Bayes for Big Data

David Ríos Insua

AXA-ICMAT Chair and Royal Academy

Las Palmas February 2019

Outline

- **Bayesian methods: A brief reminder**
- Bayesian methods for ML: early approaches
- Bayes for BD: The challenges
- Bayes for BD: Some partial solutions
- Adversarial machine learning
- Discussion and challenges

Bayesian inference 1

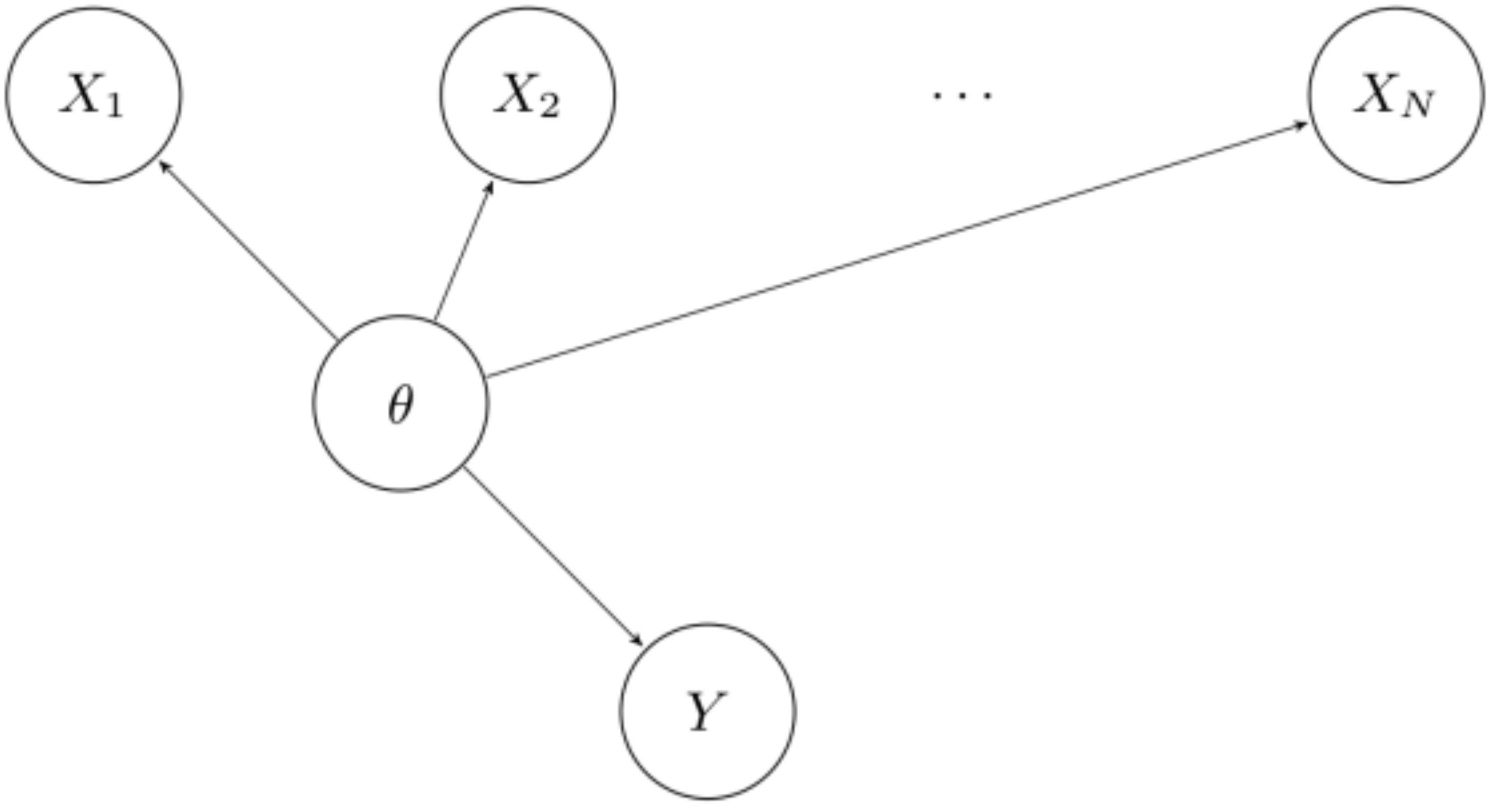
- We have iid observations providing info about a parameter of interest (cid given parameter)
- We also have prior information
- Combine both sources to obtain posterior
- The posterior summarises all info available for solving standard inference (science) problems:
  - **Point estimation**
  - **Interval estimation**
  - **Hypothesis testing**

Bayesian inference 2

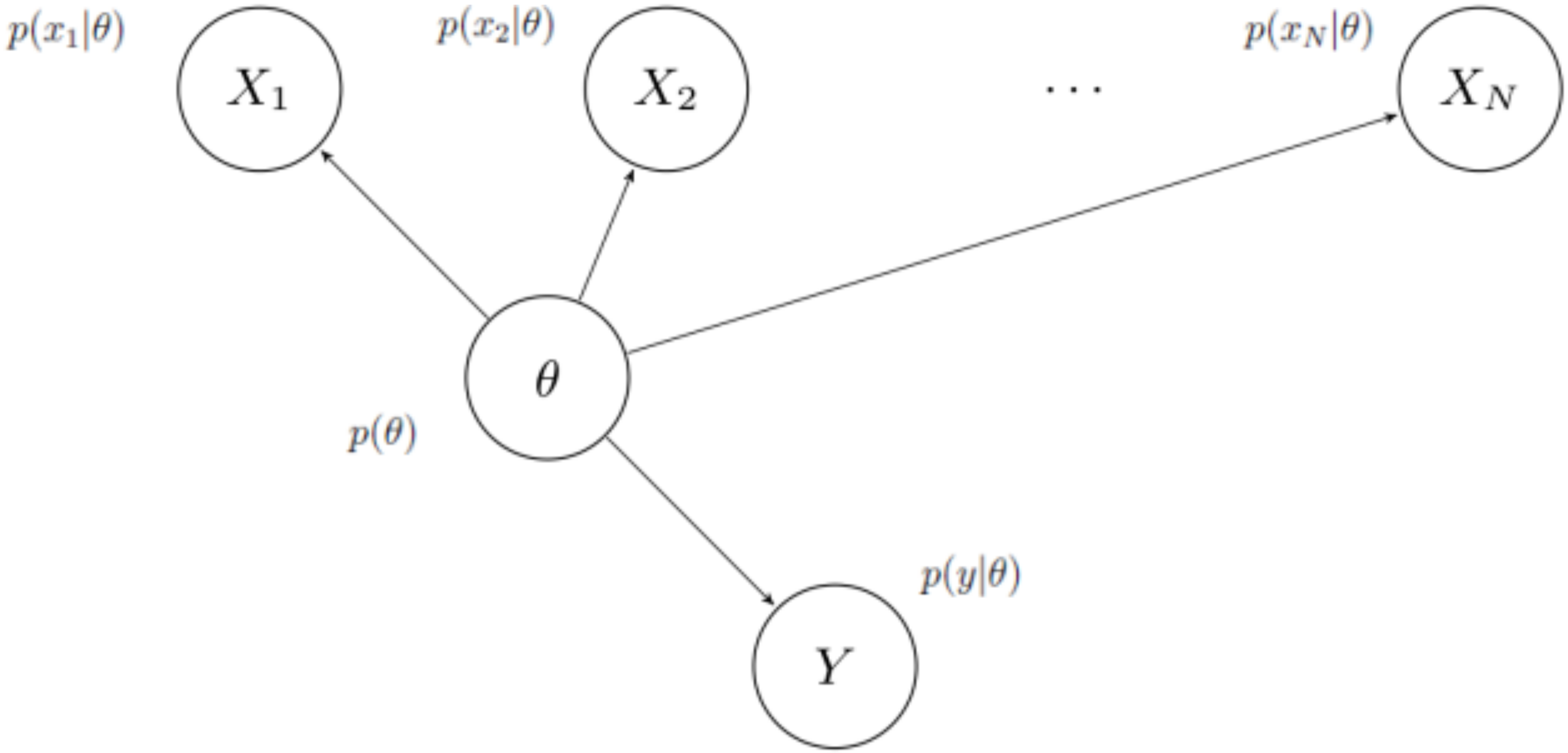
- The posterior is also key for solving two main problems in engineering, business and policy applications:

  – **Forecasting**. Through the predictive distribution (point forecast, interval forecast, predictive hypothesis testing).
  – **Decision support**. Maximising posterior (or predictive) expected utility.

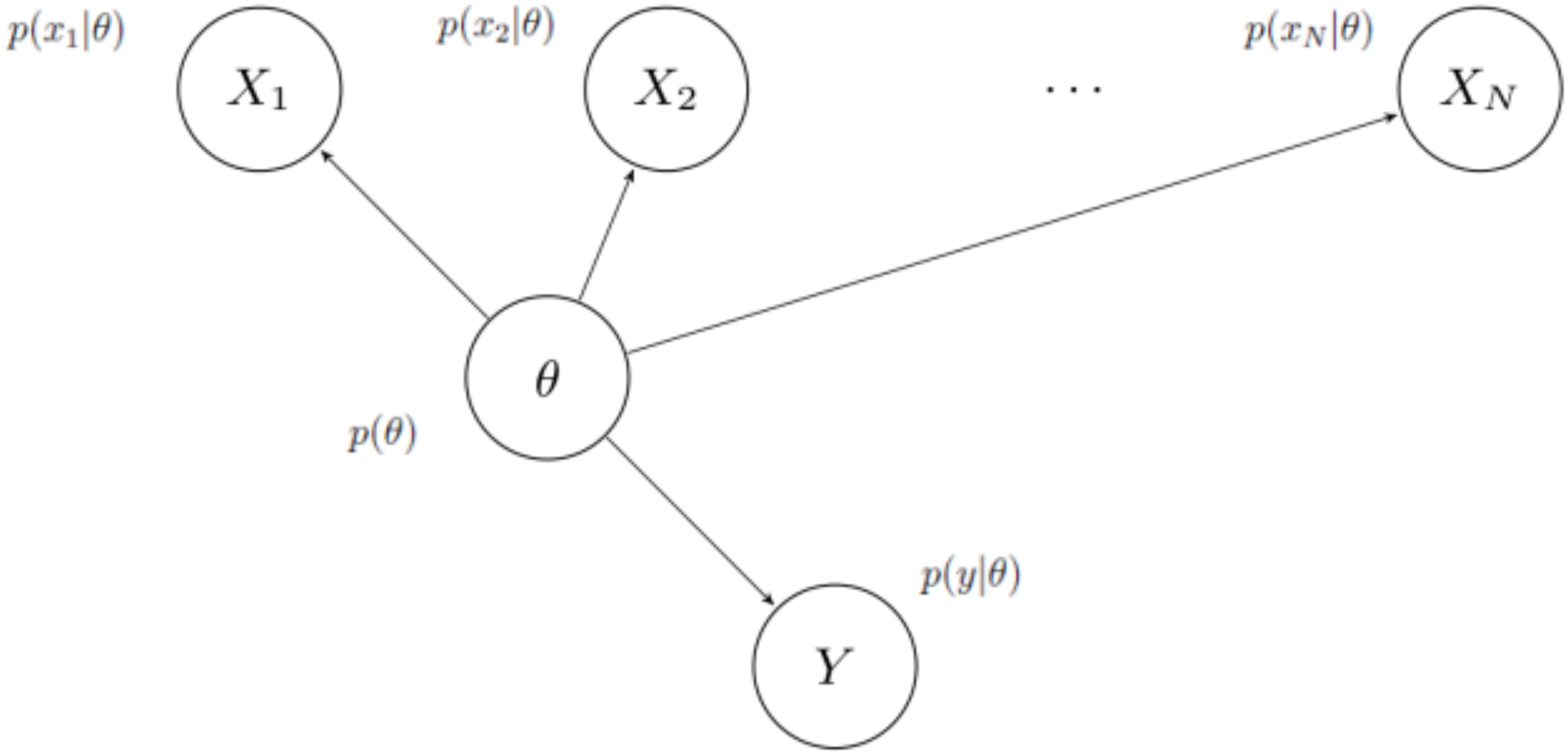
Bayesian inference 3



Bayesian inference 4



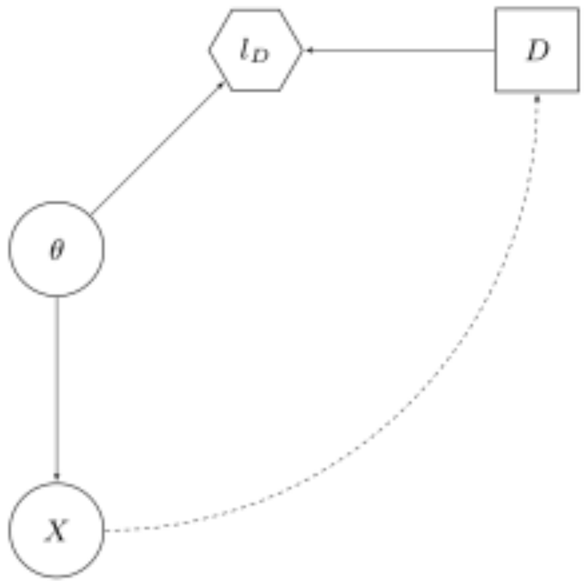
Bayesian inference 5



$$p(\theta|x_1, x_2, ..., x_N) = \frac{\Pi\, p(x_i|\theta)p(\theta)}{\int \Pi\, p(x_i|\theta)p(\theta)} = \frac{\Pi\, p(x_i|\theta)p(\theta)}{p(x_1,...,x_N)} \propto \Pi\, p(x_i|\theta)p(\theta) = l(\theta|x_1, ..., x_N)p(\theta)$$

$$p(y|x_1, x_2, ..., x_N) = \int p(y|\theta)p(\theta|x_1, x_2, ..., x_N)d\theta$$

BI as Bayesian SDT



$$d^*(x) = \arg\min_d \int l_D(d,\theta)\, p_D(\theta \mid x)\, \mathrm{d}\theta.$$

$$d^*(x) = \arg\min_d \int l_D(d,\theta)\, p_D(x \mid \theta)\, p_D(\theta)\, \mathrm{d}\theta,$$

Point estimation under quadratic loss

$$l_D(d,\theta) = (\theta - d)^2$$

$$d^*(x) = \frac{1}{p_D(x)} \int \theta\, p_D(x \mid \theta)\, p_D(\theta)\, \mathrm{d}\theta = \int \theta\, p_D(\theta \mid x)\, \mathrm{d}\theta = E\left[\theta \mid x\right]$$

Bayesian computation

Compute (posterior) maximum expected utility alternatives

$$\max_a \int u(c(a,\theta))p(\theta|x)d\theta$$

Sometimes, it may be convenient to solve

$$\max_a \int u(a,\theta)p(x|\theta)p(\theta)d\theta$$

One possibility, approximate expected utilities by Monte Carlo then optimise the MC sums… Sampling from the posterior??

1. Select a sample $\theta^1, ..., \theta^m \sim p(\theta|x)$.
2. Solve the optimisation problem

$$\max_{a \in \mathcal{A}} \frac{1}{m} \sum_{i=1}^{m} u(a, \theta^i)$$

yielding $a_m(\theta)$.

Bayesian computation: Gibbs sampler

Suppose we may sample from the marginal conditionals. Then, under certain conditions, the following scheme converges to the target distribution (the posterior)

1. Choose initial values $(\theta_2^0, \ldots, \theta_k^0)$. $i = 1$

2. Until convergence is detected, iterate through

   . Generate $\theta_1^i \sim \theta_1 | \theta_2^{i-1}, .., \theta_k^{i-1}$

   . Generate $\theta_2^i \sim \theta_2 | \theta_1^i, \theta_3^{i-1}, ..., \theta_k^{i-1}$

   . ...

   . Generate $\theta_k^i \sim \theta_k | \theta_1^i, ..., \theta_{k-1}^i$.

   . $i = i + 1$

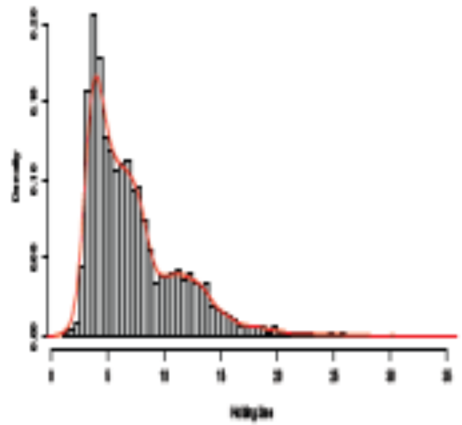
Bayesian computation: mixtures DRI et al, 2001

Modelling with mixtures provides a sound and flexible way to model uncertainty

- **Theoretically**. Any positive distribution may be approximated by a mixture of gammas; any distribution may be approximated by a mixture of normals ➔

  An approach to density estimation.

- Computationally. Ways to proceed via Markov chain Monte Carlo samplers (including uncertain number of components in mixture)
- Applications. Describe model heterogeneity (clustering), model uncertainty, …



Bayesian computation: mixtures

1. Start with arbitrary values $(\mathbf{q}^0, \boldsymbol{\mu}^0, \mathbf{z}^0)$, $i = 0$.
2. Until convergence, iterate through
   - Generate $\mathbf{z}_j^{i+1} \sim \mathbf{z}_j | t_j, \mathbf{q}^i, \boldsymbol{\mu}^i$, $j = 1, ..., n_s$.
   - Generate $\mathbf{q}^{i+1} \sim \mathbf{q} | \mathbf{t}, \mathbf{z}^{i+1}$.
   - Generate $\mu_j^{i+1} \sim \mu_j | \mathbf{t}, \mathbf{z}^{i+1}$, $j = 1, ..., k$.
   - Set $i = i + 1$.

Extended to an unknown number of components (DRI et al, 2001)

Bayesian computation: Metropolis

Sometimes, we cannot sample from conditionals.
We know, up to a constant, the target. By choosing an appropriate candidate generating distribution q(.|.), under appropriate conditions, this scheme is designed to converge to the target distribution

1. Choose initial values θ^0. $i = 0$

2. Until convergence is detected, iterate through

   . Generate a candidate $\theta^* \sim q(\theta|\theta^i)$.

   . If $p(\theta^i)q(\theta^i \mid \theta^*) > 0$, $\alpha(\theta^i, \theta^*) = \min\left(\frac{p(\theta^*)q(\theta^*|\theta^i)}{p(\theta^i)q(\theta^i|\theta^*)}, 1\right)$;

   . else, $\alpha(\theta^i, \theta^*) = 1$.

   . Do

$$\theta^{i+1} = \begin{cases} \theta^* & \text{with prob } \alpha(\theta^i, \theta^*), \\ \theta^i & \text{with prob } 1 - \alpha(\theta^i, \theta^*) \end{cases}$$

   . $i = i + 1$.

Bayesian computation: Augmented probability simulation (Bielza et al, 2000)

Frequently, the involved posterior depends on decision made. The following observation helps in this context. Define an artificial distribution such that (u, needs to be nonnegative)

$$h(a,\theta) \propto u(a,\theta) \times p_\theta(\theta \mid x, a).$$

The marginal of the artificial distribution is proportional to expected utility

$$h(a) = \int h(a,\theta)\, d\theta_{a,\theta} \propto \Psi(a).$$

This suggests

1. Generate a sample $((\theta^1, a^1), ..., (\theta^m, a^m))$ from density $h(a,\theta)$.
2. Convert it to a sample $(a^1, ..., a^m)$ from the marginal $h(a)$.
3. Find the sample mode.

Bayesian methods: Advantages (French, DRI, 2000)

- All information taken into account
- Axiomatic basis, coherent framework
- Uncertainty duly apportioned and acknowledged
- Transparent to users
- Robust
- Compatible with a wider philosophy
- Feasibility

Bayesian methods: Advantages (French, DRI, 2000)

longer major driving forces in model building. It is interesting to chart the history of applied Bayesian methods through the proceedings of the Valencia conferences from their beginnings in 1979 to their most recent in 1998. The balance has shifted from conceptual and analytical issues in theoretical models to computational aspects of applied studies. Today Bayesian methods are most certainly practicable. Indeed, for complicated models, Bayesian analysis has arguably now become the simplest (and often the only possible) method of analysis.

Bayesian methods: Advantages (French, DRI, 2000)

- All information taken into account
- Axiomatic basis, coherent framework
- Uncertainty duly apportioned and acknowledged
- Transparent to users
- Robust, robust against attacks
- Compatible with a wider philosophy
- Feasibility

Bayesian methods: Advantages TODAY

- All information taken into account
- Axiomatic basis, coherent framework
- Uncertainty duly apportioned and acknowledged
- Transparent to users
- Robust, robust against attacks
- Compatible with a wider philosophy
- Feasibility….

Outline

- Bayesian methods: A brief reminder
- **Bayesian methods for ML: early approaches**
- Bayes for BD: The challenges
- Bayes for BD: Some partial solutions
- Adversarial machine learning
- Discussion and challenges

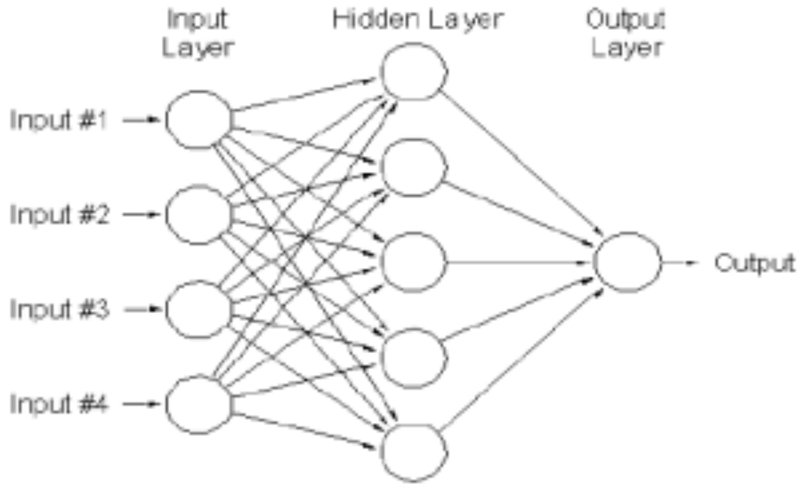
Bayes for neural nets (Muller, DRI, 2000)

$$\hat{y}(x) = \sum_{j=1}^{M} \beta_j \psi(x'\gamma_j + \delta_j)$$

$$y_i = \sum_{j=1}^{M} \beta_j \psi(x_i'\gamma_j) + \epsilon_i, \quad i = 1, \ldots, N,$$

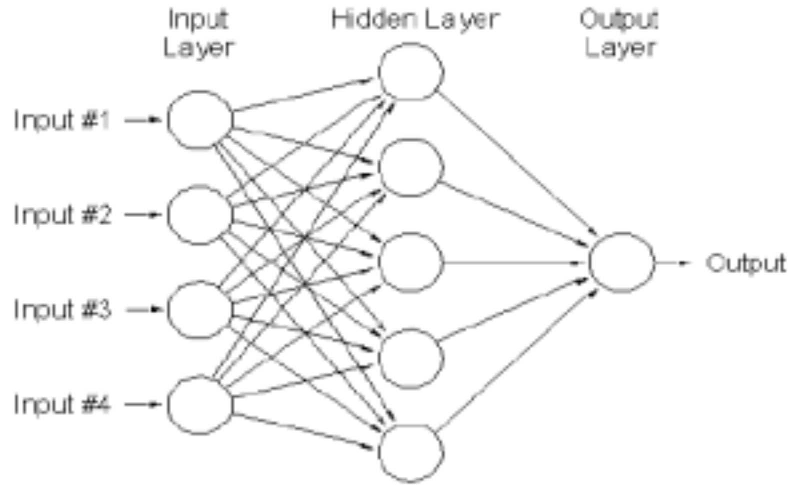
$$\epsilon_i \sim N(0, \sigma^2), \quad \psi(\eta) = \exp(\eta)/(1 + \exp(\eta))$$

$$\beta_j \sim N(\mu_\beta, \sigma_\beta^2), \quad \gamma_j \sim N(\mu_\gamma, S_\gamma), \quad j = 1, \ldots, M$$



Bayes for neural nets (Muller, DRI, 2000)

$$\hat{y}(x) = \sum_{j=1}^{M} \beta_j \psi(x'\gamma_j + \delta_j)$$



$$y_i = \sum_{j=1}^{M} \beta_j \psi(x_i'\gamma_j) + \epsilon_i, \quad i = 1, \ldots, N,$$

$$\epsilon_i \sim N(0, \sigma^2), \quad \psi(\eta) = \exp(\eta)/(1 + \exp(\eta))$$

$$\beta_j \sim N(\mu_\beta, \sigma_\beta^2), \quad \gamma_j \sim N(\mu_\gamma, S_\gamma), \quad j = 1, \ldots, M$$

$$\mu_\beta \sim N(a_\beta, A_\beta), \mu_\gamma \sim N(a_\gamma, A_\gamma), \sigma_\beta^{-2} \sim Gamma(c_b/2, c_b C_b/2)$$

$$S_\gamma^{-1} \sim Wish(c_\gamma, (c_\gamma C_\gamma)^{-1}), \text{ and } \sigma^{-2} \sim Gamma(s/2, sS/2)$$

Bayes for neural nets

1. Start with θ equal to some initial guess (for example, the prior means).

Until convergence is achieved, iterate through steps 2 through 4:

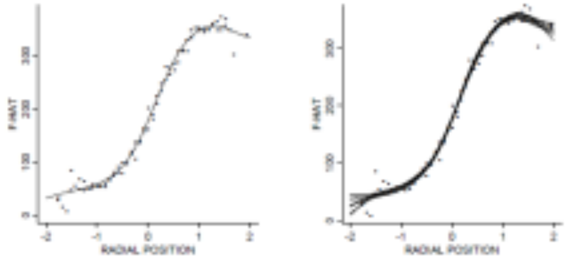
2. Given current values of ν only, (marginalizing over β) replace γ by Metropolis steps: For each γ_j, $j = 1, \ldots, M$, generate a proposal $\tilde{\gamma}_j \sim g_j(\gamma_j)$, with $g_j(\gamma_j)$ described below. Compute

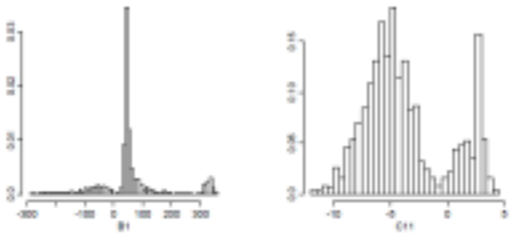
$$a(\gamma_j, \tilde{\gamma}_j) = \min\left[1, \frac{p(D|\tilde{\gamma}, \nu)p(\tilde{\gamma}|\nu)}{p(D|\gamma, \nu)p(\gamma|\nu)}\right], \tag{2.4}$$

   where $\tilde{\gamma} = (\gamma_1, \ldots, \gamma_{j-1}, \tilde{\gamma}_j, \gamma_{j+1}, \ldots, \gamma_M)$. With probability $a(\gamma_j, \tilde{\gamma}_j)$ replace γ_j by the new candidate $\tilde{\gamma}_j$. Otherwise leave γ_j unchanged. Use Lemma 2.1 to evaluate $p(D|\gamma, \nu)$.

3. Given current values of (γ, ν), generate new values for β by a draw from the complete conditional $p(\beta|\gamma, \nu, D)$. This is a multivariate normal distribution with moments described in Lemma 2.1.

4. Given current values of (β, γ), replace the hyperparameters by a draw from the respective complete conditional posterior distributions: $p(\mu_\beta|\beta, \sigma_\beta)$ is a normal distribution, $p(\mu_\gamma|\gamma, S_\gamma)$ is multivariate normal, $p(\sigma_\beta^{-2}|\beta, \mu_\beta)$ is a Gamma distribution, $p(S_\gamma^{-1}|\gamma, \mu_\gamma)$ is Wishart, and $p(\sigma^{-2}|\beta, \gamma, y)$ is Gamma, as corresponds to a normal linear model. (See Bernardo & Smith, 1994).





$$\hat{f}(x) = \hat{E}(y_{n+1}|x_{n+1}, D) = \frac{1}{k}\sum_{t=1}^{k} E(y_{N+1}|x_{n+1}, \theta = \theta_t)$$

Bayes for neural nets

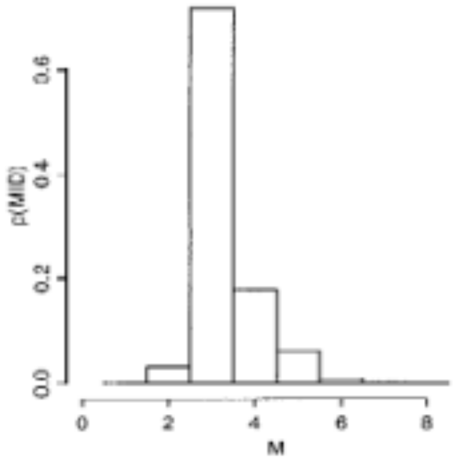
$$y_i = x_i'\lambda + \sum_{j=1}^{M^*} d_j \beta_j \psi(x_i'\gamma_j) + \epsilon_i,\ i = 1, \ldots, N,$$

$$\epsilon_i \sim N(0, \sigma^2), \quad \psi(\eta) = \exp(\eta)/(1 + \exp(\eta))$$

$$\gamma_{1p} \leq \gamma_{2p} \cdots \leq \gamma_{Mp}$$

$$Pr(d_j = d | d_{j-1} = 1) = \begin{cases} 1 - \alpha, & \text{for } d = 0 \\ \alpha & \text{for } d = 1 \end{cases} \quad j = 1, \ldots, M^*,$$

$$\beta_j \sim N(\mu_\beta, \sigma_\beta^2), \quad \lambda \sim N(\mu_\beta, \sigma_\beta^2),$$
$$\gamma_j \sim N(\mu_\gamma, S_\gamma), \quad \alpha \sim Beta(a_\alpha, b_\alpha).$$



Hyperpriors

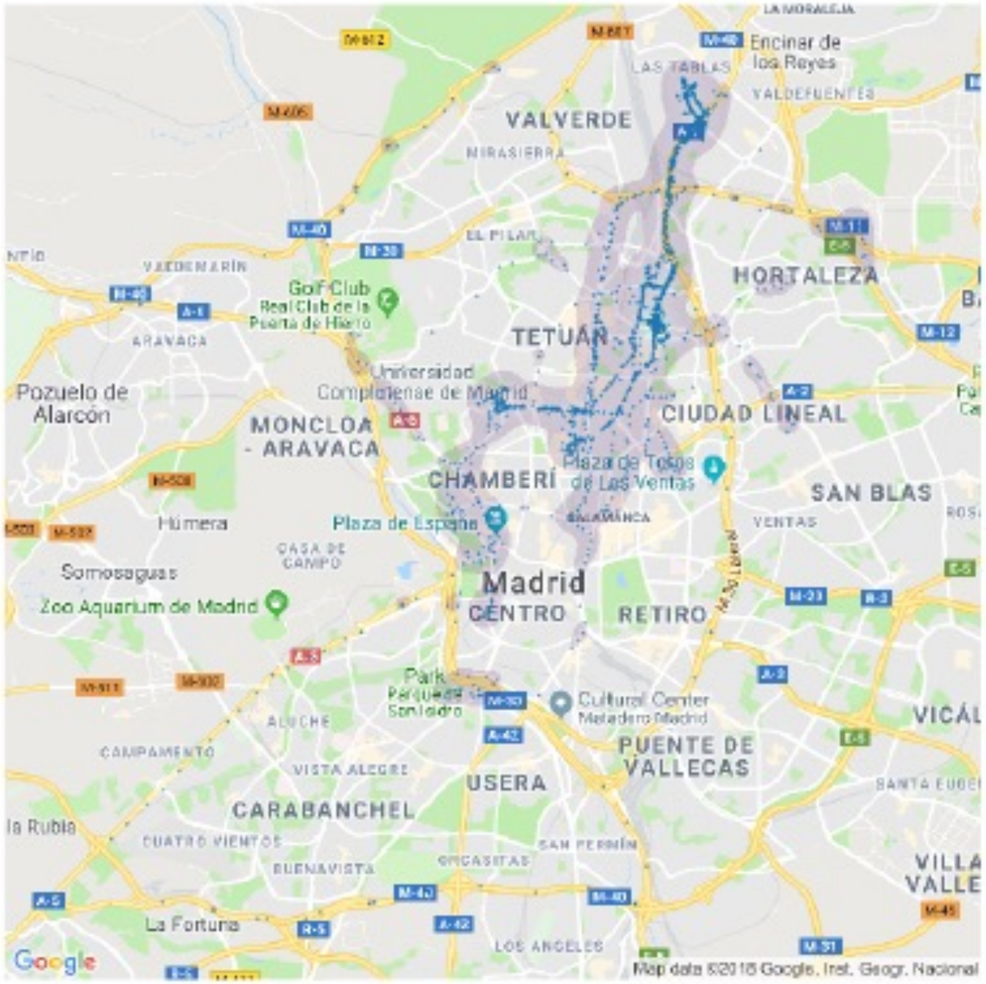
Outline

- Bayesian methods: A brief reminder
- Bayesian methods for ML: early approaches
- **Bayes methods for BD: The challenges**
- Bayes methods for BD: Some partial solutions
- Adversarial machine learning
- Discussion and challenges

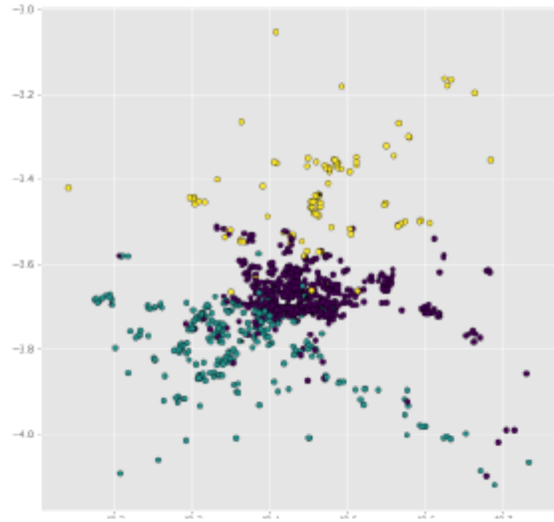
ML meets BD

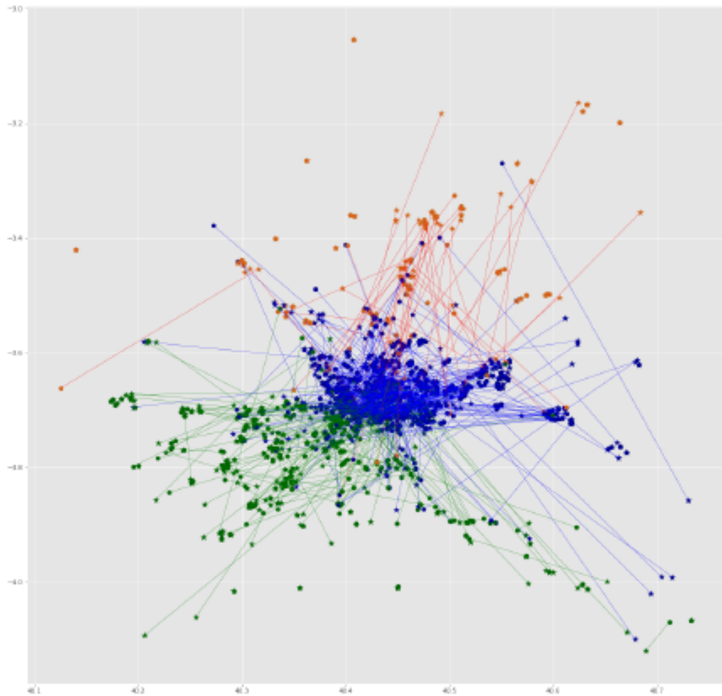
- Volume (big n, big p)
- Variety. Text, images, sound, video,….
- Velocity. High frequency, time series, dynamic models
- Value

ML meets BD



LA MORALEJA
Encinar de los Reyes
LAS TABLAS
VALDEFUENTES
VALVERDE
MIRASIERRA
EL PILAR
HORTALEZA
VALDEMARÍN
Golf Club Real Club de la Puerta de Hierro
ARAVACA
TETUÁN
Universidad Complutense de Madrid
Pozuelo de Alarcón
MONCLOA - ARAVACA
CIUDAD LINEAL
CHAMBERÍ
Plaza de Toros de Las Ventas
SAN BLAS
VENTAS
SALAMANCA
Húmera
Plaza de España
CASA DE CAMPO
Somosaguas
Madrid
CENTRO
RETIRO
Zoo Aquarium de Madrid
Park Parque de San Isidro
Cultural Center Matadero Madrid
ALUCHE
VICÁLVARO
CAMPAMENTO
PUENTE DE VALLECAS
VISTA ALEGRE
USERA
CARABANCHEL
SANTA EUGENIA
la Rubia
CUATRO VIENTOS
SAN FERMÍN
BUENAVISTA
ORCASITAS
VILLA VALLECAS
La Fortuna
LOS ANGELES
Google
Map data ©2018 Google, Inst. Geogr. Nacional





MLE meets BD

- MLE. Optimize
$$J(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x},y\sim\hat{p}_{\text{data}}} L(\boldsymbol{x}, y, \boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} L(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{\theta})$$
$$L(\boldsymbol{x}, y, \boldsymbol{\theta}) = -\log p(y \mid \boldsymbol{x}; \boldsymbol{\theta})$$

- The gradient is
$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} \nabla_{\boldsymbol{\theta}} L(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{\theta})$$

If m billion…but gradient is an expectation and may be estimated via mini-batches (SGD)

$$\boldsymbol{g} = \frac{1}{m'}\nabla_{\boldsymbol{\theta}}\sum_{i=1}^{m'} L(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{\theta})$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon \boldsymbol{g}$$

Require: Learning rate ϵ_k.
Require: Initial parameter $\boldsymbol{\theta}$
while stopping criterion not met **do**
  Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
  Compute gradient estimate: $\hat{\boldsymbol{g}} \leftarrow +\frac{1}{m}\nabla_{\boldsymbol{\theta}}\sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$
  Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon\hat{\boldsymbol{g}}$
end while

$$\sum_{k=1}^{\infty} \epsilon_k = \infty$$

$$\sum_{k=1}^{\infty} \epsilon_k^2 < \infty$$

Bayes meets BD

1. Start with arbitrary values $(\mathbf{q}^0, \boldsymbol{\mu}^0, \mathbf{z}^0)$, $i = 0$.

2. Until convergence, iterate through

   . Generate $\mathbf{z}_j^{i+1} \sim \mathbf{z}_j | t_j, \mathbf{q}^i, \boldsymbol{\mu}^i$, $j = 1, ..., n_s$.

   . Generate $\mathbf{q}^{i+1} \sim \mathbf{q} | \mathbf{t}, \mathbf{z}^{i+1}$.

   . Generate $\mu_j^{i+1} \sim \mu_j | \mathbf{t}, \mathbf{z}^{i+1}$, $j = 1, ..., k$.

   . Set $i = i + 1$.

Bayes meets BD

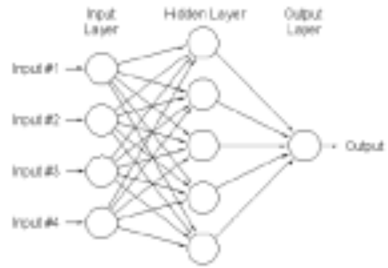
1. Start with θ equal to some initial guess (for example, the prior means).

Until convergence is achieved, iterate through steps 2 through 4:

2. Given current values of ν only, (marginalizing over β) replace γ by Metropolis steps: For each γ_j, $j = 1, \ldots, M$, generate a proposal $\tilde{\gamma}_j \sim g_j(\gamma_j)$, with $g_j(\gamma_j)$ described below. Compute

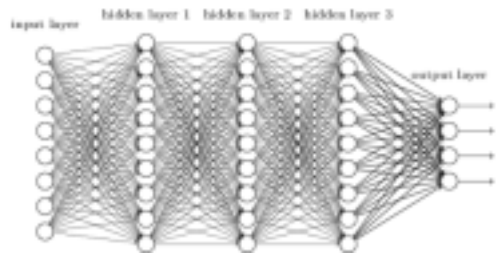
$$a(\gamma_j, \tilde{\gamma}_j) = \min\left[1, \frac{p(D|\tilde{\gamma}, \nu)p(\tilde{\gamma}|\nu)}{p(D|\gamma, \nu)p(\gamma|\nu)}\right], \tag{2.4}$$

where $\tilde{\gamma} = (\gamma_1, \ldots, \gamma_{j-1}, \tilde{\gamma}_j, \gamma_{j+1}, \ldots, \gamma_M)$. With probability $a(\gamma_j, \tilde{\gamma}_j)$ replace γ_j by the new candidate $\tilde{\gamma}_j$. Otherwise leave γ_j unchanged. Use Lemma 2.1 to evaluate $p(D|\gamma, \nu)$.



3. Given current values of (γ, ν), generate new values for β by a draw from the complete conditional $p(\beta|\gamma, \nu, D)$. This is a multivariate normal distribution with moments described in Lemma 2.1.

4. Given current values of (β, γ), replace the hyperparameters by a draw from the respective complete conditional posterior distributions: $p(\mu_\beta|\beta, \sigma_\beta)$ is a normal distribution, $p(\mu_\gamma|\gamma, S_\gamma)$ is multivariate normal, $p(\sigma_\beta^{-2}|\beta, \mu_\beta)$ is a Gamma distribution, $p(S_\gamma^{-1}|\gamma, \mu_\gamma)$ is Wishart, and $p(\sigma^{-2}|\beta, \gamma, y)$ is Gamma, as corresponds to a normal linear model. (See Bernardo & Smith, 1994).



$$\hat{f}(x) = \hat{E}(y_{n+1}|x_{n+1}, D) = \frac{1}{k}\sum_{t=1}^{k} E(y_{N+1}|x_{n+1}, \theta = \theta_t)$$

Bayes meets BD

- Computational bottlenecks per iteration
  - Evaluating the likelihood
  - Visiting all parameters
  - Visiting all data
- Slow mixing rate

Outline

- Bayesian methods: A reminder
- Bayesian methods for ML: early approaches
- Bayes for BD: The challenges
- **Bayes for BD: Some partial solutions**
- Adversarial machine learning
- Discussion and challenges

Large scale network monitoring (Naveiro et al, 2018)

- Objective: Monitor **safety** and **security** of **several hundred thousands** of ICDs generating **tens of variables every few minutes**

Huge number of high frequency time series

- **Automatic**, **Scalable** , **Versatile (and Accurate)** approach

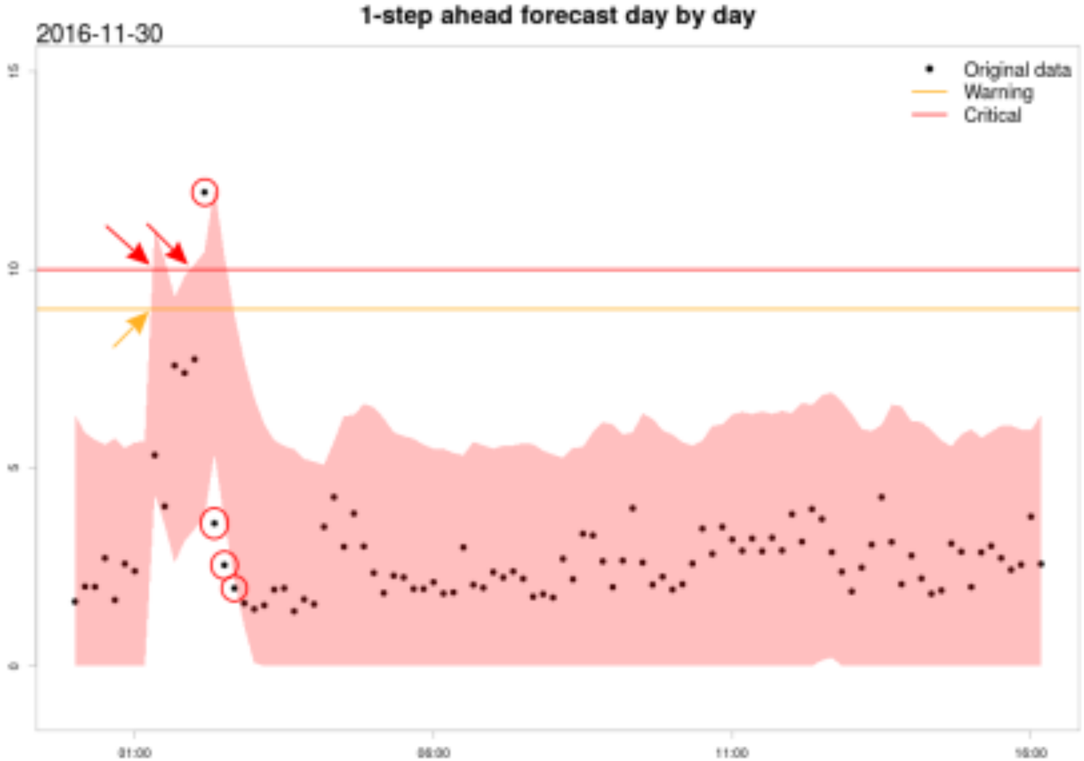
- Idea: Dynamic Linear Models (trend+seasonal) + Outbursts:

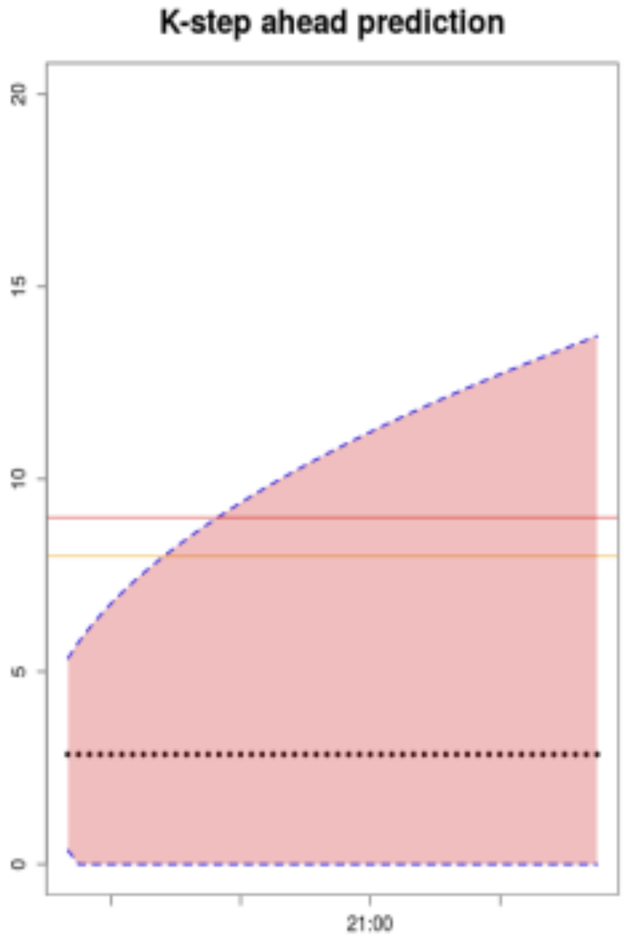
- Constructed blockwise (relatively easy to automate)
- Each block captures some feature of the series (versatility)
- Store a few parameters (space scalability)
- Conjugacy, fast posterior computation (time scalability)

Large scale network monitoring (Naveiro et al, 2018)

- **Automatized estimating**
- **Sequential learning**: parameters are updated as new data is incorporated
- Produce **full predictive distributions.** Thi y:

- Aberrant behaviour detection
- Foresee critical/warning values





Bayes meets BD

- MLE . Optimize

$$J(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x},\mathrm{y}\sim\hat{p}_{\mathrm{data}}} L(\boldsymbol{x}, y, \boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} L(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{\theta})$$

$$L(\boldsymbol{x}, y, \boldsymbol{\theta}) = -\log p(y \mid \boldsymbol{x}; \boldsymbol{\theta})$$

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} \nabla_{\boldsymbol{\theta}} L(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{\theta})$$

Bayes meets BD

- MLE +regulariser . Optimize

$$J(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x},\mathrm{y}\sim\hat{p}_{\text{data}}} L(\boldsymbol{x}, y, \boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} L(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{\theta}) \quad +h(\theta)$$

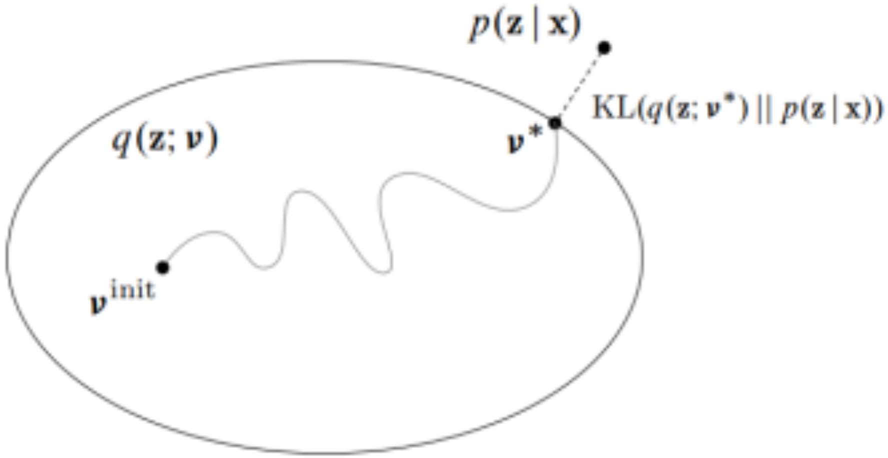
$$L(\boldsymbol{x}, y, \boldsymbol{\theta}) = -\log p(y \mid \boldsymbol{x}; \boldsymbol{\theta})$$

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} \nabla_{\boldsymbol{\theta}} L(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{\theta}) \quad +\nabla_{\theta} h(\theta)$$

MAP. Uncertainty….

Variational Bayes (Blei et al 2018)

- **Variational principle:** approximate a complex density with a member of a family of simpler, tractable densities.



- Inference as an **optimization problem**:
  - KL divergence measures distance between distributions
  - ν are the variational parameters
  - Goal: $\nu^* = \mathrm{argmin}_\nu \mathrm{KL}\left(q(z;\nu)||p(z|x)\right)$
  - But how to compute the previous KL divergence?

Variational Bayes

$$KL(q||p) \equiv \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{x}, \mathbf{z})] + \log p(\mathbf{x})$$

- Instead, optimize an alternative objective equivalent to the KL divergence up to a constant. Evidence lower bound

$$\text{ELBO}(\nu) = \mathbb{E}_q\left[\log p(z, x)\right] - \mathbb{E}_q\left[\log q(z|\nu)\right]$$

- Simplest choice for q is a factorized Gaussian
  - The second ELBO term is tractable.
- Ongoing research:
  - more flexible and tractable posterior approximations:
    - auxiliary variables, normalizing flows, etc..
  - optimize other divergences:
    - alpha-divergences, Stein discrepancies, etc

- **Advantages** over MCMC (as of today): much faster inference times, scales better
- **Problems**: biased, underestimates variances

Parallelising MCMC

- Increase availability of computer power have changed the way statistical analyses are carried out
- Parallel processing exploited dividing task into subtasks executed in parallel
- Extremely useful tool in BD
- MC trivially to parallelize

$$\widehat{h} = \frac{1}{N} \sum_{i=1}^{N} h(X_i)$$

- Divide the sum into P>2 components and assign one processor to evaluate each component
- MCMC harder to parallelize. Elements of sequence are not independent. I need X_i to compute X_{i+1}
- Compute several chains in several processors and mix. (Scott et al 2013)

Parallel MCMC via Weierstrass Sampling (Wang, Dunson, 2016)

- Idea: Partition data into m non-overlapping subsets
- Under independence assumption

$$p(\theta|X) \propto p(\theta|X_1)p(\theta|X_2)\cdots p(\theta|X_m) = \prod_{i=1}^{m} p(\theta|X_i)$$

- Weierstrass transformation $W_h f(\theta) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}h} e^{-\frac{(\theta-t)^2}{2h^2}} f(t)dt$ converges to $f(\theta)$ as h tends to 0.
- Define $f_i(\theta) = p(\theta|X_i)$ then full posterior approximated by

$$\prod_{i=1}^{m} f_i(\theta) \approx \prod_{i=1}^{m} W_{h_i} f_i(\theta) = \prod_{i=1}^{m} \int \frac{1}{\sqrt{2\pi}h_i} e^{-\frac{(\theta-t_i)^2}{2h_i^2}} f(t_i)dt_i$$

$$\propto \int \exp\left\{-\frac{(\theta-\bar{t})^2}{2h_0^2}\right\} \exp\left(-\frac{\bar{t^2}-\bar{t}^2}{2h_0^2}\right) f_1(t_1)\cdots f_m(t_m)dt$$

- This is the marginal of $\boldsymbol{\theta}$ with $p(\theta, t_1, \ldots, t_m)$

$$\exp\left\{-\frac{(\theta-\bar{t})^2}{2h_0^2}\right\} \exp\left(-\frac{\bar{t^2}-\bar{t}^2}{2h_0^2}\right) \cdot f_1(t_1) \cdot f_2(t_2) \cdots f_m(t_m)$$

Enables subset-based posterior sample that can be parallelized!!

GIBBS Sampling

$$\theta | t_i \sim N(\bar{t}, h_0^2)$$

$$t_i | \theta \sim \frac{1}{\sqrt{2\pi}h_i} e^{-\frac{(t_i-\theta)^2}{2h_i^2}} f_i(t_i) \quad i = 1, 2, \ldots, m.$$

Speeding MCMC

- Stochastic gradient Langevin dynamics (SGLD) (Teh et al, 2016)
- Hamiltonian Monte Carlo (Chen et al, 2014)
- SGLD+Repulsion (Gallego et al, 2018)

Algorithm 1 Bayesian Inference via SGLD+R

Input: A target distribution with density function $\pi(\boldsymbol{z}) \propto \exp(-H(\boldsymbol{z}))$.
Output: A set of particles $\{\boldsymbol{z}_i\}_{i=1}^{MK}$ that approximates the target distribution.
Sample initial set of particles from prior: $\boldsymbol{z}_1^0, \boldsymbol{z}_2^0, \dots \boldsymbol{z}_K^0 \sim \pi(\boldsymbol{z})$.
for each iteration t **do**

$$\boldsymbol{z}_i^{t+1} \leftarrow \boldsymbol{z}_i^t - \epsilon_t \frac{1}{K} \sum_{j=1}^{K} \left[k(\boldsymbol{z}_j^t, \boldsymbol{z}_i^t) \nabla_{\boldsymbol{z}_j^t} H(\boldsymbol{z}_j^t) + \nabla_{\boldsymbol{z}_j^t} k(\boldsymbol{z}_j^t, \boldsymbol{z}_i^t) \right] + \boldsymbol{\eta}_i^t \tag{6}$$

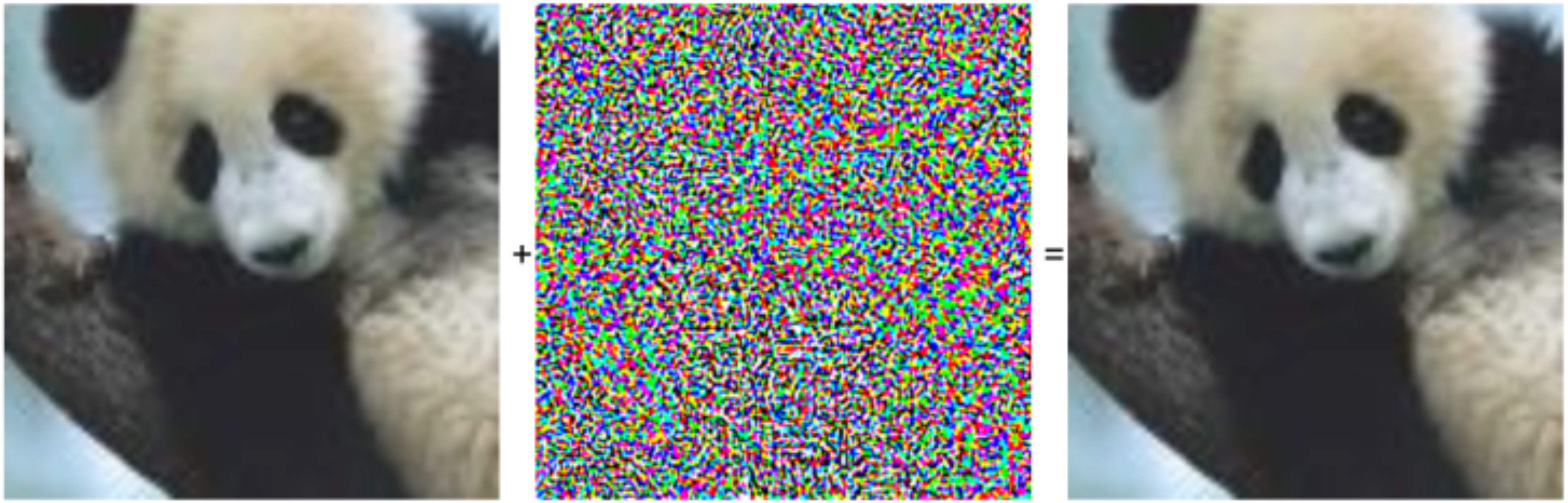
where $\boldsymbol{\eta}_i^t$ is the noise for particle i defined as in Eq (5).
After a burn-in period, start collecting particles: $\{\boldsymbol{z}_i\}_{i=1}^{NK} \leftarrow \{\boldsymbol{z}_i\}_{i=1}^{(N-1)K} \cup \{\boldsymbol{z}_1^{t+1}, \dots, \boldsymbol{z}_K^{t+1}\}$
end for

Exploit gradient info to generate samples far from current point with high posterior density

Outline

- Bayesian methods: A brief reminder
- Bayesian methods for ML: early approaches
- Bayes for BD: The challenges
- Bayes for BD: Some partial solutions
- **Adversarial machine learning**
- Discussion and challenges

ML meets security



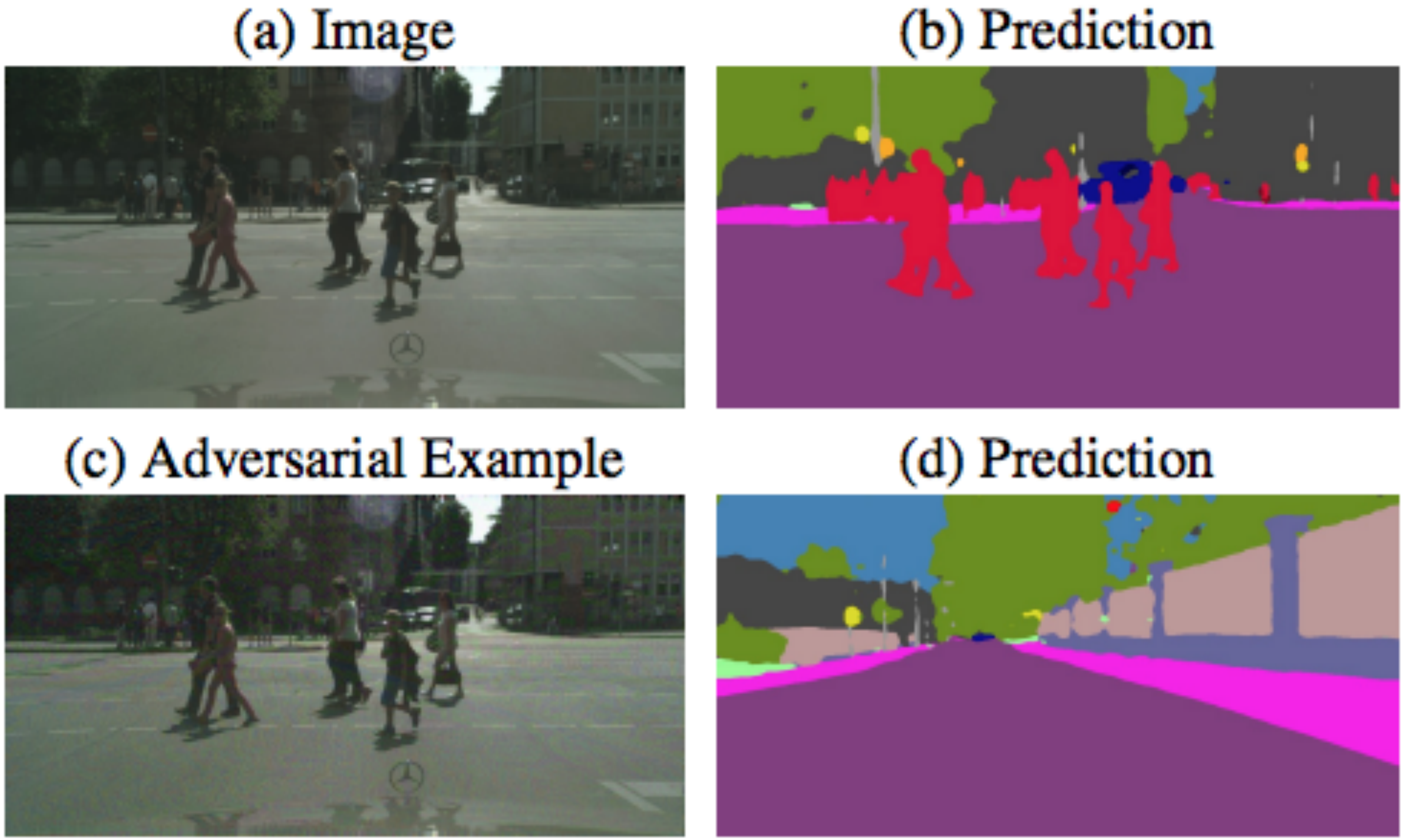
Original image classified as a panda with 60% confidence.

Tiny adversarial perturbation.

Imperceptibly modified image, classified as a gibbon with 99% confidence.

So what?

ML meets security

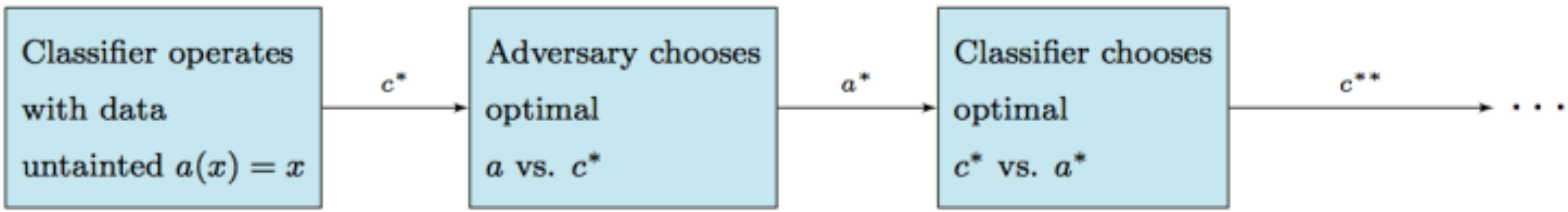


Adversarial classification

- C, classifier. A, adversary
- Two classes: + malicious; - innocent.
- C and A **maximise expected utility** under **common knowledge** conditions
- Finding **Nash equilibria** extremely **complex**

Adversarial classification Dalvi et al 2004

Proposed scheme (based on utility sensitive NB classifier)



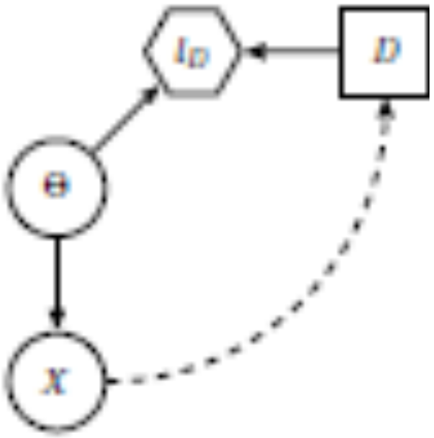
Forward myopic approach under **strong common knowledge!**

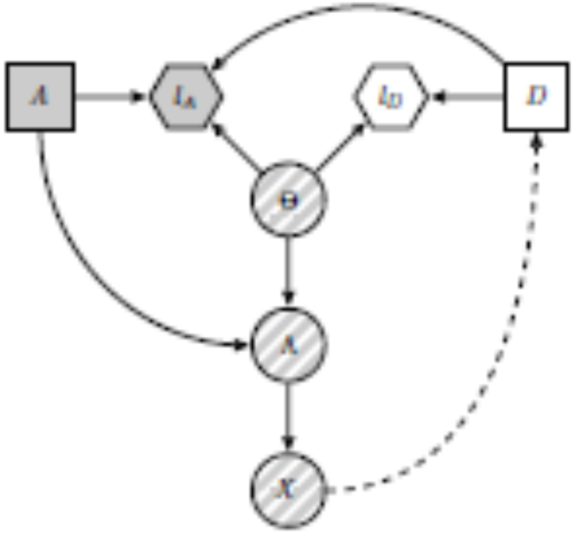
AML

- Everything adversarial: adversarial SVM's,….
- Mostly viewed from a game theoretic perspective.
- Common knowledge too commonly assumed. **Uncertainty about attackers**
- A very difficult area.
- Very relevant area from an applied point of view: security and cybersecurity
- Google, AICS competitions

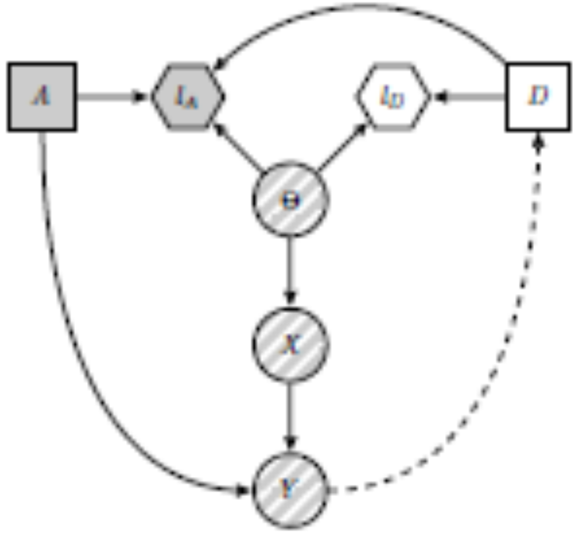
Adversarial Statistical Decision Theory

Ortega et al, 2018

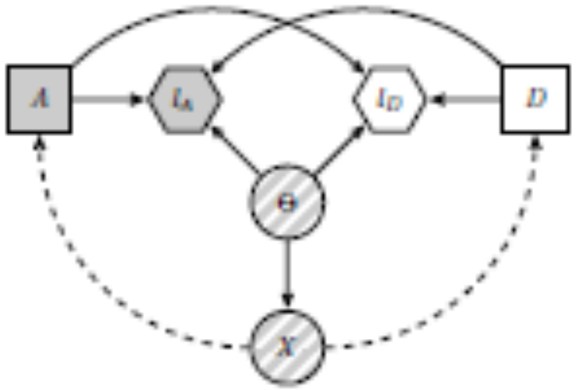




(a) Structural attacker

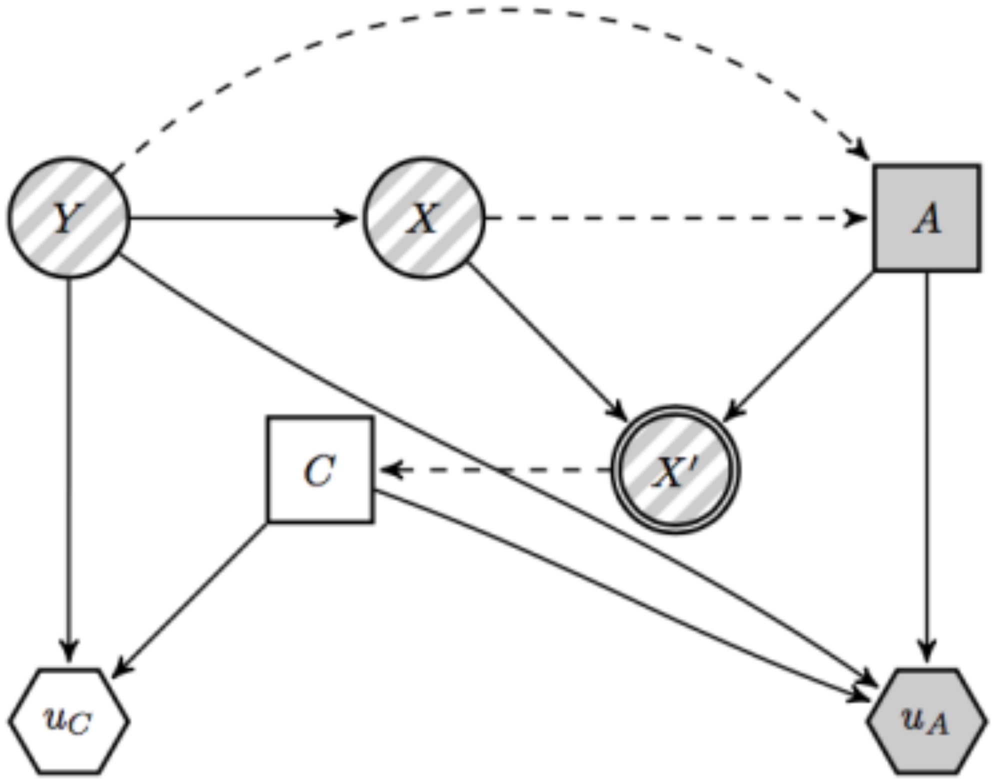


(b) Data-fiddler attacker



(c) Simultaneous ASDT problem

Adversarial classification through ARA. ACRA Naveiro et al 2018



Malicious (+) or innocent (-)

ACRA. Classifier problem

$$\begin{aligned} c(x') &= \arg\max_{y_C} \sum_{y \in \{+,-\}} u_C(y_C, y) p_C(y|x') \\ &= \arg\max_{y_C} \sum_{y \in \{+,-\}} u_C(y_C, y) p_C(y) p_C(x'|y) = \\ &= \arg\max_{y_C} \sum_{y \in \{+,-\}} u_C(y_C, y) p_C(y) \sum_{x \in \mathcal{X}'} \sum_{a \in \mathcal{A}(x)} p_C(x', x, a|y) \end{aligned}$$

(...)

$$= \arg\max_{y_C} \left[u_C(y_C, +) p_C(+) \sum_{x \in \mathcal{X}'} p_C(a_{x \to x'} | x, +) p_C(x|+) + u_C(y_C, -) p_C(x'|-) p_C(-) \right]$$

ACRA. Adversary problem

$$a^*(x,y) \;=\; \arg\max_a \int \Big[u_A(c(a(x)) = +, y, a)\, p + u_A(c(a(x)) = -, y, a)\, (1-p) \Big] f_A(p|a(x))dp$$

$$\int \Big[u_A(+,+,a)\ p + u_A(-,+,a)\ (1-p) \Big] f_A(p|a(x))dp =$$

$$\Big[u_A(+,+,a) - u_A(-,+,a) \Big] p^A_{a(x)} + u_A(-,+,a)$$

$$A^*(x,+) = \arg\max_a \left([U_A(+,+,a) - U_A(-,+,a)]\, P^A_{a(x)} + U_A(-,+,a) \right)$$

random version of

$$p_C(a|x,+) = Pr(A^*(x,+) = a)$$

$$p^A_{a(x)} = \int p f_A(p|a(x))dp$$

$$P_A(c|x') \sim \beta e(\delta_1, \delta_2) \longrightarrow \frac{\delta_1}{\delta_1+\delta_2} = Pr_A(c(x') = +)$$

ACRA. Operational framework

1. PREPROCESSING
   Train **generative classifier** to estimate $p_C(y)$ and $p_C(x|y)$

1. OPERATION
   Read x'.

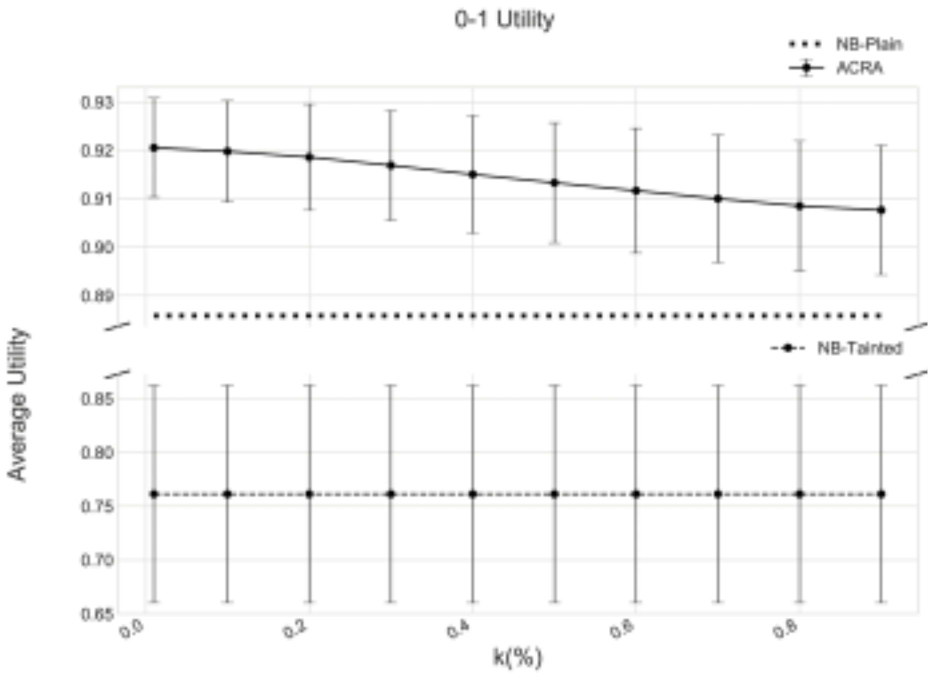
   Estimate $p_C(a_{x \to x'}|x, +)$

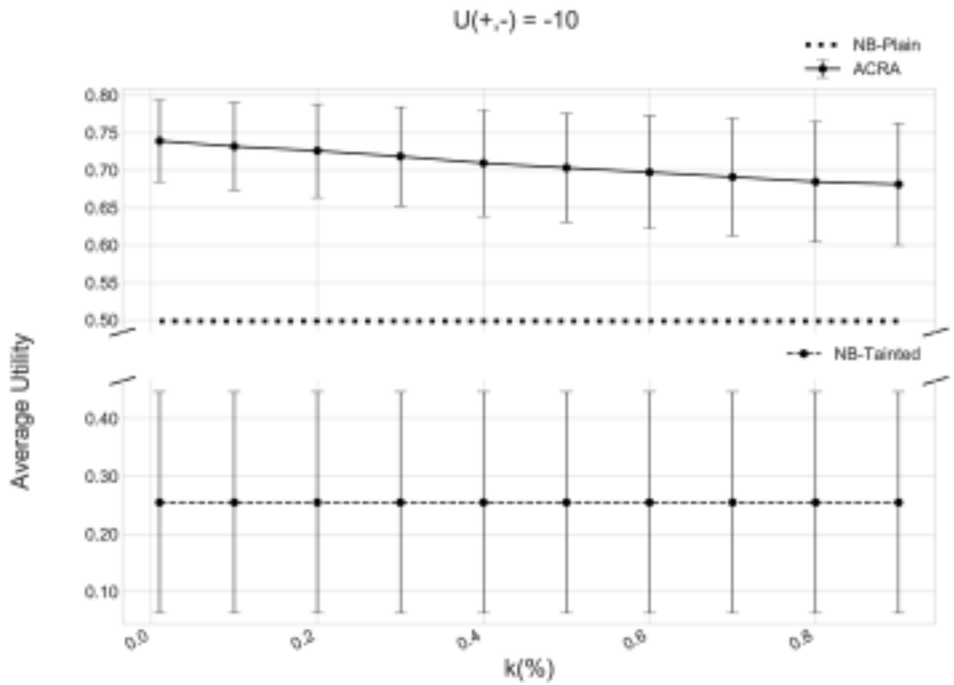
   Solve

$$
\begin{aligned}
c(x') \quad &= \quad \arg\max_{y_C} \Big[u(y_C, +)\widehat{p}_C(+) \sum_{x \in \mathcal{X}'} \widehat{p}_C(a_{x \to x'}|x, +)\widehat{p}_C(x|+) \\
&+ \quad u(y_C, -)\widehat{p}_C(x'|-)\widehat{p}_C(-) \Big].
\end{aligned}
$$

   Output $c(x')$

ACRA. Spam detection





ACRA. Computational enhancements

Main optimization problem

$$\begin{aligned} c(x') &= \arg\max_{y_C} \Bigg[u(y_C,+)\widehat{p}_C(+) \sum_{x\in\mathcal{X}'} \widehat{p}_C(a_{x\to x'}|x,+)\widehat{p}_C(x|+) \\ &+ u(y_C,-)\widehat{p}_C(x'|-)\widehat{p}_C(-) \Bigg]. \end{aligned}$$

Equivalent to $c(x') = +$ if and only if $\sum_{x\in\mathcal{X}'} p_C(a_{x\to x'}|x,+)p_C(x|+) > t$

$$t = \frac{\Big[u_C(-,-) - u_C(+,-)\Big] p_C(x'|-)p_C(-)}{\Big[u_C(+,+) - u_C(-,+)\Big] p_C(+)}$$

MC estimate + importance sampling. In addition, sequentially decide

$$I = \frac{1}{N}\sum_n p_C(a_{x_n\to x'}|x_n,+)I(x_n\in\mathcal{X}') > t$$

ACRA. Computational enhancements

Estimate $p_C(a_{x \to x'} | x, +)$ using small MC size

$$\widehat{p}_C(a_{x \to x'} \mid x, +) = \frac{\#\{a_k^* = a_{x \to x'}\} + 1}{K + |(\mathcal{A}(x))|}$$

Regression Metamodel

Parallel processing

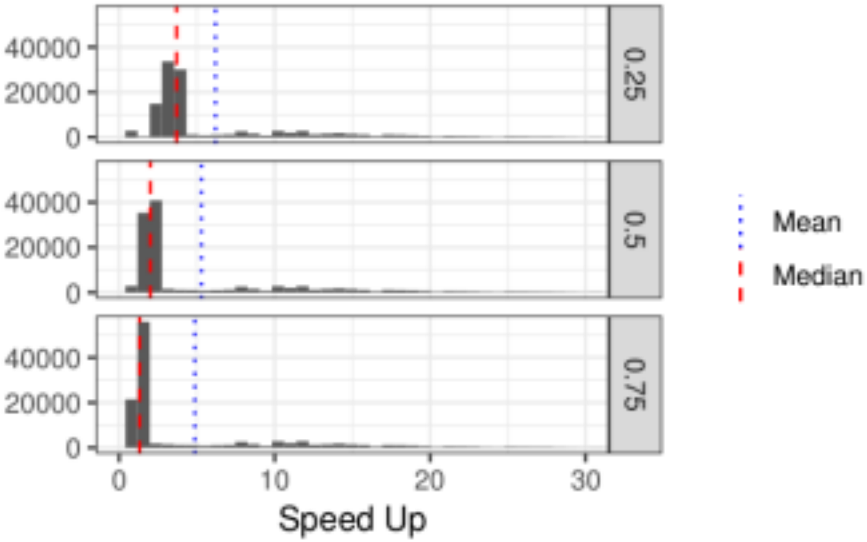
ACRA. Computational enhancements

	Size	Accuracy	FPR	FNR
ACRA	1.00	0.919 ± 0.010	0.0187 ± 0.0076	0.177 ± 0.022
MC ACRA	0.75	0.912 ± 0.012	0.0320 ± 0.0091	0.174 ± 0.023
MC ACRA	0.50	0.905 ± 0.016	0.0270 ± 0.0086	0.199 ± 0.032
MC ACRA	0.25	0.885 ± 0.029	0.0209 ± 0.0072	0.260 ± 0.067
NB-Tainted	-	0.761 ± 0.101	0.68 ± 0.10	0.50 ± 0.25

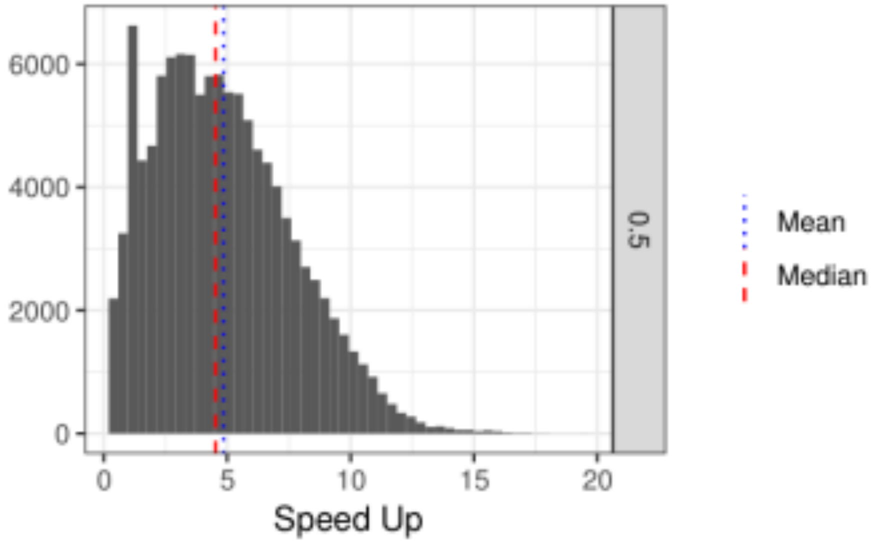
	Dataset	Accuracy	FPR	FNR
MC ACRA	UCI	0.904 ± 0.012	0.0369 ± 0.0070	0.187 ± 0.023
NB-Tainted	UCI	0.724 ± 0.088	0.0656 ± 0.0079	0.601 ± 0.022
MC ACRA	Enron-Spam	0.824 ± 0.017	0.132 ± 0.012	0.305 ± 0.073
NB-Tainted	Enron-Spam	0.534 ± 0.011	0.283 ± 0.013	1.00 ± 0.00
MC ACRA	Ling-Spam	0.958 ± 0.008	0.0390 ± 0.0011	0.057 ± 0.030
NB-Tainted	Ling-Spam	0.800 ± 0.016	0.0400 ± 0.0013	1.00 ± 0.00

Table 3: Comparison between MC ACRA with size 0.5 and NB under 2-GWI attacks.

ACRA. Computational enhancements



MC approximation



Parallelization strategy

Size	Mean	Median
0.25	6.20	3.69
0.50	5.30	2.00
0.75	4.86	1.31

Table 2: Mean and median speed ups.

Outline

- Bayesian methods: A brief reminder
- Bayesian methods for ML: early approaches
- Bayes for BD: The challenges
- Bayes for BD: Some partial solutions
- Adversarial Machine Learning
- **Discussion and challenges**

Discussion

Review of Bayesian methods

Challenges for Bayes and Big Data

- Computational bottlenecks per iteration
  - Evaluating the likelihood
  - Visiting all parameters
  - Visiting all data
- Slow mixing rate

Potential advantages

- Coherent framework, embedding decision making
- Uncertainty duly apportioned
- Forecasting
- Robustness against attacks
- Interpretability
- Fasibility….

Bayesian Nonparametrics

Research agenda

- Generic methods
  - Freeze some parameters at MLE/VB and do full Bayes over the others (Gallego et al, 2018)
  - Parallelization
- Dynamic problems (Naveiro et al, 2018; Berry, West, 2018)
- Decision support
- Probabilistic programming. STAN, AVDI,
- Priors
- Robustness
- Small and Big Data
- Big n, big p
- Decision making

Research agenda

- Multiple attackers, Multiple defenders
- Competition and cooperation
- Attacker models
- Discriminative models
- Generic approach: Point estimation, Interval estimation, Hypothesis testing, Forecasting, Classification
- Multiagent reinforcement learning
- Efficient computational schemes
- Computational environment

- Fake news
- Malware detection (Redondo et al, 2018)

Some refs

- Statistics in the big data era: Failures of the machine. Dunson
- http://bayesiandeeplearning.org/ Neurips 2018
- https://arxiv.org/pdf/1601.00670.pdf Variational inference

- https://arxiv.org/abs/1812.00071 Speeding MCMC
- https://arxiv.org/abs/1809.01560 RL under threats
- https://arxiv.org/abs/1802.07513 Adversarial classification
- https://arxiv.org/abs/1802.06678 Large Scale monitoring

Some events

- BISP11 June 12-14 Madrid

https://www.methaodos.org/congresos-methaodos/index.php/bisp11/bisp11

- GDRR @ SAMSI 19-20 (+Deep learning)

https://www.samsi.info/programs-and-activities/year-long-research-programs/2019-2020-program-on-games-decisions-risk-and-reliability/

Thanks!!!

Collabs welcome

david.rios@icmat.es

SPOR DataLab https://www.icmat.es/spor/
It’s a risky life @YouTube
Aisoy Robotics https://www.aisoy.com
CYBECO https://www.cybeco.eu/