F - Mémoire Flash USB

Matériel requis : • Windows XP ou supérieur - Mac OS X ou supérieur • Une prise USB disponible sur l'ordinateur. **- Concernant l'installation :** Aucune installation de pilote n'est requise. Branchez la mémoire au port USB de votre ordinateur pour qu'elle soit prête à l'emploi. **- Caractéristiques Techniques :** Branchement sur un port USB 1.0 et 1.1 : taux de transfert théorique maximum de 1 Mo/s – branchement sur un port USB 2.0 : taux de transfert théorique maximum de 60 Mo/s – Témoin lumineux d'activité (sur certains modèles). **- Utilisation :** > Branchez votre support de stockage amovible sur une prise USB de votre ordinateur. Utilisez un câble prolongateur USB si l'emplacement est peu accessible. > La clé est alors automatiquement reconnue et un nouveau disque apparaît. Sur certains modèles, plusieurs disques peuvent apparaître au branchement d'une seule clé USB. > Pour copier ou lire des informations sur la clé USB, procédez comme avec n'importe quel disque dur. Si un lecteur de CD-Rom apparaît au branchement de la clé, les documents qu'il contient ne pourront être ni modifiés, ni supprimés. **- Déconnexion :** > Sous MacOS : Éjectez-la comme un CD-Rom. Une fois le disque disparu, ôtez la clé de la prise USB. > Sous Windows : Sur la barre d'état en bas de l'écran à droite, cliquez sur l'icône où figure une flèche verte. Sélectionnez alors l'option «retirer le périphérique de stockage de masse USB». Si vous voyez apparaître des instructions complémentaires, suivez-les jusqu'à ce que Windows vous indique «le matériel peut être retiré en toute sécurité» (Le texte peut être différent selon la version de Windows). Vous pouvez alors débrancher la clé de la prise USB.

ATTENTION : ***Cette clé USB a été conçue pour être utilisée dans le cadre de votre activité professionnelle. L'usage du support à des fins de copie privée pour la reproduction d'œuvres littéraires et artistiques sur le territoire français doit être signalé à votre entreprise et est assujetti à l'indemnisation pour copie privée. Pour plus d'information sur cette indemnisation, veuillez consulter le site de Copie France (http://www.sorecop.fr/lv_particulier.htm).***

UK - USB Flash Memory

System requirements: • Operating systems:- Windows XP or higher – Mac OS X or higher. • One available USB port on the computer. **- Installation instructions:** It is not necessary to install a driver. Plug the memory into the USB port on your computer and it will be ready for immediate use. **- Technical Features:** When plugged in an USB 1.0 or 1.1 port: theoretical maximum transfer rate 1MB/s – When plugged in an USB 2.0 port: theoretical maximum transfer rate 60 MB/s • Device activity light (on some models) **- Using your Flash Memory :** > To use it, plug it into a USB port on your computer. You may use a USB extension cable if the location is difficult to access. > The key will then be detected automatically and a new disk will appear. With some models, several disks may appear. > To copy or read the information on the USB key, proceed as with any hard drive. If a CD-Rom drive appeared, you won't be able to modify or delete the files it contains **- Disconnecting your Flash Memory :** > Under MacOS: eject it in the same way as a CD-ROM. Once the disk has disappeared, remove the key from the USB port. > Under Windows: on the status bar on the bottom right of the screen, click on the icon where a green arrow appears. Then choose the option "eject the USB mass storage support". If you see additional instructions, follow them until Windows informs you that "it is safe to eject the device". (Note that the text may vary according to the version of Windows). You may then unplug the key from the USB port.

D - USB Flash-Speicher

Systemvoraussetzungen : • Windows XP oder höher- Mac OS X oder höher. • Ein USB-Anschluss am Computer. **- Hinweise zur Installation:** • Es ist keine Installation eines Drivers erforderlich. Verbinden Sie den Speicher mit dem USB-Anschluss Ihres Computers, damit er laufbereit ist. **- Technische Merkmale:** Verbindung USB 1.0 und 1.1: maximale Übertragungsgeschwindigkeit von 1 MB/s - Verbindung 2.0: maximale Übertragungsgeschwindigkeit 60 MB/s • Leuchtanzeige für den Betrieb (auf einigen Modellen) **- Verwendung Ihres Flash-Speichers :** Schließen Sie Ihren USB Flash-Speicher (USB-Stick) an einen USB-Anschluss Ihres Computers an. > Verwenden Sie ein USB-Verlängerungskabel, wenn er sich an einer unzugänglichen Stelle befindet. > Der Stick wird nun automatisch erkannt, und es erscheint ein neues Gerätsympol. Bei einigen Modellen können mehrere Symbole, durch einen USB Stick erscheinen. > Um Informationen auf den USB-Stick zu kopieren oder daraus zu lesen, verfahren Sie wie mit jeder anderen Festplatte. Wenn ein CD-Rom Laufwerk erscheint, können die beinhaltenden Dokumenten nicht modifiziert oder gelöscht werden**- Abschalten Ihres Flash-Speichers :** > Unter MacOS : Werfen Sie ihn wie eine CD-Rom aus: nachdem die Platte verschwunden ist, entfernen Sie den Stick aus dem USB-Anschluss. > Unter Windows : Klicken Sie auf der Statusleiste rechts unten auf dem Bildschirm auf das Symbol, in dem sich ein grüner Pfeil befindet. Wählen Sie dann die Option „Peripheriegerät USB-Massenspeicher entfernen". Wenn zusätzliche Anweisungen erscheinen, befolgen Sie sie, bis Windows anzeigt, dass „das Gerät sicher entfernt werden kann" (Berücksichtigen Sie, dass der Text je nach der Windows-Version verschieden sein kann). Sie können nun den Stick vom USB-Anschluss entfernen.

ES – Memoria Flash USB

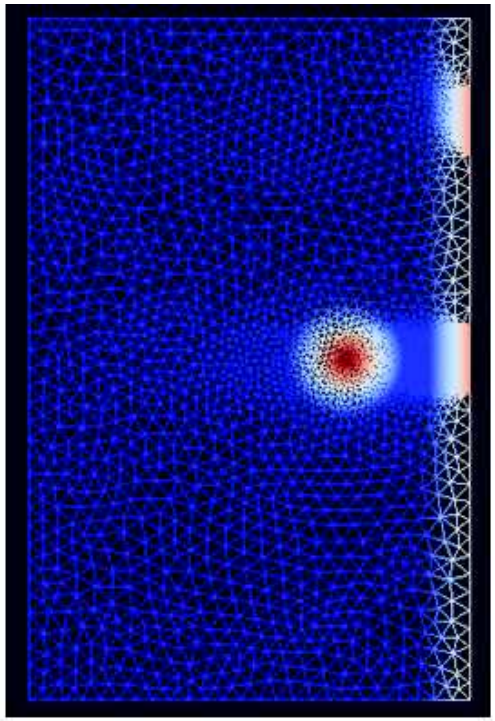
Sistemas necesarios: Windows XP en adelante – Mac OS X en adelante • Un puerto USB disponible en el ordenador. **- Instrucciones de instalación:** no es necesaria la instalación de controladores de dispositivos (drivers). Enchufar la memoria en el puerto USB de su ordenador y estará lista para ser utilizada. **- Aspectos técnicos:** Conexión en un puerto USB 1.0 y 1.1: Tasa de transferencia teórica máxima de 1 MB/s. Conexión en un puerto USB 2.0: transferencia teórica máxima de 60MB/s • Señal luminosa de actividad (sobre algunos modelos) **- Modo de uso:** > Enchufar la memoria flash USB en el puerto USB de su ordenador. Utilice un cable de extensión si el lugar es de difícil acceso. > La memoria USB será automáticamente reconocida y un nuevo disco aparecerá. Con algunos modelos, varios discos pueden aparecer al conectar una sola unidad. > Para copiar o leer información de la memoria USB, procede como con cualquier disco duro. Si un lector de CD-ROM aparece al enchufar la memoria, los documentos que contiene no podrán ser ni modificados ni borrados. **- Desconexión:** > Mac: sacarla de la misma manera que un CD-ROM. Una vez desaparecido el disco, retire la memoria del puerto USB. > Windows: En la barra de estado al pie de la pantalla sobre la derecha, pinchar en el ícono donde aparece la flecha verde. Luego elija la opción "retirar el dispositivo en toda seguridad". si observa instrucciones adicionales, prosiga con las mismas hasta que Windows le confirme que "es seguro retirar el dispositivo" (note que el texto puede variar según la versión de Windows). Luego puede desenchufar la memoria del puerto USB.

I - Memoria Flash USB

Materiale requisito : • Windows XP o superiore - Mac OS X o superiore • Una presa USB disponibile sul computer. **- A proposito dell'istallazione :** Nessuna istallazione è necessaria. Collegate la memoria alla porta USB del Vostro computer affinché sia pronta per l'uso. – **Caratteristiche tecniche :** Collegamento su una porta USB 1.0 e 1.1 : tasso di trasferimento teorico massimo di 1 Mo/s – Collegamento su una porta USB 2.0 : tasso di trasferimento massimo di 60 Mo/s • Testimone luminoso di attività (su certi modelli). - Utilizzo : > Collegate il Vostro supporto di stoccaggio amovibile su una presa USB del Vostro computer. Utilizzare un cavo USB se la posizione è poco accessibile. > La chiavetta è dunque automaticamente riconosciuta ed un nuovo disco appare. > Su certi modelli, alcuni dischi possono possono apparire al collegamento di una sola chiavetta USB. > Per copiare o leggere informazioni sulla chiavetta USB procedete come con qualsiasi disco rigido. Se un lettore di CD-Rom appare al collegamento della chiavetta, I documenti che contiene non potranno essere ne modificati, ne cancellati. **- Disconnessione :** > Su Mac OS : Toglietela come un CD-Rom. Una volta chiuso il CD-Rom, togliete la chiavetta della presa USB. > Su Windows : Sulla sbarra di stato in basso dello schermo a destra, cliccate sull' icona in cui figura una freccia verde. Selezionate allora l'opzione "togliere l'unità periferica di stoccaggio di massa USB". Se vedete apparire istruzioni complementari, seguitele fino a che Windows vi indica "il supporto puo' essere tolto in sicurezza" (Il testo puo' essere diverso secondo la versione di Windows). Potete allora scollegare la chiavetta della presa USB.

Example 1 - Experimental design for tsunami simulation

Tsunami Wave Basin at Oregon State University



Adaptive mesh grid of the VOLNA solver
[Dutykh et al., 2011]

Joint work with: Emile Contal*, Frédéric Dias, Themis Stefanakis*, Costas Synolakis

Experimental design - Goals and constraints

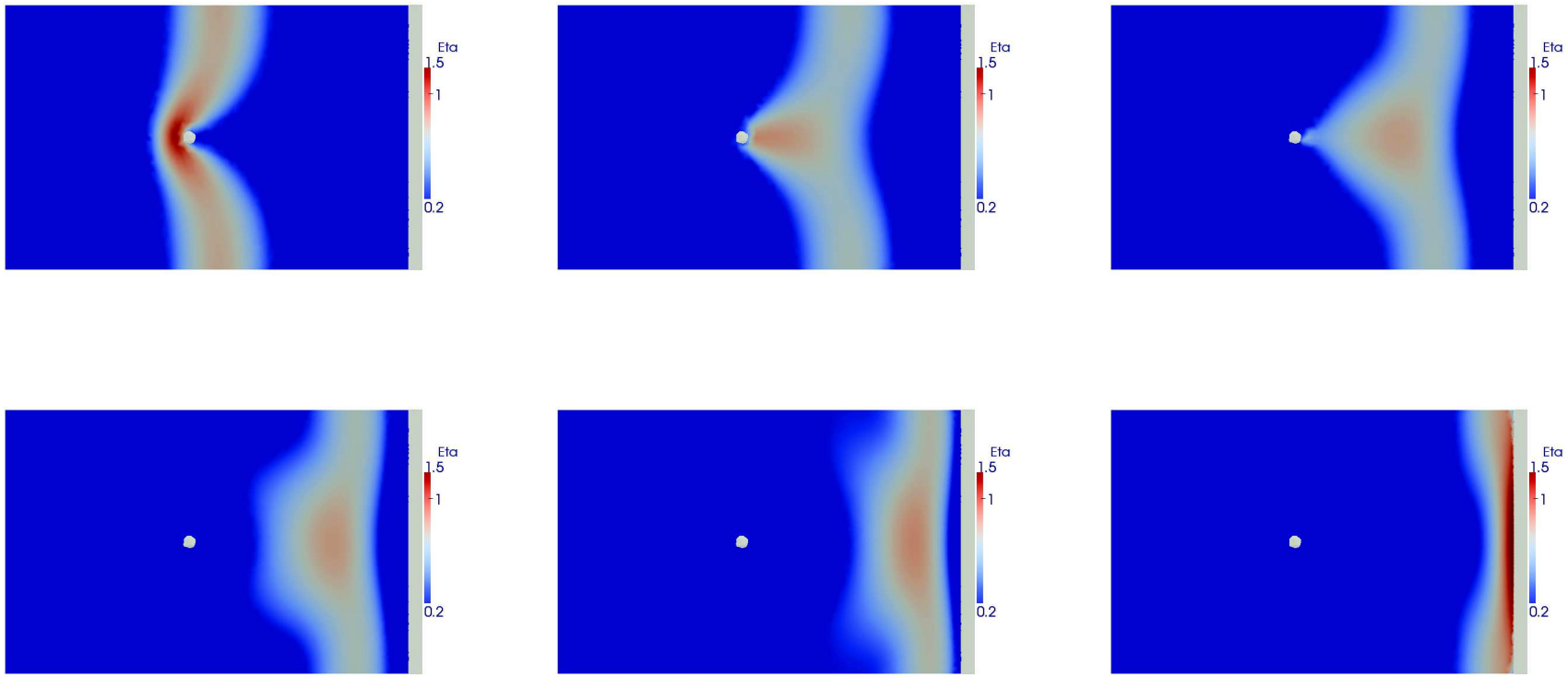
Possible goals

- Analysis/Control of the system
- Inverse problem and complex system design
- Optimization of the output $\Leftarrow$

Constraints

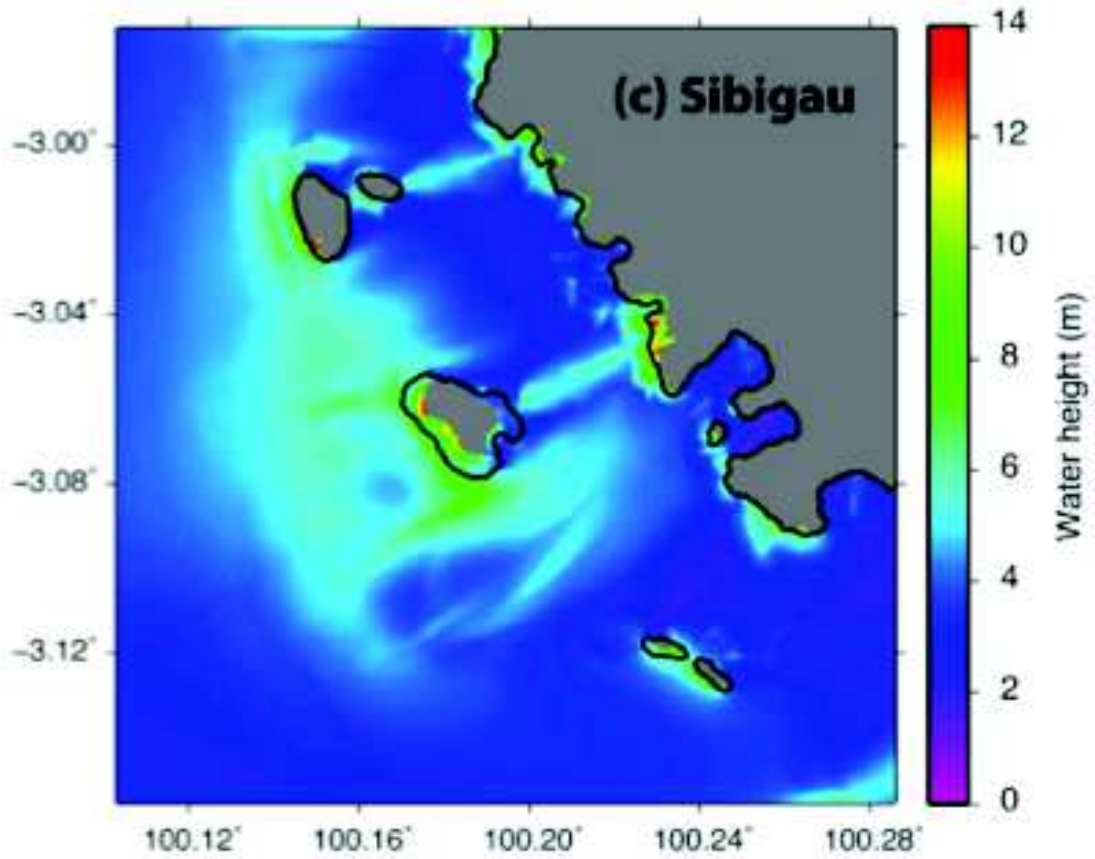
- Many input variables (i.e. parameters that drive the simulation)
- High cost of one experiment (gives one data point)
- Overall budget constraints: time and resources

Tsunamis amplification phenomena



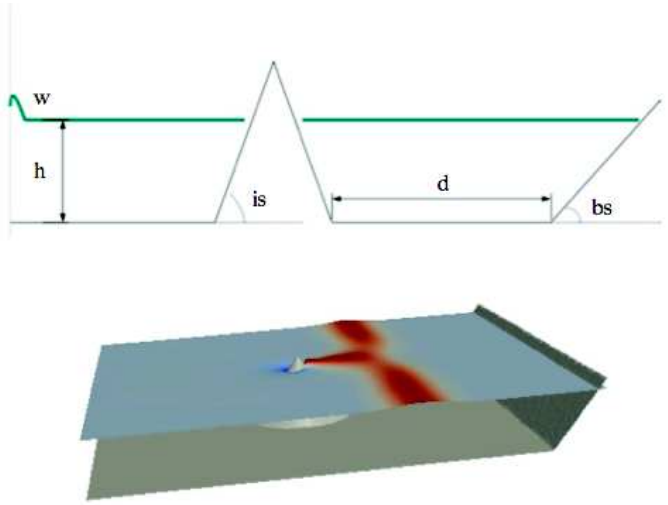
Numerical simulations of a tsunami amplification generated by a conical island

Real Scenario



2010 Sumatra tsunami and the Mentawai Islands [Hill *et al.*,, 2012]

Tsunami modeling example - Simulation setup



Five parameters modelling the geometry stored in a vector x

Exploration of the simulation output

- $d = 5$ parameters
- Each simulation takes 2 hours of computation
- A regular grid with 10 values per parameters needs 10^5 points
- A naive approach would take 23 years of computation

Problem Statement

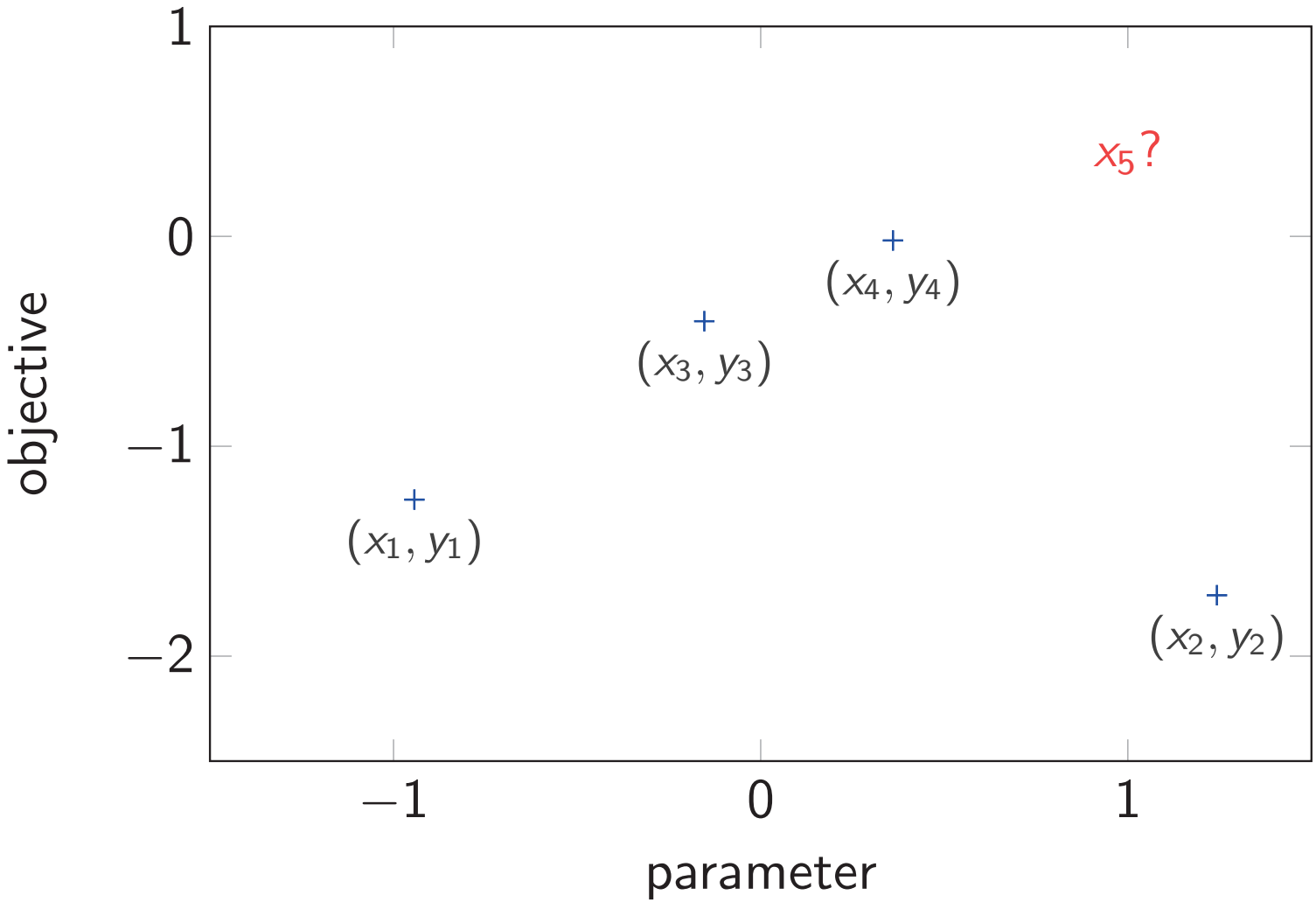
Sequential Optimization

- d real parameters denoted by d-dimensional vectors $x \in \mathcal{X}$
- $\mathcal{X} \subseteq \mathbb{R}^d$ compact and convex
- Unknown objective function $f(x) \in \mathbb{R}$ for all $x \in \mathcal{X}$
- Noisy measurement $y = f(x) + \epsilon$, where $\epsilon \overset{\text{iid}}{\sim} \mathcal{N}(0, \eta^2)$
- Find the parameters x maximizing $f(x)$

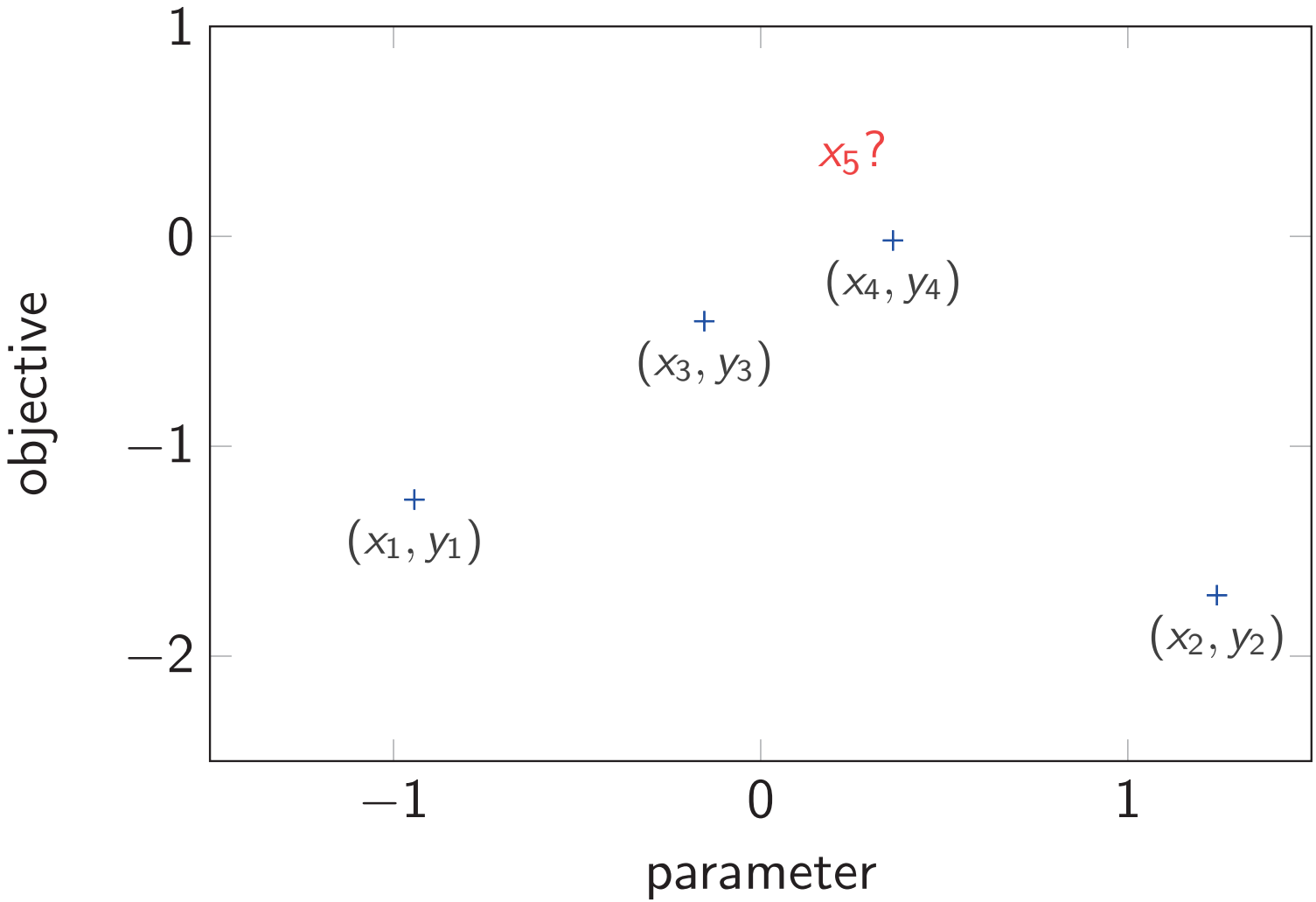
Goal

- Denote by f the unknown function relating topographic parameters x to **runup amplification** y
- Consider access to $K \geq 2$ processors with time horizon $T \geq 2$
- Find the maximal value of f with T batches of size K

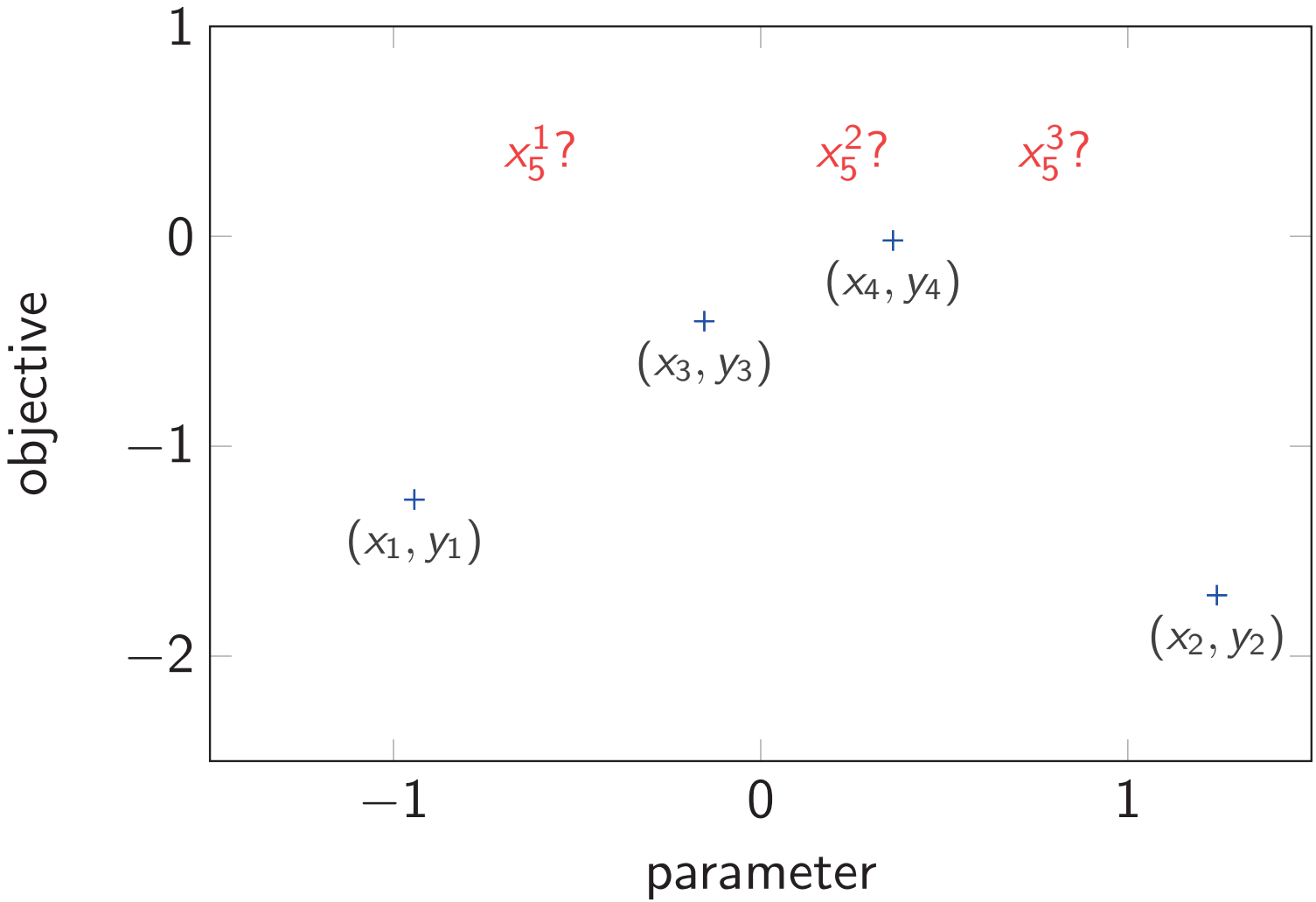
Sequential Optimization



Sequential Optimization



Batch-Sequential Optimization



Gaussian Processes Framework

Definition

$f \sim \mathcal{GP}(m, k)$, with mean function $m : \mathcal{X} \to \mathbb{R}$ and covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$, when for all $x_1, \ldots, x_n$,

$$\big(f(x_1), \ldots, f(x_n)\big) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C}) \ ,$$

$$\text{with } \boldsymbol{\mu}[x_i] = m(x_i) \text{ and } \mathbf{C}[x_i, x_j] = k(x_i, x_j) \ .$$

Probabilistic smoothness assumption

- Nearby location are highly correlated
- Large local variation have low probability

Typical Kernels

- Polynomial with degree $\alpha \in \mathbb{N}$: for $c \in \mathbb{R}$

$$\forall x_1, x_2 \ , \quad k(x_1, x_2) = (x_1^T x_2 + c)^\alpha$$

- Radial Basis Function with length-scale parameter $b > 0$:

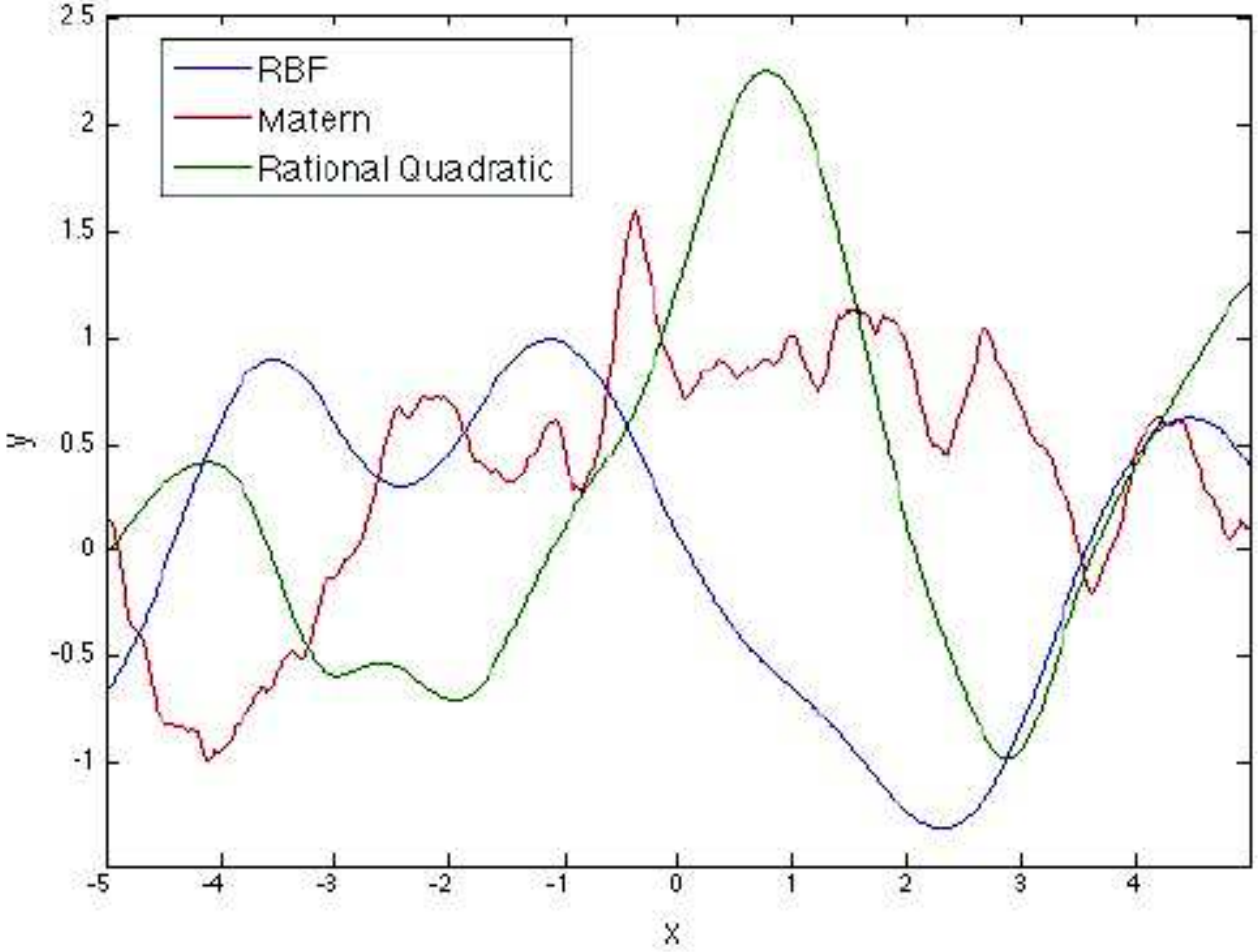
$$\forall x_1, x_2 \ , \quad k(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2b^2}\right)$$

- Matérn with length-scale $b > 0$ and order ν:

$$\forall x_1, x_2 \ , \quad k(x_1, x_2) = \frac{2^{1-\nu}}{\Gamma(\nu)} \Phi_\nu\left(\frac{\sqrt{2\nu}\|x_1 - x_2\|}{b}\right)$$

where $\Phi_\nu(z) = z^\nu K_\nu(z)$ and K_ν is a Bessel function of the second kind with order ν.

Gaussian Processes Examples



1D Gaussian Processes with different covariance functions

Gaussian Process Interpolation

Bayesian Inference [Rasmussen and Williams, 2006]

At iteration t, with observations $\mathbf{Y}_t$ for the query points $\mathbf{X}_t$, the posterior mean and variances are given at all point x in the search space by:

$$\mu_t(x) = \mathbf{k}_t(x)^\top \mathbf{C}_t^{-1} \mathbf{Y}_t \tag{1}$$

$$\sigma_t^2(x) = k(x,x) - \mathbf{k}_t(x)^\top \mathbf{C}_t^{-1} \mathbf{k}_t(x) \ , \tag{2}$$

where $\mathbf{C}_t = \mathbf{K}_t + \eta^2 \mathbf{I}$, and $\mathbf{k}_t(x) = [k(x_\tau, x)]_{1 \leq \tau \leq t}$, and $\mathbf{K}_t = [k(x_\tau, x_{\tau'})]_{1 \leq \tau, \tau' \leq t}$.

Interpretation

- posterior mean μ_t: prediction
- posterior variance σ_t^2: uncertainty

Upper and Lower Confidence Bounds

Definition

Fix $0 < \delta < 1$, and consider upper/lower confidence bounds on f:

$$f_t^+(x) = \mu_t(x) + \sqrt{\beta_t(\delta)\sigma_t^2(x)}$$

$$f_t^-(x) = \mu_t(x) - \sqrt{\beta_t(\delta)\sigma_t^2(x)}$$

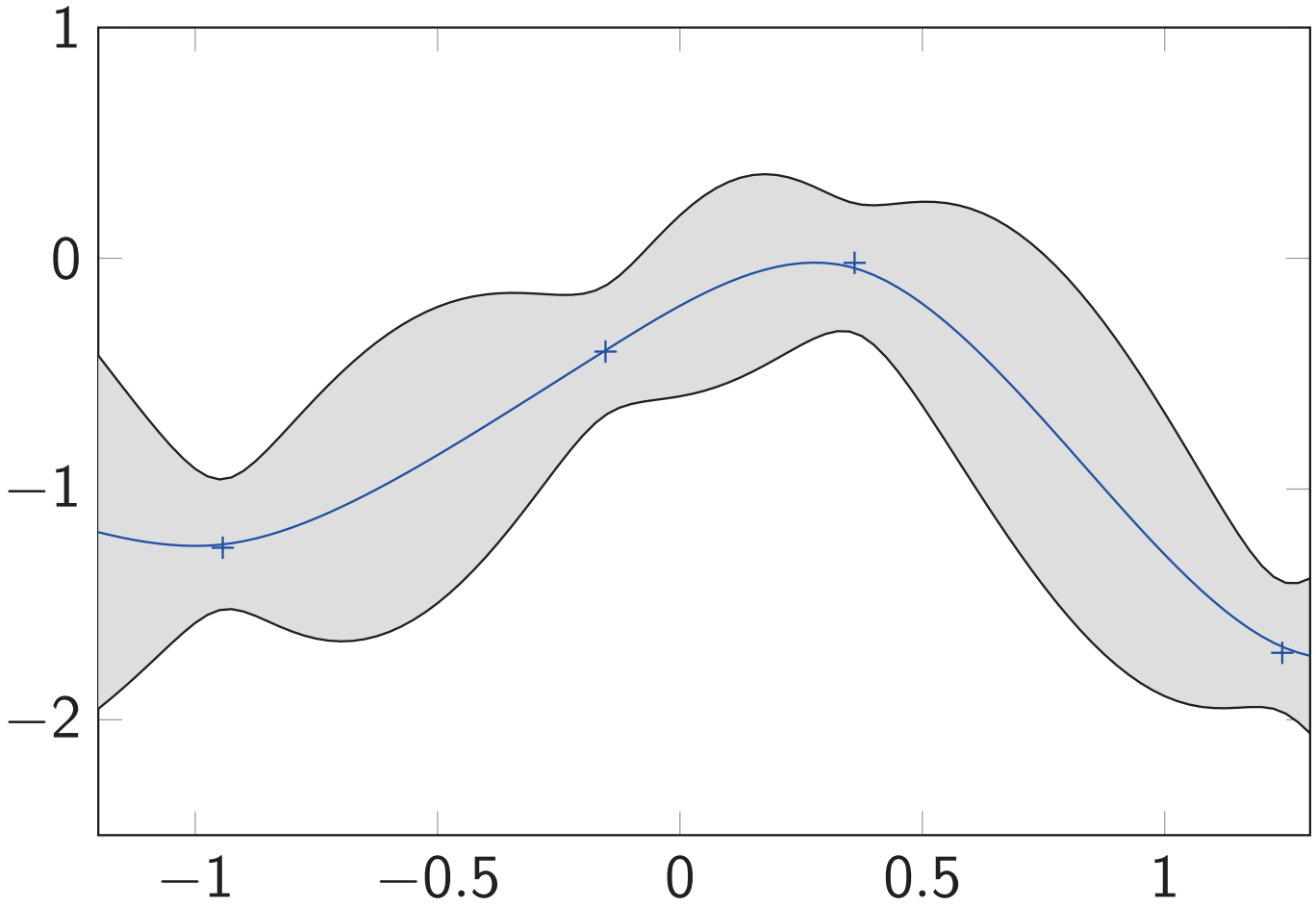
with $\beta_t(\delta) = \mathcal{O}\big(\log(t/\delta)\big)$ defined in [Srinivas, 2012].

Property

We have with probability at least $(1-\delta)$*:* $\forall x \in \mathcal{X}$, $\forall t \geq 1$,

$$f(x) \in \left[f_t^-(x), f_t^+(x)\right] \ .$$

Key step - Confidence bands based on gaussian processes



After bayesian inference obtained with four points on a 1D toy example

Relevant Region

Definition

The Relevant Region $\mathfrak{R}_t$ is defined by,

$$y_t^\bullet = \max_{x \in \mathcal{X}} f_t^-(x) \ ,$$

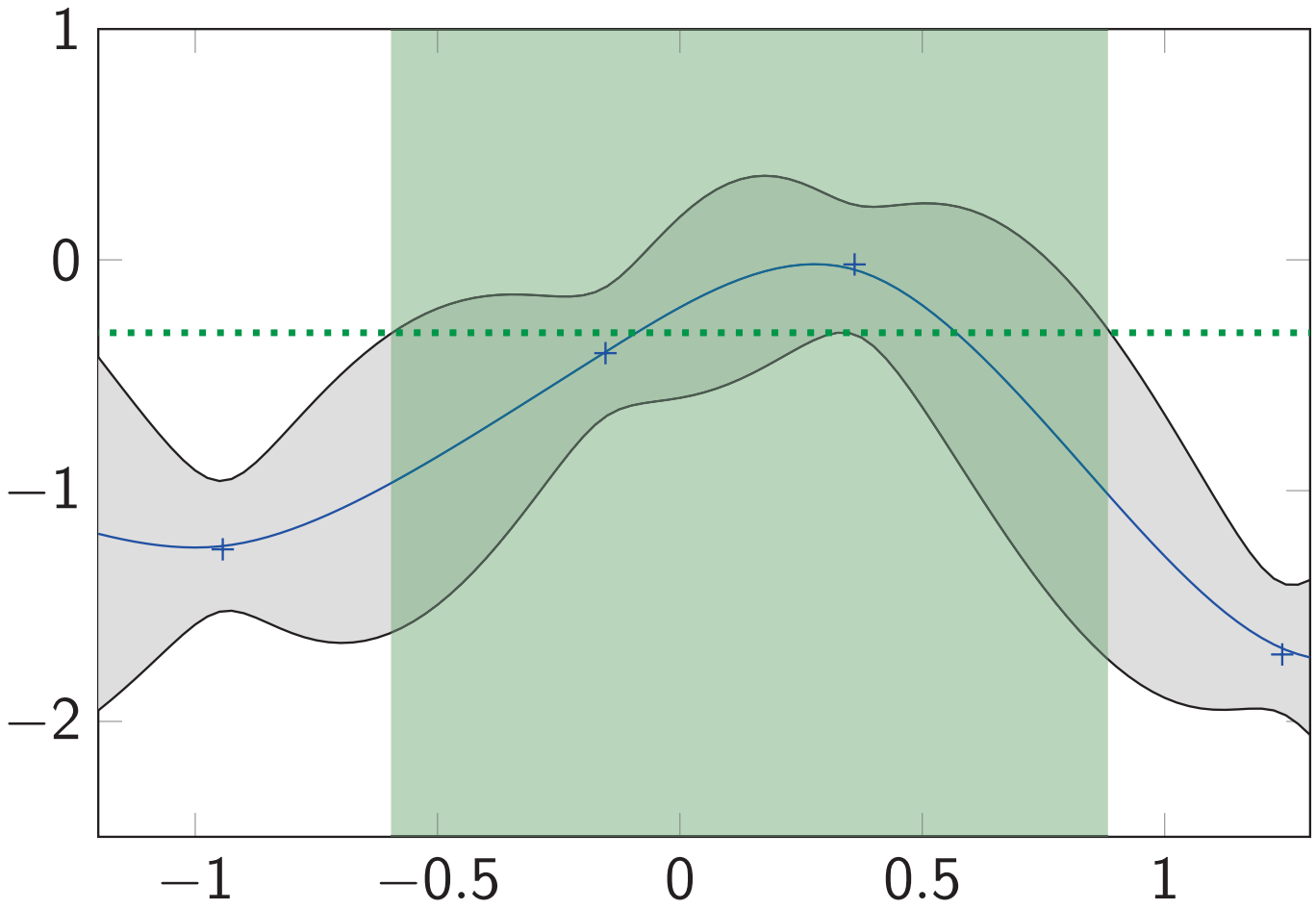
$$\mathfrak{R}_t = \left\{ x \in \mathcal{X} \mid f_t^+(x) \geq y_t^\bullet \right\} \ .$$

Property

We have, with probability at least $(1 - \delta)$*:*

$$x^\star \in \mathfrak{R}_t \ .$$

Relevant Region



Based on the level set corresponding to the max of the lower bound

Upper Confidence Bound and Pure Exploration

UCB policy: $k = 1$

Achieves tradeoff between exploitation *vs.* exploration (μ_t *vs.* σ_t^2):

$$x_{t+1}^1 \leftarrow \underset{x \in \mathfrak{R}_t^+}{\operatorname{argmax}} f_t^+(x)$$

where $\mathfrak{R}_t^+ = \left\{ x \in \mathcal{X} \mid \mu_t(x) + 2\sqrt{\beta_t(\delta)\sigma_t^2(x)} \geq y_t^\bullet \right\}$

PE policy: $k = 2, \ldots, K$

Selects the most uncertain points inside the Relevant Region:

$$x_{t+1}^k \leftarrow \underset{x \in \mathfrak{R}_t^+}{\operatorname{argmax}} \sigma_t^{(k)}(x)\,, \text{ for } 2 \leq k \leq K\,,$$

where $\sigma_t^{(k)}(x)$ is the updated uncertainty using $x_{t+1}^1, \ldots, x_{t+1}^{k-1}$

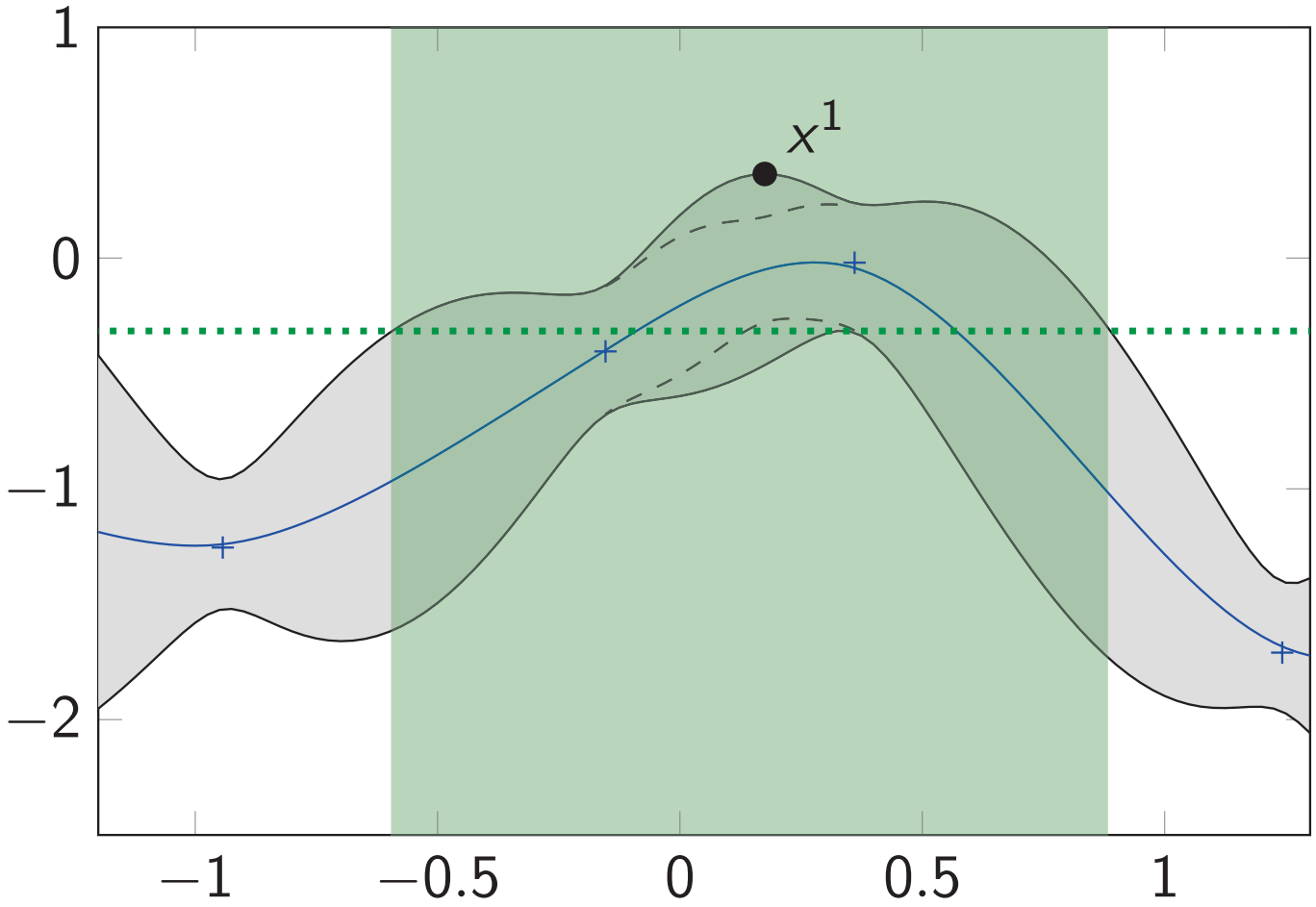
GP-UCB-PE pseudocode

Algorithm 1: GP-UCB-PE

for $t = 1, 2, \ldots$ **do**

- Compute μ_t and σ_t^2 with Bayesian inference on $y_1^1, \ldots, y_{t-1}^K$
- Compute $\mathfrak{R}_t^+$
- $x_{t+1}^1 \leftarrow \text{argmax}_{x \in \mathfrak{R}_t^+} f_t^+(x)$
- **for** $k = 2, \ldots, K$ **do**
  - Update $\sigma_t^{(k)}$
  - $x_{t+1}^k \leftarrow \text{argmax}_{x \in \mathfrak{R}_t^+} \sigma_t^{(k)}(x)$
- Query $x_{t+1}^1, \ldots, x_{t+1}^K$
- Observe $y_{t+1}^1, \ldots, y_{t+1}^K$

The GP-UCB-PE algorithm [Contal *et al.*, 2013]

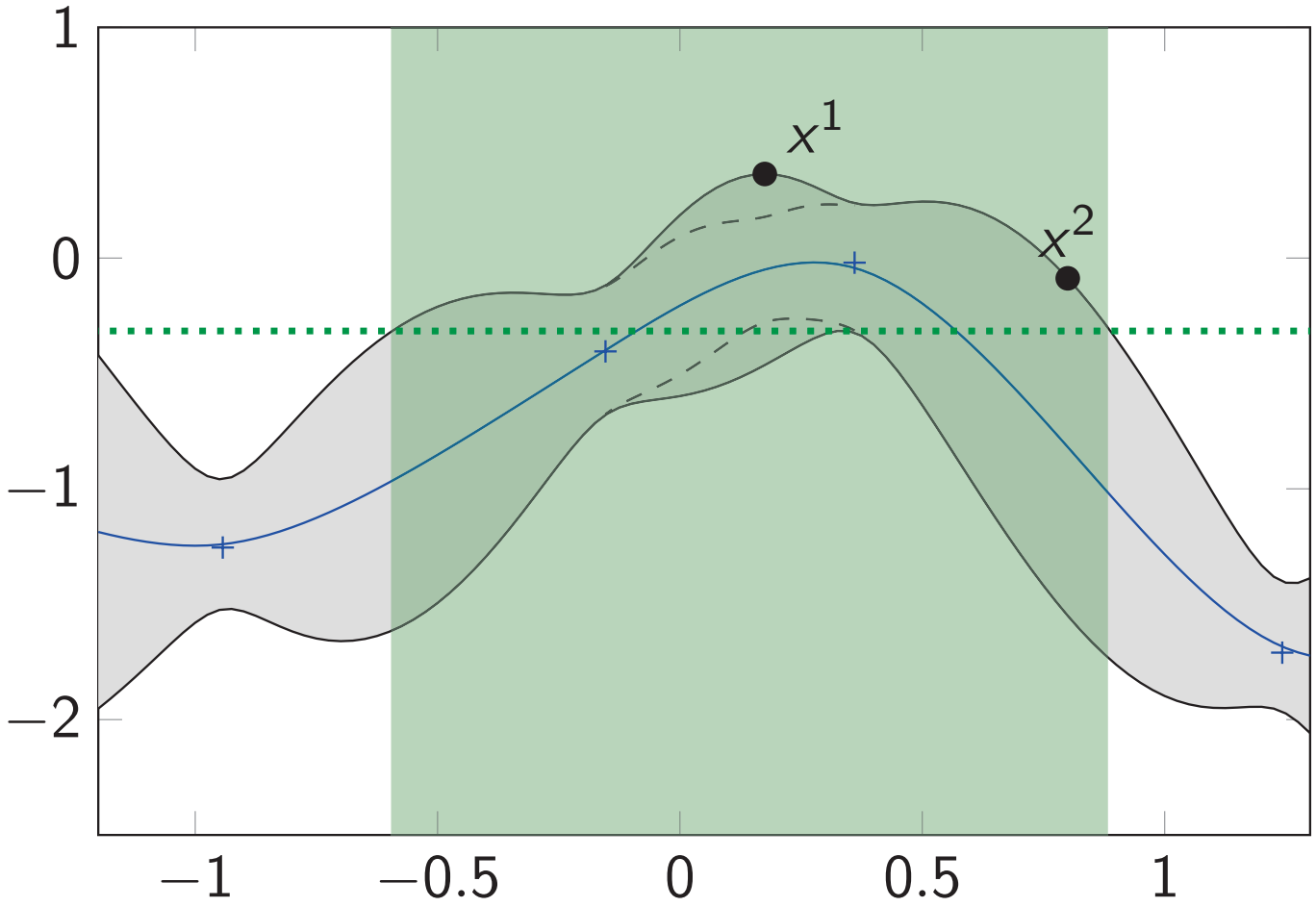


UCB = Upper-Confidence-Bound $\Rightarrow$ Exploitation (1 point out of K)

PE = Pure exploration $\Rightarrow$ Exploration

($K-1$ remaining points in the batch)

The GP-UCB-PE algorithm [Contal *et al.*, 2013]



UCB = Upper-Confidence-Bound $\Rightarrow$ Exploitation (1 point out of K)

PE = Pure exploration $\Rightarrow$ Exploration

($K-1$ remaining points in the batch)

Mutual Information – an important concept

Information Gain

The information gain on f at X_T is the mutual information between f and Y_T. For a GP distribution with $\mathbf{K}_T$ the kernel matrix of X_T:

$$I_T(X_T) = \frac{1}{2} \log \det(\mathbf{I} + \eta^{-2}\mathbf{K}_T) \,.$$

We define $\gamma_T = \max_{|X|=T} I_T(X)$ the maximum information gain by a sequence of T queries points.

Empirical Lower Bound

For GPs with bounded variance, we have: [Srinivas et al. 2012]

$$\widehat{\gamma}_T = \sum_{t=1}^{T} \sigma_t^2(x_t) \leq C\gamma_T \text{ where } C = \frac{2}{\log(1+\eta^{-2})}$$

Mutual Information – examples

The parameter γ_T is the maximum mutual information about f obtainable by a sequence of T queries.

- Linear kernel: $\gamma_T = \mathcal{O}(d \log T)$
- RBF kernel: $\gamma_T = \mathcal{O}\big((\log T)^{d+1}\big)$
- Matérn kernel:

$$\gamma_T = \mathcal{O}\big(T^{\alpha} \log T\big) \ ,$$

where

$$\alpha = \frac{d(d+1)}{2\nu + d(d+1)} \leq 1 \ .$$

Regret bound on GP-UCB-PE

General result

Consider $f \sim \mathcal{GP}(0, k)$ with $k(x, x) \leq 1$ for all x, and $x^\star = \text{argmax}_{x \in \mathcal{X}} f(x)$, then we have, with high probability:

$$R_T^K \doteq \sum_{t=1}^{T} \left(f(x^\star) - \max_{1 \leq k \leq K} f(x_t^k) \right) = \mathcal{O}\left(\sqrt{\left(\frac{T}{K} \right) \gamma_{TK} \log T} \right)$$

Specialized results

- Linear kernel: $R_T^K = \mathcal{O}\left(\log(TK) \sqrt{dT/K} \right)$
- RBF kernel: $R_T^K = \mathcal{O}\left(\sqrt{(T/K)(\log(TK))^{d+2}} \right)$
- Matérn kernel: $R_T^K = \mathcal{O}\left(\log(TK) \sqrt{T^{\alpha+1} K^{\alpha-1}} \right)$

Improvement of Batch-Sequential over Sequential

Impact on Regret

Take $K \ll T$, then the improvement of the parallel strategy over the sequential one is $\sqrt{K}$ for R_T^K.

Complexity

Note that $\text{Cost(GP)} = \mathcal{O}(n^2)$ (Osborne, 2010), where n number of candidate evaluation points:

$$\text{Sequential} = n\ \text{Cost}(f) + n\ \text{Cost(GP)}$$

$$\text{Batch-Sequential} = (n/K)\ \text{Cost}(f) + n\ \text{Cost(GP)}$$

For large n, practical approaches are: Lazy Variance Computation, MCMC sampling, random projection...

Two Competitors for Batch-Sequential Strategies

GP-BUCB = GP Batch UCB [Desautels *et al.*, 2012]

- Batch estimation based on updates $\mu_t^k(x)$ of $\mu_t(x)$
- Regret bound with RBF kernel due to initialization:

$$\mathcal{O}\left(\exp\left(\left(\frac{2d}{e}\right)^d\right)\sqrt{\left(\frac{T}{K}\right)\log(TK)}\right)$$

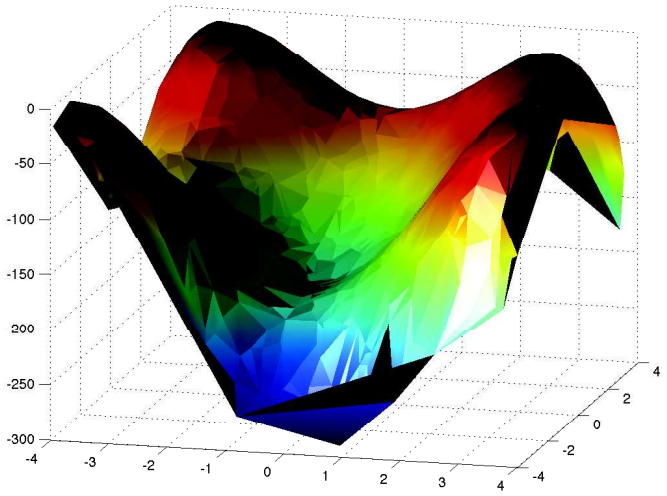
SM-UCB = Simulation Matching with UCB [Azimi *et al.*, 2010]

- Select batch of points that matches expected behavior
- Based on a greedy K-medoid algorithm to screen irrelevant data points
- No regret bound available

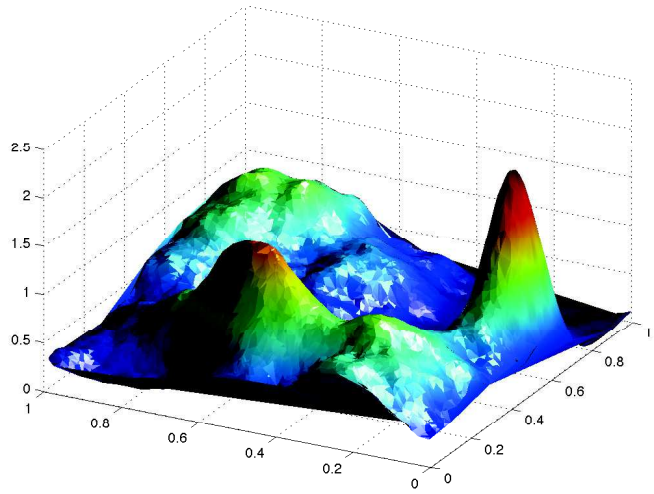
Experiments

Setup

- Competitors: GP-BUCB and SM-UCB
- Assessment: 3 synthetic problems and 3 real applications

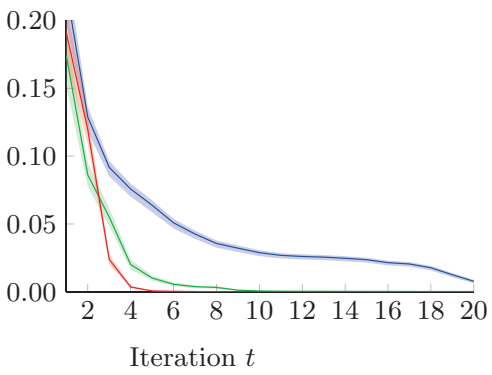


(a) Himmelblau's function

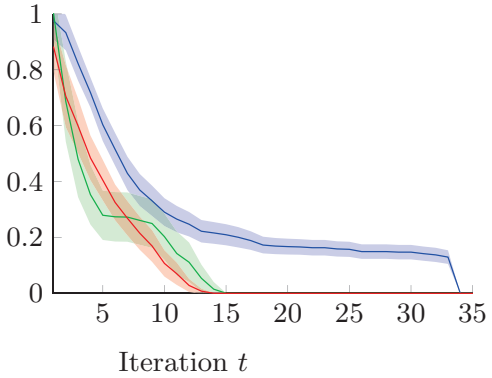


(b) Gaussian Mixture

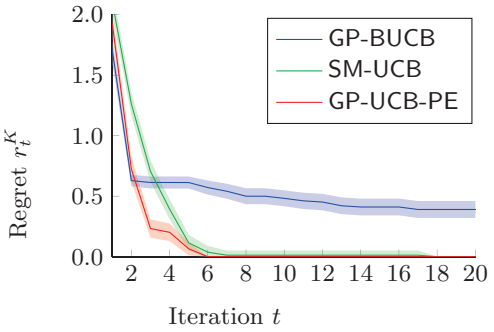
Results: mean instantaneous batch regret and confidence interval over 64 experiments



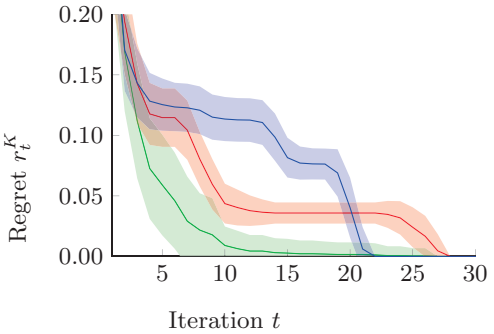
(a) Himmelblau



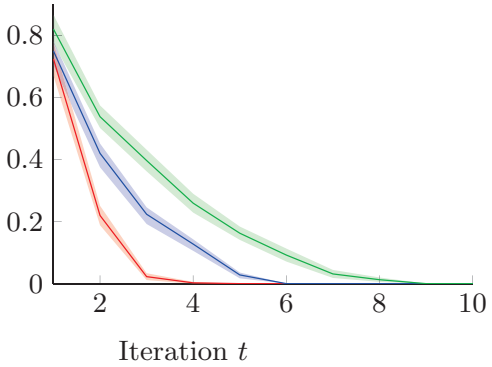
(b) Gaussian mixture



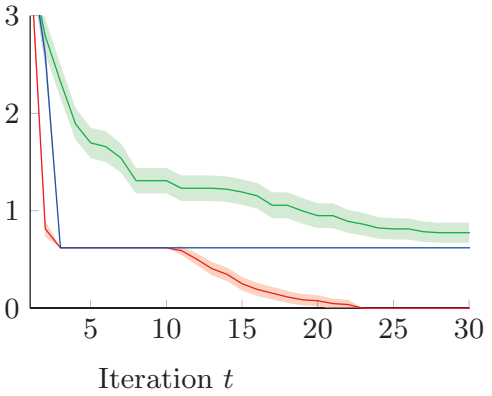
(c) Generated GP



(d) Mackey-Glass

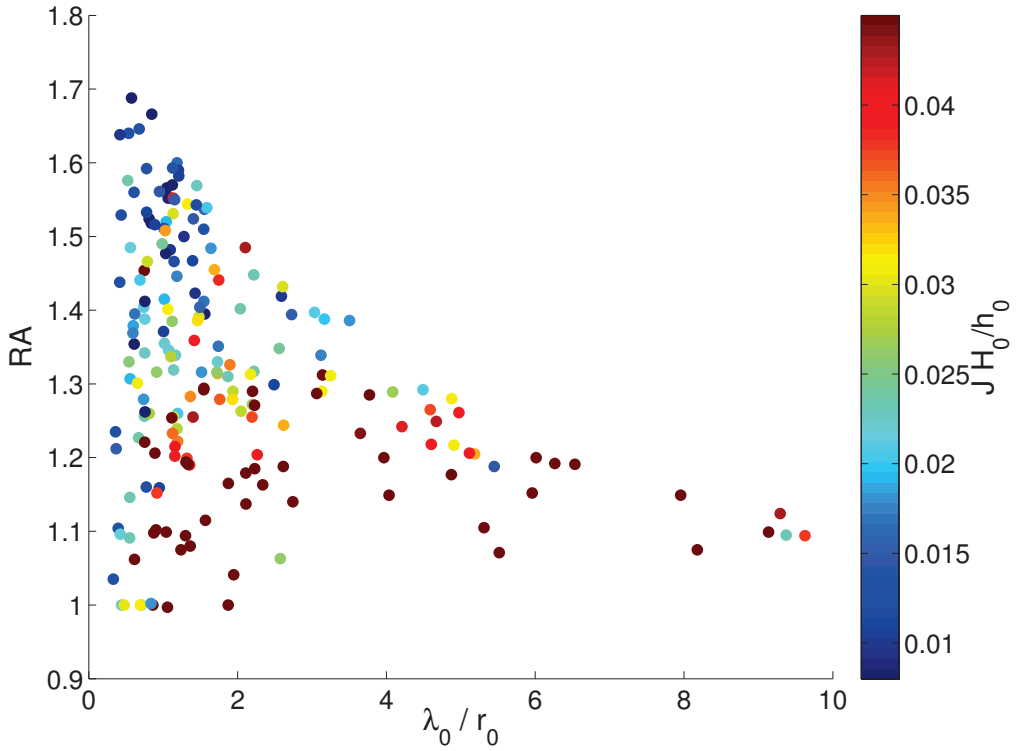


(e) Tsunamis



(f) Abalone

Proof of runup amplification and physical priors



Run-up amplification (RA) as a function of the wavelength to the island radius (at its base) ratio. The color code indicates the surf similarity (Iribarren number) computed with the beach slope and multiplied with the relative wave amplitude (wave amplitude to water depth ratio).

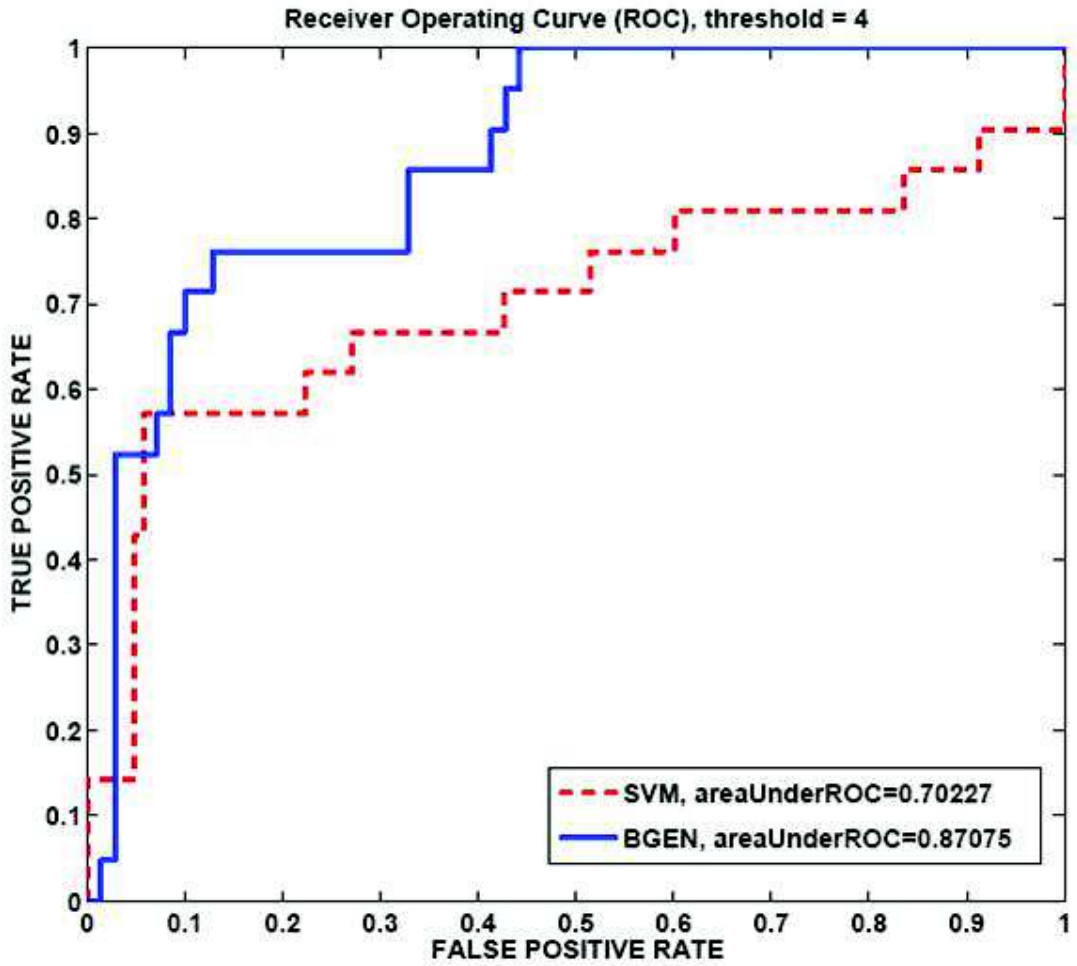
Conclusion on Example 1

GP-UCB-PE

- Generic optimization method
- Good theoretical guarantees
- Efficient in practice
- Easy to implement

Matlab source code online at:
`http://econtal.perso.math.cnrs.fr/software/`

Receiver Operating Characteristic (ROC) curve

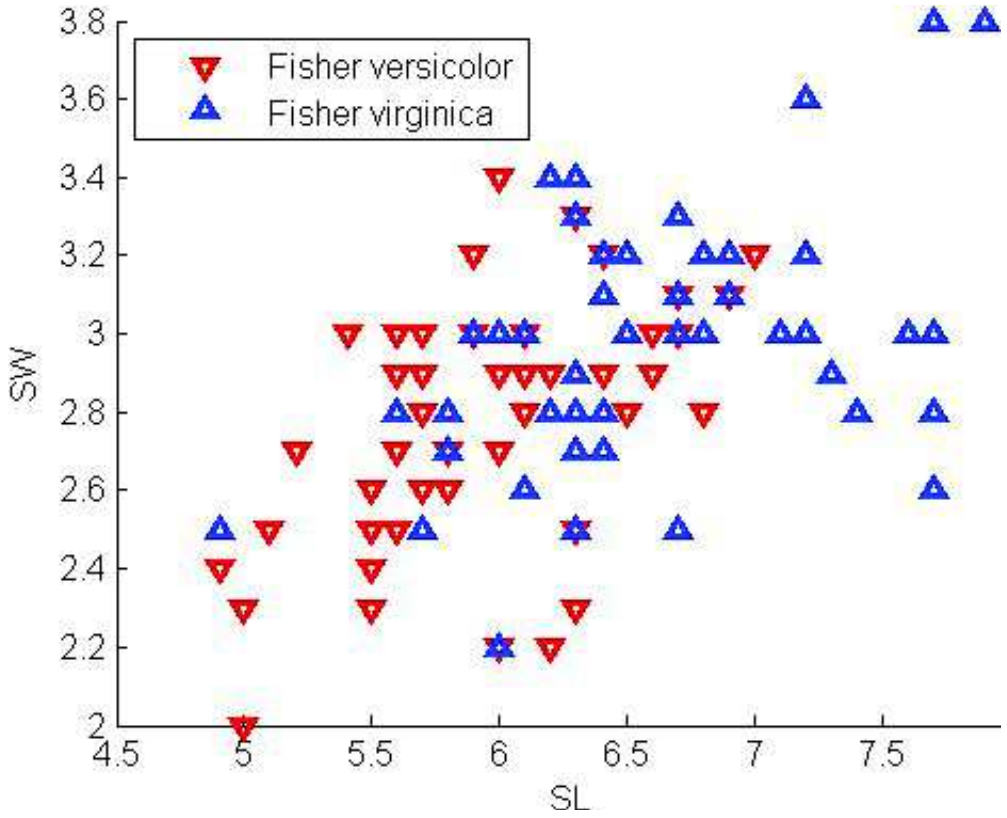


Motivations: Predictive analysis on high dimensional data

- Applications:
  - credit risk screening, medical diagnosis, churn prediction, spam filtering, ...
- Advances in prediction models:
  - parametric estimation vs. risk optimization
- Goals:
  - Performance, stability, scalability, interpretability

Main Example: Learning from classification data

- Observe a collection of data: $(X_i, Y_i) \in \mathbb{R}^d \times \{-1, +1\}$, $i = 1, \ldots, n$



Which decision?

1. Predictive Classification

   Given a new X', predict the label Y'

   Decision rule: $g : \mathbb{R}^d \to \{-1, +1\}$

   Happy if classification error is low

2. Predictive Ranking/Scoring

   Given new data $\{X'_1, \ldots, X'_m\}$, predict a ranking $(X'_{i_1}, \ldots, X'_{i_m})$

   Decision rule: $s : \mathbb{R}^d \to \mathbb{R}$

   Happy if many $Y'_i = +1$ at the top of the ordered list

The Classification Problem

Statistical Model for Classification Data - Two views

- (X, Y) random pair with unknown distribution P over $\mathbb{R}^d \times \{-1, +1\}$

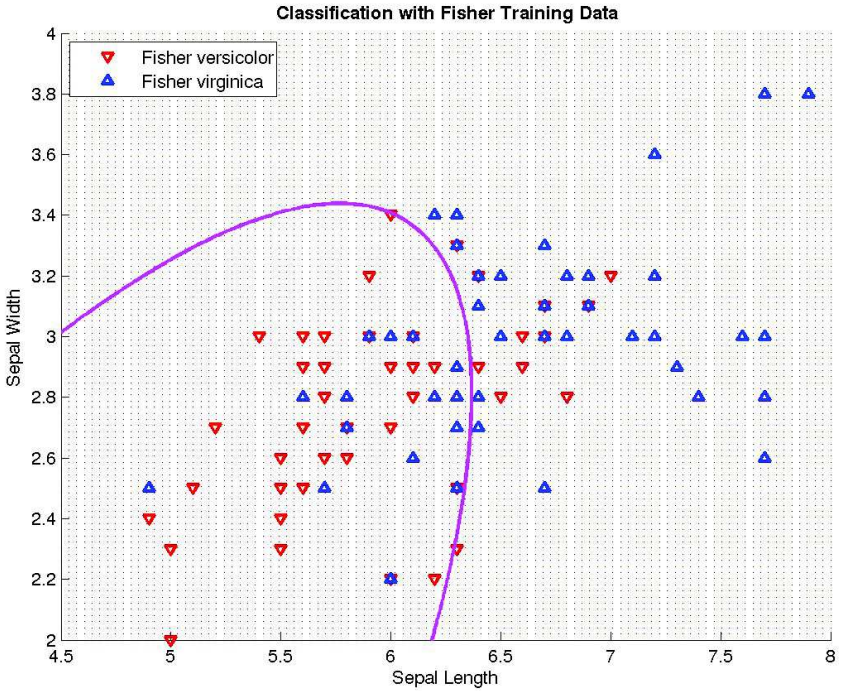
1. **Generative view** - Joint distribution P as a mixture
   - Class-conditional densities: f_+ and f_-
   - Mixture parameter: $p = \mathbb{P}\{Y = +1\}$
2. **Discriminative view** - Joint distribution P described by (P_X, η)
   - Marginal distribution: $X \sim P_X$, density f_X
   - Posterior probability function:

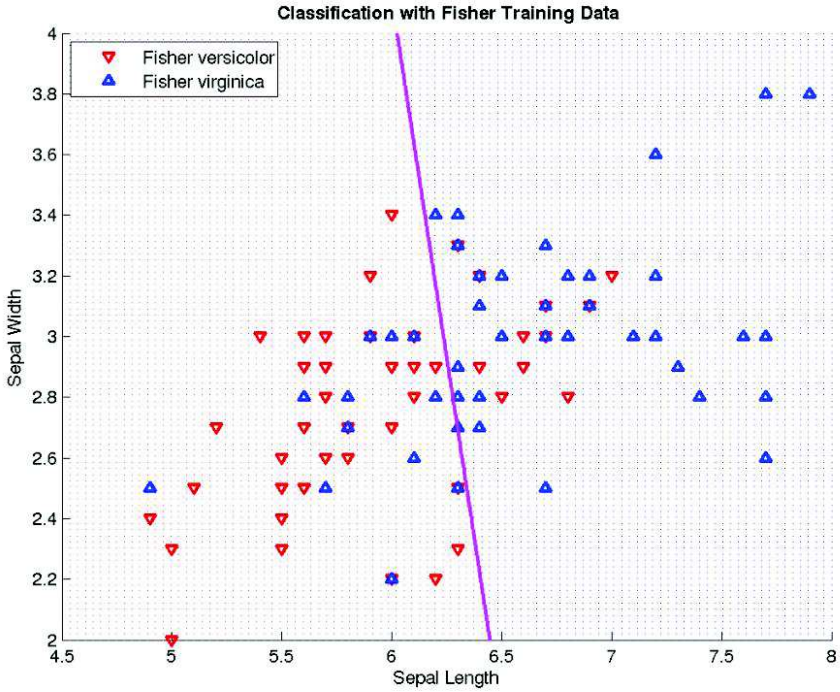
$$\eta(x) = \mathbb{P}\{Y = 1 \mid X = x\} \ , \quad \forall x \in \mathbb{R}^d$$

- Marginal distribution has density: $f_X = pf_+ + (1 - p)f_-$
- Posterior probability is given by: $\eta = pf_+/f_X$

Parametric classification with Discriminant Analysis

- Mixture model with gaussian class-conditional distributions f_+ and f_-
- Linear or Quadratic Discriminant Analysis





Principle of Discriminant Analysis

- Use estimates of posterior probabilities: $\forall x \in \mathbb{R}^d$

$$\eta(x) = \frac{p f_+(x)}{f_X(x)} \ , \quad 1 - \eta(x) = \frac{(1-p) f_-(x)}{f_X(x)}$$

- Decision function = plug-in estimate of

$$g^*(x) = 2\mathbb{I}\left\{\eta(x) > 1 - \eta(x)\right\} - 1$$

- Discriminant Analysis: use $f_+ = \mathcal{N}_d(\mu_+, \Sigma_+)$ and $f_- = \mathcal{N}_d(\mu_-, \Sigma_-)$
- If d large, apply dimension reduction techniques (PCA, ...)

Parametric classification with Logistic Regression

- Consider a family $\{\eta_\theta \ : \ \theta \in \mathbb{R}^d\}$ such that:

$$\log\left(\frac{\eta_\theta(x)}{1-\eta_\theta(x)}\right) = \theta^T x$$

- This is equivalent to:

$$\eta_\theta(x) = \frac{\exp(\theta^T x)}{1+\exp(\theta^T x)}$$

- Estimation $\widehat{\theta}$ by conditional likelihood maximization (Newton-Raphson)
- Plug-in classification rule: $\widehat{g}(x) = 2\mathbb{I}\{\eta_{\widehat{\theta}}(x) > 1/2\} - 1$

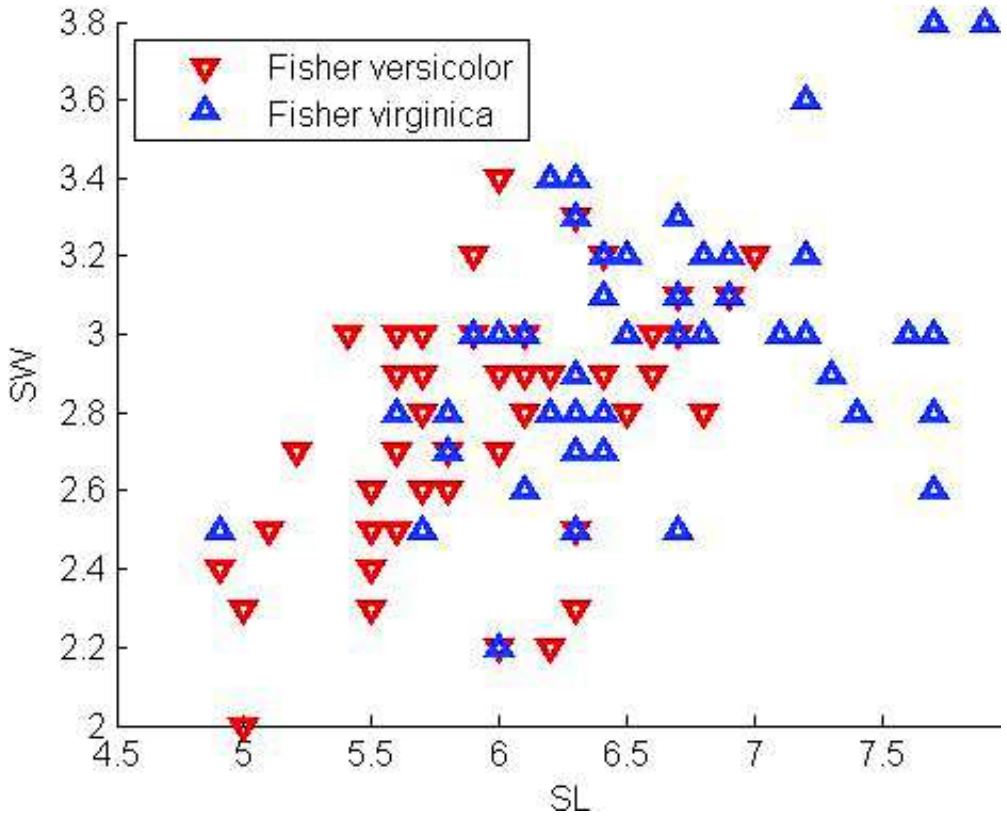
Efficient Classification for High Dimensional Data

- Local averaging
  - Histogram or Kernel rules
  - Nearest Neighbors
  - Partitioning methods: decision trees (CART, C4.5, ...)
- Global methods
  - Neural Networks: minimize (smooth version of) classification error
  - Support Vector Machines, Boosting - minimize convex surrogate of classification error
- Aggregation and randomization
  - Bagging, Random Forests - use aggregation, resampling and randomization

The scoring problem

Scoring binary classification data

- From small scores (most likely -1) to high scores (most likely $+1$)



Motivations

- Learn a preorder on a measurable space $\mathcal{X}$ (e.g. $\mathbb{R}^d$)
- Alternative approach to parametric modeling of the posterior probability (e.g. Logistic Regression)
- The special nature of the scoring problem:
  - between classification and regression function estimation

Main issues

- Optimal elements
- Performance measures
- ERM principles and statistical theory
- Design of efficient algorithms
- Meta-algorithms and aggregation principle

Modeling issue: Nature of feedback information?

- Preference model:
  - (X, X', Z) with label $Z = Y - Y'$ over $\{-1, 0, +1\}$
- Plain regression:
  - (X, Y) with label Y over $\mathbb{R}$
- Bipartite scoring:
  - (X, Y) with binary label in $\{-1, +1\}$
- K-partite scoring:
  - (X, Y) with ordinal label Y over $\{1, \ldots, K\}$, $K > 2$

The scoring problem

The bipartite case

Optimal elements for scoring ($K = 2$)

- $X \in \mathbb{R}^d$ - observation vector in a high dimensional space
- $Y \in \{-1, +1\}$ - binary diagnosis
- Key theoretical quantity (posterior probability)

$$\eta(x) = \mathbb{P}\{Y = 1 \mid X = x\} \ , \quad \forall x \in \mathbb{R}^d$$

- Optimal scoring rules:

  $\Rightarrow$ increasing transforms of η

Representation of optimal scoring rules ($K = 2$)

- Note that if $U \sim \mathcal{U}([0,1])$

$$\forall x \in \mathcal{X}\ , \quad \eta(x) = \mathbb{E}\left(\mathbb{I}\{\eta(x) > U\}\right)$$

- If $s^* = \psi \circ \eta$ with ψ strictly increasing, then:

$$\forall x \in \mathcal{X}\ , \quad s^*(x) = c + \mathbb{E}\left(w(V) \cdot \mathbb{I}\{\eta(x) > V\}\right)$$

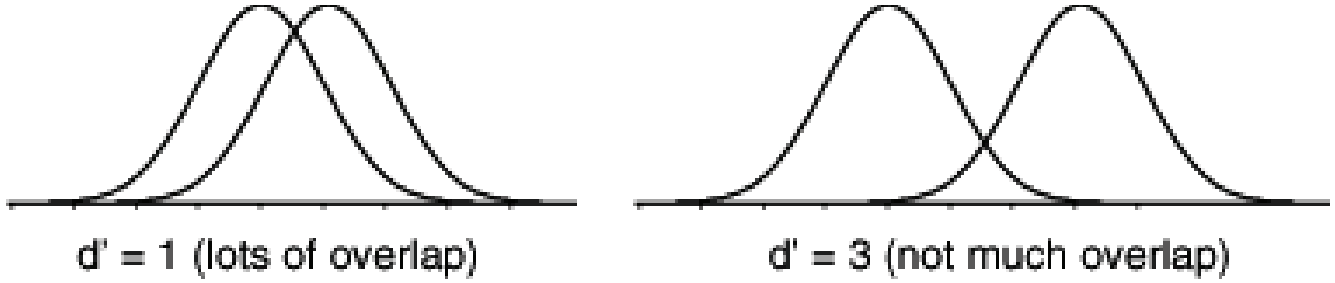
  for some:

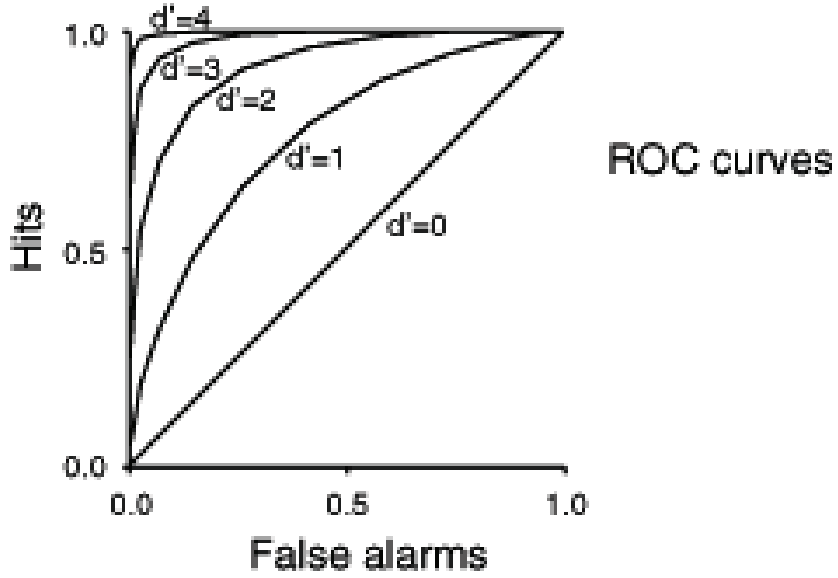
  - $c \in \mathbb{R}$,
  - V continuous random variable in $[0,1]$
  - $w : [0,1] \to \mathbb{R}_+$ integrable.

- Optimal scoring amounts to recovering the level sets of η:

$$\{x : \eta(x) > q\}_{q \in (0,1)}$$

The Gold Standard for Scoring: the ROC Curve ($K = 2$)





ROC optimality = Neyman-Pearson theory

- **Power curve** of the test statistic $s(X)$ when testing

$$\mathcal{H}_0 : X \sim P_- \quad \text{against} \quad \mathcal{H}_1 : X \sim P_+$$

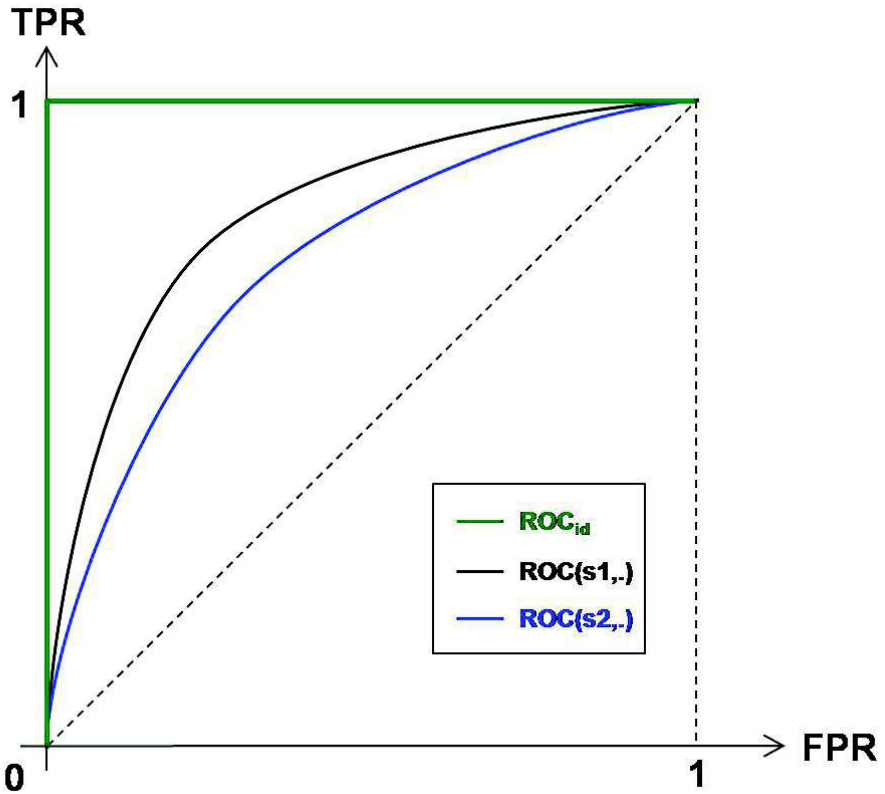
- Likelihood ratio $\phi(X)$ yields a **uniformly most powerful** test

$$\phi(X) = \frac{dP_+}{dP_-}(X) = \frac{1-p}{p} \times \frac{\eta(X)}{1-\eta(X)}.$$

- Optimal scoring rules are optimal in the sense of the ROC curve

Performance measures for scoring ($K = 2$)

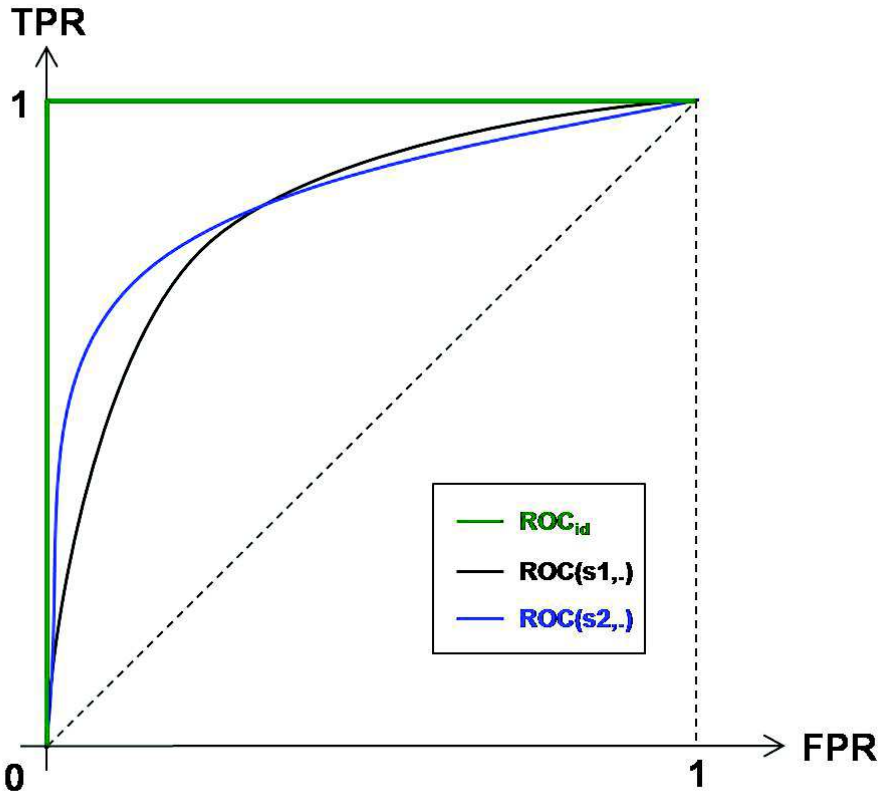
- Curves:
  - **ROC curve**
  - Precision-Recall curve
- Summaries (global vs. best scores):
  - **AUC** (global measure)
  - Partial AUC (Dodd and Pepe '03)
  - **Local AUC** (Clémençon and Vayatis '07)
- Other measures:
  - Average Precision, Hit Rate, Discounted Cumulative Gain, ...



ROC curves.

Performance measures for scoring ($K = 2$)

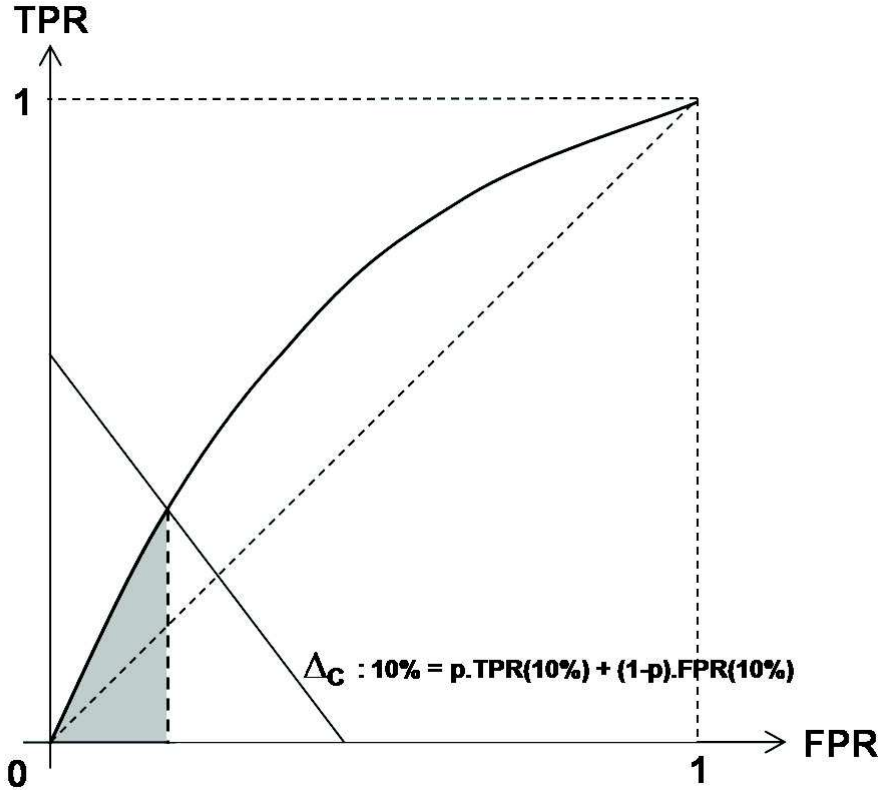
- Curves:
  - **ROC curve**
  - Precision-Recall curve
- Summaries (global vs. best scores):
  - **AUC** (global measure)
  - Partial AUC (Dodd and Pepe '03)
  - **Local AUC** (Clémençon and Vayatis '07)
- Other measures:
  - Average Precision, Hit Rate, Discounted Cumulative Gain, ...



ROC curves.

Performance measures for scoring ($K = 2$)

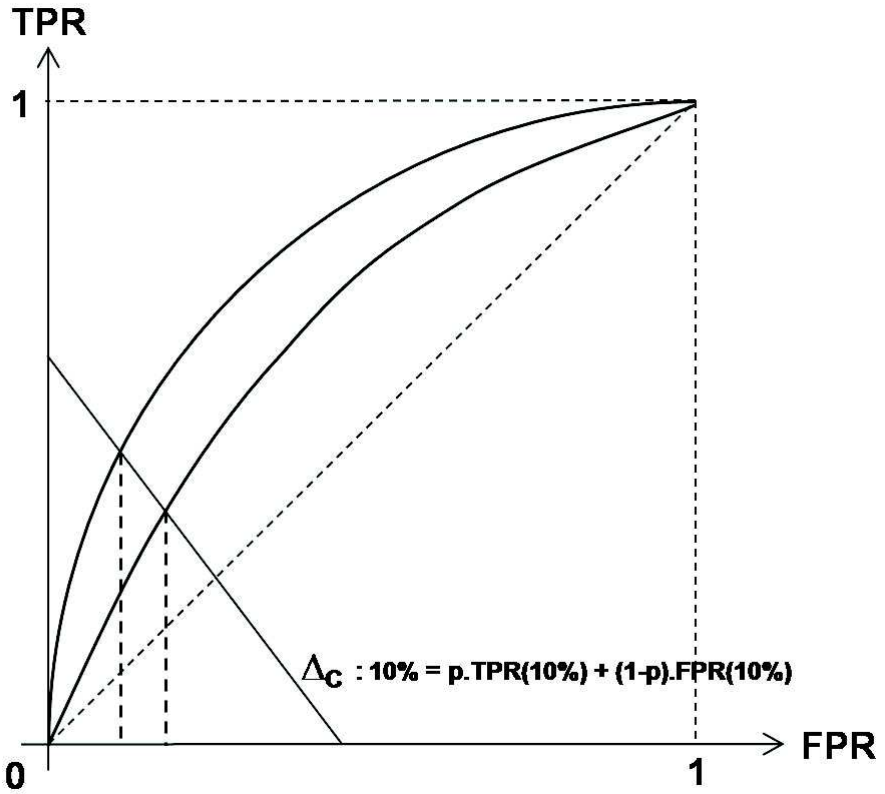
- Curves:
  - **ROC curve**
  - Precision-Recall curve
- Summaries (global vs. best scores):
  - **AUC** (global measure)
  - Partial AUC (Dodd and Pepe '03)
  - **Local AUC** (Clémençon and Vayatis '07)
- Other measures:
  - Average Precision, Hit Rate, Discounted Cumulative Gain, ...



Partial AUC.

Performance measures for scoring ($K = 2$)

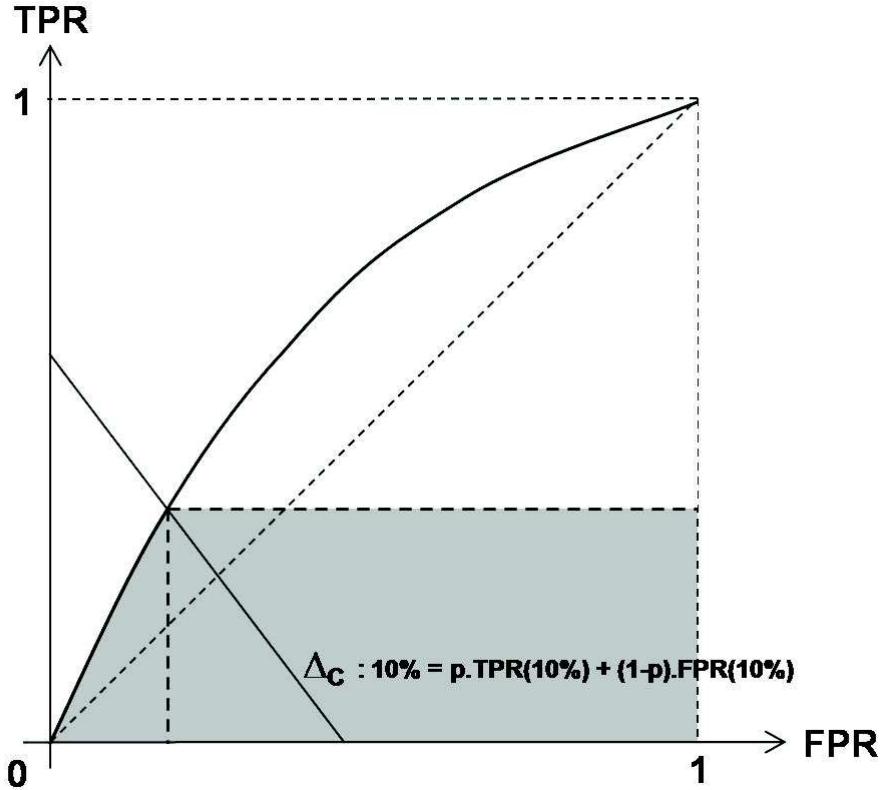
- Curves:
  - **ROC curve**
  - Precision-Recall curve
- Summaries (global vs. best scores):
  - **AUC** (global measure)
  - Partial AUC (Dodd and Pepe '03)
  - **Local AUC** (Clémençon and Vayatis '07)
- Other measures:
  - Average Precision, Hit Rate, Discounted Cumulative Gain, ...



Inconsistency of Partial AUC.

Performance measures for scoring ($K = 2$)

- Curves:
  - **ROC curve**
  - Precision-Recall curve
- Summaries (global vs. best scores):
  - **AUC** (global measure)
  - Partial AUC (Dodd and Pepe '03)
  - **Local AUC** (Clémençon and Vayatis '07)
- Other measures:
  - Average Precision, Hit Rate, Discounted Cumulative Gain, ...



Local AUC.

The TreeRank algorithm

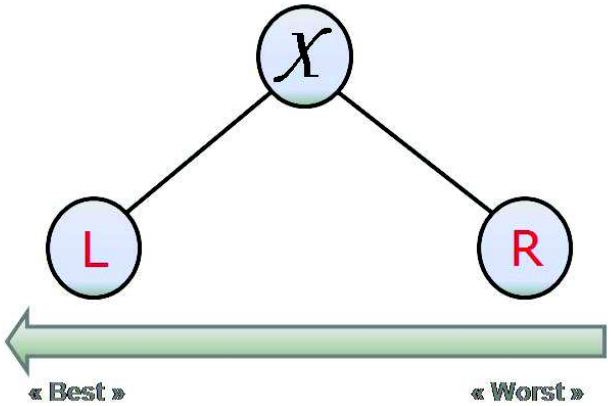
Recursive partitioning for nonparametric scoring

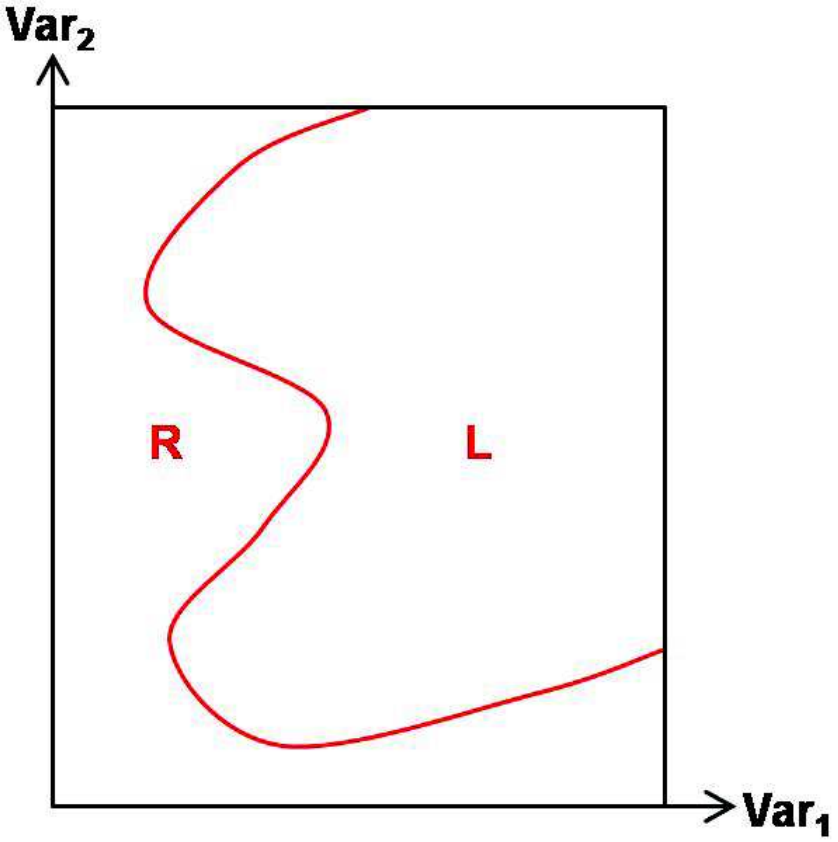
Principles of TreeRank - Clémençon and Vayatis (2009)

- Focus on the ROC curve optimization
- Decision tree heuristic based on three algorithms
  - TreeRank - Recursive partitioning step through local maximization of the AUC
  - LeafRank - Nonlocal splitting rule (operates cell permutation)
  - RankingForest - Aggregation of ranking trees by resampling and randomization
- Sound theoretical properties
- Numerical and statistical efficiency
- Analysis of variable importance (global and local)

TreeRank - building ranking (binary) trees

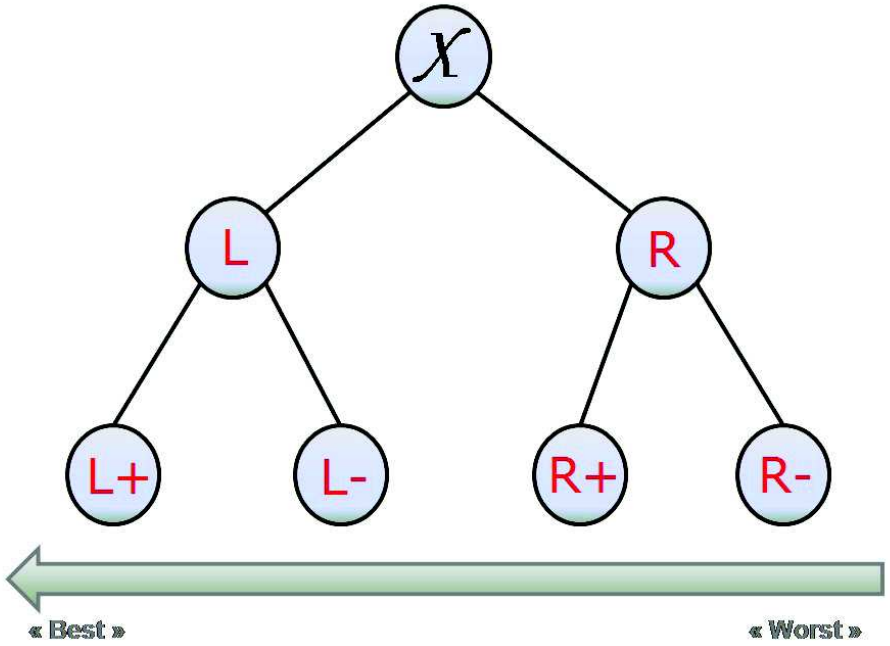
- Assume $\mathcal{X} = [0,1] \times [0,1]$

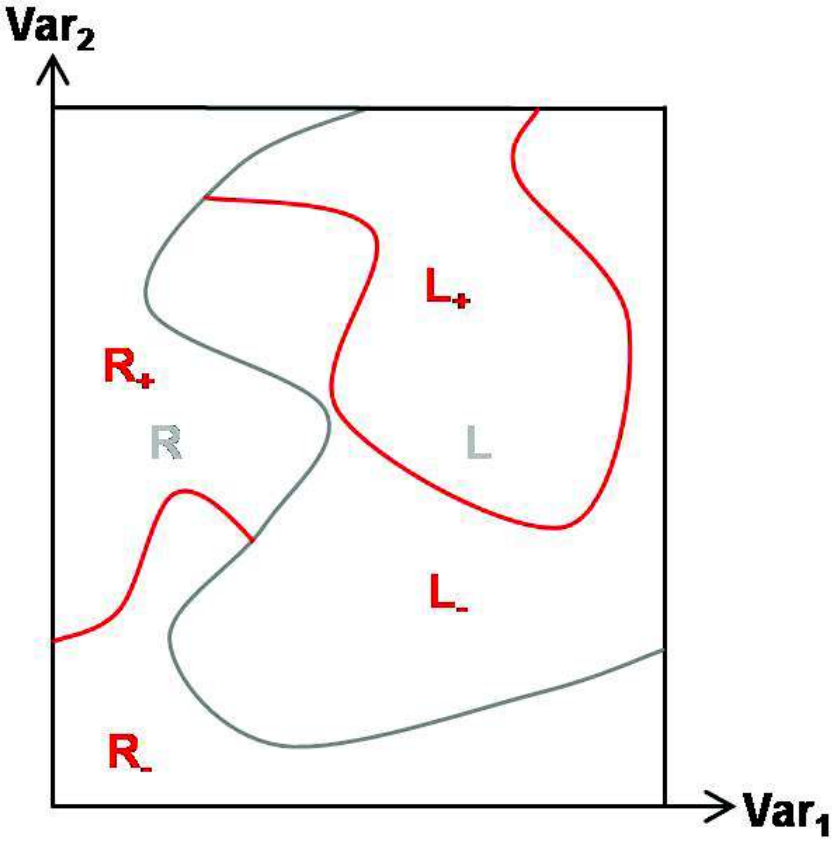




TreeRank - building ranking (binary) trees

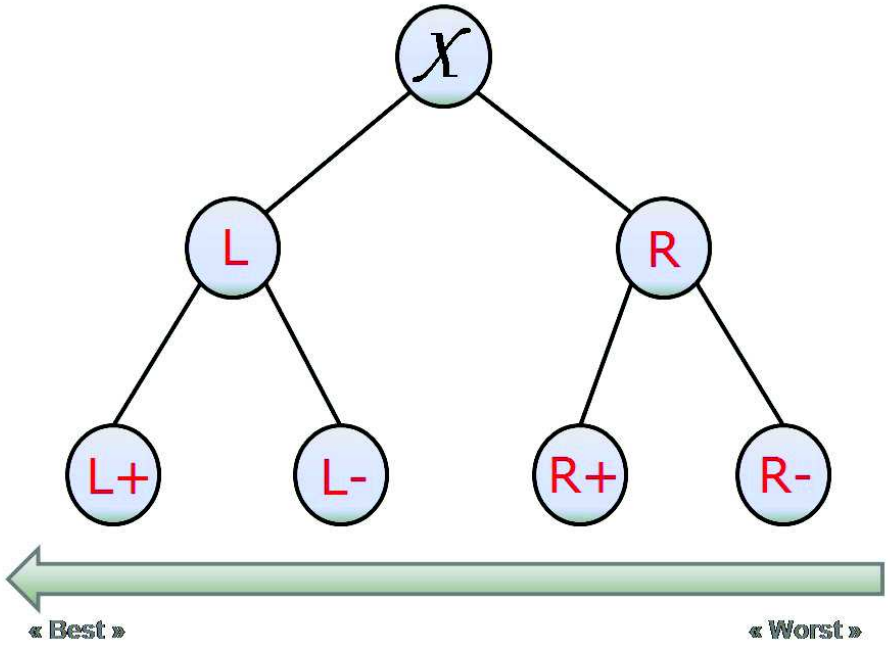
- Assume $\mathcal{X} = [0,1] \times [0,1]$

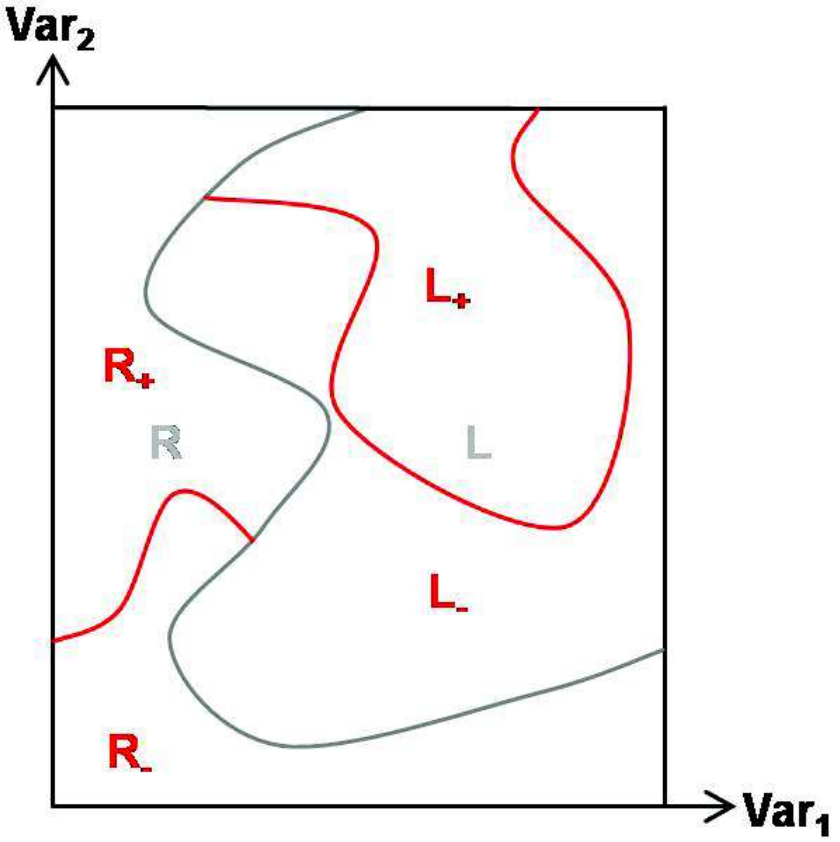




TreeRank - building ranking (binary) trees

- Assume $\mathcal{X} = [0,1] \times [0,1]$





- A wiser option: use orthogonal splits!

Notations

- Data: $(X_1, Y_1), \ldots, (X_n, Y_n)$
- Take a class $\mathcal{C}$ of sets defining the (orthogonal) splits in input space
- Empirical versions of FPR and TPR:

$$\hat{\alpha}(C) = \frac{1}{n_-} \sum_{i=1}^{n} \mathbb{I}\{X_i \in C,\ Y_i = -1\}$$

$$\hat{\beta}(C) = \frac{1}{n_+} \sum_{i=1}^{n} \mathbb{I}\{X_i \in C,\ Y_i = +1\}$$

Purity criterion for splitting

- Mother cell C with FPR $\alpha(C)$ and TPR $\beta(C)$
- Class Γ of splitting rules
- Purity measure

$$\Lambda_C(\gamma) = \alpha(C) \cdot \hat{\beta}(\gamma) - \beta(C) \cdot \hat{\alpha}(\gamma)$$

- Find the left offspring of C as the best subset C_+

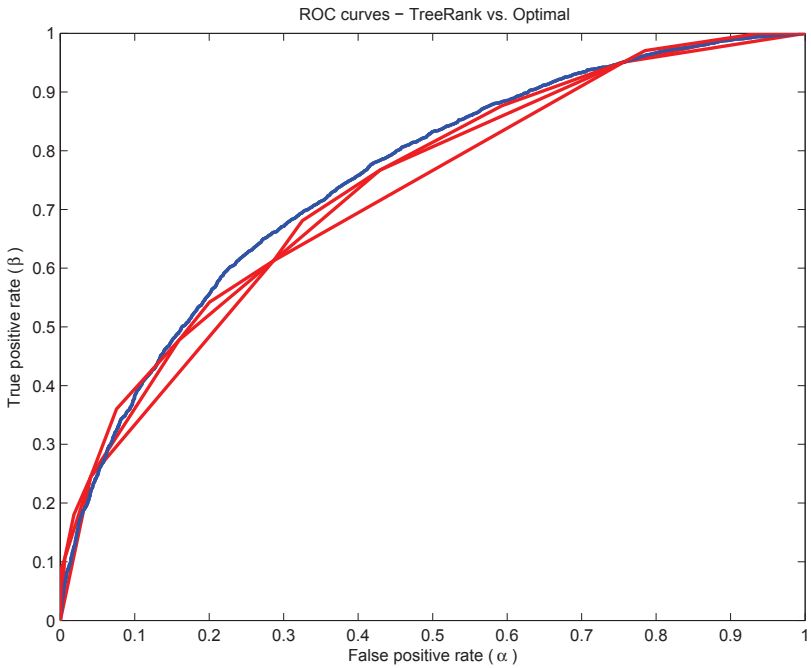
$$C_+ = \underset{\gamma \in \Gamma,\ \gamma \subset C}{\operatorname{argmax}} \Lambda_C(\gamma)$$

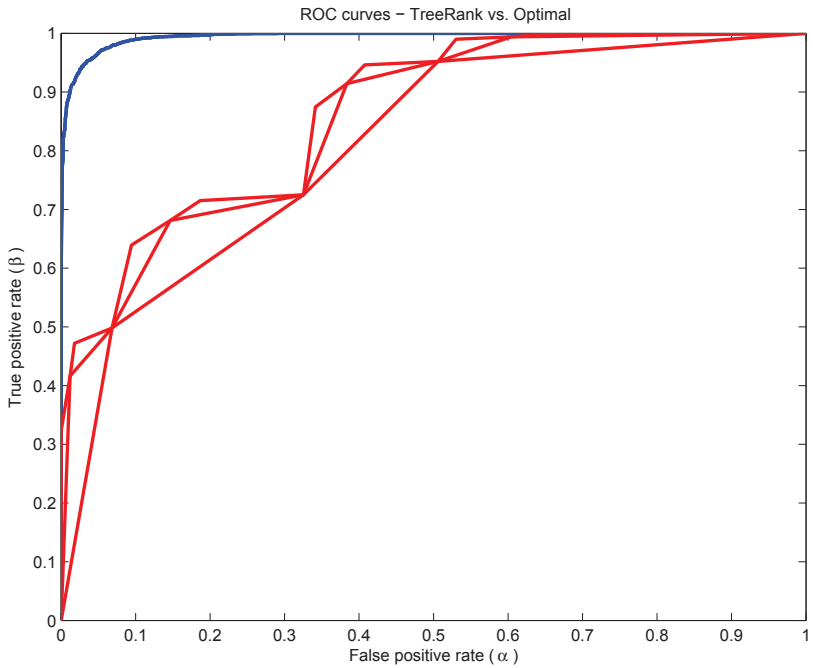
- Amounts to solving an asymmetric classification classification problem with data-dependent cost.
- Amounts to maximizing the local increments of AUC

Empirical performance of TreeRank

Gaussian mixture with orthogonal splits

easy with overlap **vs.** difficult and no overlap



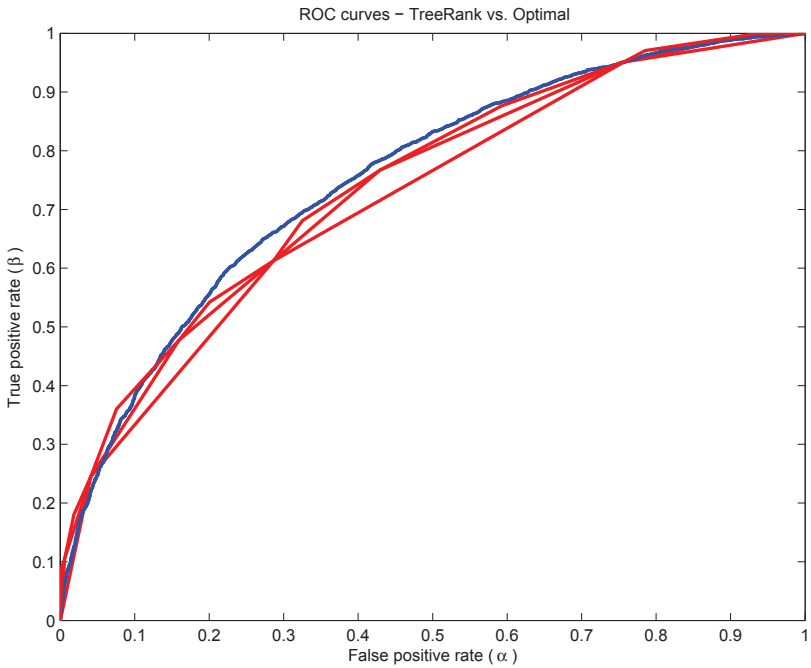


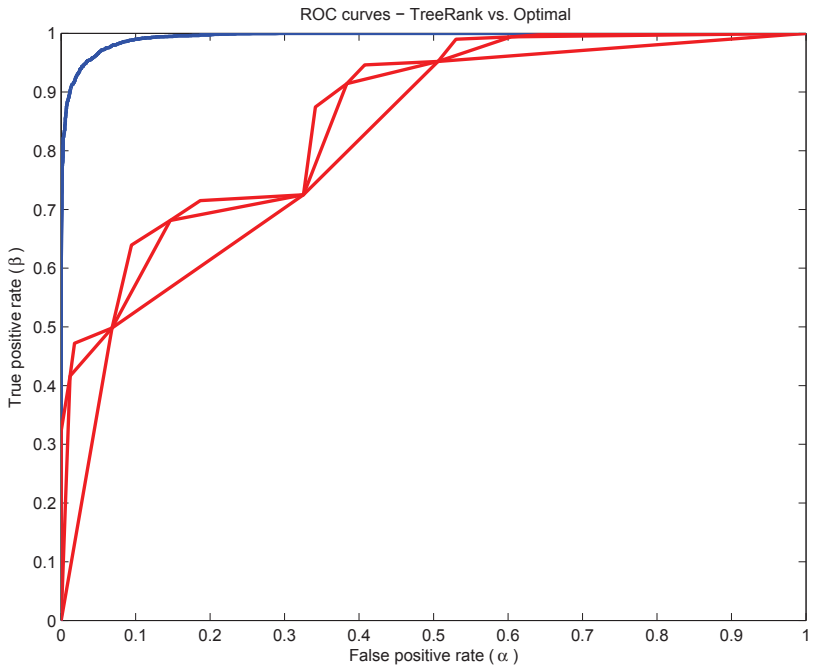
- Concavity of the ROC curve estimate only if Γ is union stable

Empirical performance of TreeRank

Gaussian mixture with orthogonal splits

easy with overlap **vs.** difficult and no overlap





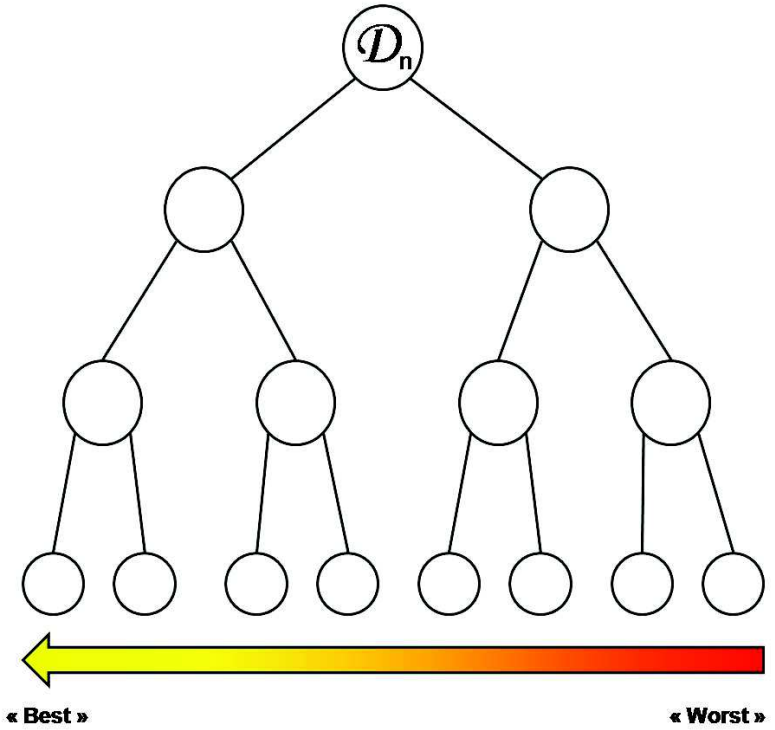
- Concavity of the ROC curve estimate only if Γ is union stable

TreeRank and the problem with recursive partitioning

- The TreeRank algorithm:
  - implements an empirical version of local AUC maximization procedure
  - yields AUC- **and** ROC- **consistent** scoring rules (Clémençon-Vayatis '09)
  - boils down to solving a collection of **nested** optimization problems

- **Main goal:**
  - Global performance in terms of the ROC curve

- **Main issue:**
  - Recursive partitioning not so good when the nature of the problem is not local



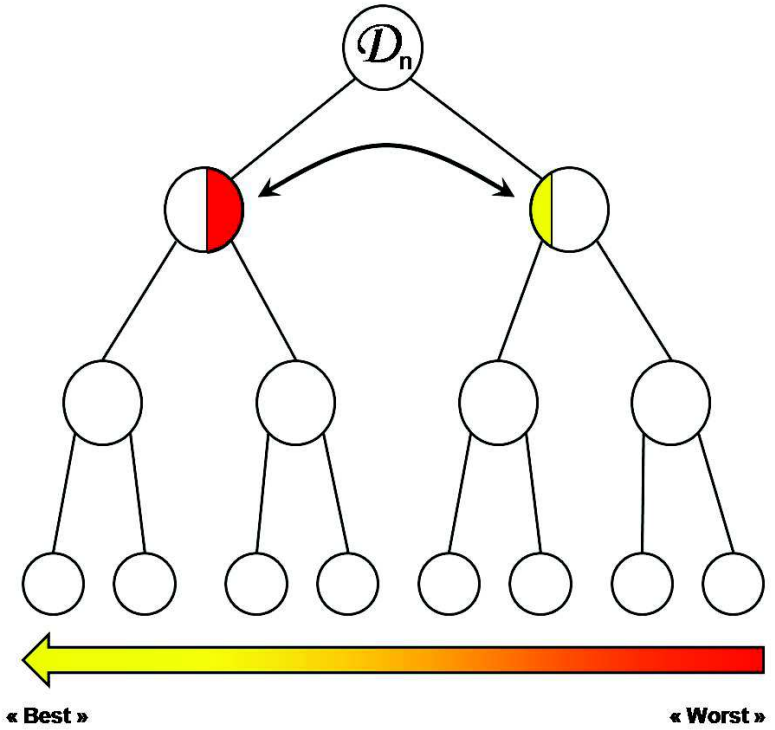
- **Key point:** choice of a splitting rule for the AUC optimization

TreeRank and the problem with recursive partitioning

- The TreeRank algorithm:
  - implements an empirical version of local AUC maximization procedure
  - yields AUC- **and** ROC- **consistent** scoring rules (Clémençon-Vayatis '09)
  - boils down to solving a collection of **nested** optimization problems

- **Main goal:**
  - Global performance in terms of the ROC curve

- **Main issue:**
  - Recursive partitioning not so good when the nature of the problem is not local



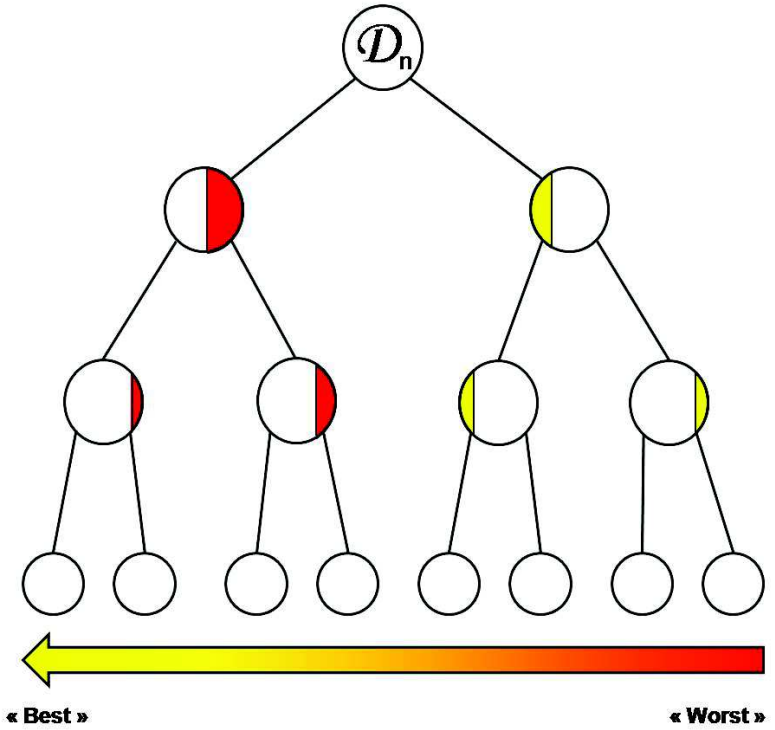
- **Key point:** choice of a splitting rule for the AUC optimization

TreeRank and the problem with recursive partitioning

- The TreeRank algorithm:
  - implements an empirical version of local AUC maximization procedure
  - yields AUC- **and** ROC- **consistent** scoring rules (Clémençon-Vayatis '09)
  - boils down to solving a collection of **nested** optimization problems

- **Main goal:**
  - Global performance in terms of the ROC curve

- **Main issue:**
  - Recursive partitioning not so good when the nature of the problem is not local



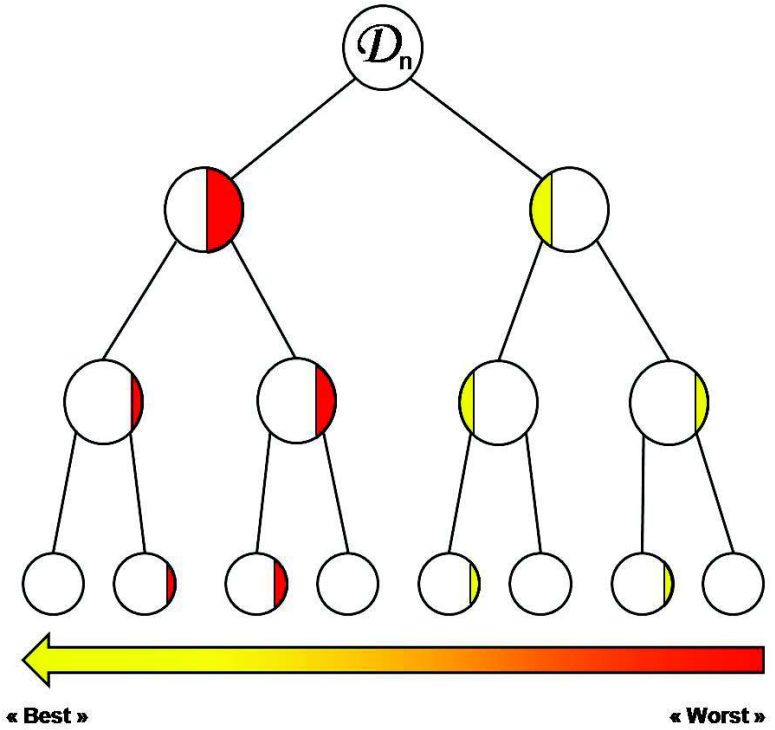
- **Key point:** choice of a splitting rule for the AUC optimization

TreeRank and the problem with recursive partitioning

- The TreeRank algorithm:
  - implements an empirical version of local AUC maximization procedure
  - yields AUC- **and** ROC- **consistent** scoring rules (Clémençon-Vayatis '09)
  - boils down to solving a collection of **nested** optimization problems

- **Main goal:**
  - Global performance in terms of the ROC curve

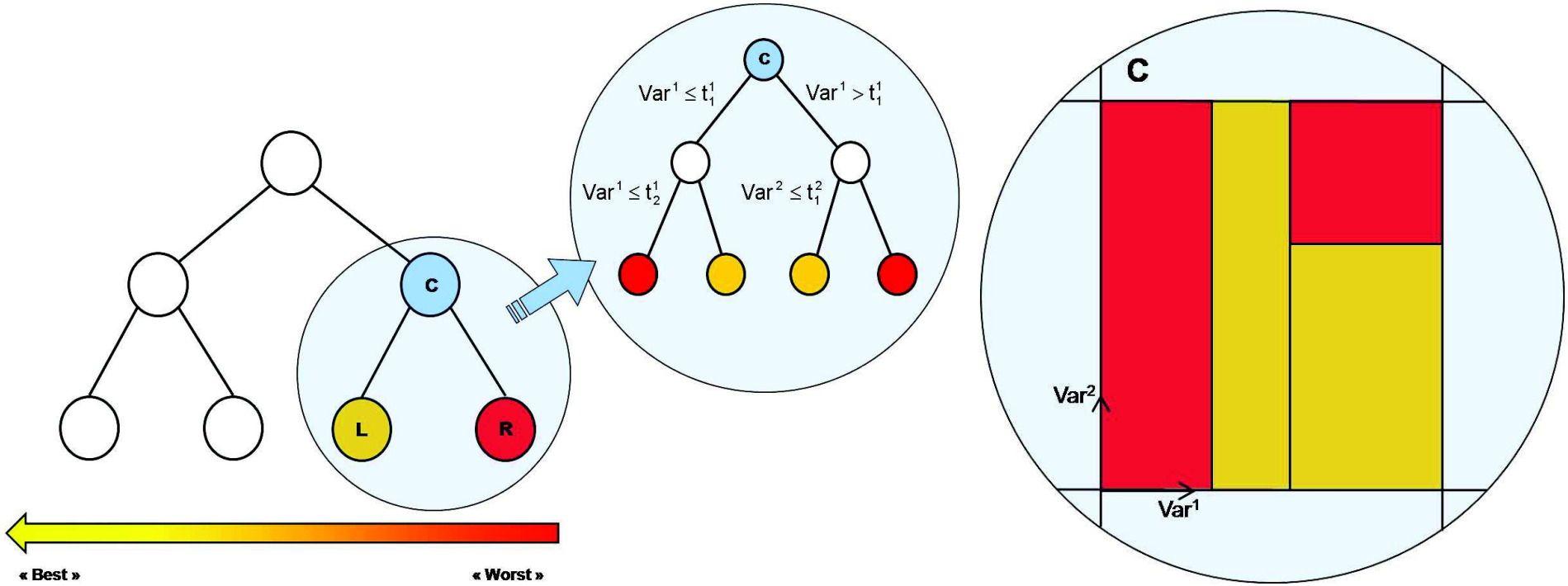
- **Main issue:**
  - Recursive partitioning not so good when the nature of the problem is not local



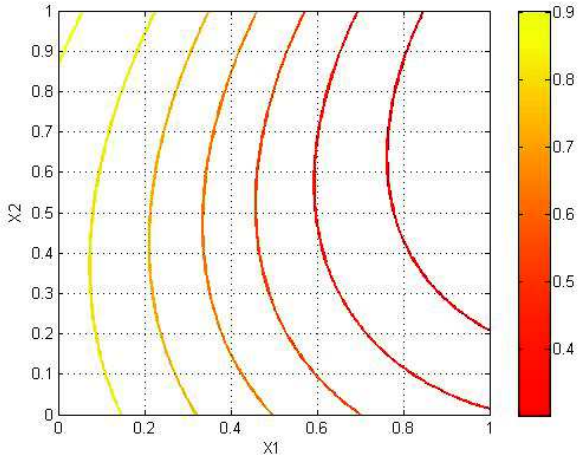
- **Key point:** choice of a splitting rule for the AUC optimization

Nonlocal splitting rule - The LeafRank Procedure

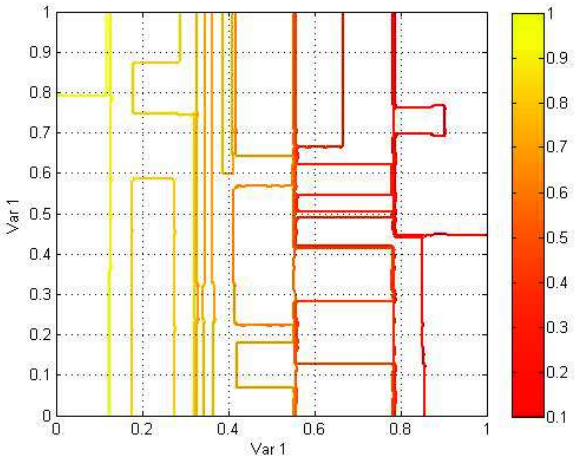
- Any classification method can be used as a splitting rule
- Our choice: the LeafRank procedure
  - Use classification tree with orthogonal splits (CART)
  - Find optimal cell permutation for a fixed partition
  - Improves representation capacity and still permits interpretability



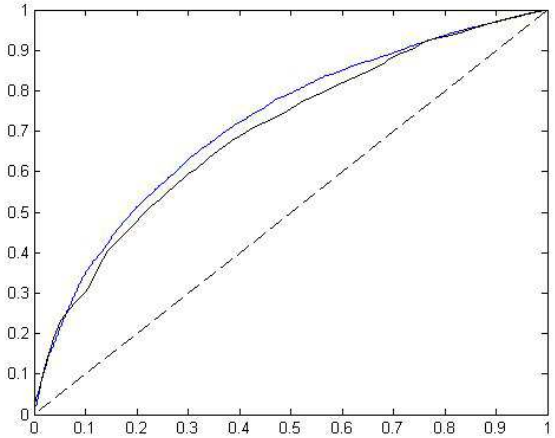
Iterative TreeRank in action- synthetic data set



a. Level sets of the true regression function η.



b. Level sets of the estimated regression function η.

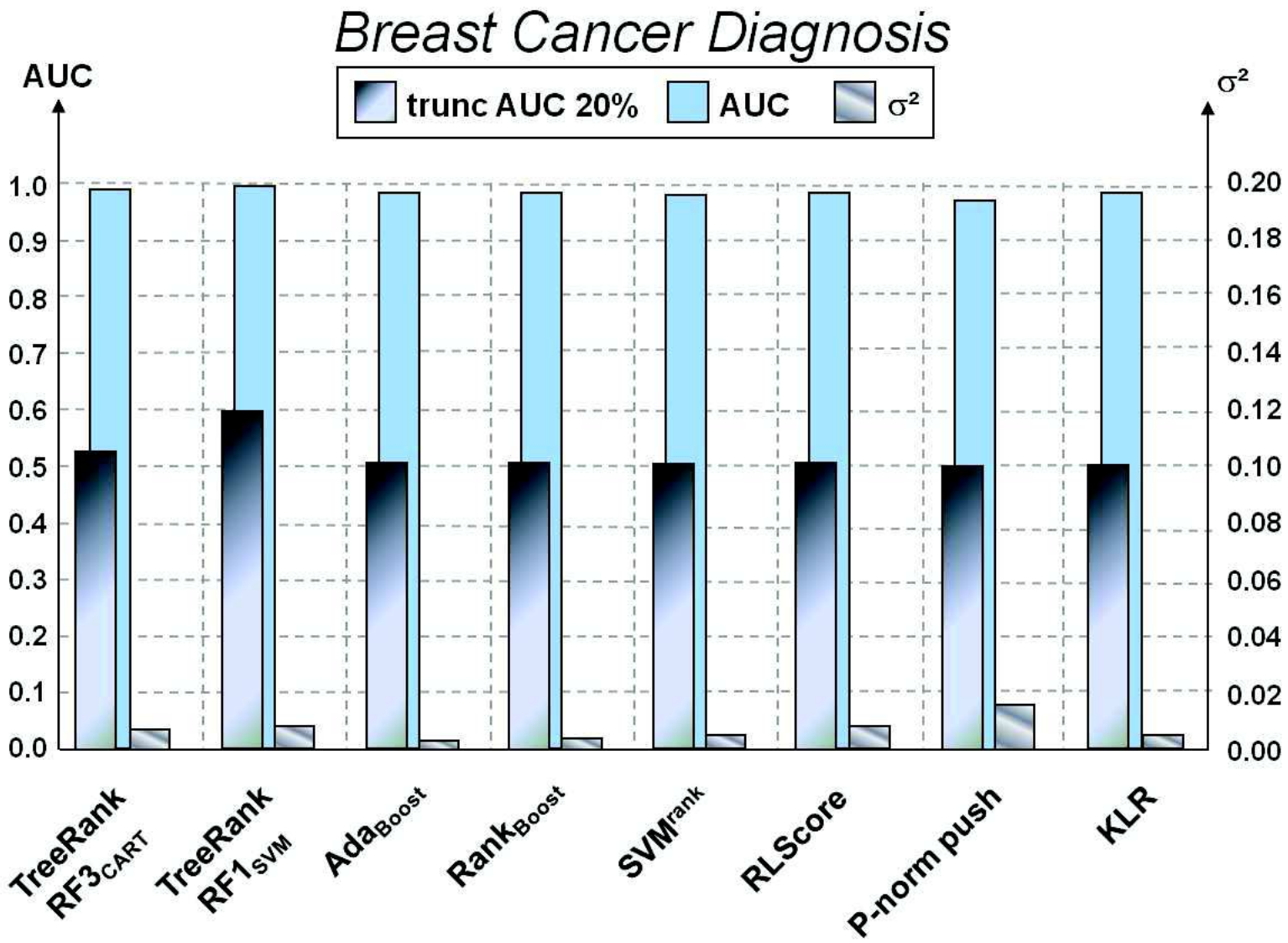


c. True (blue) and Estimated (black) Roc Curve.

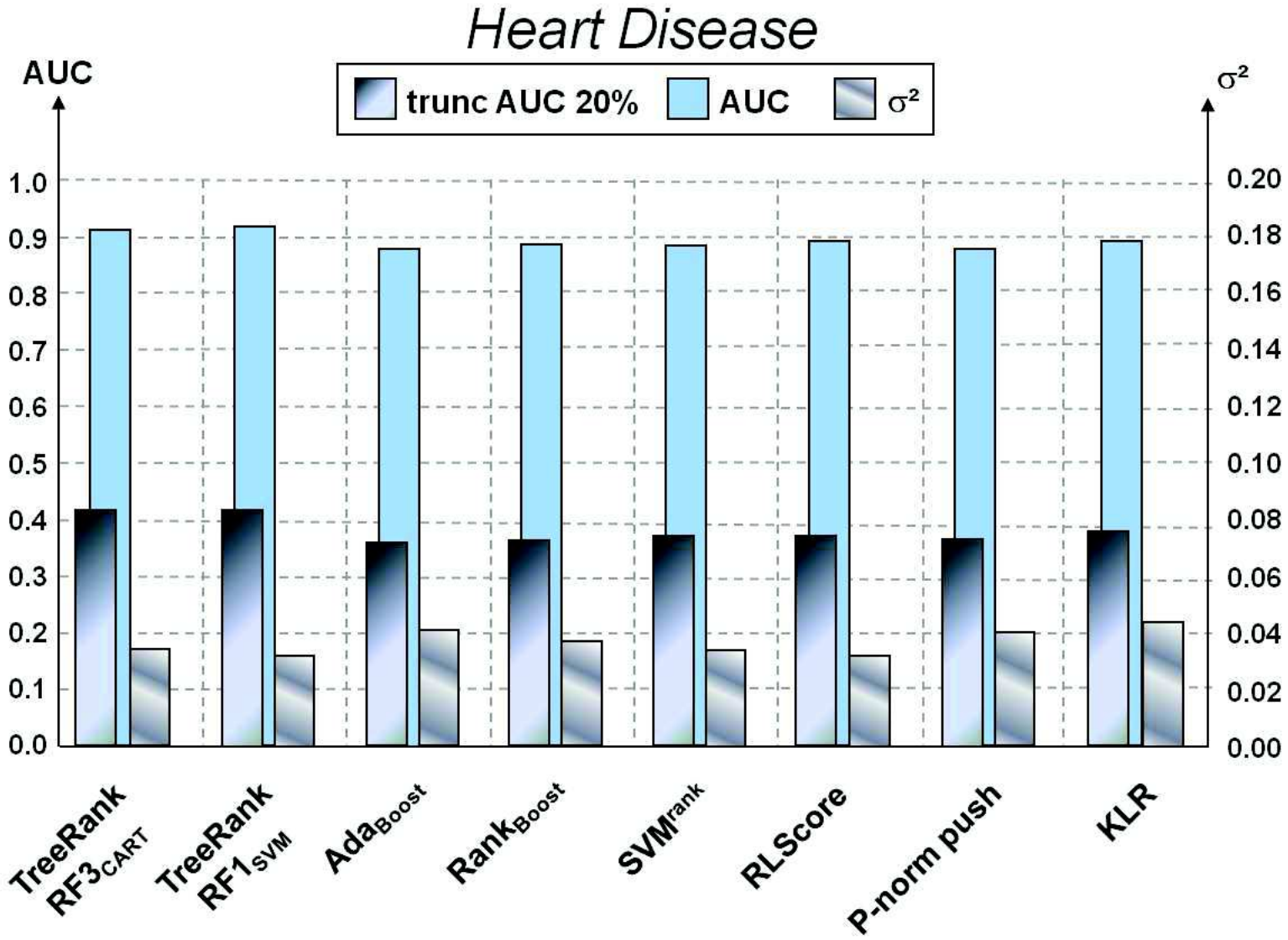
RankForest and competitors on UCI data sets (1)

- Data sets from the UCI Machine Learning repository
  - Breast Cancer
  - Heart Disease
  - Hepatitis
- **Competitors:**
  - AdaBoost (Freund and Schapire '95)
  - RankBoost (Freund *et al.* '03)
  - RankSvm (Joachims '02, Rakotomamonjy '04)
  - RankRLS (Pahikkala *et al.* '07)
  - KLR (Zhu and Hastie '01)
  - P-normPush (Rudin '06)

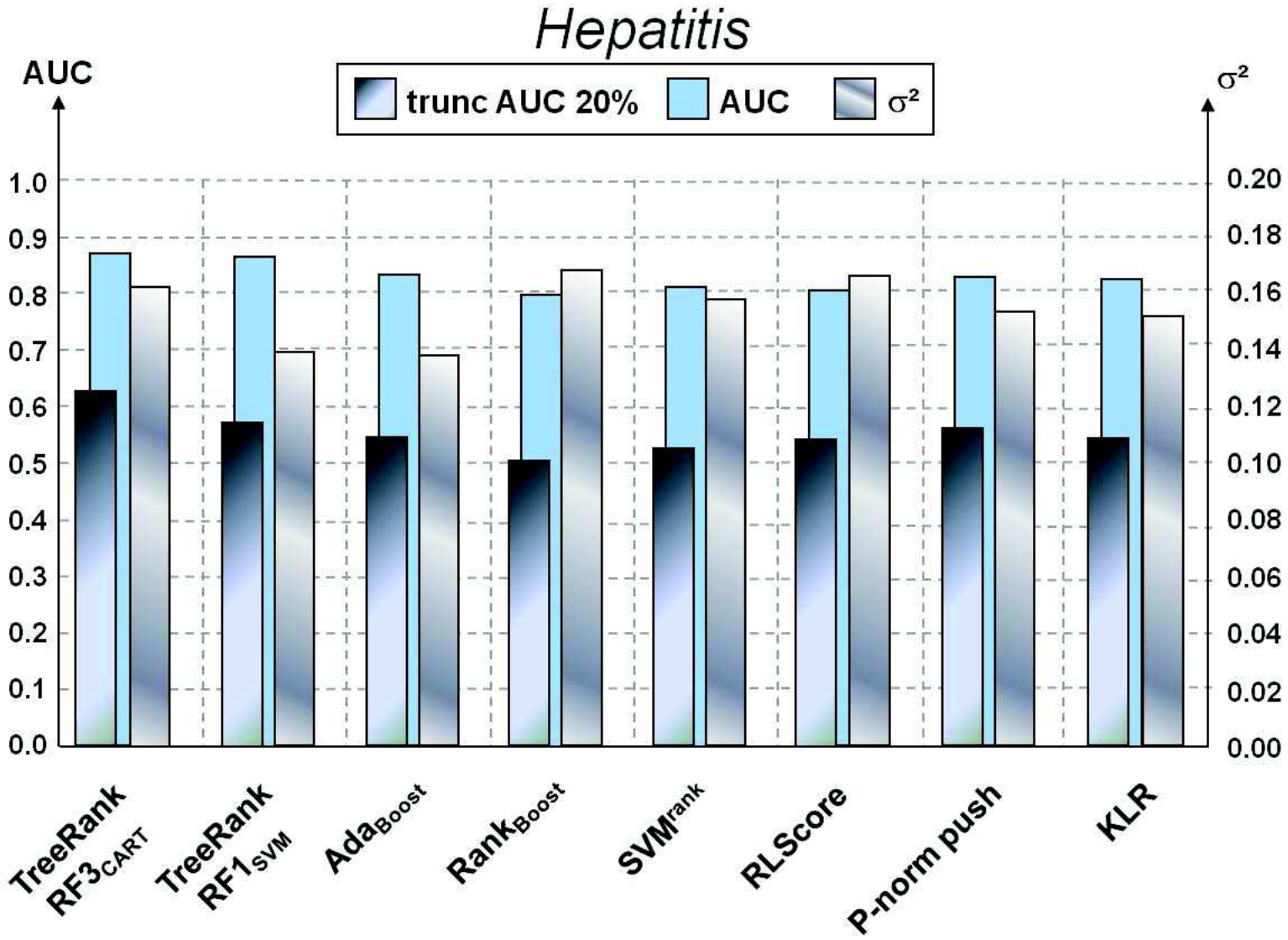
RankForest and competitors (2)



RankForest and competitors (2)



RankForest and competitors (2)



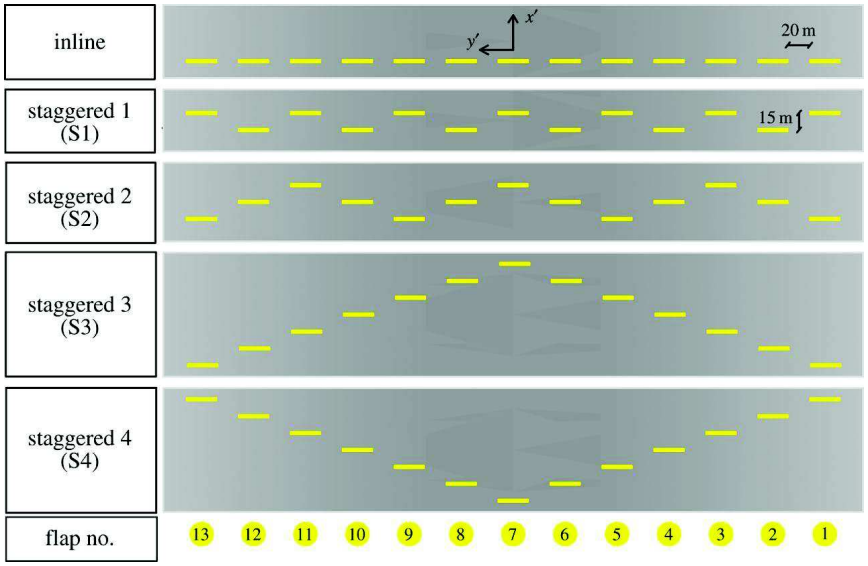
Local AUC $u = 0.5$ $u = 0.2$ $u = 0.1$	TreeRank	RankBoost	RankSVM
Australian Credit	0.425 ($\pm$0.012) 0.248 ($\pm$0.039) 0.111 ($\pm$0.002)	0.412 ($\pm$0.014) 0.206 ($\pm$0.013) 0.103 ($\pm$0.011)	0.404 ($\pm$0.024) 0.204 ($\pm$0.013) 0.103 ($\pm$0.010)
Ionosphere	0.494 ($\pm$0.062) 0.156 ($\pm$0.002) 0.078 ($\pm$0.001)	0.288 ($\pm$0.005) 0.144 ($\pm$0.003) 0.072 ($\pm$0.003)	0.263 ($\pm$0.044) 0.131 ($\pm$0.024) 0.065 ($\pm$0.014)
Breast Cancer	0.559 ($\pm$0.010) 0.442 ($\pm$0.076) 0.146 ($\pm$0.010)	0.534 ($\pm$0.018) 0.265 ($\pm$0.012) 0.132 ($\pm$0.014)	0.537 ($\pm$0.017) 0.271 ($\pm$0.009) 0.137 ($\pm$0.012)
Heart Disease	0.416 ($\pm$0.027) 0.273 ($\pm$0.070) 0.118 ($\pm$0.017)	0.361 ($\pm$0.041) 0.176 ($\pm$0.027) 0.089 ($\pm$0.017)	0.371 ($\pm$0.035) 0.188 ($\pm$0.022) 0.094 ($\pm$0.011)
Hepatitis	0.572 ($\pm$0.240) 0.413 ($\pm$0.138) 0.269 ($\pm$0.190)	0.504 ($\pm$0.225) 0.263 ($\pm$0.115) 0.133 ($\pm$0.057)	0.526 ($\pm$0.248) 0.272 ($\pm$0.125) 0.137 ($\pm$0.062)

Sytem design example

Exemple de mise en œuvre n°2
aide à la conception de système

Exemple :

→ Systèmes hydroliens off-shores
(WEC = Wave energy converter)



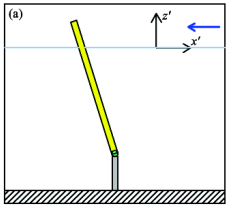
Question :

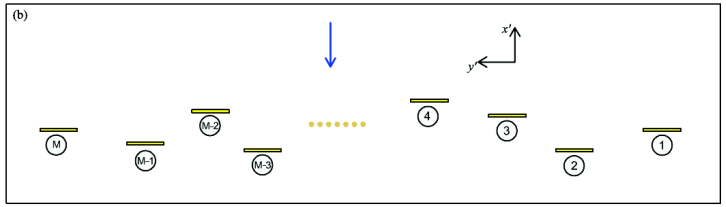
→ Configuration optimale du système ?

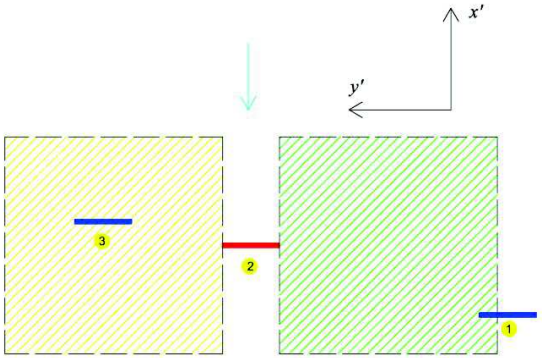
Exemple de mise en œuvre n°2
aide à la conception de système

Input X

- Caractéristiques typiques des vagues
- Bathymétrie
- Position relative des WEC







Output Y

- Energie produite (‘q factor’) :

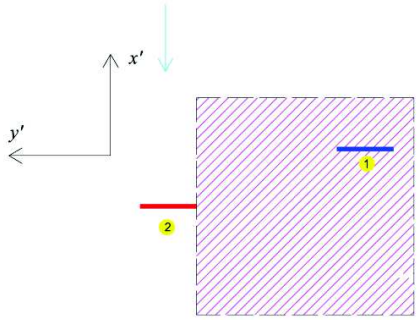
$$q = \frac{P_{groupe} - P_{isolé}}{P_{isolé}}$$

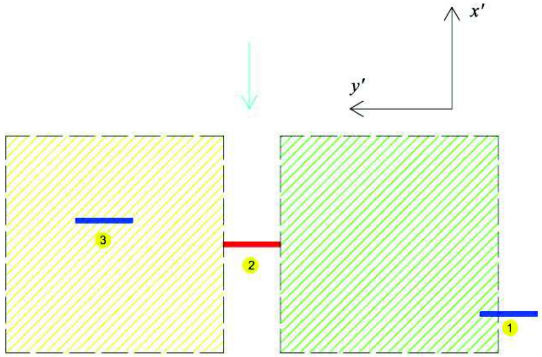
Exemple de mise en œuvre n°2
sélection de la configuration optimale

Méthode

- Approximation de l'énergie produite par une fonction additive de modules de 3-WEC et 2-WEC
- Recherche des maxima des modules 3-WEC et 2-WEC par la méthode précédente
- Utilisation d'un algorithme génétique 'sur-mesure' pour optimiser l'énergie globale du vecteur de WEC

$$\widetilde{q}(x'_1, y'_1, \ldots x'_M, y'_M) = 1 + \frac{1}{M}\left(q_2^{mod}(x'_2 - x'_1, y'_2 - y'_1) + q_2^{mod}(x'_M - x'_{M-1}, y'_M - y'_{M-1}) + \sum_{i=2}^{M-1} q_3^{mod}(x'_i - x'_{i-1}, y'_i - y'_{i-1}, x'_{i+1} - x'_i, y'_{i+1} - y'_i) \right)$$

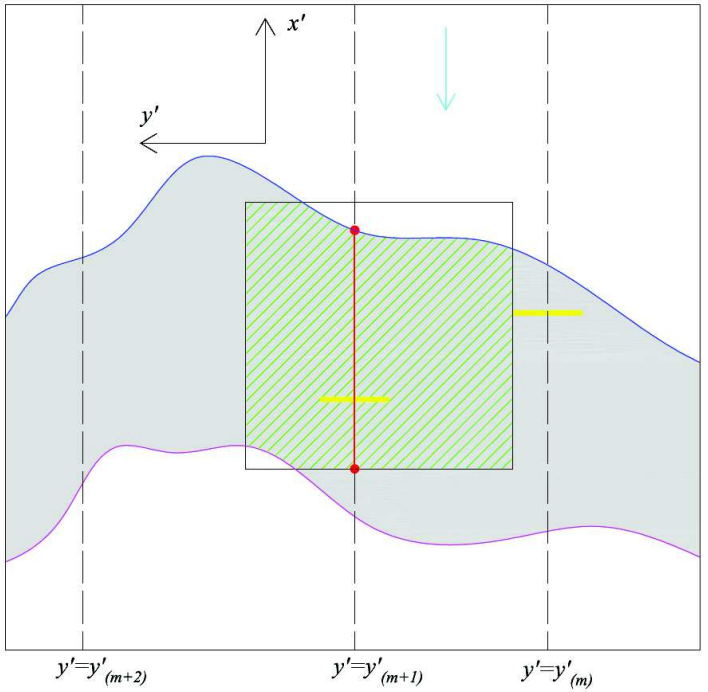




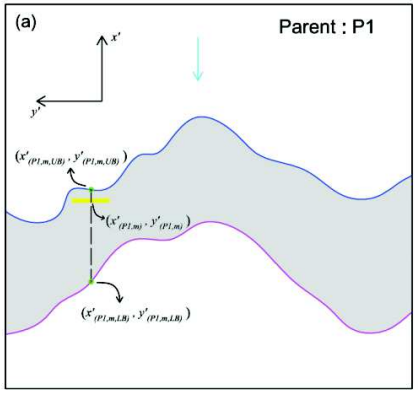
Exemple de mise en œuvre n°2
Utilisation d'un algorithme génétique (1)

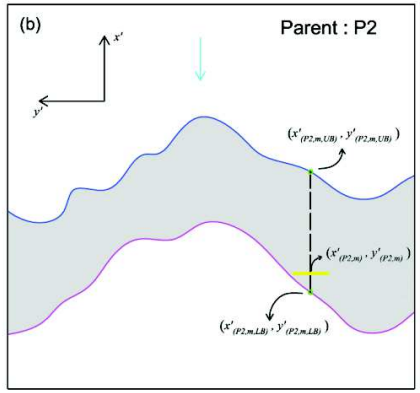
Points initiaux et mutation

- Sélection des points initiaux dans la zone de bathymétrie admissible
- Contrainte = distance minimale entre éléments
- Mutation = tirage aléatoire (uniforme) à x fixé dans la zone admissible (en rouge)



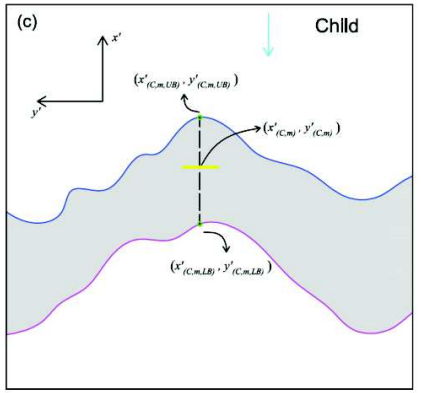
Exemple de mise en œuvre n°2
Utilisation d'un algorithme génétique (2)





Cross-over

- Moyenne normalisée (relativement à la bande de confiance des parents) des positions des parents



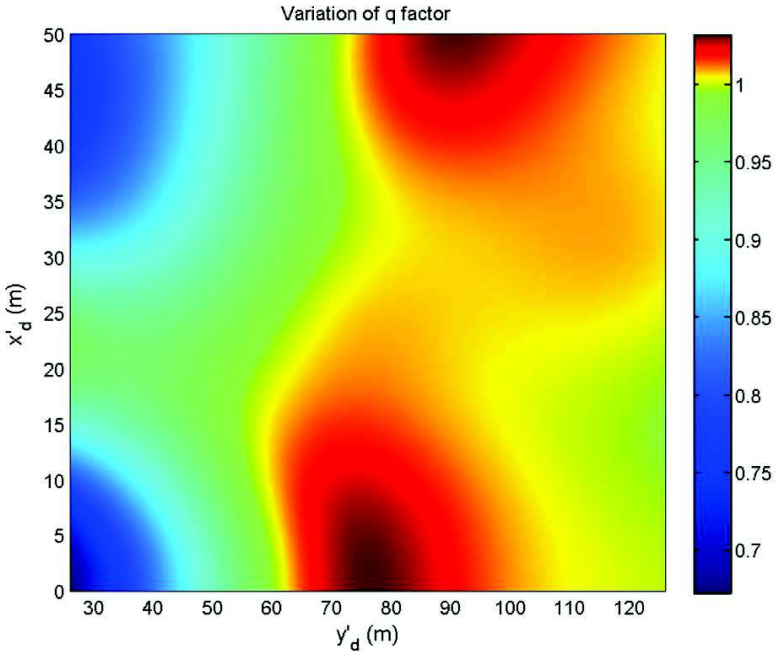
Exemple de mise en œuvre n°2
Utilisation d'un algorithme génétique (3)

Résultats

- Ferme de 40 WEC, distances optimales sur une répartition équidistante dans une configuration en quinconce

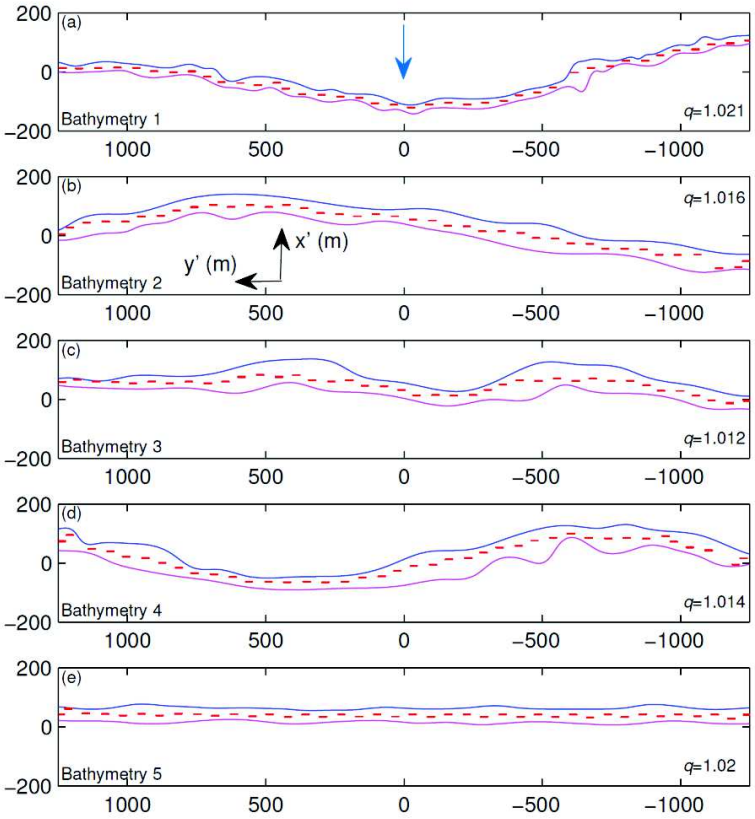
Table 3: Optimal $\hat{q}$ factor with 16 million predictions

Bathymetry no.	Genetic Algorithm	Monte Carlo
1	1.021	0.988
2	1.016	0.986
3	1.012	0.987
4	1.014	0.987
5	1.02	0.99

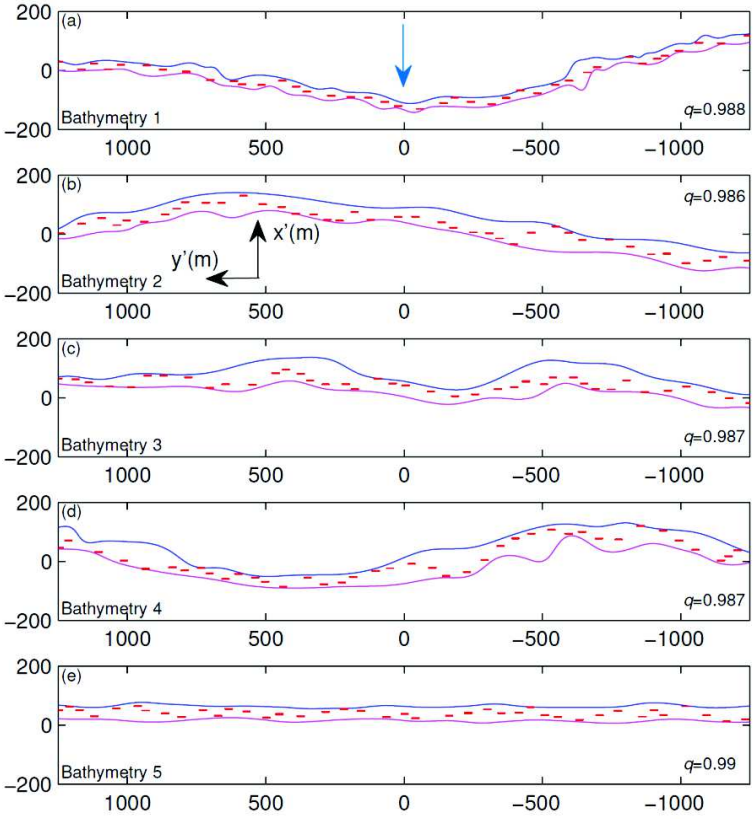


Exemple de mise en œuvre n°2
comparaison avec l'approche monte-carlo

Approche Algorithme Génétique



Approche Monte-Carlo



Nouvelles perspectives sur l'optimisation séquentielle et les plans d'expérience grâce au machine learning

Nicolas Vayatis (CMLA, ENS Paris-Saclay & CNRS)

Séminaire IRT SystemX - 28 juin 2017

Who’s painting is this?

Example 1 - The next Rembrandt

Example 2 - Deep hyper-learning

Examples of hyper-parameters in DL

- Initial learning rate
- Learning rate decrease rate
- Number of layers
- Layer size
- Non-linear activation function
- Output non-linearity
- Output cost function
- Minibatch size
- Skip connections
- Dropout probability
- L1 regularizer, L2 regularizer
- Max weight vector norm
- Pre-training algorithm

Other hyper-parameters:

- Pre-training hyper-parameters
- Momentum
- Gradient clipping norm
- Early stopping patience
- Input normalization
- Input dimensionality reduction
- Convolution kernels widths
- Convolutions stride
- Pooling windows sizes
- Pooling strides
- Number of shared layers in multi-task settings
- Output layer regularizer
- Embeddings dimension

- From Bengio, ICML'14, AutoML workshop

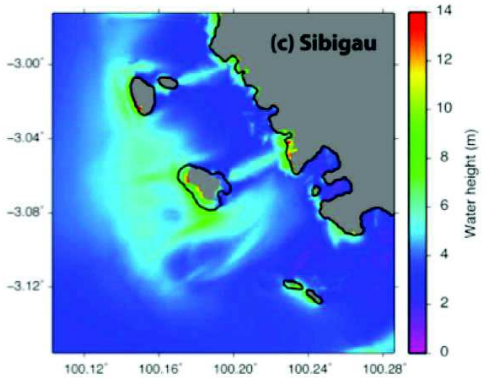
The problem: some type of sequential « regression »

- Training data are evaluations of past ‘prototypes’
- Learn/Control/Design amounts to further sampling given performance feedback
- Regression: mapping of a design space (input) on a performance space (output)
- Four important characteristics
  - Unknown regularity of the objective
  - Small samples
  - The user controls the sampling of the design space
  - Sequential aspect of scientific exploration
- Often the problem is reduced to optimization rather than regression…

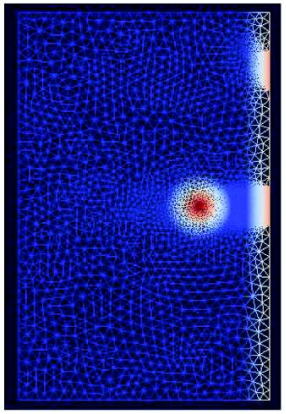
Example 3 – computer experiments

Contexte :

→ Impact de la présence d'un obstacle au large de la côte



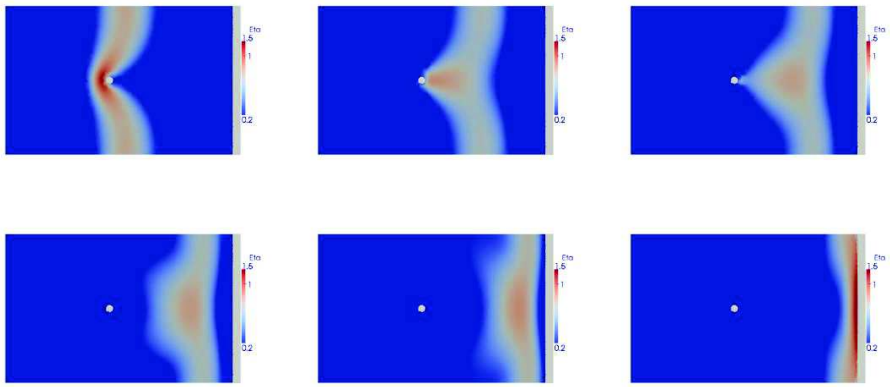
2010 Sumatra tsunami and the Mentawai Islands (Hill et al., 2012



Adaptive mesh grid of the VOLNA solver

Implémentation :

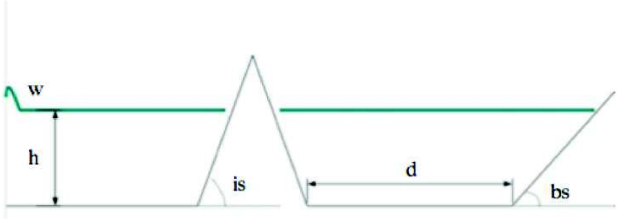
→ Equations de Saint-Venant
→ Solveur VOLNA (2007-)
Travaux de Dias-Dutykh-Poncet
→ Adaptation par T. Stefanakis (2013)

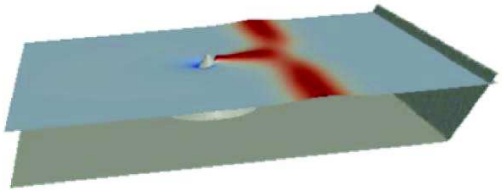


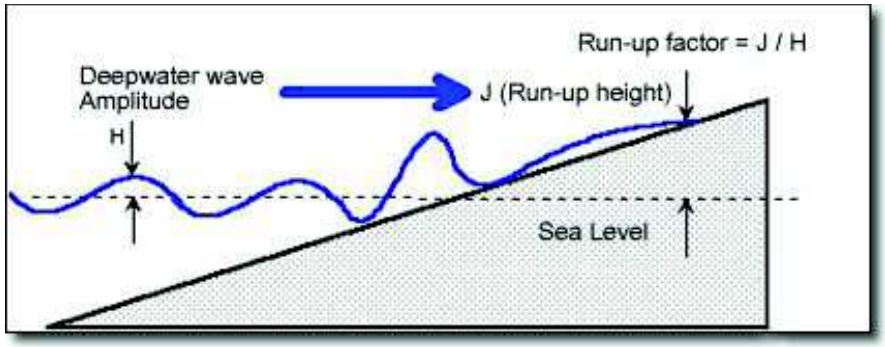
Example 3 – computer experiments

Input X

- Caractéristiques de la vague (conditions initiales)
- Topographie de l'îlot
- Bathymétrie







Output Y

- Amplification du runup = ratio du runup entre une position derrière l'îlot et une position éloignée

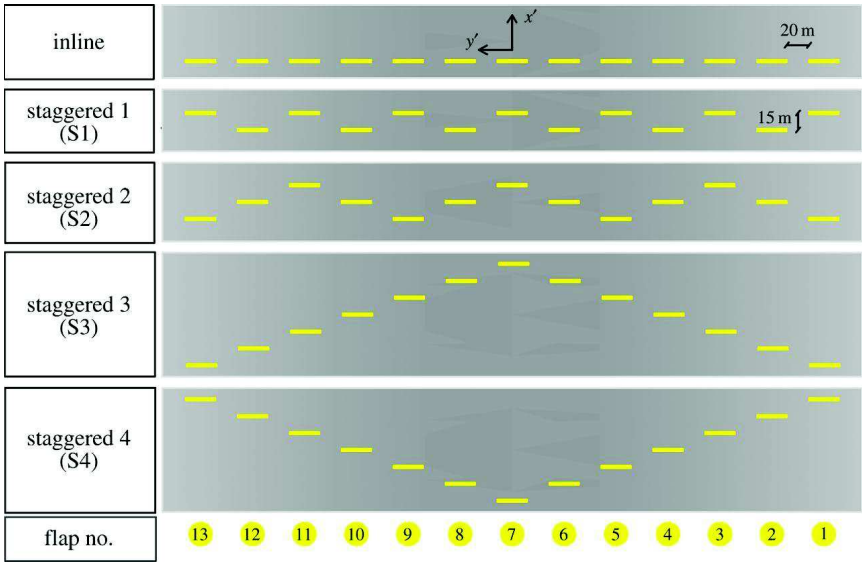
Example 4 – system design

Contexte :

→ Optimisation de la production d'énergie par une ferme de WEC

Exemple :

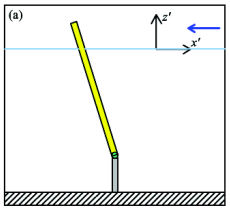
→ Systèmes hydroliens off-shores
(WEC = Wave energy converter)

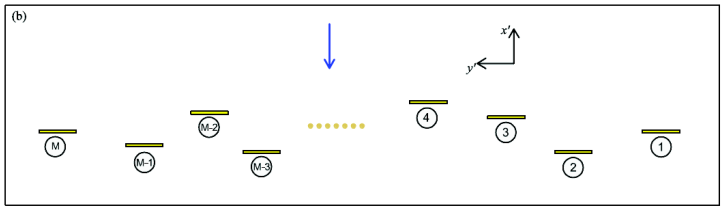


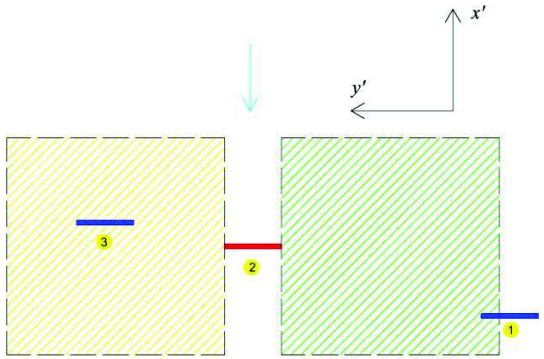
Example 4 – system design

Input X

- Caractéristiques typiques des vagues
- Bathymétrie
- Position relative des WEC







Output Y

- Energie produite (‘q factor’) :

$$q = \frac{P_{groupe} - P_{isolé}}{P_{isolé}}$$

Overview/Mathematical topics

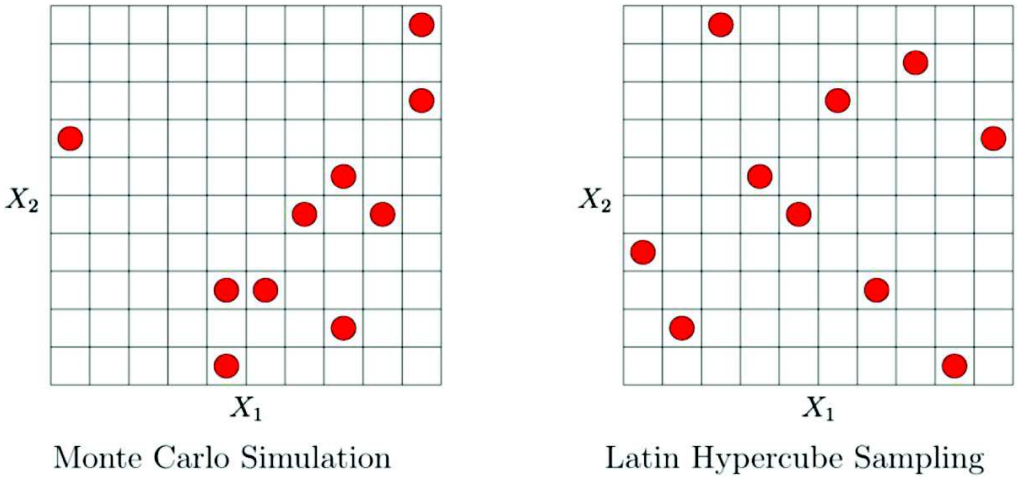
1. Experimental design → if no supervision
2. Scoring and ranking → if weak supervision
3. Sequential (global) optimization → if full supervision
    a) Parametric approach: gaussian processes
    b) Nonparametric approach: machine learning
        i. Global optimization of Lipschitz functions
        ii. Ranking strategy for nonsmooth functions

Overview/Mathematical topics

1. Experimental design
2. Scoring and ranking
3. Sequential (global) optimization
    a) Parametric approach: gaussian processes
    b) Nonparametric approach: machine learning
        i. Global optimization of Lipschitz functions
        ii. Ranking strategy for nonsmooth functions

Experimental design (or DOE)

- Schemes: random vs. deterministic, optimal designs, etc
- Quite popular: Latin Hybercube Sampling



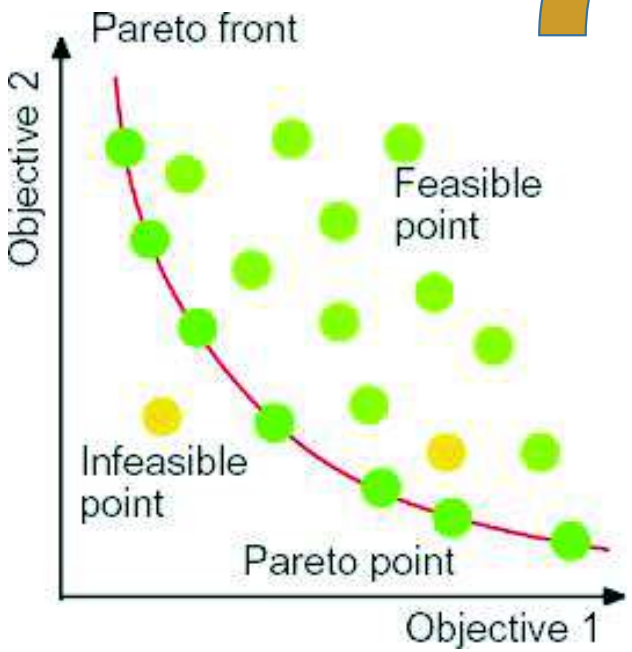
- Trade-off: Space-filling vs. objective-driven designs
- What if : high dimensional or non-euclidean design space?

Overview/Mathematical topics

1. Experimental design
2. Scoring and ranking
3. Sequential (global) optimization
    a) Parametric approach: gaussian processes
    b) Nonparametric approach: machine learning
        i. Global optimization of Lipschitz functions
        ii. Ranking strategy for nonsmooth functions

De l’optimisation multi-critères au scoring

Espace des sorties Y



Binarisation de l’espace

- On remplace l’objectif bi-critère par un indicateur binaire Z après partitionnement de l’espace des Y
  - Si Y est admissible alors Z=1
  - Sinon Z=0
- On applique une méthode de machine learning ‘sur-mesure’ pour la classification binaire sur les couples de données (X,Z)
- La classification binaire est un problème traité efficacement par les algorithmes de machine learning

A Machine Learning approach to Scoring

Scoring presentation

Overview/Mathematical topics

1. Experimental design
2. Scoring and ranking
3. Sequential (global) optimization
    a) Parametric approach: gaussian processes
    b) Nonparametric approach: machine learning
        i. Global optimization of Lipschitz functions
        ii. Ranking strategy for nonsmooth functions

Gaussian processes

GP presentation

Back to example 3 – system design

System design presentation

Overview/Mathematical topics

1. Experimental design
2. Scoring and ranking
3. Sequential (global) optimization
    a) Parametric approach: gaussian processes
    b) Nonparametric approach: machine learning
        i. Global optimization of Lipschitz functions
        ii. Ranking strategy for nonsmooth functions

Global optimization

- Maximization of a deterministic function f over $\mathcal{X}$

$$x^\star = \arg\max_{x \in \mathcal{X}} f(x)$$

An algorithm *sequentially* returns X_{n+1} given $\{(X_i, f(X_i))\}_{i=1}^n$

- Assumptions
  - $\mathcal{X}$ is a compact and convex subset of $\mathbb{R}^d$ (e.g. $\mathcal{X} = [0,1]^d$)
  - $f \in \mathcal{C}^0(\mathcal{X}, \mathbb{R})$
  - f admits a unique maximum $x^\star$ over $\mathcal{X}$

- Notations
  - d is the dimension (nbr of parameters we want to optimize)
  - $f_{\max} = f(x^\star)$
  - $X_1, X_2, \dots X_n$ refers to a sequence generated by an algorithm

Consistency and Pure Random Search

- We say that an algorithm is consistent if for any $f \in \mathcal{C}^0(\mathcal{X}, \mathbb{R})$,

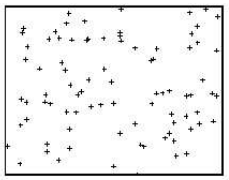
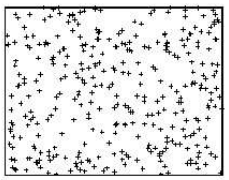
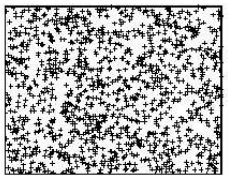
$$f_{\max} - \max\{f(X_i)\}_{i=1}^n \xrightarrow{\mathbb{P}} 0.$$

- Equivalently, an algorithm is consistent iff. for any $f \in \mathcal{C}^0(\mathcal{X}, \mathbb{R})$

$$\sup_{x \in \mathcal{X}} \min\{\|X_i - x\|_2\}_{i=1}^n \xrightarrow{\mathbb{P}} 0.$$

- Pure Random Search (PRS): Let $\mathcal{X} = [0,1]^d$ and $\{X_i\}_{i=1}^n \overset{\text{iid}}{\sim} \mathcal{U}(\mathcal{X})$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$,

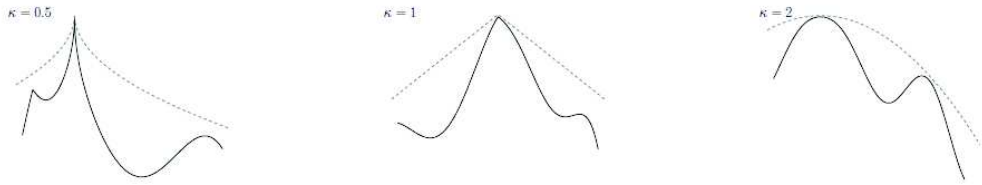
$$\sup_{x \in \mathcal{X}} \min\{\|x - X_i\|_2\}_{i=1}^n \leq 2\sqrt{d}\left(\frac{\ln(n/\delta)}{n}\right)^{\frac{1}{d}}.$$

Pure Random Search – convergence analysis

- Regularity around the maximum: Assume that there exists $c_1, c_2 > 0$ and $\kappa > 0$ such that for any $x \in \mathcal{X}$

$$c_1 \|x - x^\star\|_2^\kappa \leq f_{\max} - f(x) \leq c_2 \|x - x^\star\|_2^\kappa$$



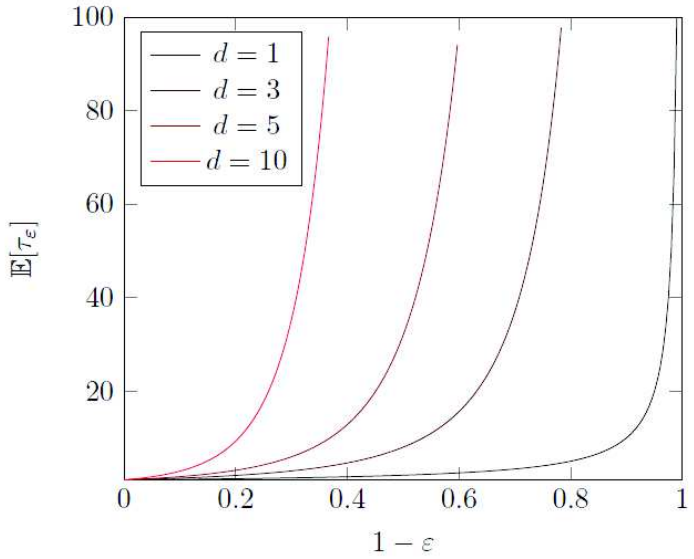
- The ratio κ/d controls the rate of convergence. If $\{X_i\}_{i=1}^n \overset{\text{iid}}{\sim} \mathcal{U}(\mathcal{X})$ and f satisfies the previous assumption then for any $\delta \in (0,1)$ there exists $c_{1,\delta}, c_{2,\delta}$ such that with probability at least $1 - 2\delta$,

$$c_{1,\delta} \left(\frac{\delta}{n}\right)^{\frac{\kappa}{d}} \leq f_{\max} - \max\{f(X_i)\}_{i=1}^n \leq c_{2,\delta} \left(\frac{\ln(1/\delta)}{n}\right)^{\frac{\kappa}{d}}.$$

Pure Random Search – empirical performance

- We want to optimize $f(x) = 1 - \|x\|_2$ over $\mathcal{X} = B_{\|\cdot\|_2}(\mathbf{0}, 1)$.
- Let τ_ε be the waiting time to have a precision ε

$$\tau_\varepsilon = \inf\{n \in \mathbb{N}^\star : \max\{f(X_i)\}_{i=1}^n \geq f_{\max} - \varepsilon\}$$



- Here $\mathbb{E}[\tau_\varepsilon] = \frac{1}{\varepsilon^d}$

Overview/Mathematical topics

1. Experimental design
2. Scoring and ranking
3. Sequential (global) optimization
    a) Parametric approach: gaussian processes
    b) Nonparametric approach: machine learning
        i. Global optimization of Lipschitz functions
        ii. Ranking strategy for nonsmooth functions

The case of Lipschitz functions

Lipschitz continuity:

- We want to take into account that for any close points x, x' with regards to $\|\cdot\|_2$ the function f has close associated values
- It can be formulated in terms of Lipschitz continuity.
  Assumption: there exists $k \in \mathbb{R}^+$ such that

$$|f(x) - f(x')| \leq k \|x - x'\|_2 \text{ for any } (x, x') \in \mathcal{X}^2$$

Questions:

1. If k is known, what kind of improvement can we have ?
2. If k is unkown: does it really help ?

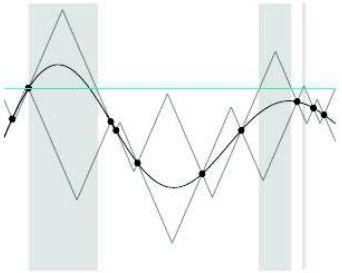
Notations

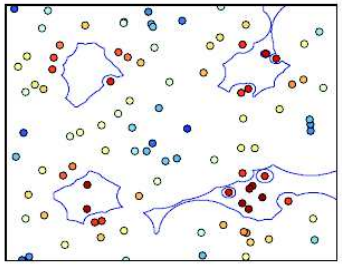
- Assumption: $f \in Lip(k)$ with known k. What can we say given a sample $\{(X_i, f(X_i))\}_{i=1}^n$?
- We define the set of functions that explains the sample

$$\mathcal{F}_n = \{f' \in Lip(k) : \ \forall i \in \{1 \dots n\}, \ f'(X_i) = f(X_i)\}$$

- The set of points still on the run to maximize f

$$\textsc{ArgMax}(\mathcal{F}_n) = \{x \in \mathcal{X} : \exists f \in \mathcal{F}_n \text{ s.t. } x \in \arg\max_{x \in \mathcal{X}} f(x)\}$$



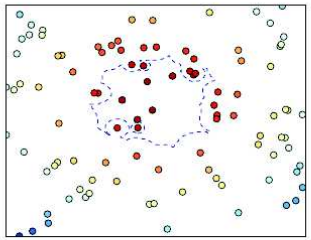


Algorithmic principle when k is known

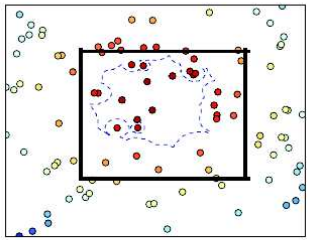
LipOpt(k, n, f, $[0,1]^d$)

Init: $\hat{\mathcal{X}} = [0,1]^d$, $t, u \leftarrow 0$, $\mathcal{F}_0 \leftarrow Lip(k)$
While $t < n$
  Let $X_{t+1} \sim \mathcal{U}(\hat{\mathcal{X}})$, request $f(X_{t+1})$, $t \leftarrow t+1$
  $\mathcal{F}_t = \{f' \in \mathcal{F}_{t-1} : f'(X_t) = f(X_t)\}$
  If $\mathrm{diam}(\textsc{ArgMax}(\mathcal{F}_t)) \leq 2^{-(u+2)}$
    $\hat{\mathcal{X}} \leftarrow$ hypercube of diameter $\sqrt{d}2^{-(u+1)}$ containing $\textsc{ArgMax}(\mathcal{F}_t)$, $u \leftarrow u+1$

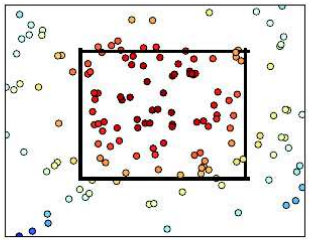
- LipOpt iterates 3 steps



Compute $\textsc{ArgMax}$



Update $\hat{\mathcal{X}}$



Sample in $\hat{\mathcal{X}}$

Rates of convergence when k is known

- Under the same assumptions, if $\{X_i\}_{i=1}^n$ is a sequence generated by LipOpt$(n, f, k, \mathcal{X})$, for any $\delta \in (0,1)$, there exists c_4 such that with probability at least $1-\delta$,

$$c_4 \cdot \exp\left(-\frac{n}{d}\right) \leq f_{\max} - \max\{f(X_i)\}_{i=1}^n$$

- There also exists c_1, c_2, c_3 such that with probability at least $1-\delta$

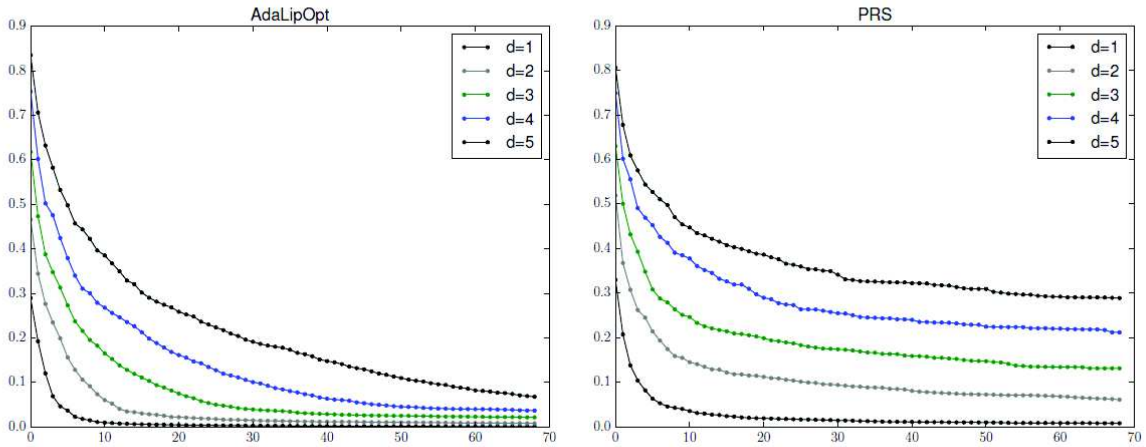
$$f_{\max} - \max\{f(X_i)\}_{i=1}^n \leq \begin{cases} c_1 \cdot \exp\left(-\frac{n\ln(2)}{\ln(n/\delta)(32k\sqrt{d}/c)^d}\right) & \text{if } \kappa = 1 \\ c_2 \cdot n^{-\frac{\kappa}{d(\kappa-1)}} & \text{if } \kappa \in (1,2) \\ c_3 \cdot n^{-\frac{\kappa}{d}} & \text{if } \kappa > 2 \end{cases}$$

Adaptive case when k is not known

- An adaptive version of the algorithm has been proposed in the case the Lipschitz constant is not known
- The adaptive version is consistent
- Surprisingly, convergence rates have the same order of magnitude with worst constants

Numerical example

- We consider $f(x) = 1 - \|x\|_2$, where $d = 1, 2, 3, 4, 5$.



Regret as a function of the number of iterations

Overview/Mathematical topics

1. Experimental design
2. Scoring and ranking
3. Sequential (global) optimization
    a) Parametric approach: gaussian processes
    b) Nonparametric approach: machine learning
        i. Global optimization of Lipschitz functions
        ii. Ranking strategy for nonsmooth functions

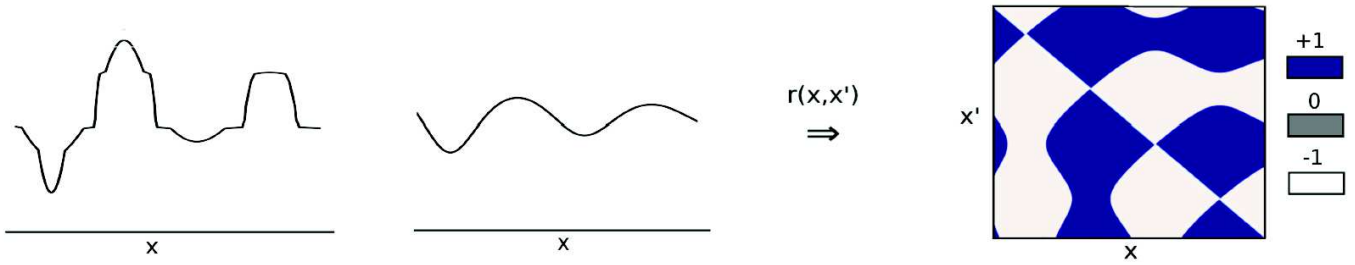
A ranking approach to global optimization

Maximizing f $\approx$ learning the induced ranking rule r_f

Definition: The ranking rule $r_f : \mathcal{X} \times \mathcal{X} \to \{-1, 0, +1\}$ induced by a function $f : \mathcal{X} \to \mathbb{R}$ is defined as

$$r_f : (x, x') \mapsto \text{sign}\left(f(x) - f(x')\right)$$

Example: different functions with the **same ranking rule**

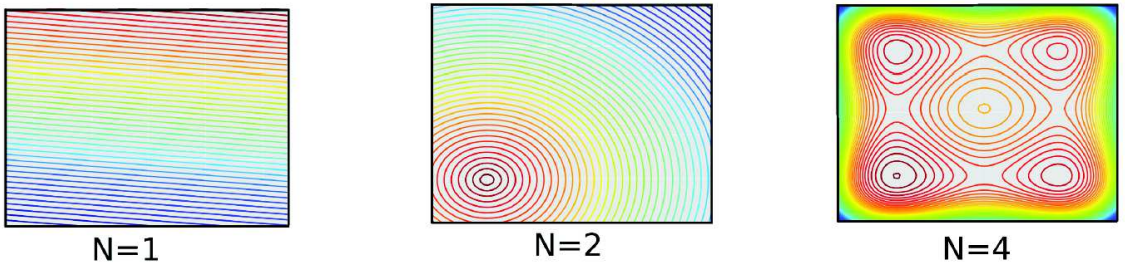


Ranking rules

- We **define collections of ranking rules** to characterize the complexity of a function

> **Example:** The set of **polynomial ranking rules** of degree N is defined as
>
> $$\mathcal{R}_N = \{r(x, x') = \mathrm{sgn}(f(x) - f(x')) \mid f \text{ polynom of degree } N\}$$

- **Tuning N** allows to adjust the complexity of the level sets of f



Some 2-d function with polynomial ranking

Algorithmic principle

- **Assumption:** the *true* ranking rule r_f induced by the unknown function belongs to $\mathcal{R}$

Algorithm: given $\{(X_i, f(X_i))\}_{i=1}^n$ and $\mathcal{R}$ repeat

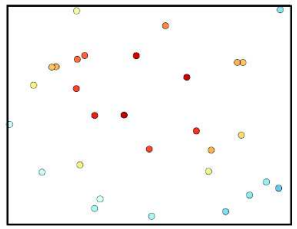
1. Define the set of **consistent ranking rules** [Dasgupta 2009]:

$$\mathcal{R}_n = \{r \in \mathcal{R} : \sum_{i,j}^{n} \mathbb{I}\{r(X_i, X_j) \neq \text{sgn}(f(X_i) - f(X_j))\} = 0\}$$

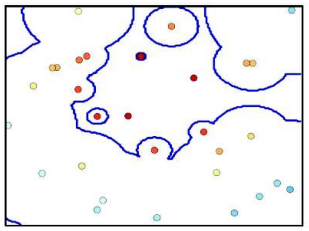
2. Estimate the set of **potential maximizers**

$$\hat{\mathcal{X}}_n = \{x \in \mathcal{X} : \exists r \in \mathcal{R}_n \text{ s.t. } r(x, X_i) \geq 0, \forall i \leq n\}$$

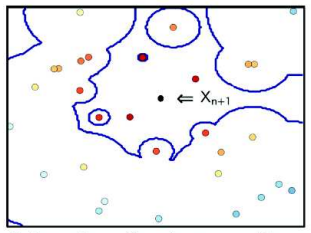
3. Sample $X_{n+1} \sim U(\hat{\mathcal{X}}_n)$ and evaluate $f(X_{n+1})$



1. Estimate consistent rankings



2. Estimate potential maxima



3. Sample uniformly among the potential maxima

Theoretical guarantees

Proposition: Let X_n be the ouput of the algorithm. If:

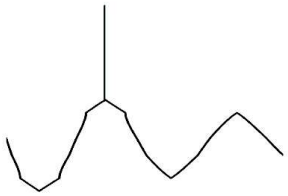
1. the *hypothesis* is true: $r_f \in \mathcal{R}$

2. the maximum of f is not *a spike*

then, we have that,

$$f(X_n) \xrightarrow{\mathbb{P}} \max_{x \in \mathcal{X}} f(x).$$

Good



Not Good

Main theorem

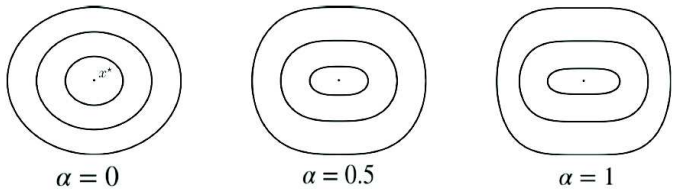
Theorem: Let X_n be the ouput of the algorithm. If:

1. the *hypothesis* is true: $r_f \in \mathcal{R}$
2. f has a **unique** maximum $x^\star$
3. f has α-**regular level-sets**, i.e. $\forall f^{-1}(y) = \{x \in \mathcal{X} : f(x) = y\}$

$$\max_{x \in f^{-1}(y)} \|x^\star - x\|_2 \leq c_\alpha \cdot \min_{x \in f^{-1}(y)} \|x^\star - x\|_2^{1/(1+\alpha)},$$

then, with probability at least $1 - \delta/2$,

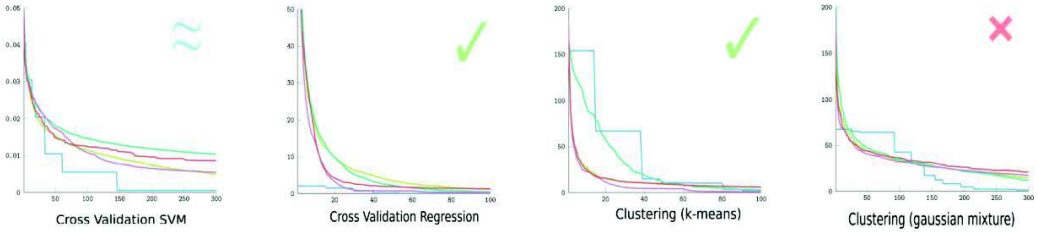
$$c_1 \cdot \delta^{1/d} \cdot e^{-\frac{n(1+\alpha)^2}{d}} \leq \|x^\star - X_n\|_2 \leq c_2 \cdot \left(\frac{\ln(1/\delta)}{n}\right)^{\frac{1}{d(1+\alpha)^2}}$$



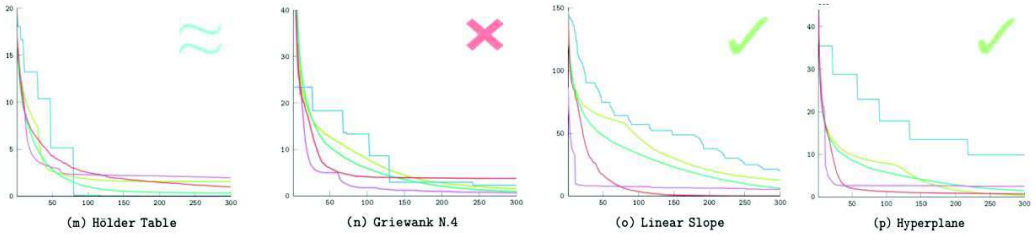
Numerical experiments

- **Comparison** with stat-of-the-art algorithms:
  - AdaRank [This talk]
  - BayesOpt [Martinez 2014]
  - CMA-ES [Hansen 2003]
  - CRS [Price 1983]
  - DIRECT [Jones 1993]

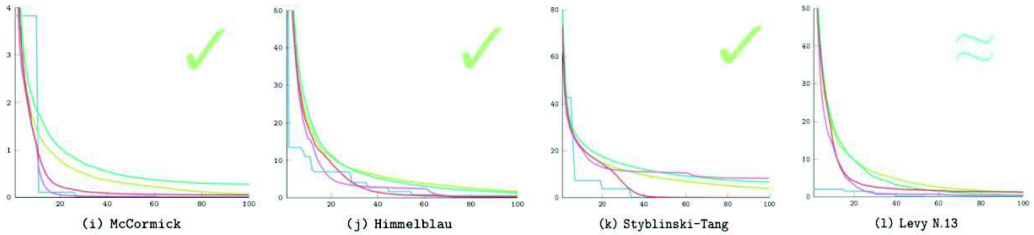
- **Cross validations problems (Real data sets)**



- **Synthetic** high-dimensional problems [AdaRank, BayesOpt, CMA-ES, CRS, DIRECT]



- **Synthetic** 2-d non-convex problems [AdaRank, BayesOpt, CMA-ES, CRS, DIRECT]



Difference between the true maximum and estimation in terms of iterations

Some other hot topics

- Privacy preserving learning
- Transfer learning
- Continuous learning
- Explainable learning

Nouvelles perspectives sur l'optimisation séquentielle et les plans d'expérience grâce au machine learning

Nicolas Vayatis (CMLA, ENS Paris-Saclay & CNRS)

Séminaire IRT SystemX - 28 juin 2017

Who’s painting is this?

Example 1 - The next Rembrandt

Example 2 - Deep hyper-learning

Examples of hyper-parameters in DL

- Initial learning rate
- Learning rate decrease rate
- Number of layers
- Layer size
- Non-linear activation function
- Output non-linearity
- Output cost function
- Minibatch size
- Skip connections
- Dropout probability
- L1 regularizer, L2 regularizer
- Max weight vector norm
- Pre-training algorithm

Other hyper-parameters:

- Pre-training hyper-parameters
- Momentum
- Gradient clipping norm
- Early stopping patience
- Input normalization
- Input dimensionality reduction
- Convolution kernels widths
- Convolutions stride
- Pooling windows sizes
- Pooling strides
- Number of shared layers in multi-task settings
- Output layer regularizer
- Embeddings dimension

- From Bengio, ICML'14, AutoML workshop

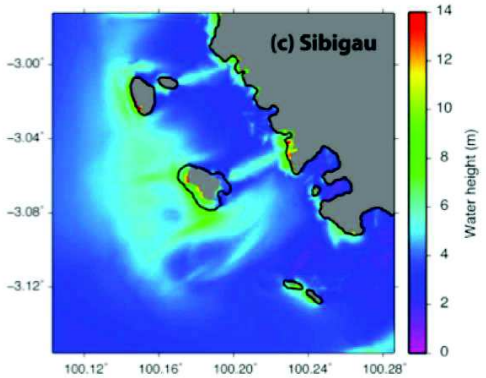
The problem: some type of sequential « regression »

- Training data are evaluations of past ‘prototypes’
- Learn/Control/Design amounts to further sampling given performance feedback
- Regression: mapping of a design space (input) on a performance space (output)
- Four important characteristics
  - Unknown regularity of the objective
  - Small samples
  - The user controls the sampling of the design space
  - Sequential aspect of scientific exploration
- Often the problem is reduced to optimization rather than regression…

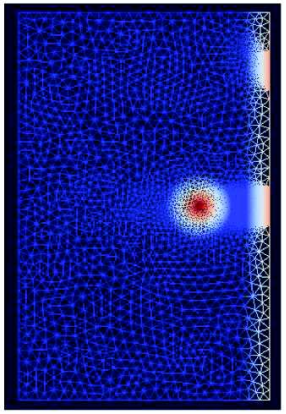
Example 3 – computer experiments

Contexte :

→ Impact de la présence d'un obstacle au large de la côte



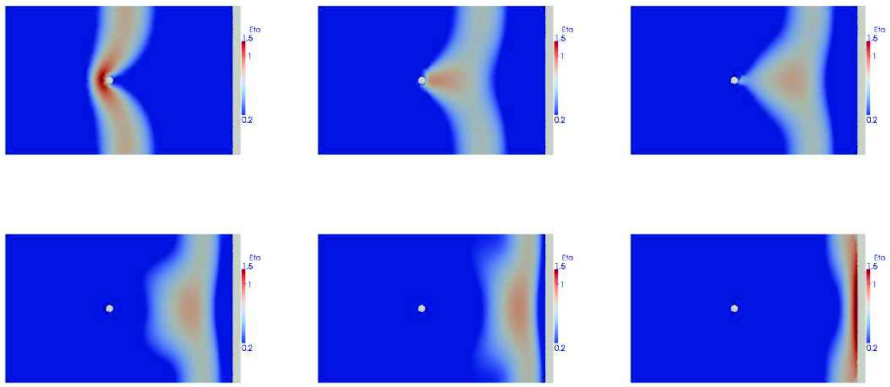
2010 Sumatra tsunami and the Mentawai Islands (Hill et al., 2012



Adaptive mesh grid of the VOLNA solver

Implémentation :

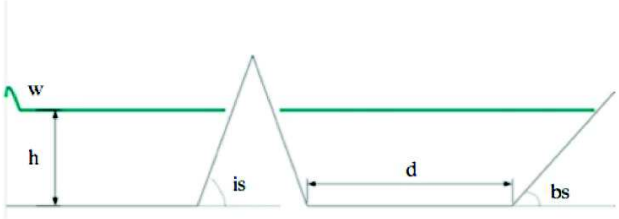
→ Equations de Saint-Venant
→ Solveur VOLNA (2007-)
Travaux de Dias-Dutykh-Poncet
→ Adaptation par T. Stefanakis (2013)

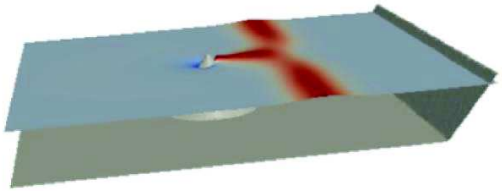


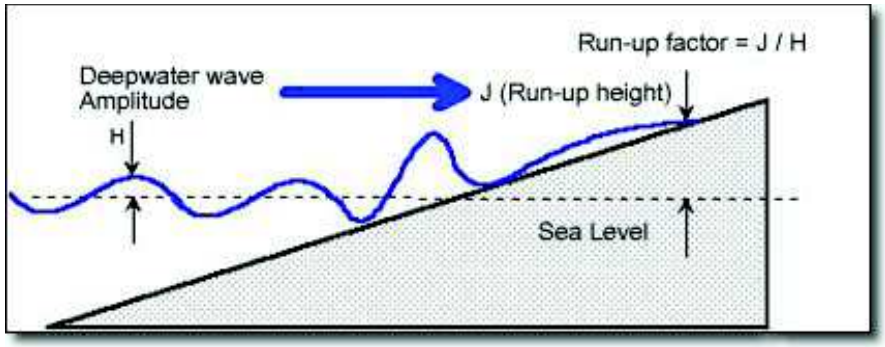
Example 3 – computer experiments

Input X

- Caractéristiques de la vague (conditions initiales)
- Topographie de l'îlot
- Bathymétrie







Output Y

- Amplification du runup = ratio du runup entre une position derrière l'îlot et une position éloignée

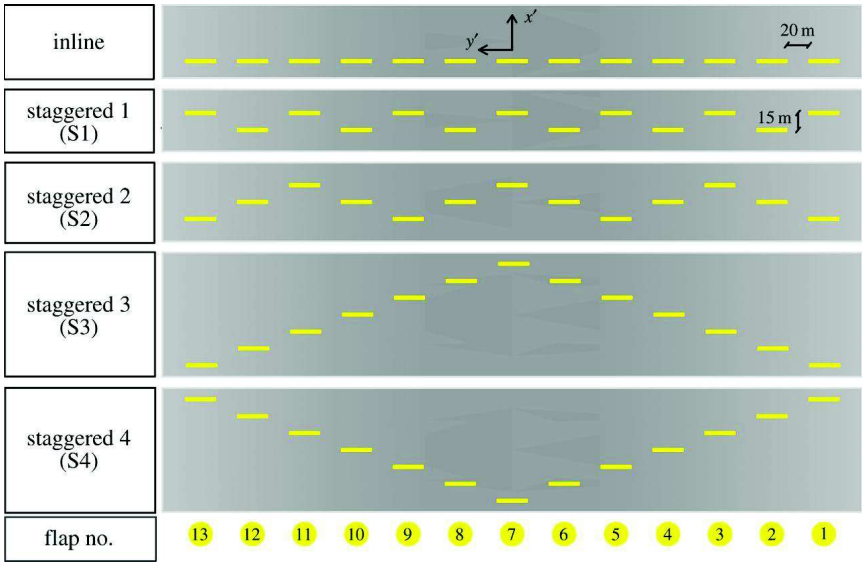
Example 4 – system design

Contexte :

→ Optimisation de la production d'énergie par une ferme de WEC

Exemple :

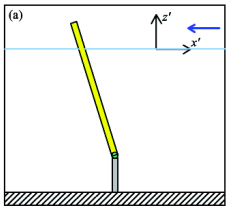
→ Systèmes hydroliens off-shores
(WEC = Wave energy converter)

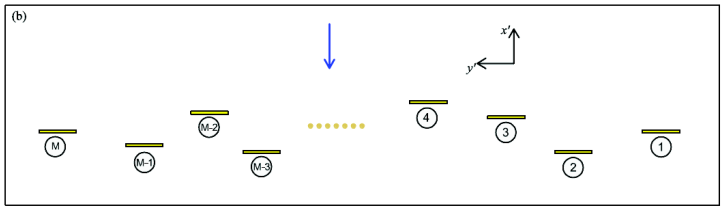


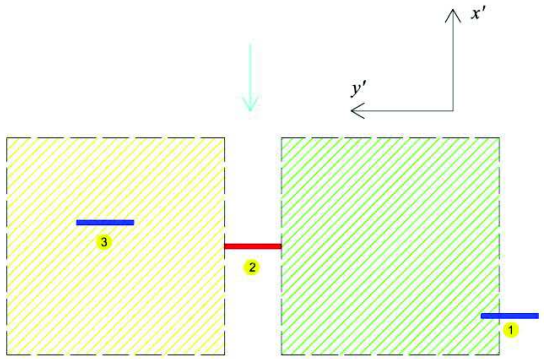
Example 4 – system design

Input X

- Caractéristiques typiques des vagues
- Bathymétrie
- Position relative des WEC







Output Y

- Energie produite (‘q factor’) :

$$q = \frac{P_{groupe} - P_{isolé}}{P_{isolé}}$$

Overview/Mathematical topics

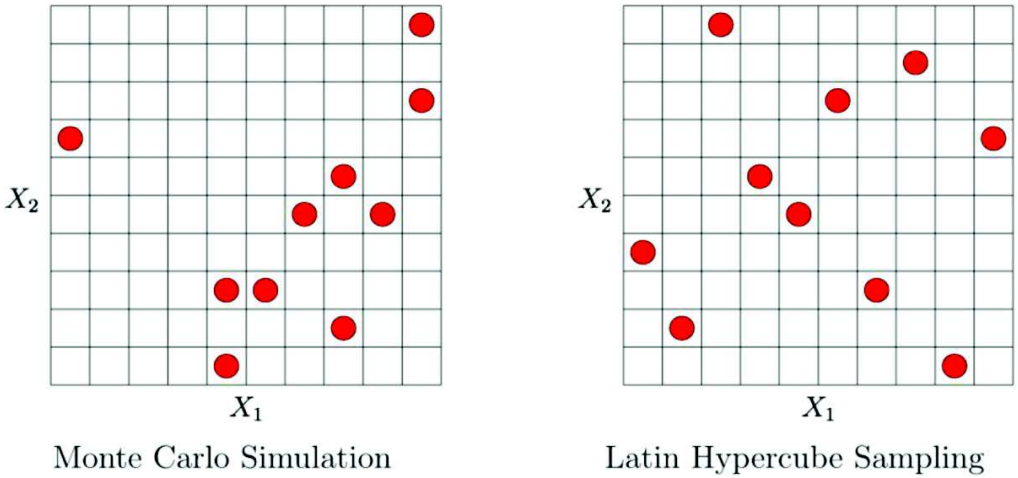
1. Experimental design → if no supervision
2. Scoring and ranking → if weak supervision
3. Sequential (global) optimization → if full supervision
    a) Parametric approach: gaussian processes
    b) Nonparametric approach: machine learning
        i. Global optimization of Lipschitz functions
        ii. Ranking strategy for nonsmooth functions

Overview/Mathematical topics

1. Experimental design
2. Scoring and ranking
3. Sequential (global) optimization
    a) Parametric approach: gaussian processes
    b) Nonparametric approach: machine learning
        i. Global optimization of Lipschitz functions
        ii. Ranking strategy for nonsmooth functions

Experimental design (or DOE)

- Schemes: random vs. deterministic, optimal designs, etc
- Quite popular: Latin Hybercube Sampling



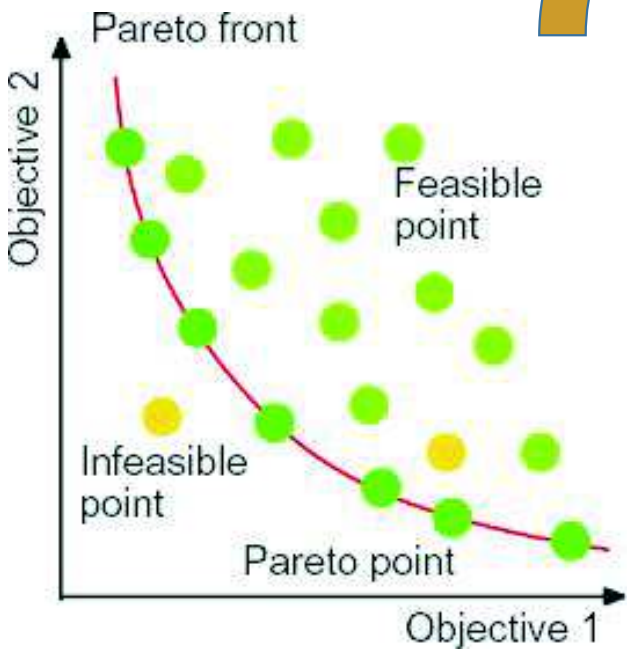
- Trade-off: Space-filling vs. objective-driven designs
- What if : high dimensional or non-euclidean design space?

Overview/Mathematical topics

1. Experimental design
2. Scoring and ranking
3. Sequential (global) optimization
    a) Parametric approach: gaussian processes
    b) Nonparametric approach: machine learning
        i. Global optimization of Lipschitz functions
        ii. Ranking strategy for nonsmooth functions

De l’optimisation multi-critères au scoring

Espace des sorties Y



Binarisation de l’espace

- On remplace l’objectif bi-critère par un indicateur binaire Z après partitionnement de l’espace des Y
  - Si Y est admissible alors Z=1
  - Sinon Z=0
- On applique une méthode de machine learning ‘sur-mesure’ pour la classification binaire sur les couples de données (X,Z)
- La classification binaire est un problème traité efficacement par les algorithmes de machine learning

A Machine Learning approach to Scoring

Scoring presentation

Overview/Mathematical topics

1. Experimental design
2. Scoring and ranking
3. Sequential (global) optimization
    a) Parametric approach: gaussian processes
    b) Nonparametric approach: machine learning
        i. Global optimization of Lipschitz functions
        ii. Ranking strategy for nonsmooth functions

Gaussian processes

GP presentation

Back to example 3 – system design

System design presentation

Overview/Mathematical topics

1. Experimental design
2. Scoring and ranking
3. Sequential (global) optimization
    a) Parametric approach: gaussian processes
    b) Nonparametric approach: machine learning
        i. Global optimization of Lipschitz functions
        ii. Ranking strategy for nonsmooth functions

Global optimization

- Maximization of a deterministic function f over $\mathcal{X}$

$$x^\star = \arg\max_{x \in \mathcal{X}} f(x)$$

An algorithm *sequentially* returns X_{n+1} given $\{(X_i, f(X_i))\}_{i=1}^n$

- Assumptions
  - $\mathcal{X}$ is a compact and convex subset of $\mathbb{R}^d$ (e.g. $\mathcal{X} = [0,1]^d$)
  - $f \in \mathcal{C}^0(\mathcal{X}, \mathbb{R})$
  - f admits a unique maximum $x^\star$ over $\mathcal{X}$

- Notations
  - d is the dimension (nbr of parameters we want to optimize)
  - $f_{\max} = f(x^\star)$
  - $X_1, X_2, \dots X_n$ refers to a sequence generated by an algorithm

Consistency and Pure Random Search

- We say that an algorithm is consistent if for any $f \in \mathcal{C}^0(\mathcal{X}, \mathbb{R})$,

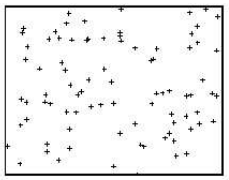
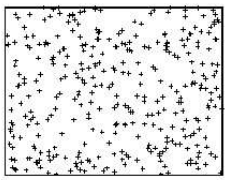
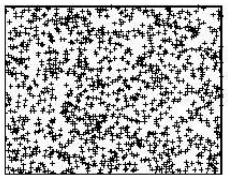
$$f_{\max} - \max\{f(X_i)\}_{i=1}^n \xrightarrow{\mathbb{P}} 0.$$

- Equivalently, an algorithm is consistent iff. for any $f \in \mathcal{C}^0(\mathcal{X}, \mathbb{R})$

$$\sup_{x \in \mathcal{X}} \min\{\|X_i - x\|_2\}_{i=1}^n \xrightarrow{\mathbb{P}} 0.$$

- Pure Random Search (PRS): Let $\mathcal{X} = [0,1]^d$ and $\{X_i\}_{i=1}^n \overset{\text{iid}}{\sim} \mathcal{U}(\mathcal{X})$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$,

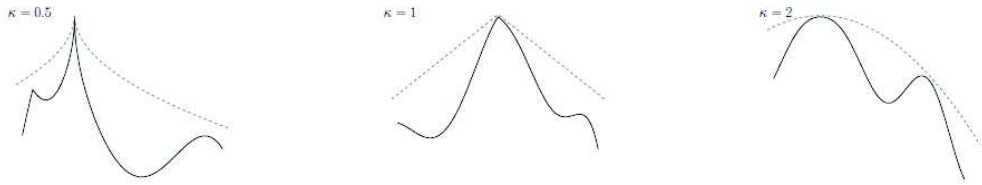
$$\sup_{x \in \mathcal{X}} \min\{\|x - X_i\|_2\}_{i=1}^n \leq 2\sqrt{d}\left(\frac{\ln(n/\delta)}{n}\right)^{\frac{1}{d}}.$$

Pure Random Search – convergence analysis

- Regularity around the maximum: Assume that there exists $c_1, c_2 > 0$ and $\kappa > 0$ such that for any $x \in \mathcal{X}$

$$c_1 \|x - x^\star\|_2^\kappa \leq f_{\max} - f(x) \leq c_2 \|x - x^\star\|_2^\kappa$$



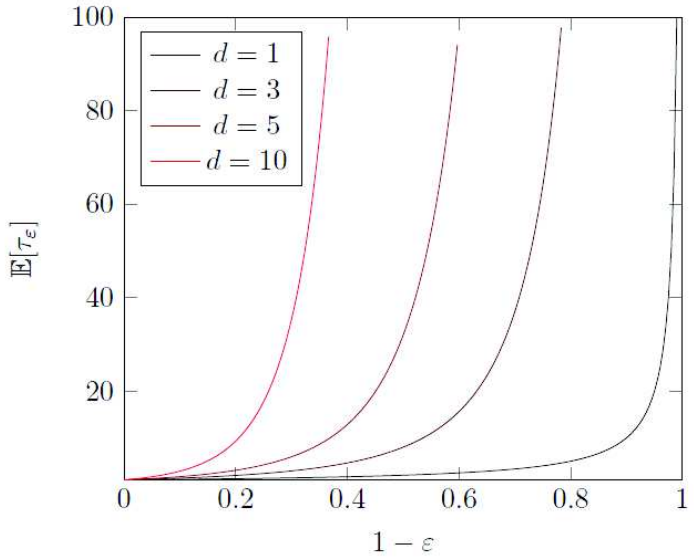
- The ratio κ/d controls the rate of convergence. If $\{X_i\}_{i=1}^n \overset{\text{iid}}{\sim} \mathcal{U}(\mathcal{X})$ and f satisfies the previous assumption then for any $\delta \in (0,1)$ there exists $c_{1,\delta}, c_{2,\delta}$ such that with probability at least $1 - 2\delta$,

$$c_{1,\delta} \left(\frac{\delta}{n}\right)^{\frac{\kappa}{d}} \leq f_{\max} - \max\{f(X_i)\}_{i=1}^n \leq c_{2,\delta} \left(\frac{\ln(1/\delta)}{n}\right)^{\frac{\kappa}{d}}.$$

Pure Random Search – empirical performance

- We want to optimize $f(x) = 1 - \|x\|_2$ over $\mathcal{X} = B_{\|\cdot\|_2}(\mathbf{0}, 1)$.
- Let τ_ε be the waiting time to have a precision ε

$$\tau_\varepsilon = \inf\{n \in \mathbb{N}^\star : \max\{f(X_i)\}_{i=1}^n \geq f_{\max} - \varepsilon\}$$



- Here $\mathbb{E}[\tau_\varepsilon] = \frac{1}{\varepsilon^d}$

Generic scheme based on active learning idea

Input: $n \in \mathbb{N}^\star$, $\mathcal{F}$, $\mathcal{X} \subset \mathbb{R}^d$, $f \in \mathcal{F}$

1. Initialization: Let $X_1 \sim \mathcal{U}(\mathcal{X})$
Evaluate $f(X_1)$, $t \leftarrow 1$
Let $\mathcal{F}_1 = \{g \in \mathcal{F} : g(X_1) = f(X_1)\}$

2. Iterations: Repeat while $t < n$:
Let $X_{t+1} \sim \mathcal{U}(\mathcal{X})$
If there exists $g \in \mathcal{F}_t$ such that $g(X_{t+1}) \geq \max_{i=1\ldots t} f(X_i)$
Evaluate $f(X_{t+1})$, $t \leftarrow t+1$
Let $\mathcal{F}_t = \{g \in \mathcal{F}_{t-1} : g(X_t) = f(X_t)\}$

3. Output: Return $X_{\hat{\imath}_n}$ where $\hat{\imath}_n \in \arg\max_{i=1\ldots n} f(X_i)$

Overview/Mathematical topics

1. Experimental design
2. Scoring and ranking
3. Sequential (global) optimization
    a) Parametric approach: gaussian processes
    b) Nonparametric approach: machine learning
        i. Global optimization of Lipschitz functions
        ii. Ranking strategy for nonsmooth functions

The case of Lipschitz functions

Lipschitz continuity:

- We want to take into account that for any close points x, x' with regards to $\|\cdot\|_2$ the function f has close associated values
- It can be formulated in terms of Lipschitz continuity.
  Assumption: there exists $k \in \mathbb{R}^+$ such that

$$|f(x) - f(x')| \leq k \|x - x'\|_2 \text{ for any } (x, x') \in \mathcal{X}^2$$

Questions:

1. If k is known, what kind of improvement can we have ?
2. If k is unkown: does it really help ?

Algorithmic principle when k is known

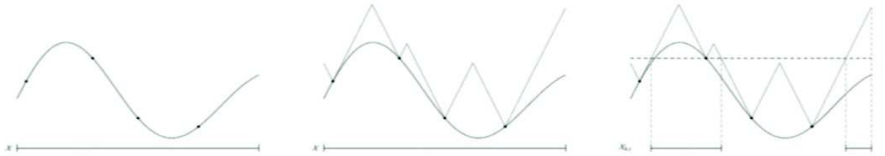


Figure 2.2: *Left:* A Lipschitz function and a sample of $t = 4$ evaluations. *Middle:* In grey, the upper bound $UB : x \mapsto \min_{i=1\ldots t} f(X_i) + k \cdot \|x - X_i\|_2$. *Right:* the set of points $\mathcal{X}_{k,t} := \{x \in \mathcal{X} : UB(x) \geq \max_{i=1\ldots t} f(X_i)\}$ which satisfy the decision rule.

Input: $n \in \mathbb{N}^\star$, $k \geq 0$, $\mathcal{X} \subset \mathbb{R}^d$, $f \in \mathrm{Lip}(k)$

1. Initialization: Let $X_1 \sim \mathcal{U}(\mathcal{X})$
Evaluate $f(X_1)$, $t \leftarrow 1$

2. Iterations: Repeat while $t < n$:
Let $X_{t+1} \sim \mathcal{U}(\mathcal{X})$
If $\min_{i=1\ldots t}(f(X_i) + k \cdot \|X_{t+1} - X_i\|_2) \geq \max_{i=1\ldots t} f(X_i)$ {Decision rule}
Evaluate $f(X_{t+1})$, $t \leftarrow t + 1$

3. Output: Return $X_{\hat{\imath}_n}$ where $\hat{\imath}_n \in \arg\max_{i=1\ldots n} f(X_i)$

Rates of convergence when k is known

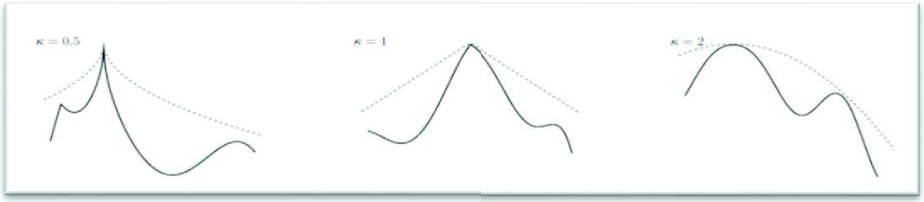
Corollary 2.1. (Upper bound) *For any $f \in \mathrm{Lip}(k)$, any $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, we have with probability at least $1-\delta$,*

$$\max_{x \in \mathcal{X}} f(x) - \max_{i=1\ldots n} f(X_i) \leq k \cdot \mathrm{diam}(\mathcal{X}) \cdot \left(\frac{\ln(1/\delta)}{n}\right)^{\frac{1}{d}}.$$

Condition 2.1. (Decreasing rate around the maximum) *A function $f : \mathcal{X} \to \mathbb{R}$ is (κ, c_κ)-decreasing around its maximum for some $\kappa \geq 0$, $c_\kappa \geq 0$ if:*

1. *The global optimizer $x^\star \in \mathcal{X}$ is unique;*
2. *For all $x \in \mathcal{X}$, we have that:*

$$f(x^\star) - f(x) \geq c_\kappa \cdot \|x - x^\star\|_2^\kappa.$$



Fast rates of convergence

Theorem 2.1. (FAST RATES) *Let $f \in \mathrm{Lip}(k)$ be any Lipschitz function satisfying Condition 2.1 for some $\kappa \geq 1, c_\kappa > 0$. Then, for any $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, we have with probability at least $1-\delta$,*

$$\max_{x\in\mathcal{X}} f(x) - \max_{i=1\ldots n} f(X_i) \leq k \cdot \mathrm{diam}(\mathcal{X}) \times \begin{cases} \exp\left\{ - C_{k,\kappa} \cdot \dfrac{n \ln(2)}{\ln(n/\delta) + 2(2\sqrt{d})^d} \right\}, & \kappa = 1, \\ \dfrac{2^\kappa}{2} \left(1 + C_{k,\kappa} \cdot \dfrac{n(2^{d(\kappa-1)} - 1)}{\ln(n/\delta) + 2(2\sqrt{d})^d} \right)^{-\frac{\kappa}{d(\kappa-1)}}, & \kappa > 1, \end{cases}$$

where $C_{k,\kappa} = (c_\kappa \max_{x\in\mathcal{X}} \|x - x^\star\|^{\kappa-1} / 8k)^d$.

Adaptive version based on a nested sequence of functional classes

Input: $n \in \mathbb{N}^\star$, $k_{i\in\mathbb{Z}}$, $\mathcal{X} \subset \mathbb{R}^d$, $f \in \bigcup_{k\geq 0} \mathrm{Lip}(k)$

1. Initialization: Let $X_1 \sim \mathcal{U}(\mathcal{X})$
Evaluate $f(X_1)$, $t \leftarrow 1$, $\hat{k}_1 \leftarrow 0$

2. Iterations: Repeat while $t < n$
Let $B_{t+1} \sim \mathcal{B}(p)$
If $B_{t+1} = 1$ (Exploration)
Let $X_{t+1} \sim \mathcal{U}(\mathcal{X})$
If $B_{t+1} = 0$ (Exploitation)
Let $X_{t+1} \sim \mathcal{U}(\mathcal{X}_{\hat{k}_t,t})$ where $\mathcal{X}_{\hat{k}_t,t}$ denotes the set of potential maximizers of Definition 2.4 computed with k set to $\hat{k}_t$

Evaluate $f(X_{t+1})$, $t \leftarrow t+1$
Let $\hat{k}_t := \inf\left\{ k_{i\in\mathbb{Z}} : \max_{i\neq j} \dfrac{|f(X_i) - f(X_j)|}{\|X_i - X_j\|_2} \leq k_i \right\}$

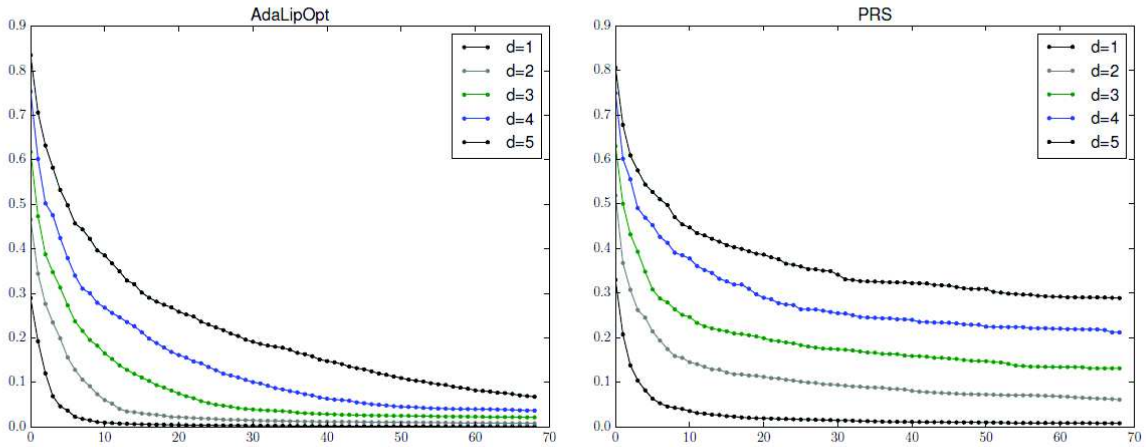
3. Output: Return $X_{\hat{\imath}_n}$ where $\hat{\imath}_n \in \arg\max_{i=1\ldots n} f(X_i)$

Adaptive case when k is not known

- An adaptive version of the algorithm has been proposed in the case the Lipschitz constant is not known
- The adaptive version is consistent
- Surprisingly, convergence rates have the same order of magnitude with worst constants

Numerical example

- We consider $f(x) = 1 - \|x\|_2$, where $d = 1, 2, 3, 4, 5$.



Regret as a function of the number of iterations

Overview/Mathematical topics

1. Experimental design
2. Scoring and ranking
3. Sequential (global) optimization
    a) Parametric approach: gaussian processes
    b) Nonparametric approach: machine learning
        i. Global optimization of Lipschitz functions
        ii. Ranking strategy for nonsmooth functions

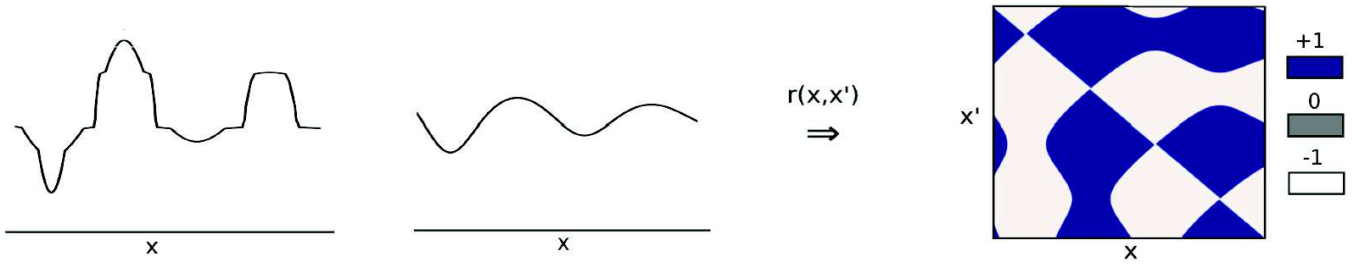
A ranking approach to global optimization

Definition 3.1. (INDUCED RANKING RULE) *The ranking rule* $r_f : \mathcal{X} \times \mathcal{X} \to \{-1,0,1\}$ *induced by a function* $f : \mathcal{X} \to \mathbb{R}$ *is defined by:*

$$r_f(x,x') = \begin{cases} +1 & \text{if } f(x) > f(x') \\ 0 & \text{if } f(x) = f(x') \\ -1 & \text{if } f(x) < f(x') \end{cases}$$

for all $(x,x') \in \mathcal{X}^2$.

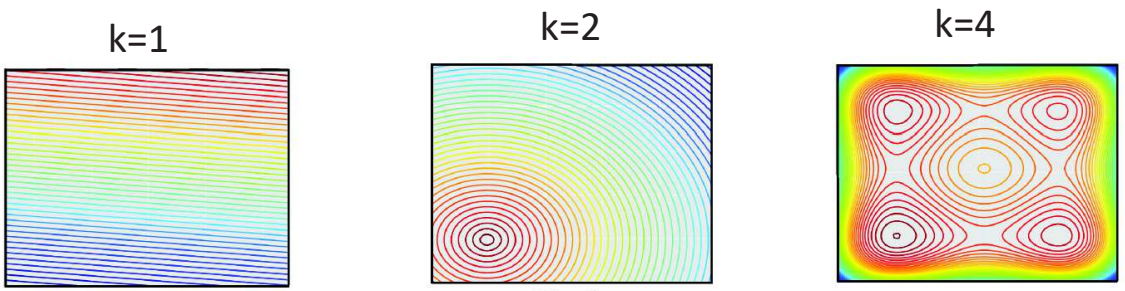
Example: different functions with the **same** ranking rule



Ranking rules - example

Example 3.1. (POLYNOMIAL RANKING RULES) *The set of polynomial ranking rules of degree* $k \geq 1$ *is defined as*

$$\mathcal{R}_{\mathcal{P}_k} := \{r_h : (x, x') \mapsto \operatorname{sgn}(h(x) - h(x')) \mid h \in \mathcal{P}_k(\mathcal{X}, \mathbb{R})\}.$$



Some 2-d function with polynomial ranking

Algorithmic principle based on ranking loss minimization

Definition 3.3. (Empirical ranking loss) *The empirical ranking loss computed over a sample* $(X_1, f(X_1)), \ldots, (X_t, f(X_t))$ *of* $t \geq 2$ *function evaluations is defined for all* $r : \mathcal{X} \times \mathcal{X} \to \{-1, 0, 1\}$ *by*

$$L_t(r) := \frac{2}{t(t-1)} \sum_{1 \leq i < j \leq t} \mathbb{I}\{r(X_i, X_j) \neq r_f(X_i, X_j)\}$$

where $r_f(X_i, X_j) = \operatorname{sgn}(f(X_i) - f(X_j))$ *for all* $(i, j) \in \{1, \ldots, t\}^2$.

Input: $n \in \mathbb{N}^\star$, $\mathcal{R} \subseteq \mathcal{R}_\infty$, $\mathcal{X} \subset \mathbb{R}^d$, $f : \mathcal{X} \to \mathbb{R}$

1. Initialization: Let $X_1 \sim \mathcal{U}(\mathcal{X})$
Evaluate $f(X_1)$, $t \leftarrow 1$
$\mathcal{R}_1 \leftarrow \mathcal{R}$, $\hat{\imath}_1 \leftarrow 1$

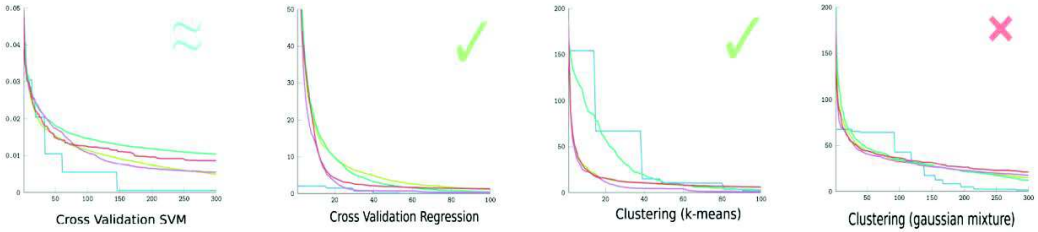
2. Iterations: Repeat while $t < n$:
Let $X_{t+1} \sim \mathcal{U}(\mathcal{X})$
If there exists $r \in \mathcal{R}_t$ such that $r(X_{t+1}, X_{\hat{\imath}_t}) \geq 0$ {Decision rule}
Evaluate $f(X_{t+1})$, $t \leftarrow t + 1$
$\mathcal{R}_t \leftarrow \{r \in \mathcal{R} : L_t(r) = 0\}$
$\hat{\imath}_t \in \arg\max_{i=1\ldots t} f(X_i)$

3. Output: Return $X_{\hat{\imath}_n}$

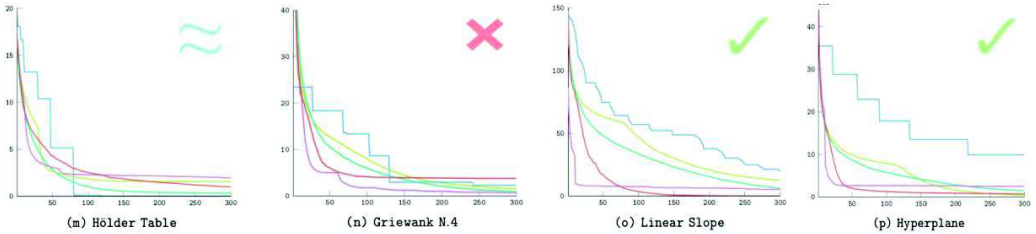
Numerical experiments

- **Comparison** with stat-of-the-art algorithms:
  - AdaRank [This talk]
  - BayesOpt [Martinez 2014]
  - CMA-ES [Hansen 2003]
  - CRS [Price 1983]
  - DIRECT [Jones 1993]

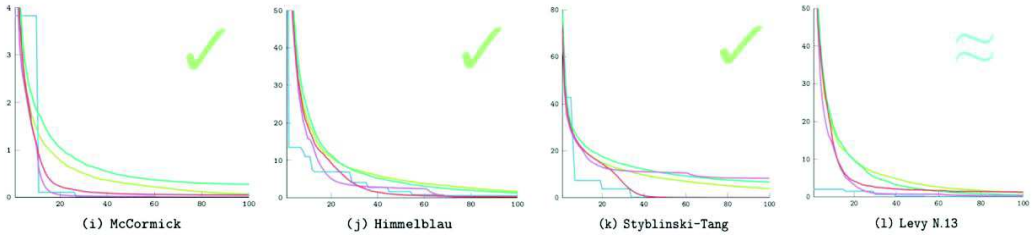
- **Cross validations problems (Real data sets)**



- **Synthetic** high-dimensional problems [AdaRank, BayesOpt, CMA-ES, CRS, DIRECT]



- **Synthetic** 2-d non-convex problems [AdaRank, BayesOpt, CMA-ES, CRS, DIRECT]



Difference between the true maximum and estimation in terms of iterations

Some other hot topics

- Privacy preserving learning
- Transfer learning
- Continuous learning
- Explainable learning