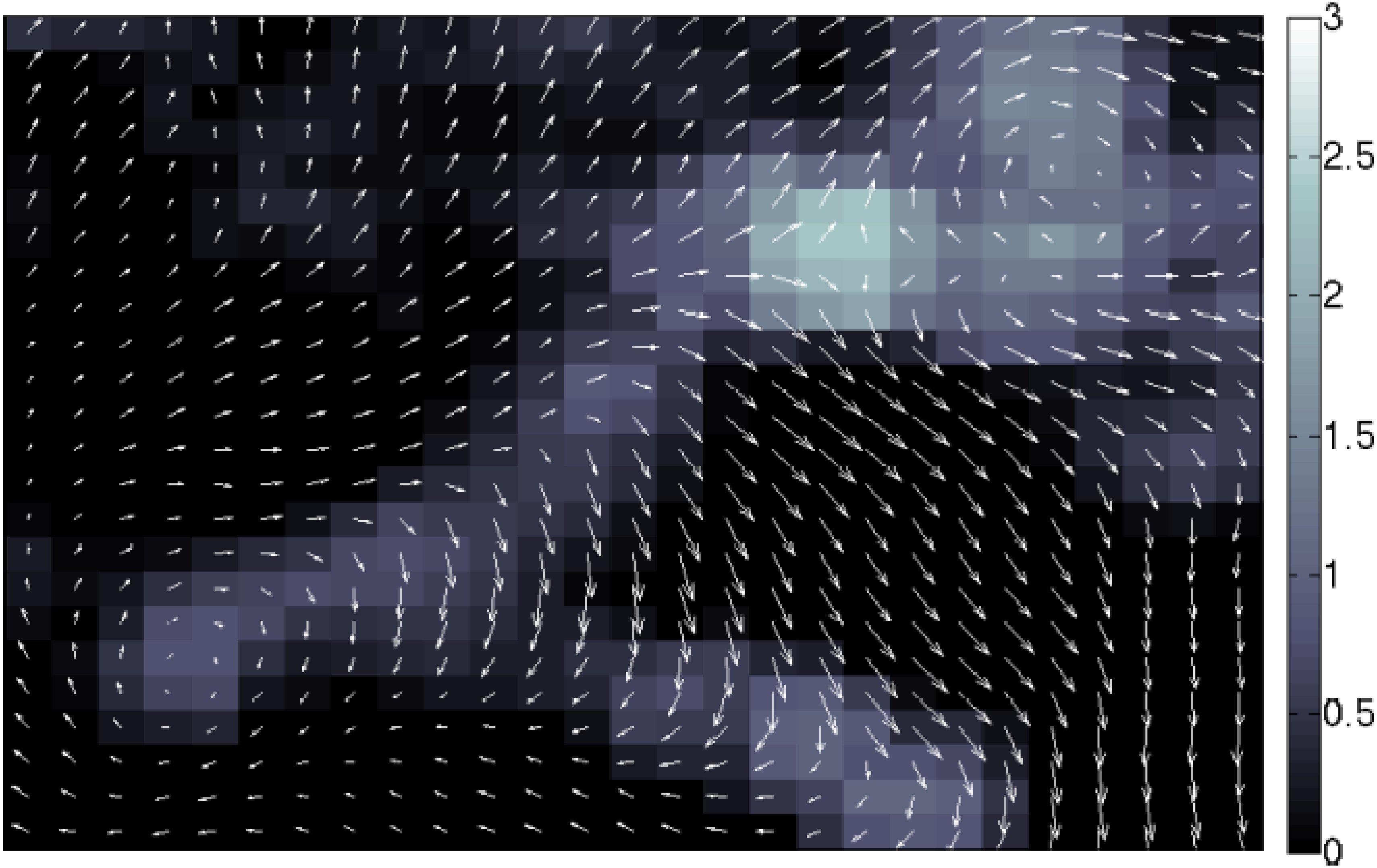
Negative



Negative

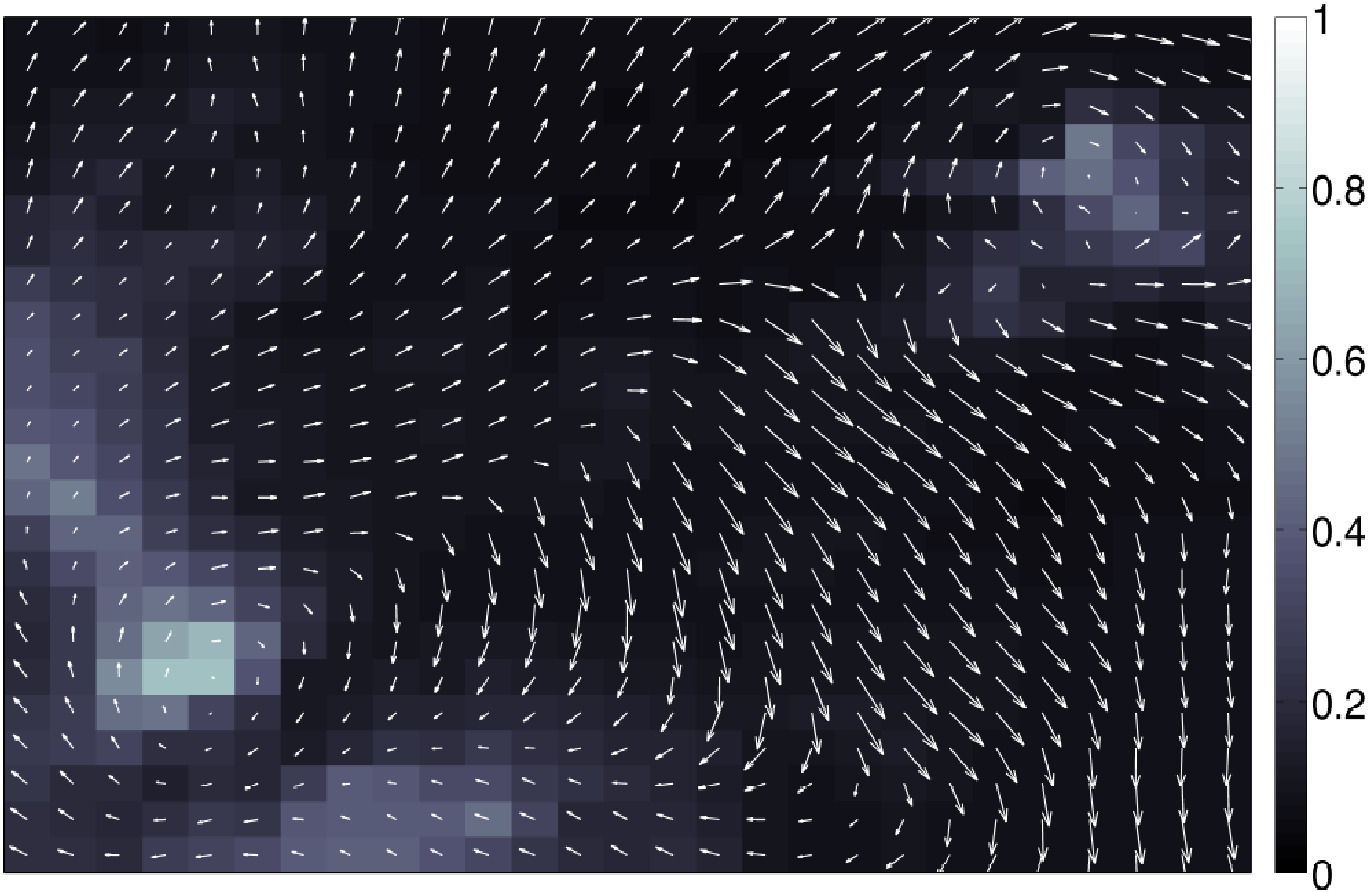
Find Interesting Phenomena in Turbulence Data

Anomaly detection



Anomaly scores

Finding Vortices

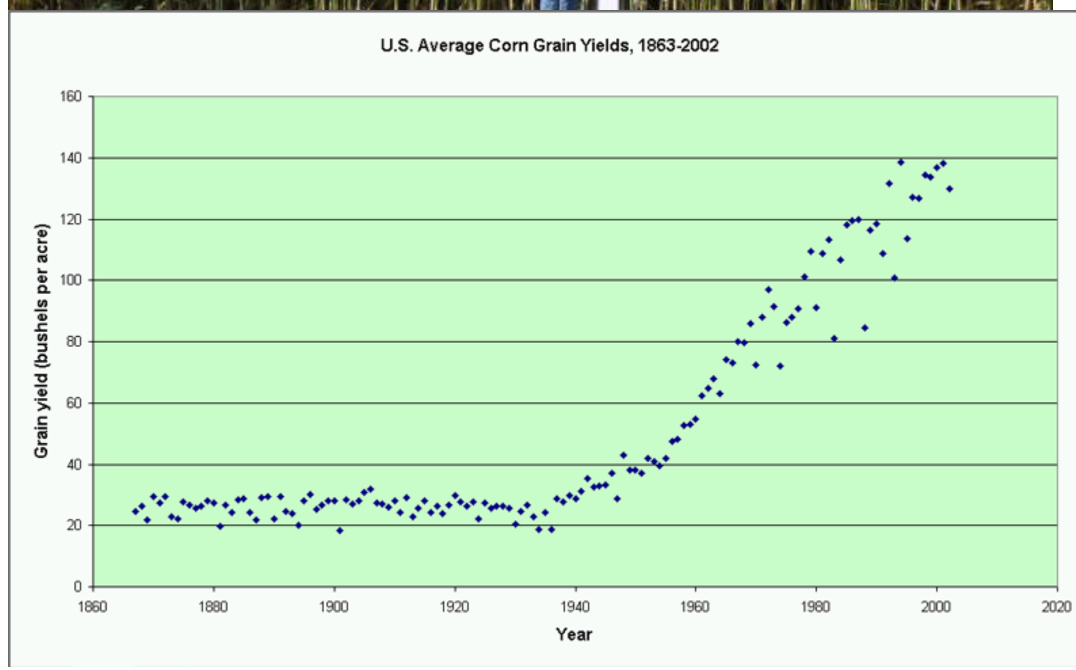


Classification probabilities

Agriculture

Agriculture

Recommend experiments (which plants to cross) to sorghum breeders.

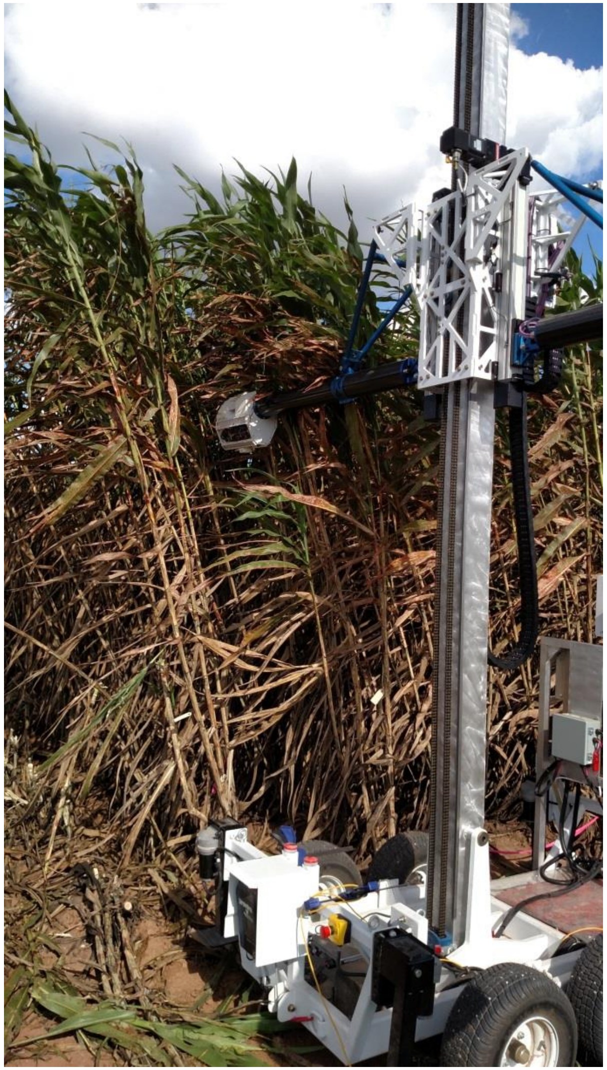


Surrogate robotic system in the field

Surrogate robotic system in the field

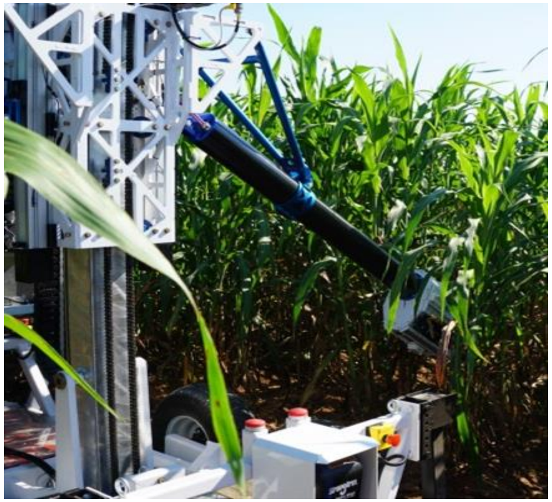
The surrogate system collecting data at the TAMU field site. The carriage supports two boom assemblies each one of which carries a sensor pod. The carriage slides up and down on the column allowing full scanning of a plant.

Surrogate robotic system in the field

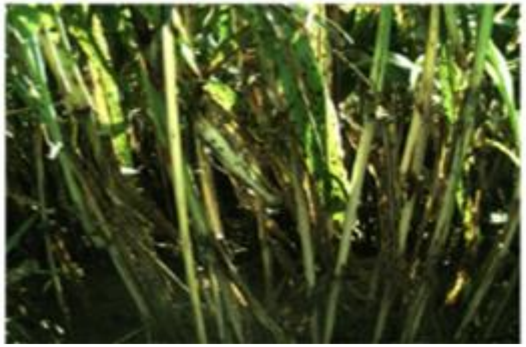


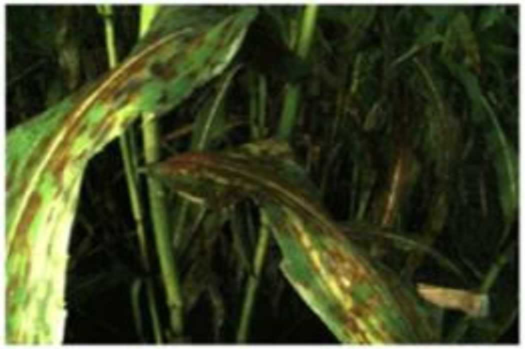
The carriage/dual-boom assembly moves up and down the column at a constant scanning speed. At its highest travel point the assembly clears the canopy (right).

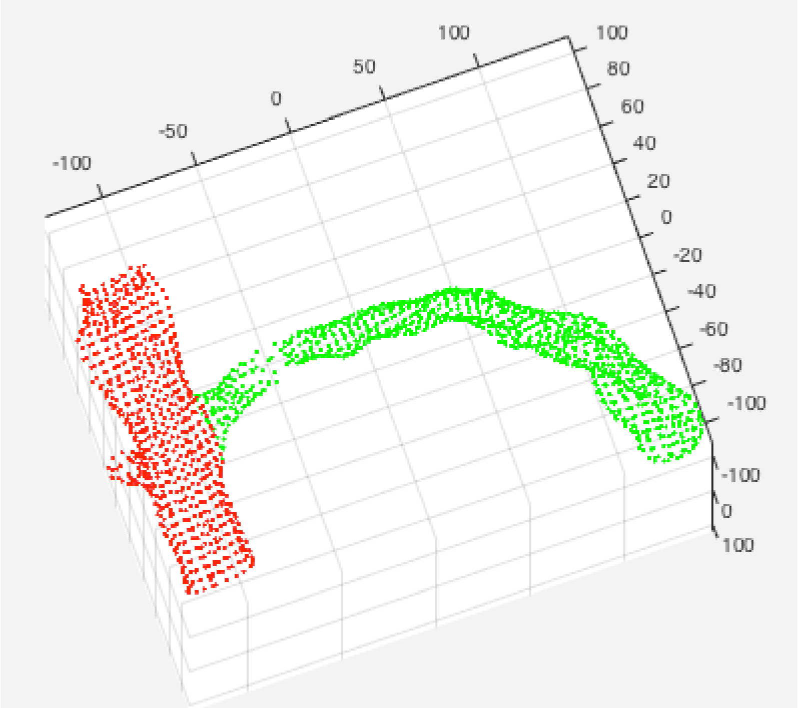
Data collection with sensor pods

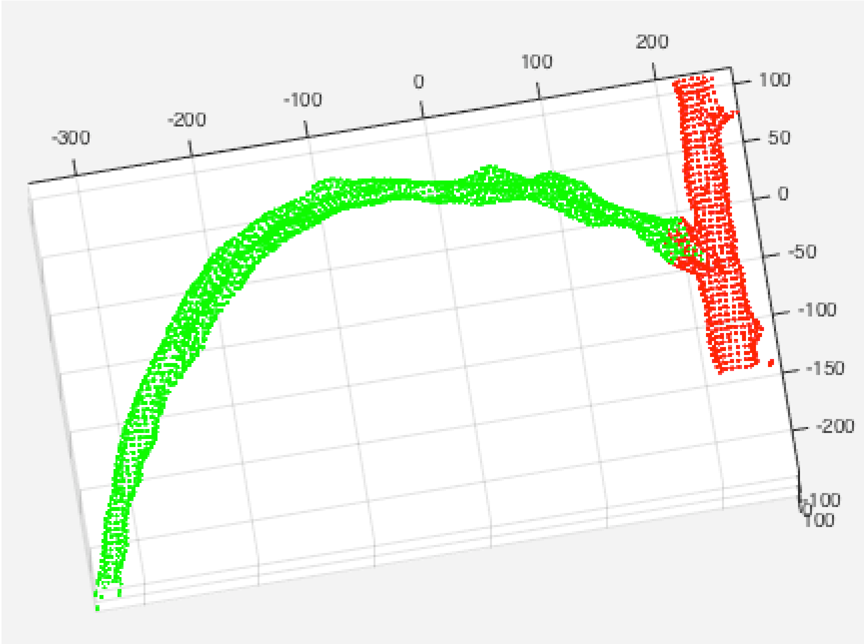


A sensor pod is deployed into a row and scans a plant







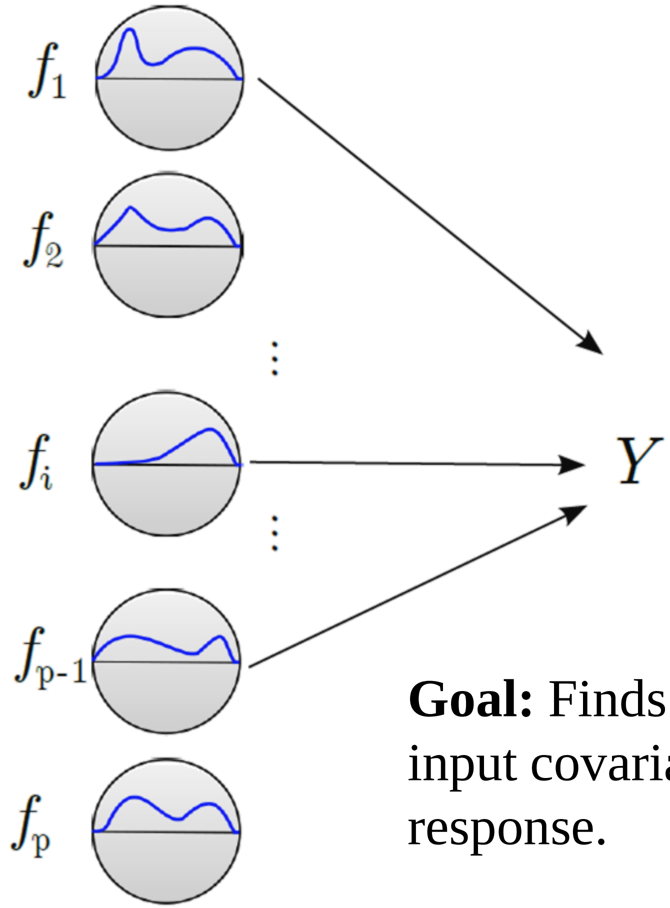


Name	**Range**	**RMSE error**
Leaf angle*	75.94	3.30 (4.35%)
Leaf radiation angle*	120.66	4.34 (3.60%)
Leaf length*	35.00	0.87 (2.49%)
Leaf width [max]	3.61	0.27 (7.48%)
Leaf width [average]	2.99	0.21 (7.02%)
Leaf area*	133.45	8.11 (6.08%)

FuSSO = Functional Shrinkage and Selection Operator
(Functional Lasso)

Sparse Functions-to-Real regression

When the number of functional input covariates may be very large, a sparse model that depends only on a few of the functional covariates may be preferred:

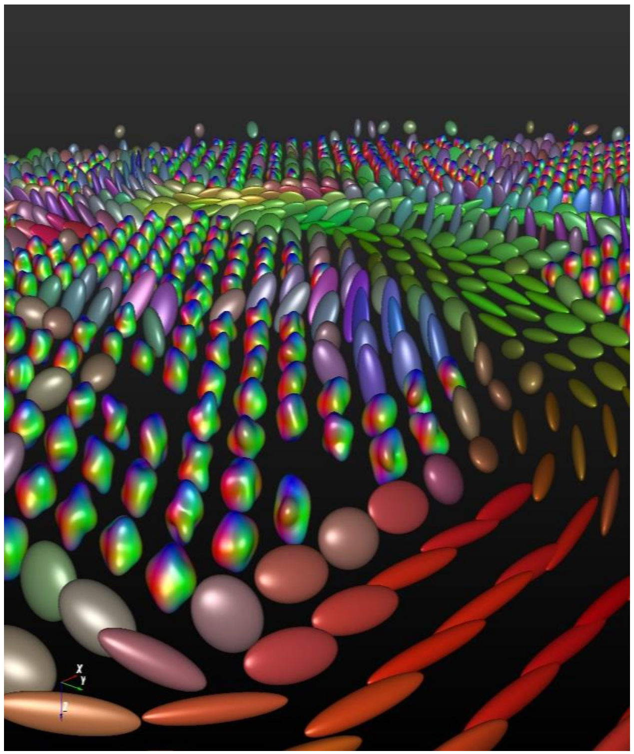


Goal: Finds a sparse set of functional input covariates to predict a real-valued response.

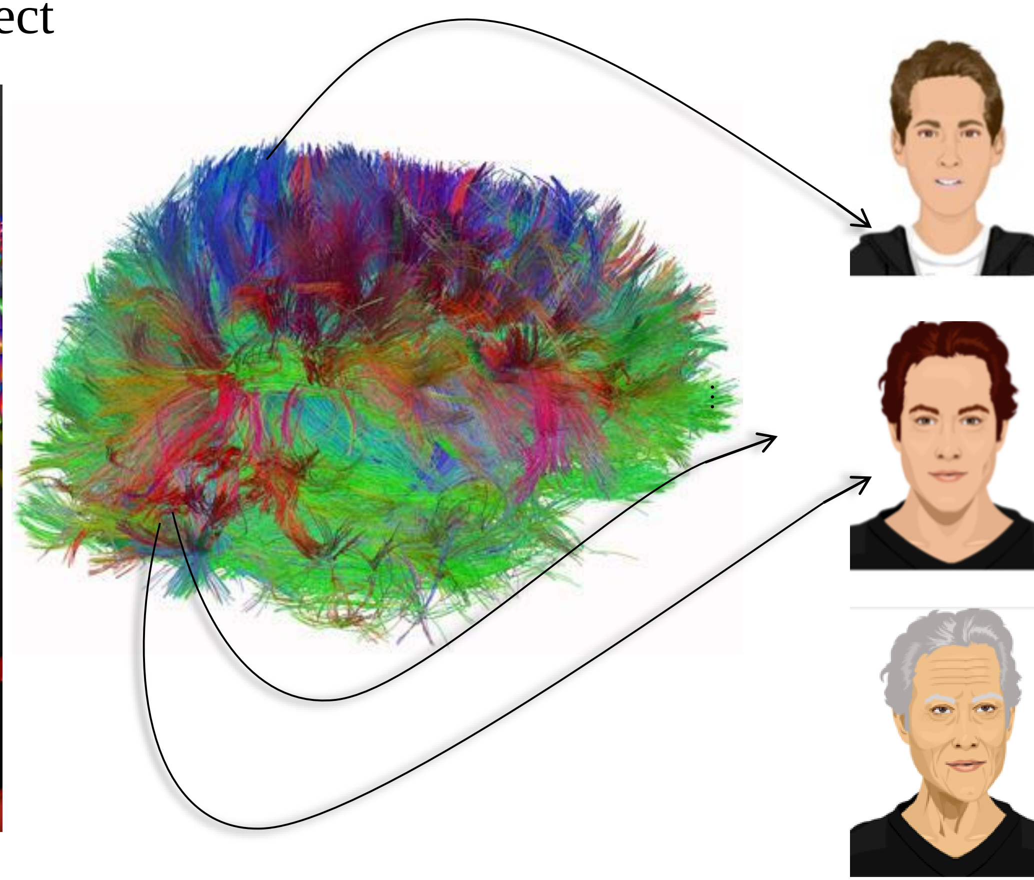
FuSSO Applications in Neuroimaging

Inputs: Functions at each voxel (e.g. orientation distribution functions)

Output: The age of the subject



Voxels' ODFs



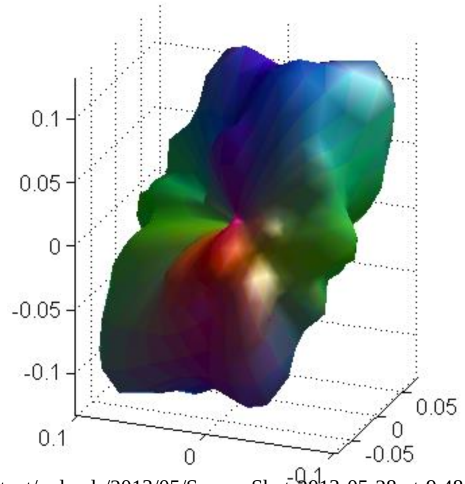
Age

Image credit: http://bmia.bmt.tue.nl/software/viste/

Results: Neuroimaging dataset

- Dataset with over 25K functions per subject for 89 total subjects (18 to 60 years old)
- Orientation distribution functions (ODF) at white matter voxels
- **Goal**: Predict the subject's age, given ODFs
- We compared to LASSO with peak ODF (quantitative anisotropy, QA) values. Finite dim non-functional data set.

Example Voxel ODF



Ages

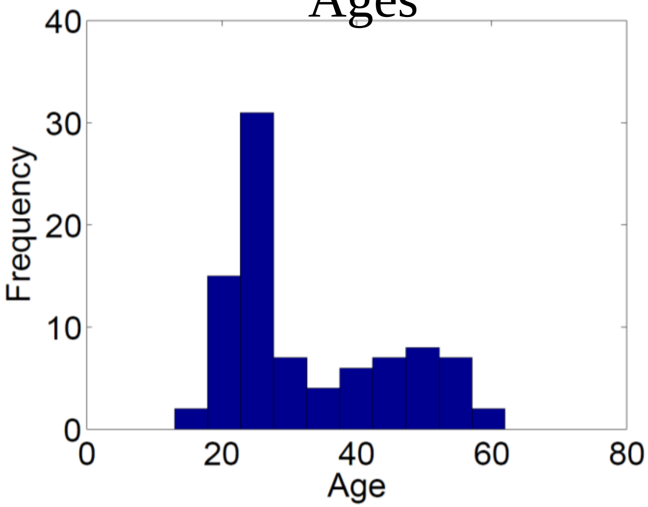


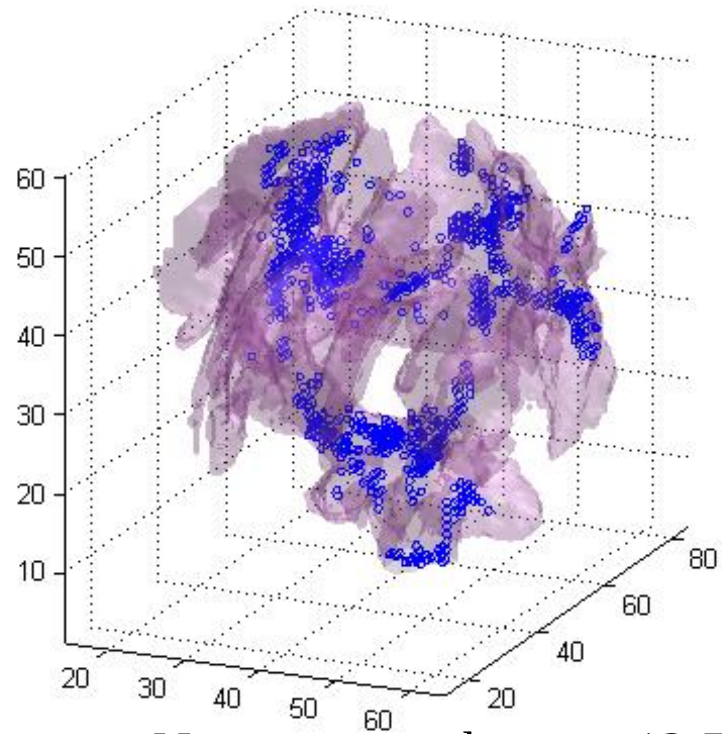
Image Sources: http://www.aging2.com/wp-content/uploads/2013/05/Screen-Shot-2013-05-28-at-9.48.49-PM.png; http://media.salon.com/2013/02/money1.jpg; http://3278as3udzze1hdk0f2th5nf18c1.wpengine.netdna-cdn.com/wp-content/uploads/2010/10/connectome-brain-diffusion-spectrum-imaging.jpg

Results: Neuroimaging dataset

Results:

Method:	FuSSO (ODFs)	LASSO (QAs)	Mean Predict
MSE:	70.85	77.13	156.43

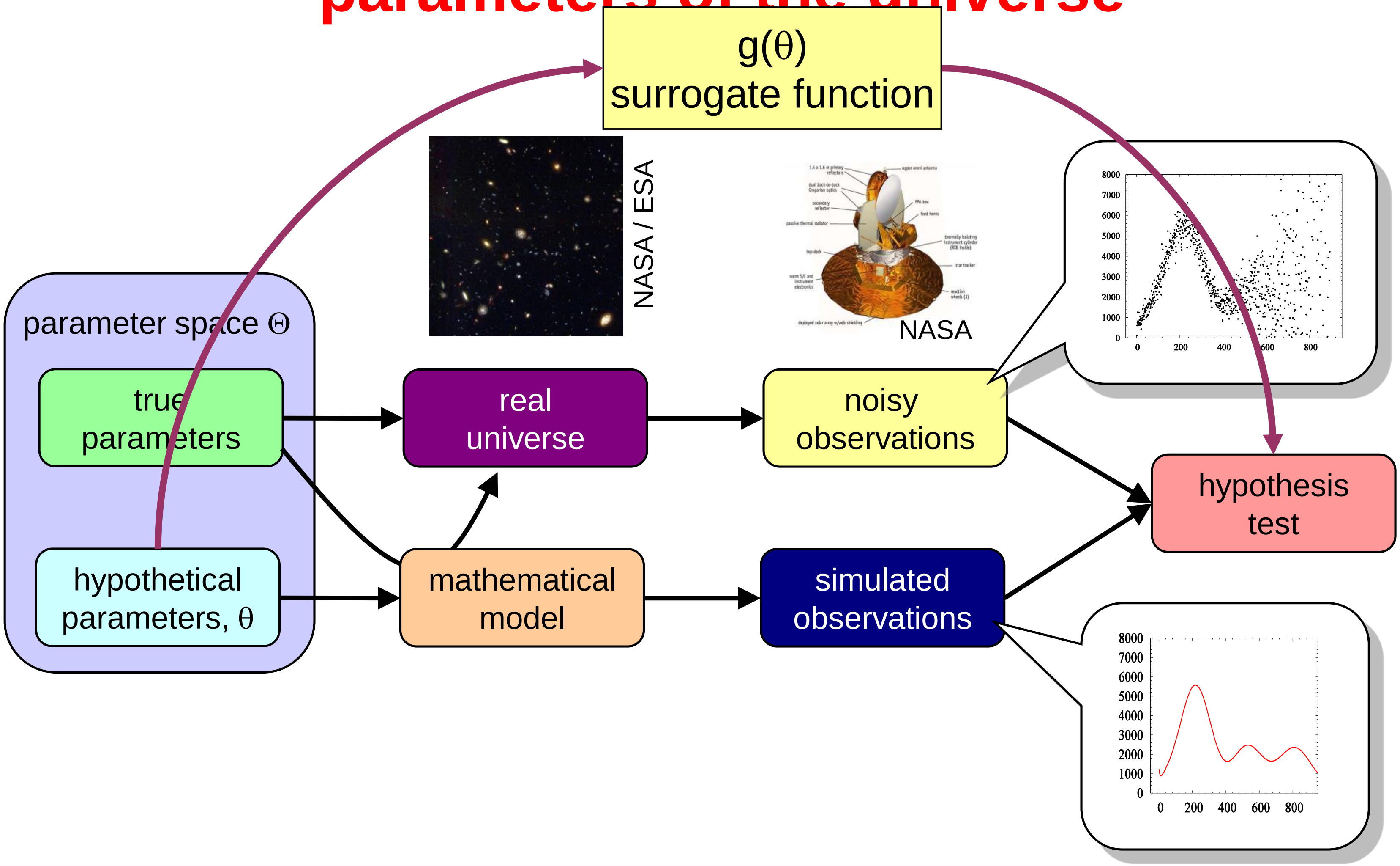
Selected Voxels



Mean error: 8.3 years, Naïve approach error: 12.5 years

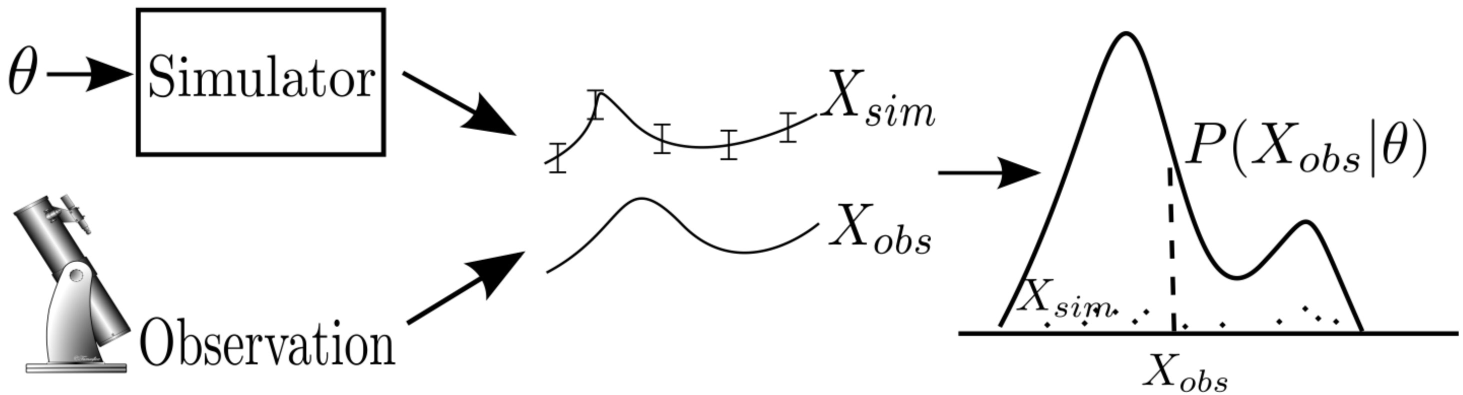
Active Learning & Design Optimization

Recommend experiments to find the true parameters of the universe



Computation problem: How to search parameter space

Solution: Learn a surrogate function and make experiment decisions using it



Question:

How well can we estimate $P_{\theta|\mathbf{X}_{\mathbf{obs}}}$ with a few queries ?

Existing methods:

- MCMC – evaluate likelihood and then keep/reject sample using a test.
- ABC – 'Likelihood Free', but sampling is also expensive.
- Nested Sampling, Kernel Bayes' Rule

None of these are designed to be query efficient.

Gaussian Processes

Main Idea

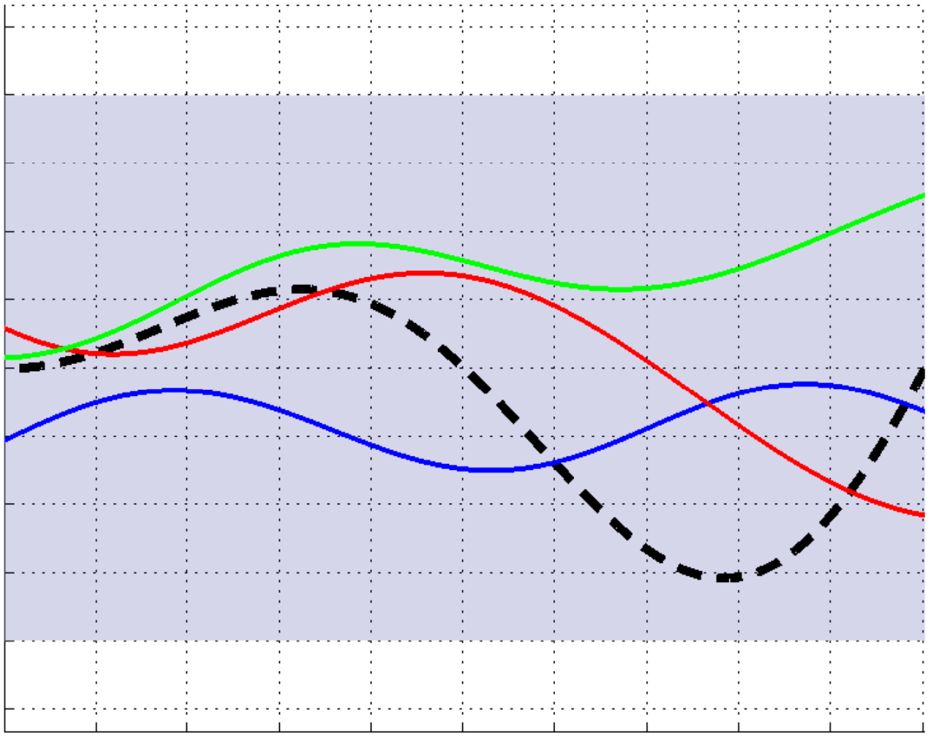
- Posterior estimation via regression.
- Actively select points based on current observations.

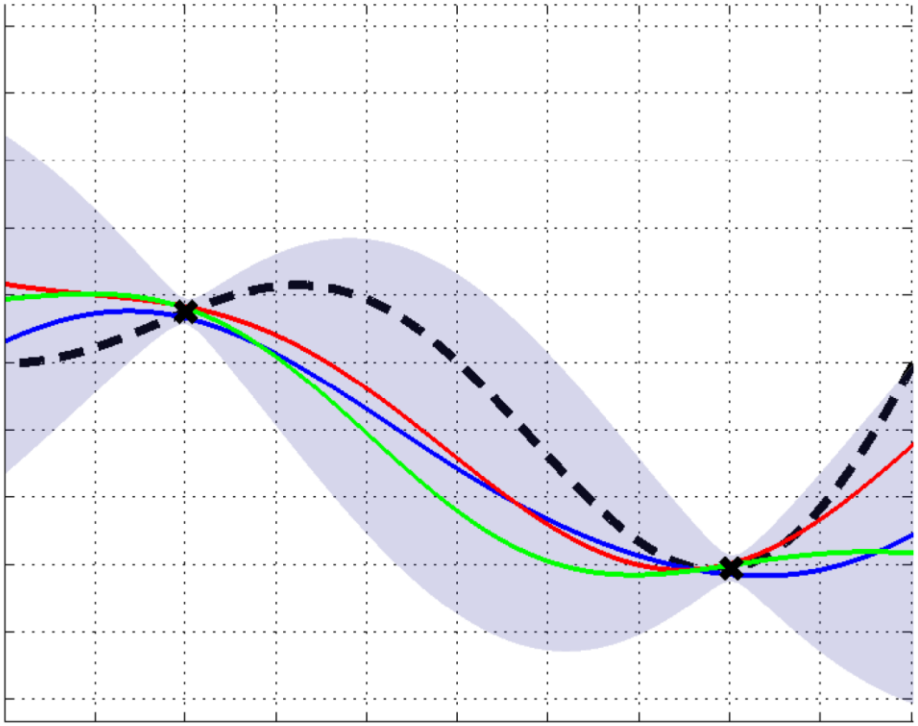
GP

- A random process on $\Theta \subset \mathbb{R}^d$.
- A distribution over functions $f : \Theta \to \mathbb{R}$
- Characterised via a mean function $\mu(\cdot)$ and a covariance kernel $k(\cdot, \cdot)$ – written $f \sim \mathcal{GP}(\mu, k)$.
- Function value at any finite set of points $\{\theta_1, \ldots, \theta_n\}$ are jointly Gaussian,

$$\begin{bmatrix} f(\theta_1) \\ \vdots \\ f(\theta_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(\theta_1) \\ \vdots \\ \mu(\theta_n) \end{bmatrix}, \begin{bmatrix} k(\theta_1, \theta_1) & \ldots & k(\theta_1, \theta_n) \\ \vdots & \ddots & \vdots \\ k(\theta_n, \theta_1) & \ldots & k(\theta_n, \theta_n) \end{bmatrix} \right)$$

Prior vs Posterior GP





Carnegie Mellon

Regression for Posterior Estimation

$$P_{\theta|\mathbf{X_{obs}}}(\theta|\mathbf{X_{obs}}) = \frac{\mathcal{L}_{\mathbf{X_{obs}}}(\theta)P_\theta(\theta)}{\int_\Theta \mathcal{L}_{\mathbf{X_{obs}}}(\theta)P_\theta(\theta)} = \frac{\mathcal{L}_{\mathbf{X_{obs}}}(\theta)P_\theta(\theta)}{P(\mathbf{X_{obs}})}$$

We work in the log joint probability space:

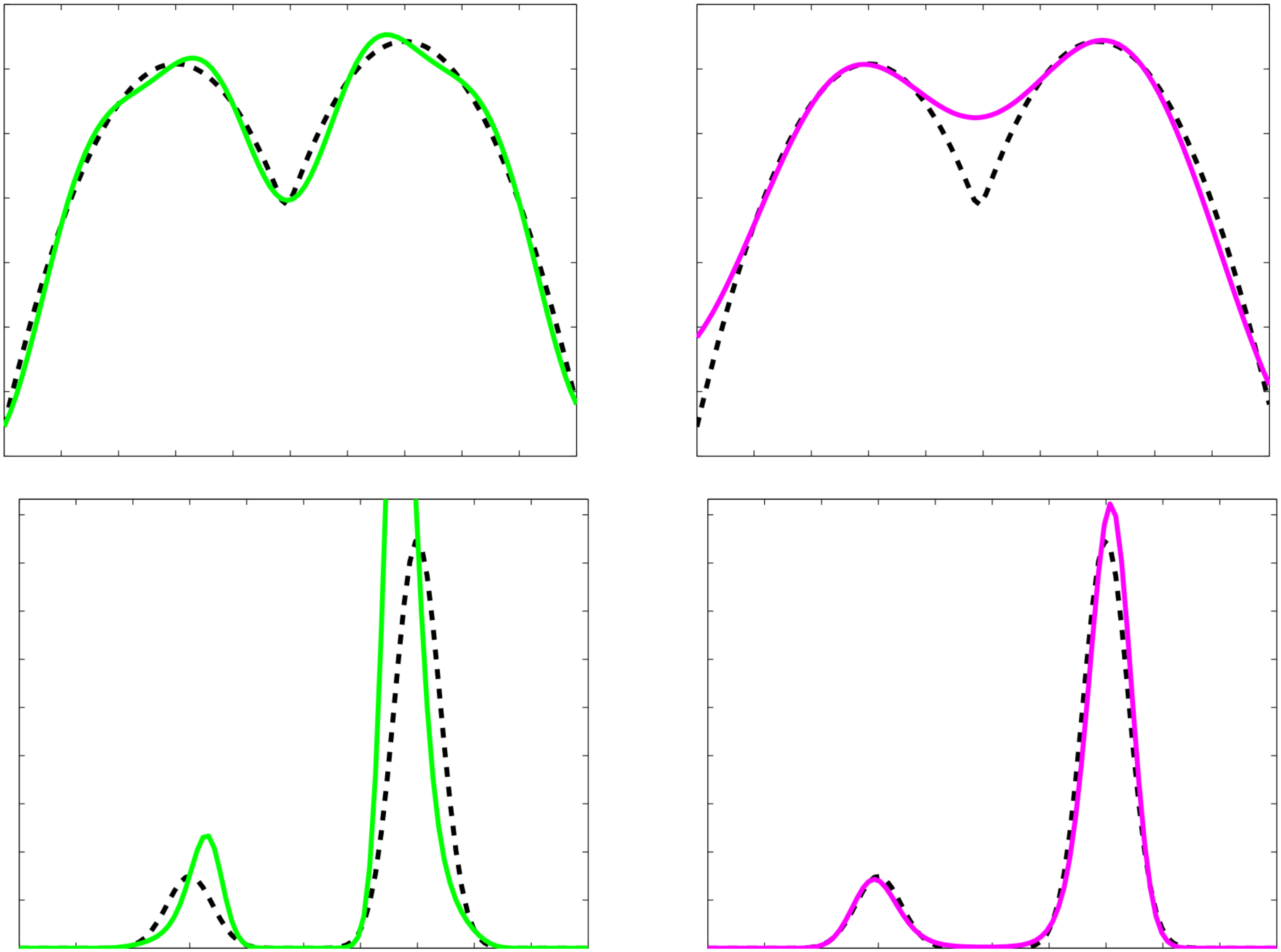
$$\log \mathcal{L}_{\mathbf{X_{obs}}}(\theta)P_\theta(\theta)$$

We have a regression algorithm,

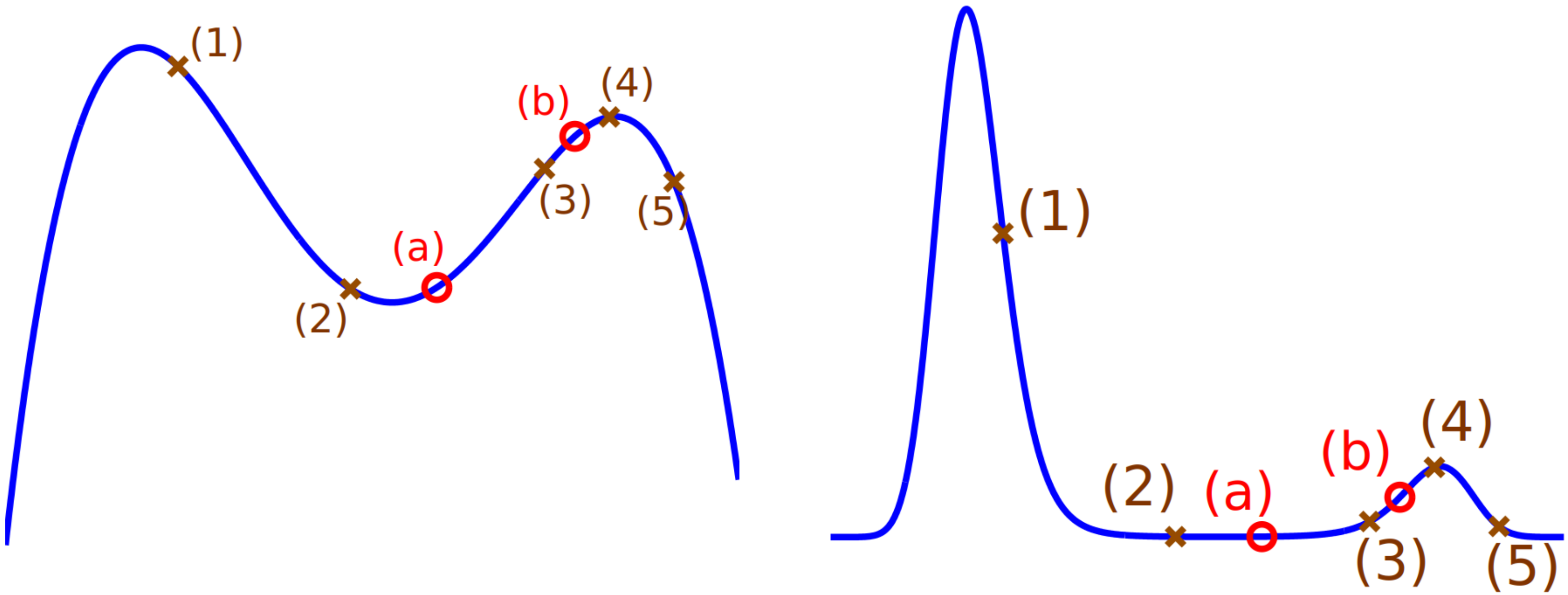
$$A_t = \{\theta_i, \log(\mathcal{L}_{\mathbf{X_{obs}}}(\theta_i)P_\theta(\theta_i))\}_{i=1}^t \longrightarrow \widehat{\mathcal{P}}^{A_t}(\theta, \mathbf{X_{obs}})$$

$$\widehat{P}^{A_t}(\theta|\mathbf{X_{obs}}) = \frac{\exp \widehat{\mathcal{P}}^{A_t}(\mathbf{X_{obs}}, \theta)}{\int_\Theta \exp \widehat{\mathcal{P}}^{A_t}(\mathbf{X_{obs}}, \theta)}$$

Which is the better estimate ?



Optimisation vs Active Regression vs Active Posterior Estimation



	(a)	(b)
Optimisation	No	No
Active Regression	Yes	Yes
Active Post-Estimation	No	Yes

A framework for Active Regression

- An iterative greedy algorithm that picks the next point based on the points we have thus far.
- At time t, we have observations at $t-1$ points: $A_{t-1} = \{\theta_i, \log(\mathcal{L}_{\mathbf{X}_{\text{obs}}}(\theta_i) P_\theta(\theta_i))\}_{i=1}^{t-1}$
- Design a utility function $u_t : \Theta \to \mathbb{R}$ using the posterior GP. $u_t(\theta)$ captures value/utility of querying at θ.
- Choose $\theta_t = \mathrm{argmax}_{\theta \in \Theta}\, u_t(\theta)$.
- Repeat.

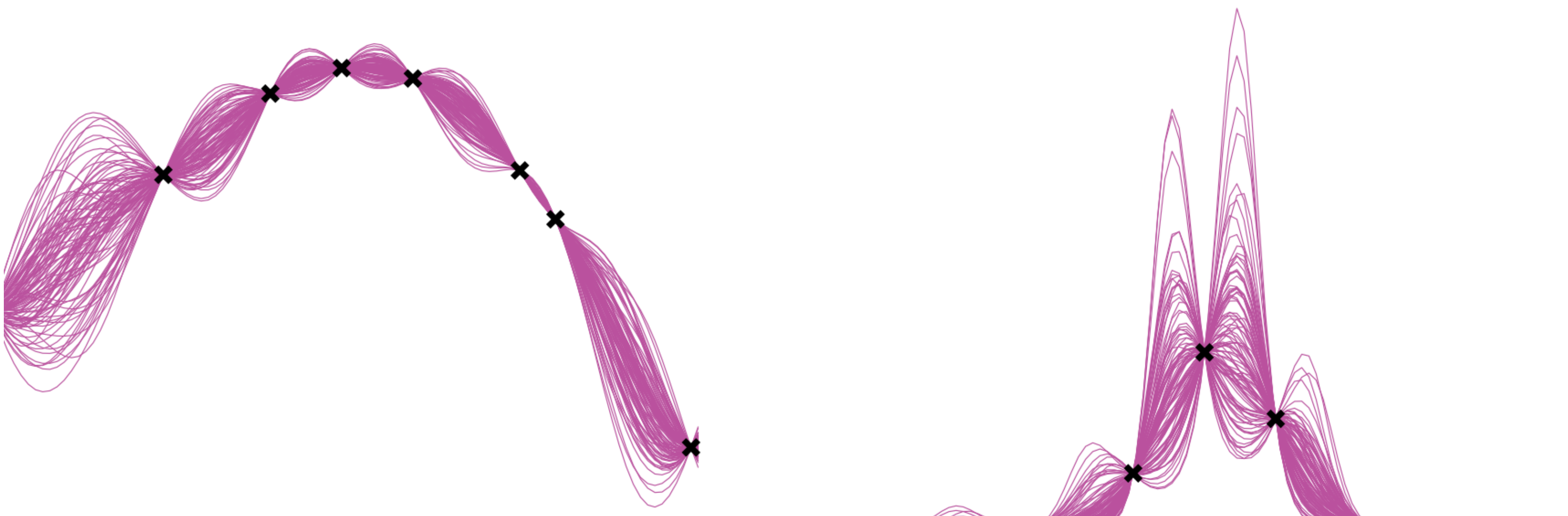
Utility:

Pick the point with the largest uncertainty

A framework for Active Regression

But .. Our GP is over the log joint probability

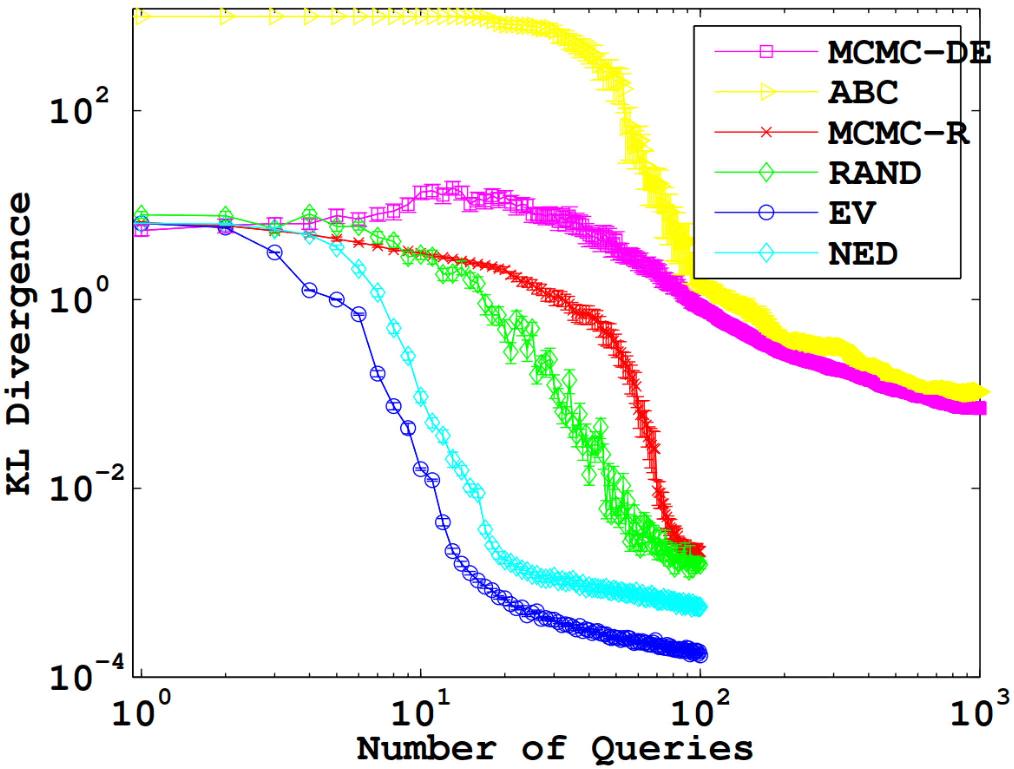
$\Longrightarrow$ pick largest variance in exponentiated GP.



Experiments

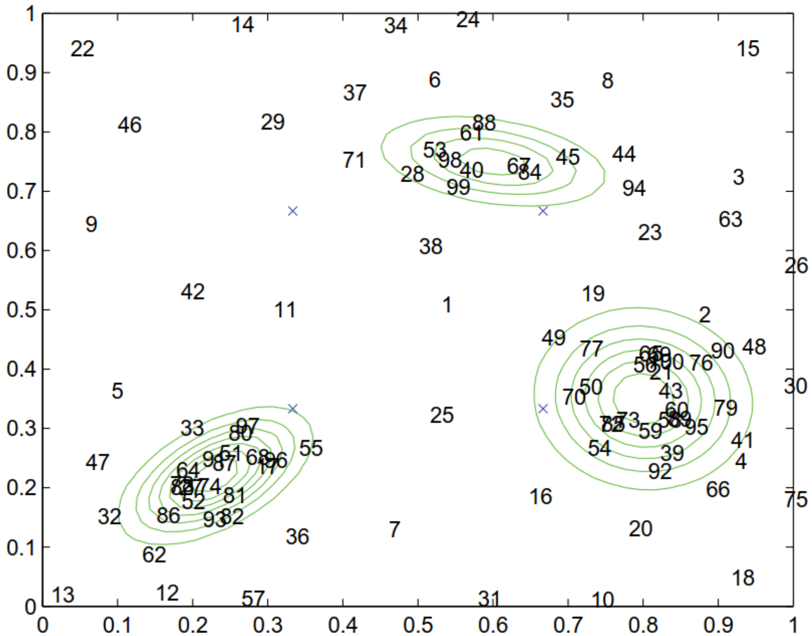
A simple one parameter problem,

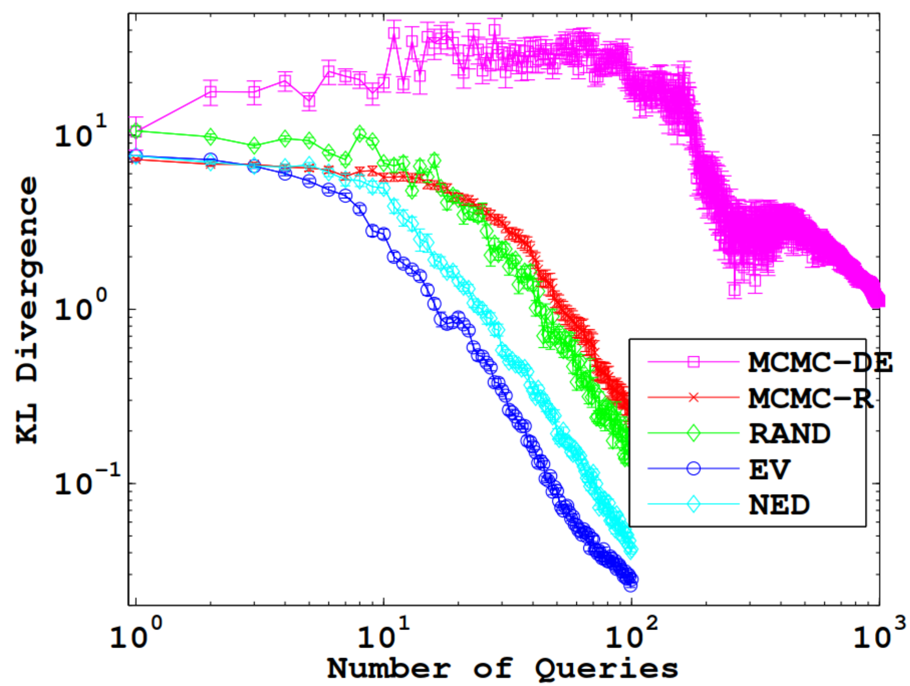
- $\Theta = (0, 1)$, $P_\theta : \mathrm{Beta}\,(1.2, 1)$
- $\mathbf{X}_{\mathbf{obs}} = \{X_1, \ldots, X_{500}\}$, $X_i \sim \mathrm{Bern}\,(\theta^2 + (1-\theta)^2)$.
- A bimodal posterior



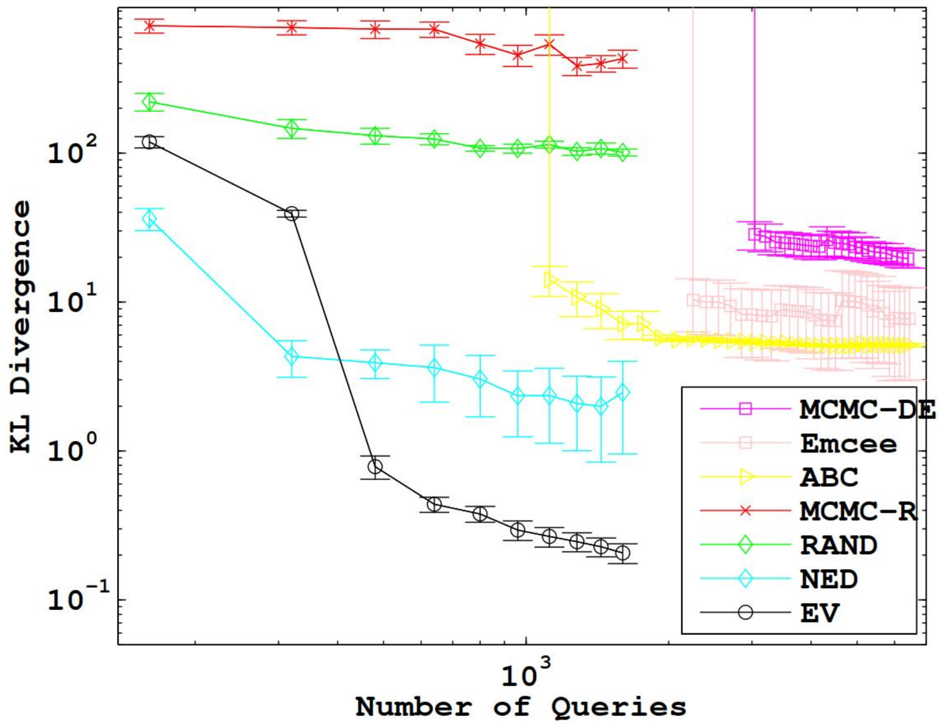
Experiments

A two parameter problem,





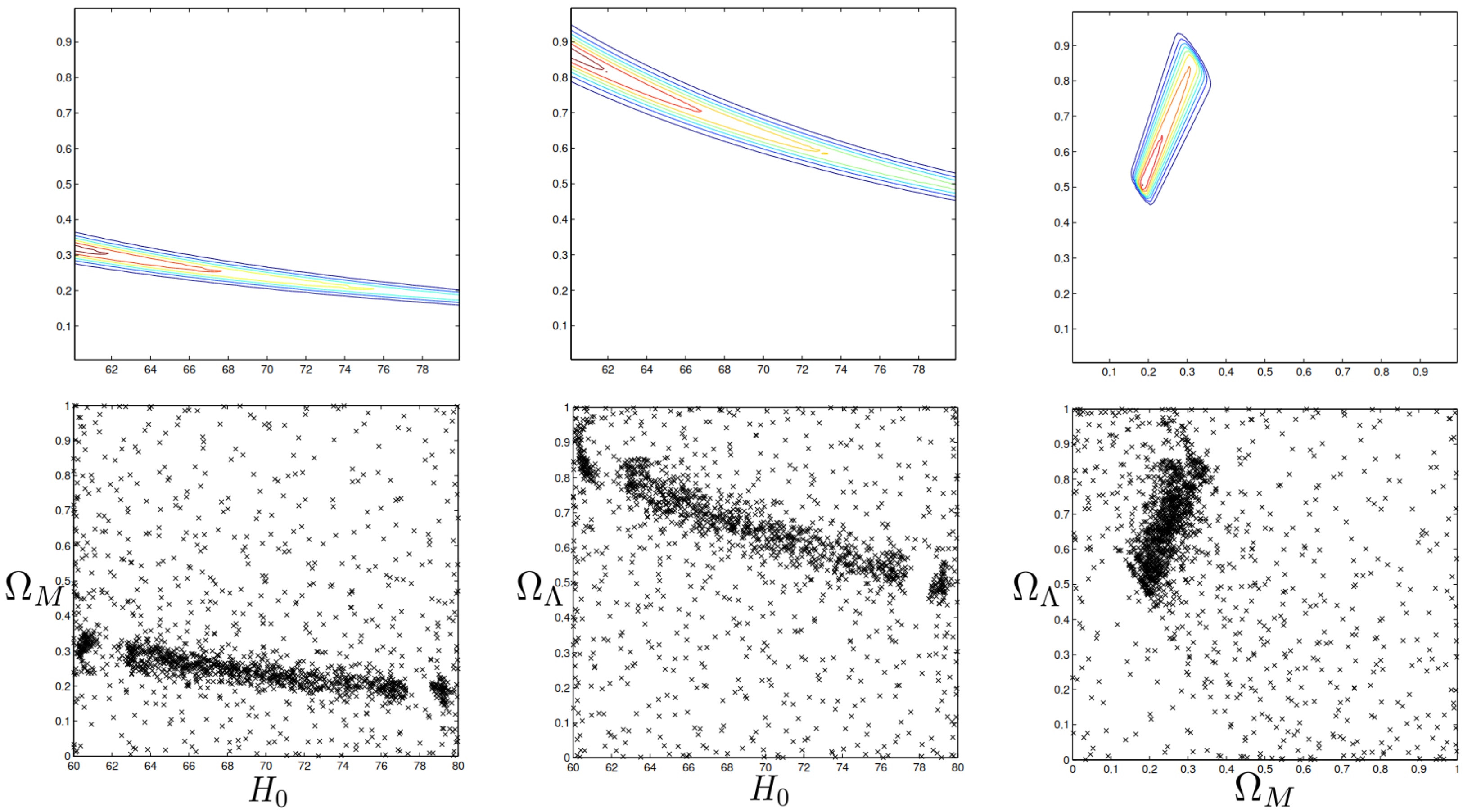
▸ Likelihood given by Robertson-Walker Metric



We use supernovae data for inference on 3 cosmological parameters: Hubble Constant (H0 ∈ (60, 80), Dark Matter Fraction ΩM ∈ (0, 1) and Dark Energy Fraction ΩΛ ∈ (0, 1).

The likelihood for the experiment is given by the Robertson– Walker metric which models the distance to a supernova given the parameters and the observed red-shift. The dataset is taken from Davis et al [2007].

Type Ia Supernovae



Carnegie Mellon

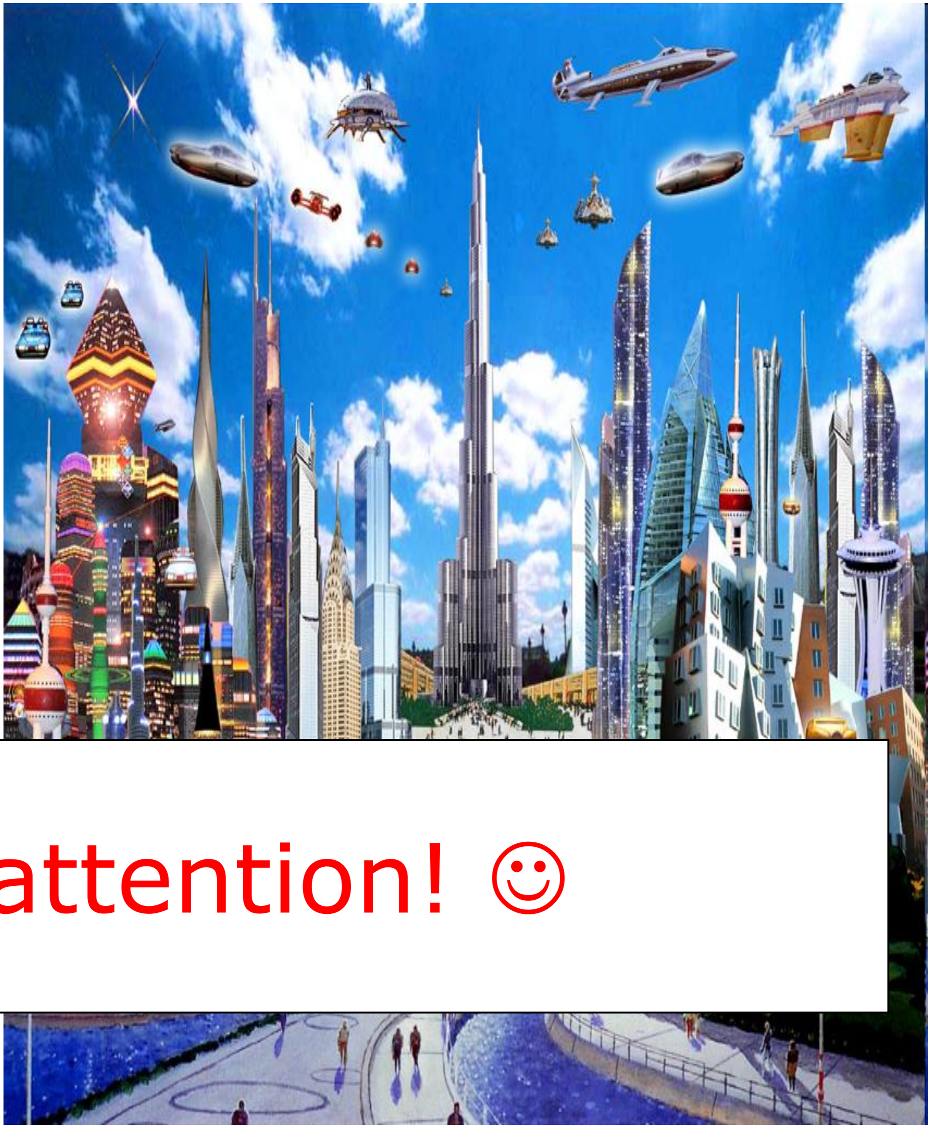
If you are interested, contact me! ☺

bapoczos@cs.cmu.edu, GHC-8231

Functional data and density functionals have so many applications!

Some results on regression/classification/anomaly detection/ Lasso

Lots of missing theoretical results:
Lov



Thanks for your attention! ☺