Technical Portfolio

Ilyas Malik

AI R&D, published in **ICML** and **NeurIPS**.

Hi ! I'm Ilyas, a top-ranked alumni from **École Polytechnique***, France's premier engineering school, known for its competitive admissions and producing innovative leaders in *STEM fields*.
University of Oxford alumni specialized in *AI*, *Statistics* and *CS*.
Developed state-of-the-art technologies with top-tier organizations like **Amazon**, **IBM** and **Oxford**.
Problem solver, market driven, entrepreneur.

*11 out of 14 of the famous Facebook LLAMA language model are Ecole Polytechnique alumni

Contents

1 Oxford Research, ICML paper – Deep Adaptive Design: Amortizing Sequential Bayesian Experimental Design

1.1 Context

During my Dissertation Thesis at the university of Oxford, I worked on Bayesian Optimal Experimental Designs (BOED) and was able to contribute to a key and innovative development of that technology, which has tremendous consequences in real world applications in numerous fields: Psychological trials, pharmaceutical trials, any kind of survey, physics experiments ...

1.2 Challenges and contribution

We were able to publish a paper in the International Conference on Machine Learning (ICML[1]).
In this work, we introduce Deep Adaptive Design (DAD), a general method for amortizing the cost of performing sequential adaptive experiments using the framework of Bayesian optimal experimental design (BOED). Traditional sequential BOED approaches require substantial computational time at each stage of the experiment. This makes them unsuitable for most real-world applications, where decisions must typically be made quickly. DAD addresses this restriction by learning an amortized design network upfront and then using this to rapidly run (multiple) adaptive experiments at deployment time. This network takes as input the data from previous steps, and outputs the next design using a single forward pass; these design decisions can be made in milliseconds during the live experiment. To train the network, we introduce contrastive information bounds that are suitable objectives for the sequential setting, and propose a customized network architecture that exploits key symmetries. We demonstrate that DAD successfully amortizes the process of experimental design, outperforming alternative strategies on a number of problems.

1.3 Results

The algorithm resulting from that paper is given in Figure 2.
In all experiments DAD performed significantly better than baselines with a comparable deployment time. Further, DAD showed competitive performance against conventional BOED approaches that do not use amortization, but make costly computations at each stage of the experiment. An example of such performance is given in a physics experiment in inspired by the acoustic energy attenuation model, we consider the problem of finding the locations of multiple hidden sources which each emits a signal whose intensity attenuates according to the inverse-square law. The total intensity is a superposition of these signals. The design problem is to choose where to make observations of the total signal to learn the locations of the sources. Results are displayed in Figure 1.

[1]ICML is the leading international academic conference in machine learning. Along with NeurIPS and ICLR, it is one of the three primary conferences of high impact in machine learning and artificial intelligence research.

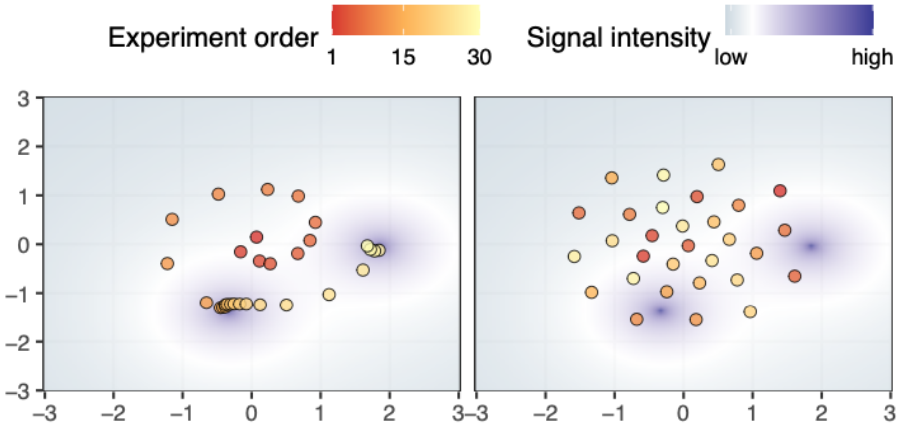


Figure 1: An example of the designs learnt by DAD [Left] and the fixed baseline [Right] for a given θ sampled from the prior.

Algorithm 1 Deep Adaptive Design (DAD)

Input: Prior $p(\theta)$, likelihood $p(y|\theta, \xi)$, number of steps T
Output: Design network π_ϕ
while Training compute budget not exceeded **do**
  Sample $\theta_0 \sim p(\theta)$ and set $h_0 = \varnothing$
  for $t = 1, ..., T$ **do**
    Compute $\xi_t = \pi_\phi(h_{t-1})$
    Sample $y_t \sim p(y|\theta_0, \xi_t)$
    Set $h_t = \{(\xi_1, y_1), ..., (\xi_t, y_t)\}$
  end
  Compute estimate for $\partial \mathcal{L}_T / \partial \phi$ as per § 4.2
  Update ϕ using stochastic gradient ascent scheme
end
At deployment time, π_ϕ is fixed and each y_t is obtained in turn by running experiment with design ξ_t.

Figure 2: Amortization technology

1.4 Technical details

We implemented DAD by extending PyTorch and Pyro[2] to provide an implementation that is abstracted from the specific problem.

[2]Pyro is a universal probabilistic programming language (PPL) written in Python and supported by PyTorch on the backend. Pyro enables flexible and expressive deep probabilistic modeling, unifying the best of modern deep learning and Bayesian modeling.

2 IBM Research, NeurIPS paper – Safe Deep Q-Learning: Designing Risk-sensitive Reinforcement Learning Agents

2.1 Context

I have worked in IBM Research Labs (Department of AI) where I designed AI agents that are risk averse on demand. This technology has numerous applications in sensitive real life scenarios, for instance in security or medical applications where the goal is to accomplish the objective but it is also necessary to avoid worst-case scenarios.

2.2 Contribution and results

We were able to publish a paper in the Conference on Neural Information Processing Systems (NeurIPS[3]).
I have used 3 novel approaches for distributional Reinforcement Learning (See RL representation in Figure 3, Deep Q-learning representation in Figure 4) to be able to recover (at least implicitly) the distribution of the return $Z(s, a)$ which is the reward of the agent where it performs an action a in a state s. The approaches I have used are:

1. GAN Q-learning
2. A Distributional Perspective on Reinforcement Learning: C51 Algorithm
3. Quantile Regression Deep Q-Learning: This is the most successful approach for which I have kept the experiments, see Algorithm 5

I have leveraged the above approaches to develop custom policies AI agents: instead of optimizing the averaged return (a.k.a Q function), the AI agent now optimizes a modified quantity, the Conditional Value at Risk which is basically a weighted average of the "extreme" cases in the tail of the distribution of possible returns.
One of the most visual experiments I have tested my code on are available to watch on **YouTube**:

- Extreme risk averse agent (Only optimizing worst-case scenarios): The lander approaches the flags but hangs in the air for the longest time to avoid crashing on an irregular ground, sometimes it just hangs in the air next to flags until the episode ends.
- Moderately risk averse agent (Optimizing worst-case scenarios and moderate scenarios): The lander has similar behaviour as the extreme risk averse case but it hangs less in the air and always lands between the flags.
- Standard agent (Optimizing over all scenarios): The Vanilla DQN version is the quickest of all but has more worst-case scenarios of crashing.

I conducted Monte Carlo experiments to compare the overall rewards of the risk averse and the standard agent, Figure 6 clearly shows that the vanilla DQN has a higher average but the safe version has significantly less worst-case or crashing scenarios.
I have written the code on Python using an object oriented logic and Tensorflow library.

[3]NeurIPS is the leading international academic conference in machine learning. Along with ICML and ICLR, it is one of the three primary conferences of high impact in machine learning and artificial intelligence research.

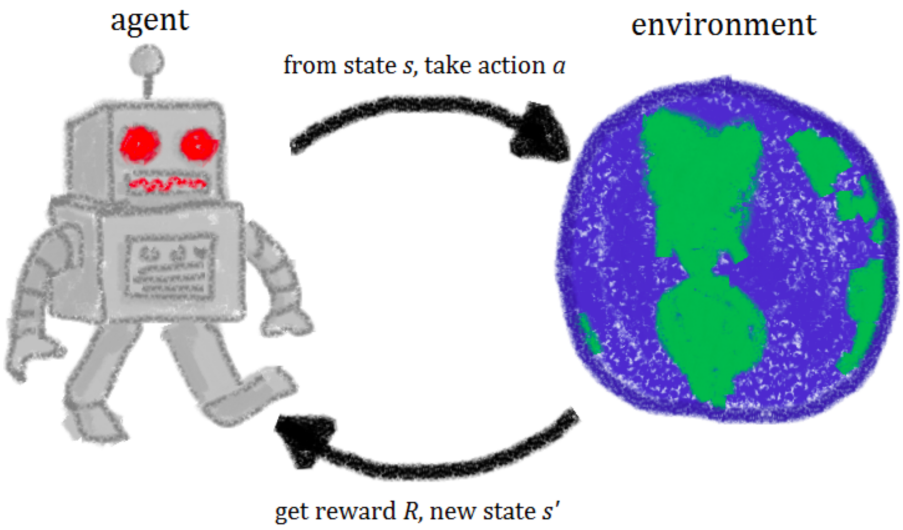


Figure 3: Reinforcement Learning representation

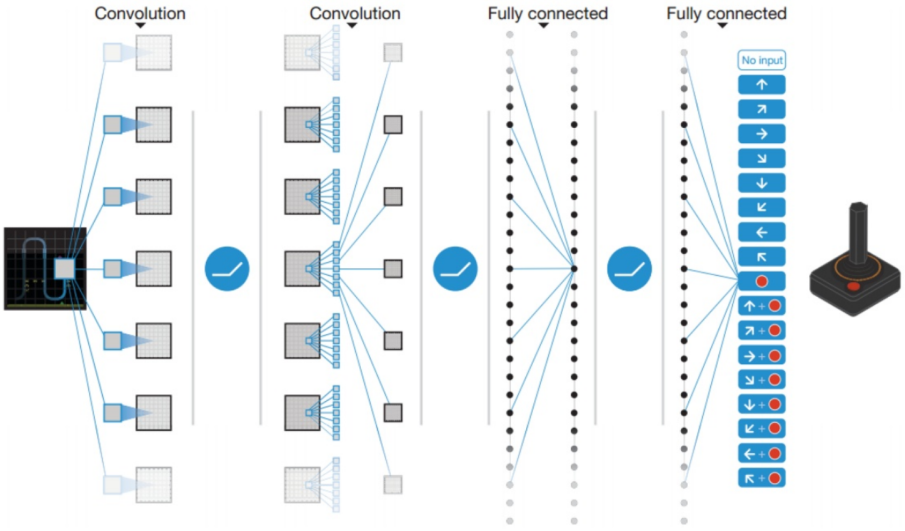


Figure 4: Deep Q-learning representation

Require: $N, \kappa\ x, a, r, x',\ \gamma \in [0, 1)$

Compute distributional Bellman target

$Q(x', a') := \sum_j q_j \theta_j(x', a')$

$a^* \leftarrow_{a'} Q(x, a')$

$\theta_j \leftarrow r + \gamma\theta_j(x', a^*), \quad \forall j$

Compute quantile regression loss (Equation ??) $\sum_{i=1\ j}^{N} \left[\rho_{\hat{\tau}_i}^{\kappa}(\theta_j - \theta_i(x, a))\right] = 0$

Figure 5: Algorithm: Quantile Regression Deep Q-learning

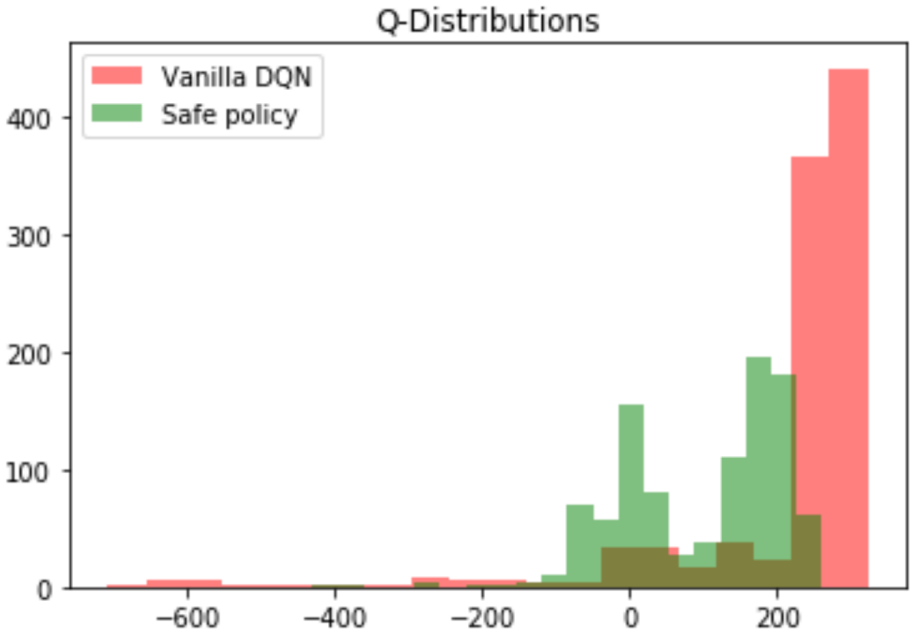


Figure 6: Comparison of Vanilla Deep Q-learning and Risk sensitive Q-learning

3 Amazon AI R&D – High Confidence Predictions leveraging Monte Carlo Dropout

3.1 Context

I have worked with Amazon Luxembourg for 6 months as an Applied Scientist intern in the supply chain Science team. In that time period, I have worked on a Deep Learning project using a multimodal Neural Network (Image branch using Computer Vision and transfer learning, Text branch using NLP, Numerical branch and high cardinality categorical branches as well as numerical branch with categorical and continuous variables) to predict the Amazon products in packaging requirements.

3.2 Challenges and results

The existing model was predicting on all the Amazon products in Europe with a reasonable accuracy, but couldn't be deployed on a very large number of products because even a small proportion of errors could lead to a large number of wrongly predicted products.
I have exclusively worked on a pipeline of tasks to return confidence scores[4] along with the predictions in order to have a subset of products for which the accuracy is much higher than the initial one. The method I have developed leverages MonteCarlo Dropout and goes further into setting and proving a rigorous mathematical and statistical framework for computing confidence scores. According to the manager of our team, this cutting edge technology is still not democratised in business and establishes a great improvement and advantage over other work done in different teams.
I have collaborated with the business team and the success of the project led to the deployment of a pilot to automatically switch products packaging.

3.3 Technical details

In this context, I have:

- Implemented Hyperparameters tuning using the Python HyperOpt library, which I used to tune the whole architecture of the network and its regularisations, I also introduced Dropout[5] layers in the Network without impairing the model thanks to the hypertuning of the Dropout rates
- Implemented Monte Carlo Dropout and introduced a confidence metric for which I built rigorous statistical framework
- Increased the F1-Score by 26 to 29 percentile points on validation set by restricting the set to high confidence predictions ($\sim$ half of the validation set)

Code was written in Python using the PyTorch library.

[4] See Figure 8
[5] See Figure 7

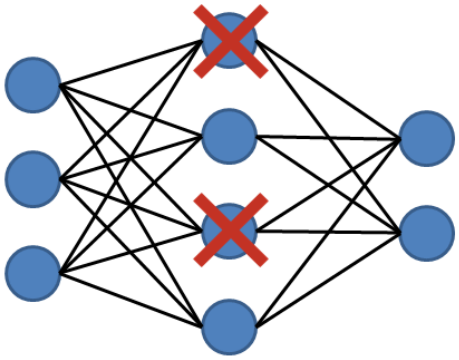


Figure 7: Dropout layers

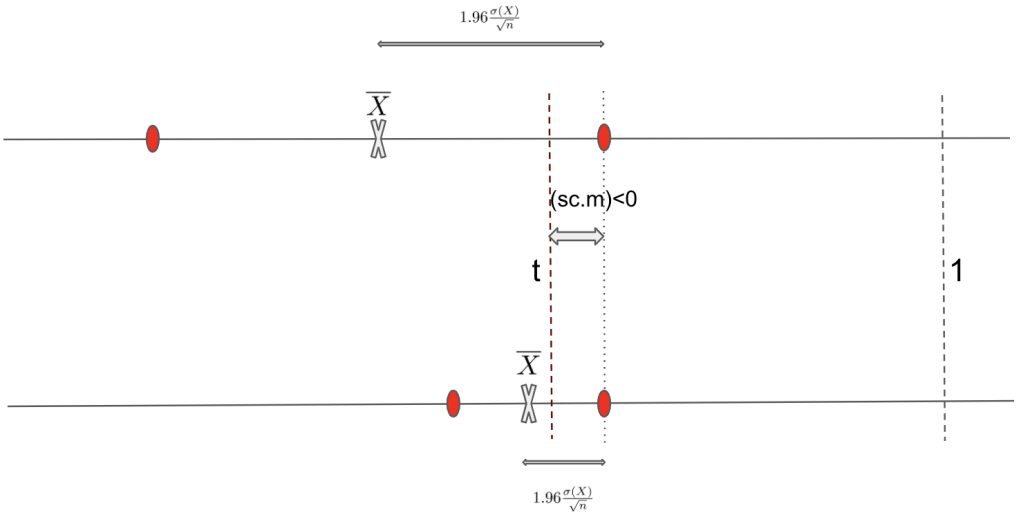


Figure 8: Statistical meaning of the predictions' confidence scores

4 Arcturus AI R&D – 3D Segmentation using Graph Neural Networks (GNNs)

4.1 Context

I have worked with Arcturus to research and develop a deep learning algorithm for 3D data samples. After doing some experiments with GNNs (See Figure 9) on 3D data, I've built a GNN model with a base layer inspired from PointNet++, implemented a customizable U-Net architecture with skip connections from scratch using Pytorch Geometric and experimented on toy datasets.

Datasets I've implemented a preprocessing algorithm to convert the raw data to features/labels data with 11 labels on point clouds, then wrapped the data samples and batches into the Pytorch Geometric data objects for optimal performances. Within the data loading pipeline, I've implemented a preprocessing step that transforms the data from meshes to uniformly sampled points to graphs with nodes and edges based on furthest point sampling.

Modeling

- I've implemented a customizable hyperparameters tuning script with Optuna using a Bayesian optimization algorithm on initial input distributions as priors, the PointNet++ model yielded a 66% accuracy with 11 classes **(the random baseline has 9% accuracy)**.
- I then wrapped the model, dataset and training/inference/hypertuning scripts into the Pytorch Lightning framework. This is a best practice which requires an initial cost setup to get rid of the boilerplate code but offers many advantages later on for future integrations as well as performance.
- The data was a bit noisy but after data annotation and reducing the number of classes to 9 (by merging high similarity classes), the model was able to reach a 72% accuracy.
- I tried additional variations and different models (PointCNN, many improvements to the Message Passing layer of the PointNet++ base, fast ad-hoc vs. pre-computed data augmentations, different pooling/clustering approaches), documented performances analyzes documented and the model ended up reaching a 75% accuracy (on 9 classes, the random baseline is 11%).

Additional features I've also built visualization tools with a link/password access to the performance of the models, their training curves, detailed insights on the forward/backward flows into the different layers, visual representation of the models and their layers as well as point cloud data samples with ground truth vs. predicted segmentation masks depicted with colors as well as an integrated comparative tool on the runs yielded by the hyperparameters tuning to help the research team get more insights on the optimization.
The whole code is optimized with vectorized operations to efficiently run mathematical operations with the native C++ loops, this is a common practice for Machine Learning algorithms in Python.

4.2 Experiments

In the following video I show and compare for each 3D data sample:

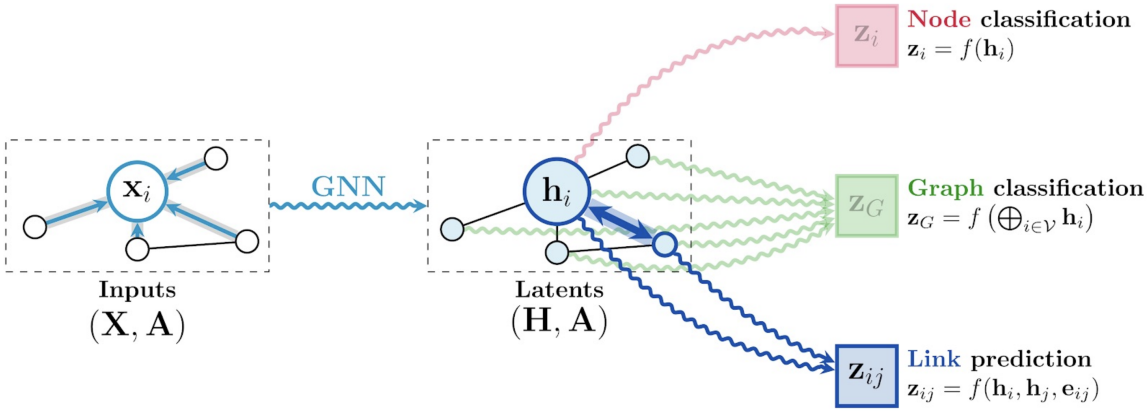


Figure 9: General blueprint for learning on graphs: A representation of a single layer of GNNs.

1. The predicted segmentation map
2. The ground truth segmentation map

4.3 Development environment

Ensured the choice and configuration of a suitable aws machine and set the remote development/debugging pipelines.

5 Data Science Mission with the hospitals of Paris – Causal inference: Assessing the effect of Tranexamic Acid on the mortality of patients with head trauma

5.1 Context

I worked in a project in conjunction with the doctors committee of the hospitals of Paris, the goal was to assess the effect of an empirical treatment (Tranexamic Acid) on the mortality of severely damaged patients with head trauma.
My teammates and I have worked on the whole pipeline of Data Science on a raw extensive Data set with 7000 patients and 350 variables, we have:

1. Preprocessed the data,
2. Performed Missing Values Analysis, imputed missing data using iterative Factorial Analysis for Mixed Data, thus taking into account the interdependance of continuous and categorical covariates
3. Performed distributions analysis to assess the quality of the data imputation
4. Compared the data imputation with the MissForest[6] imputation for further analysis
5. Performed data analysis using multiple methods: Factorial Analysis for Mixed Data[7] (See Figure 11), Hierarchical clustering (See Figure 12) as well as other custom analyses focusing on the variables of interest
6. Performed advanced causal inference method to assess the intrinsic effect of Tranexamic Acid on the survival of patients: Matching, FAMD Matching, Average Treatment Effect (See Figure 13) with Inverse Propensity Weighting (See Figure 14) and finally the Double Robust method

5.2 Challenges

The project we've worked on was a full stack Data Science project with custom uncommon problematic related to Causal Inference (See Figure 10 for illustration), we had to leverage state-of-the-art research to solve the problematic, in addition to the use of multiple Machine Learning models as intermediate steps during the project.

5.3 Technical details

The whole project was coded in R, using custom packages for the different steps and methods of the project.
The conclusion of the project (See Figure 15) led the doctors committee to pause the use of Tranexamic Acid for head trauma.

[6] Nonparametric missing value imputation for mixed-type data using Random Forests
[7] FAMD: PCA like method for data including continuous and categorical variables

Potential cause(W)	Effect (Y)	Potential bias or real cause (X)
Umbrella sales	Number of car accidents	Weather (rain)
Temperature	Weights of chicks	Gender
Treatment	Survival	Initial state of the patient

Figure 10: Causality exemples: Correlated covariate vs. real causes

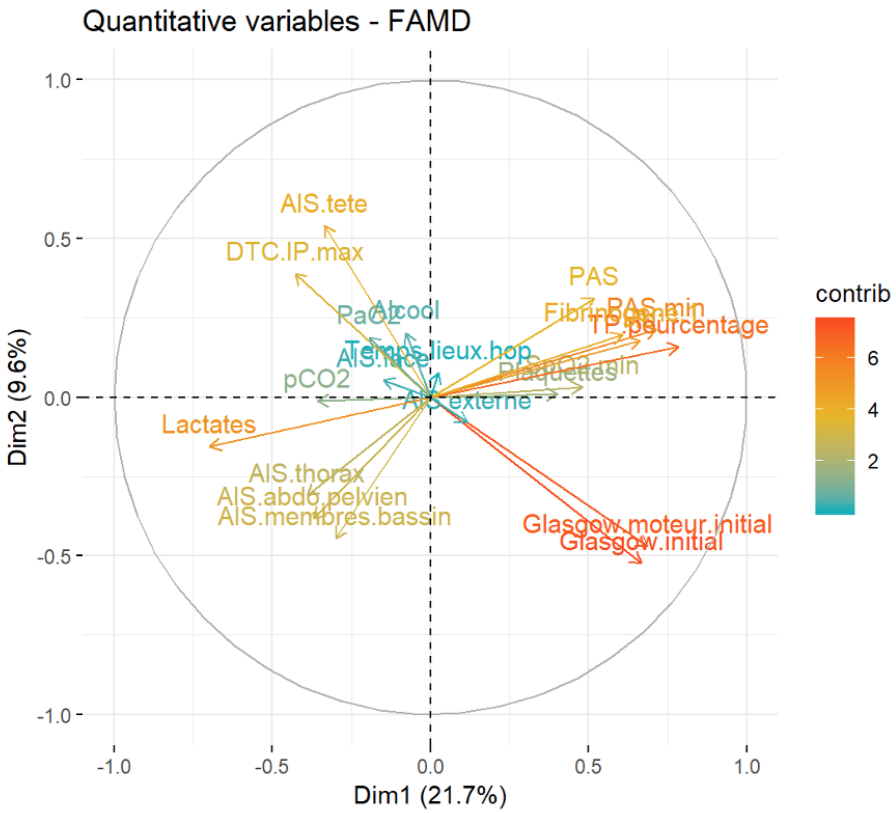


Figure 11: FAMD on the data represented on the 2 main components

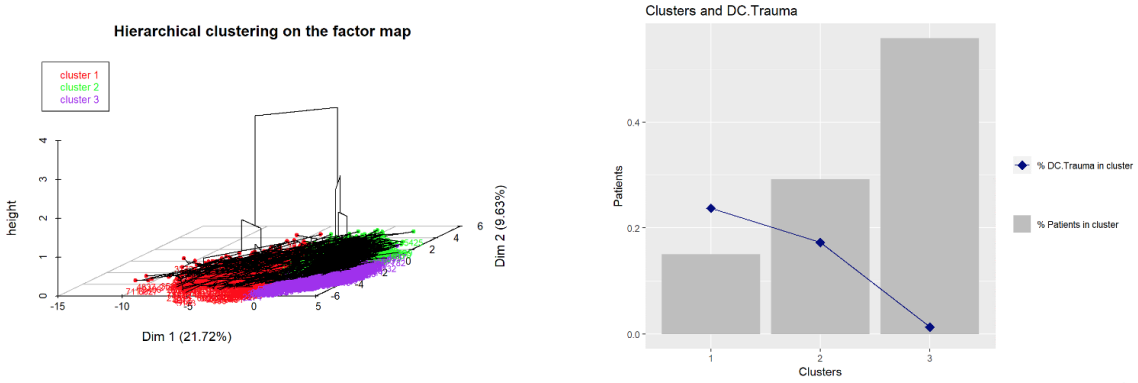
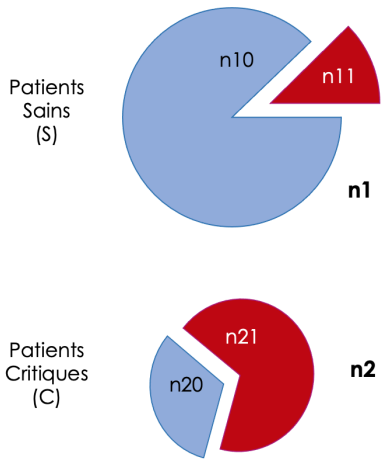


Figure 12: Hierarchical Clustering visualised for 3 clusters on the 2 main components returned by the FAMD. The mortality rate is heterogeneous across the clusters



$$\hat{\tau}(x) = \frac{1}{n_{x1}} \sum_{\{X_i = x, W_i = 1\}} Y_i - \frac{1}{n_{x0}} \sum_{\{X_i = x, W_i = 0\}} Y_i.$$

$$\hat{\tau} = \sum_{x \in \mathcal{X}} \frac{n_x}{n} \hat{\tau}(x)$$

Figure 13: The Tranexamic Acid is more often administrated to patients in critical states (The red proportion receives the treatment), which explains the need to perform causal inference before concluding its harmfulness

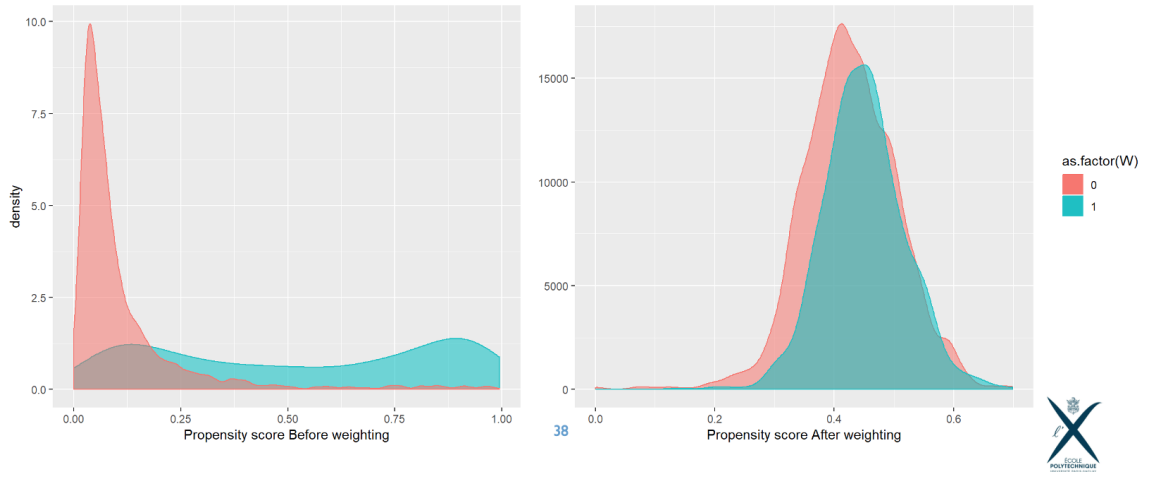


Figure 14: Distribution of the Propensity score returned by both regressions without and with the weighting

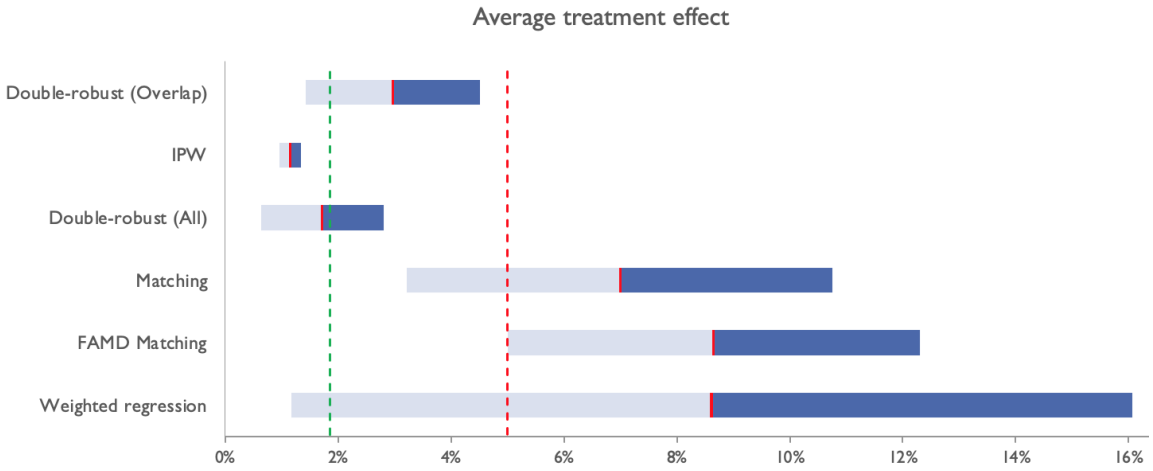


Figure 15: Comparison of the multiple computations of the ATE: They all suggest the uselessness of the treatment

6 IBM Research – Deep adversarial training for regularization

6.1 Context

I have worked in IBM Research Labs (Department of AI) where I developed a technology that uses Adversarial training to improve the generalization capability of Neural Networks.
The idea is to make use of 2 models instead of 1 model (the original classifier and the discriminator h that's only here to improve the classifier f) with a split of the data to be used in the training. These two models will have conflictual Loss functions whence the denomination Adversarial Training.

6.2 Challenges and contribution

The set of models to be trained have conflictual and custom losses that need to be addressed in specific ways (either on Tensorflow or PyTorch). Indeed, unlike in the standard or vanilla neural networks, the gradient of the losses flow backwards and update the parameters of both the networks. This work is a novel way of regularization that I compared with other regularization techniques and proved to be particularly efficient (and in many experiments outperforms the most common regularisation type: the $\mathcal{L}^2$ regularisation), see Figure 19. It is inspired by one of the applications of GANs (See general principle of GANs in Figure 16) in ML: Membership Privacy.
I have reduced overfitting by a factor 2 on different datasets ($\sim 1k \times 20$) using this novel technology.

6.3 Technical details

I have used a modified version of neural networks training as described in Figure 17. The idea is to split the training data into D and D' (See Figure 18) and only feed D to f instead of feeding it $D \cup D'$, the whole $D \cup D'$ is given to h to try to determine whether a point is in D or D'. By fooling it, f modifies the distribution of x, y to be independent from D. Thus, the output is less "overfitted" to D and more generalizable.
I have coded in Python and used ScikitLearn for the regressions and the K Nearest Neighbors, and Tensorflow for the Neural Networks and tracking the custom losses.

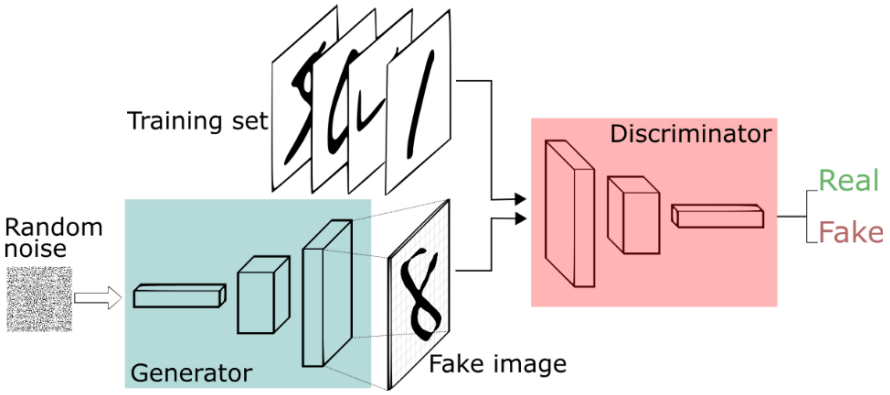


Figure: GANs principle

Figure 16: GANs principle

1: **for** number of the training epochs **do**
2: **for** k steps **do**
3: Randomly sample a mini-batch of m training data points $\{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$ from the training set D.
4: Randomly sample a mini-batch of m reference data points $\{(x'_1, y'_1), (x'_2, y'_2), \cdots, (x'_m, y'_m)\}$ from the reference set D'.
5: Update the inference model h by ascending its stochastic gradients over its parameters ω:

$$\nabla_\omega \frac{\lambda}{2m} \left(\sum_{i=1}^{m} \log(h(x_i, y_i, f(x_i))) + \sum_{i=1}^{m} (\log(1 - h(x'_i, y'_i, f(x'_i)))) \right)$$

6: **end for**
7: Randomly sample a fresh mini-batch of m training data points $\{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$ from D.
8: Update the classification model f by descending its stochastic gradients over its parameters θ:

$$\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} ((f(x_i), y_i) + \lambda \log(h(x_i, y_i, f(x_i))))$$

9: **end for**=0

Figure 17: Custom adversarial training

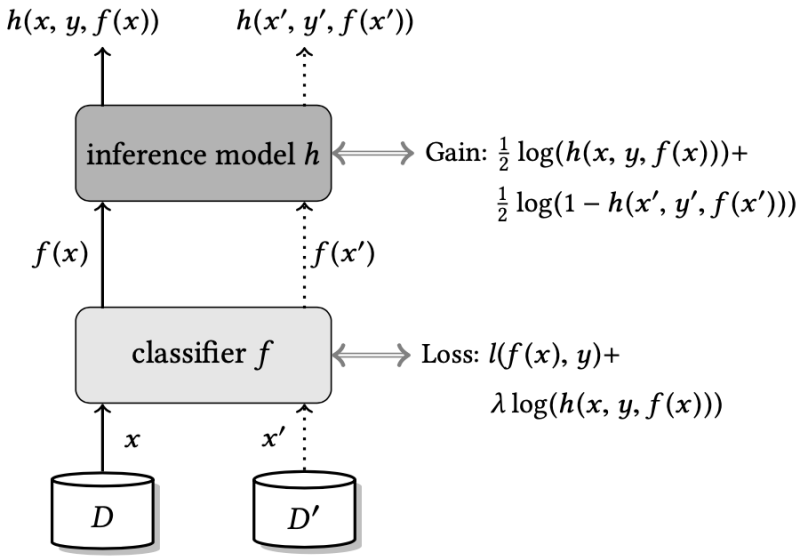


Figure 18: Classification loss and inference gain, on the training dataset D and reference dataset D', in our adversarial training. The classification loss is computed over D, but, the inference gain is computed on both sets.

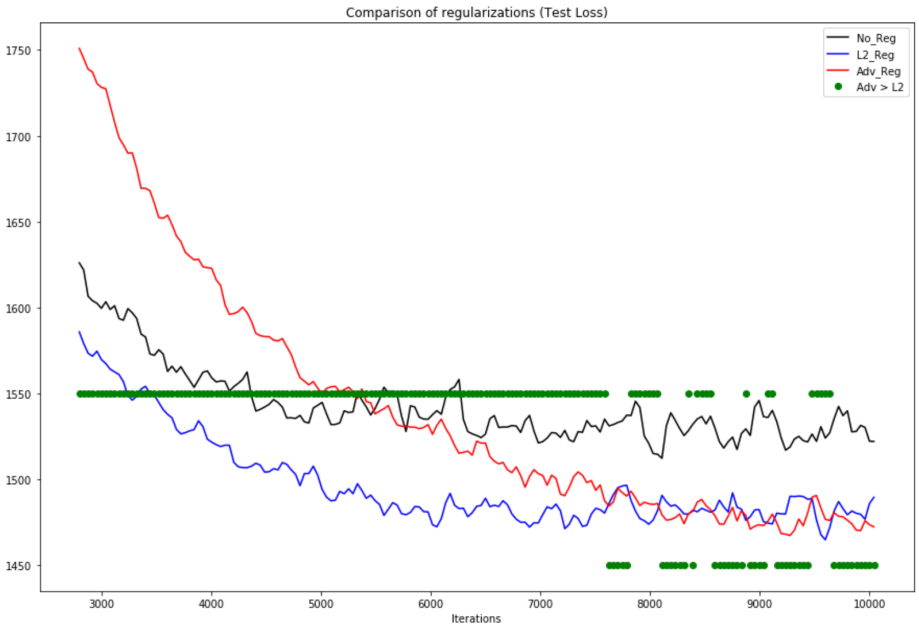


Figure 19: Comparison of the losses: Adversarial Regularization vs. $\mathcal{L}^2$ Regularization

7 Operations Research project for an anonymous energy supplier – Optimizing energy pricing in a two-layered Karush-Kuhn-Tucker model with the possibility of energy stocking

7.1 Context

I have worked on a confidential project for an energy supplier company. The project regarded the pricing of energy with respect to customers behaviour to optimize a complex high dimension cost-related quantity.

7.2 Contribution and challenges

The project treated a real-life need and the optimization problem that I formulated took into account real-life data of customers behaviours in terms of consumption and pricing decisions. The objective I formulated was a two-layered optimisation problem for which KKT conditions of the lower layer were used to simplify the problem into a one-layer, multi-parameters optimisation problem. Moreover, I have added and modelled the possibility of stocking energy by the consumer into the optimization as many clients now turn towards buying a solar panel.
Our formulation of the problem greatly improved on the existing model.

7.3 Technical details

Before solving the formulated problem, I applied the K-means ++ to cluster the clients using the scikit-learn library in order to reduce the dimension of the optimization problem. Then, I ended up with a mixed-integer quadratic problem (MIQP) that I implemented and solved using the IBM optimization libraries CPLEX and DOcplex.

8 Oxford Research project – Predicting results of a competition using Markov Chain Monte Carlo

8.1 Context

I worked on a Bayesian Statistics project where we are given data that represents records of $L = 134$ games played by $n = 24$ players, in each game a small subset of players participate and we keep track of their rankings in each game and their age seniority amongst all the players present in the field of game (not only those participating at that time), the age seniority is a number from 1 to $n_a = 16$.
I performed bayesian analysis of the dataset starting with prior elicitation under the Placket-Luce model before exploratory data analysis, I focused on 2 models that either account for the effect of seniority or not. First, I modelled the data and computed the posterior (See Figures 20 and 21) using an MCMC approach, then I used that to do parameters estimation, hypothesis testing and model selection (using Bayes Factor).

8.2 Challenges and technical details

I have performed Bayesian Data analysis starting from non-informative priors to model the outcomes of the competition taking into account skills of players and seniority ranks as covariates. The main difficulty with such a model is that the problem is high dimensional (40 dimensions). To deal with this issue, I opted for the Gibbs sampler, indeed due to geometric reasons (e.g the volume of a sphere increases exponentially with the dimension) the likelihood in such high dimensions becomes very localised, so to make sure I can go towards the support of the posterior distribution, I made sure the process makes small steps (i.e component by component).
The sequential scan is often better than the random scan Gibbs sampler, in this case this is true. Intuitively, the sequential scan makes sure to go over all the components so the autocorrelation at lag (See Figure 22) goes down lower than in the random scan.
I have used the language R for the code and the report.

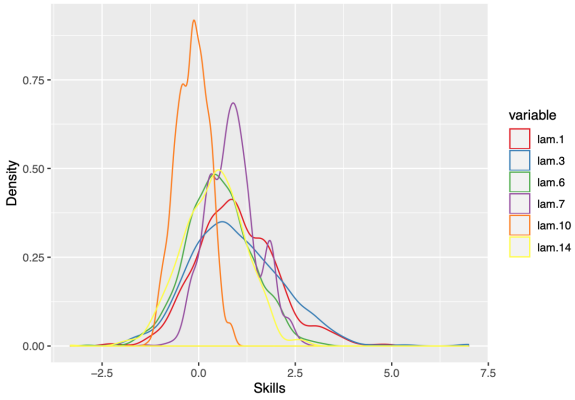


Figure 20: Posterior densities of some skills

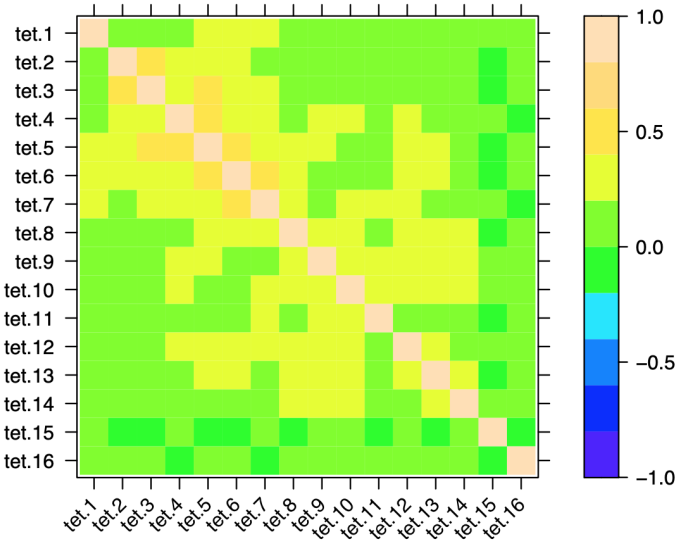


Figure 21: Posterior correlation of the theta's

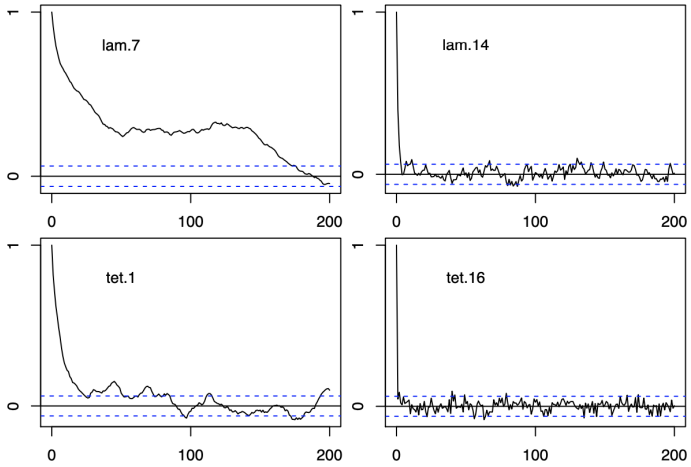


Figure 22: ACF plots

9 CMAP Research project – Probability estimation of rare events: Non-extinction of an endangered species

9.1 Context

I have worked on a research project with the CMAP[8] for which the goal is to simulate and compute probabilities of rare events ($p \sim 10^{-8}$). The case study I have used for this project is the Galton-Watson model for simulating dynamics of population (See Figure 23 for genealogical representation). At each generation t, each member of the population breeds a random number of offsprings before dying, the total number of offsprings renders the population size at generation $t+1$, this constitutes a Markov Chain for which we study the characteristics. When the average number of offsprings per individual is lower than 1, the population is doomed to the extinction.

9.2 Challenges and technical details

The difficulty that rises when estimating rare event probabilities is that standard Monte Carlo estimations no longer work because for an event of probability 10^{-6} for example, using Monte Carlo with a million simulations will only render a very few or no occurrence of the event of interest, thus making the Monte Carlo estimate noisy or even unusable.
I used mathematical properties of the Process of interest to calculate exact values of some rare events for further comparisons with the developed methods (See Figure 24 for exact probability calculations). In addition to the naive Monte Carlo estimation (See Figure 25), I have used 2 advanced simulation methods to estimate very low probabilities with reasonable computational resources:

1. Importance Sampling: I modify the probability distribution of the whole process and use Monte Carlo with the modified distribution multiplied by a likelihood term to render the original expectation. The new distribution is supposed to allow the rare event more often. See Figure 27.

2. Interacting Particle Systems[9]: I used a Markov Chain method that leverages a similar process inspired from Genetic Algorithms. The idea is to have M trajectories of the tree history of the population and at each generation 2 steps are applied: the selection step for which trajectories are sampled with respect to a potential that we define according the trajectories we want to favor for the occurrence of the rare event, the second step is the mutation for which the next generation population is computed. The estimate is then computed using an average corrected by an expression derived from the potential of the trajectories at different generations. See Figure 26.

Furthermore, we show an interesting result: if we combine 2 unviable habitats, one with good conditions but with occasional disasters and the second with constant slightly bad conditions. Then we can mathematically prove and empirically verify that there is a specific optimal proportion[10] of the population to keep in the first habitat for the population to thrive exponentially. See Figures 28, 29 and 30.

[8] Centre de Mathématiques Appliquées de l'École polytechnique
[9] This is similar to Sequential Importance Sampling with Resampling in Particle Filters
[10] See paper

Figure 23: Genealogical representation of a population evolution

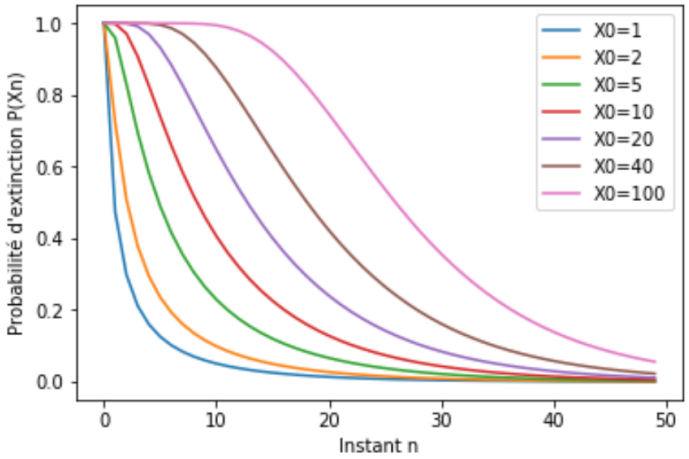


Figure 24: Probability of non extinction at generation n: $P(X_n > 0 | X_0 = X0)$ for different initial population values $X0$. The average offsprings number per individual is $m = 0.9$

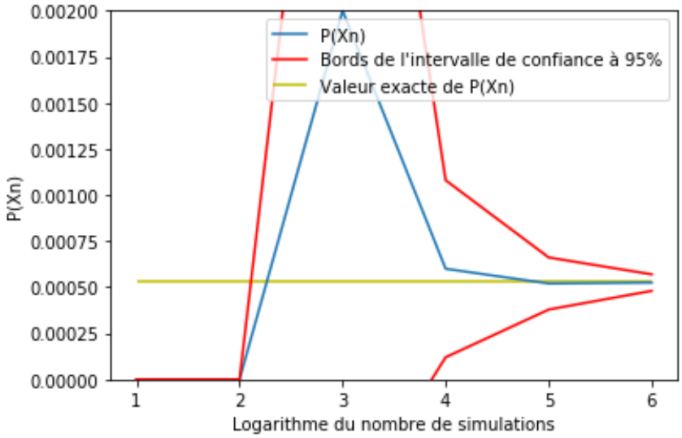


Figure 25: Naive Monte Carlo estimate: $P(X_{40} > 0 | X_0 = 20)$ for $m = 0.8$ and 10^x simulations

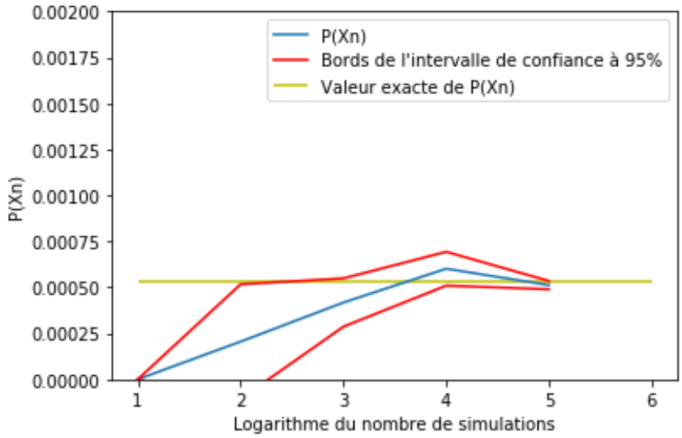


Figure 26: Interacting Particle Systems estimate: $P(X_{40} > 0 | X_0 = 20)$ for $m = 0.8, \lambda = 0.01$ and 10^x simulations

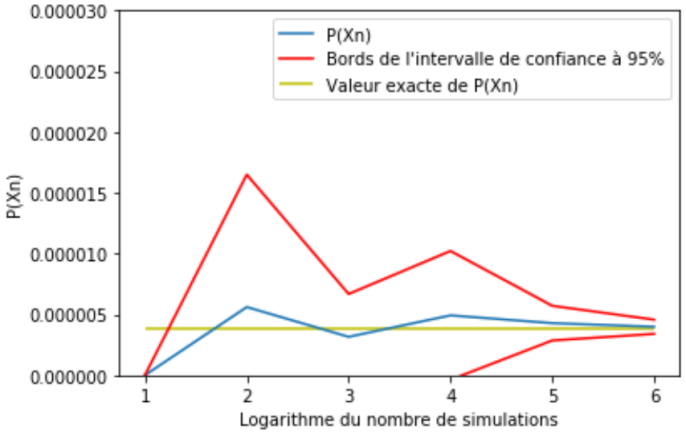


Figure 27: Importance Sampling estimate: $P(X_{40} > 0 | X_0 = 20)$ for $m = 0.7, m' = 0.97$ and 10^x simulations

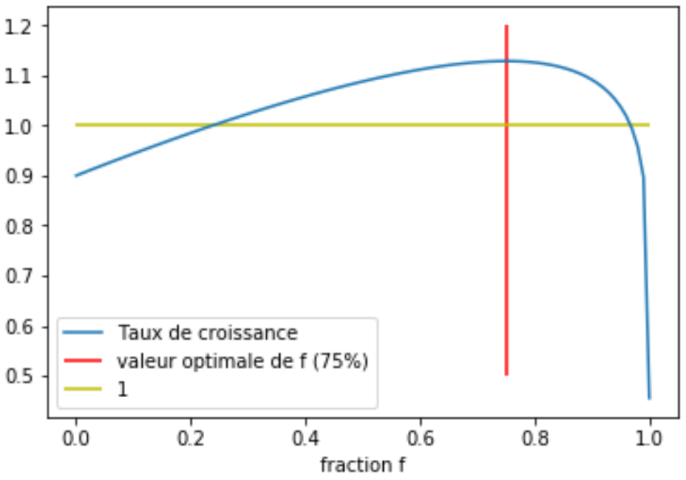


Figure 28: Optimal fraction f to put in the first habitat to maximise the growth rate

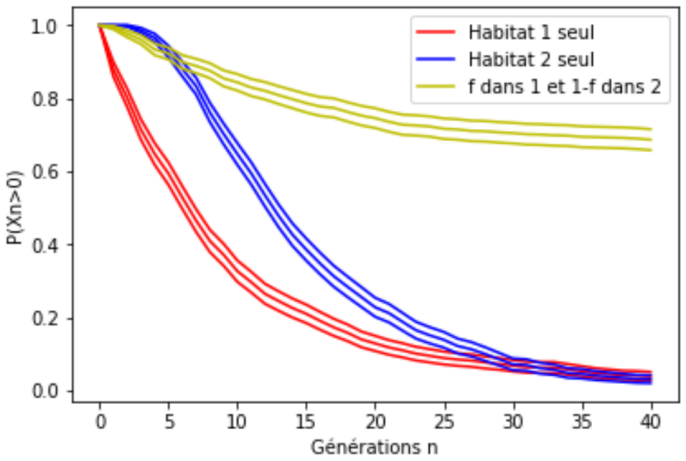


Figure 29: Optimal combination of 2 sink habitats where the population can persist

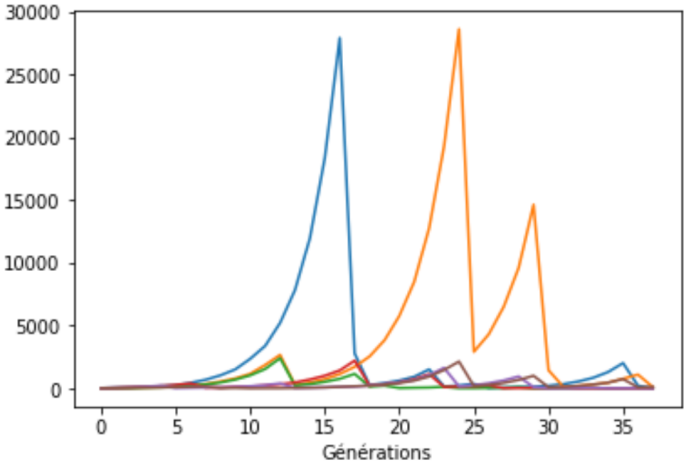


Figure 30: Examples of populations that have thrived in the combined habitats, the population stocks in the second habitat to avoid extinction when disasters occur in the first habitat

10 Oxford Machine Learning Kaggle Challenge – Predicting binding molecules

10.1 Context

I have participated in a Kaggle challenge where the goal was to predict whether a molecule will bind to proteins or not, without using any mechanical or physical law.
The data consists of different molecules that are either "active" if they can bind to different proteins, or "decoy" if they are assumed not to bind to any protein. Each molecule corresponds to one row of the data set and has 349 associated features, and a label (1 if the molecule is active, 0 if it is decoy).
We used this data to build a model that predicts the behavior of previously unseen molecules. The pipeline of the study starts with an exploratory data analysis, then a discussion of the methods used to preprocess the data, the different models built, and how the final prediction was made.

10.2 Challenges and results

The objective of this project was to find a suitable model to predict the behavior of previously unseen molecules. For a given molecule, we wanted to know the probability it had to bind to proteins. Our final model was given by an average of three classifiers: A random forest, a gradient boosting tree, and a neural network (with a specific structure to deal with the class imbalance using a custom initial bias in the penultimate layer).
One of the main challenges faced during this work was to deal with the strong class imbalance in the training data (only 6447 molecules out of 378447 are binding). To do so, we leveraged on various sampling methods, such as oversampling, undersampling or cluster-based sampling (See Figures 31 and 32). Each one of these methods had some inconveniences, but they allowed us to train our models on balanced sets. Using these strategies and refining them was the main source of improvement of our predictions.

10.3 Technical details

On the computational side, we used Python and the models were built with the libraries Scikit-Learn, XGBoost and Tensorflow. The code used for this project can be found at the end of the report.

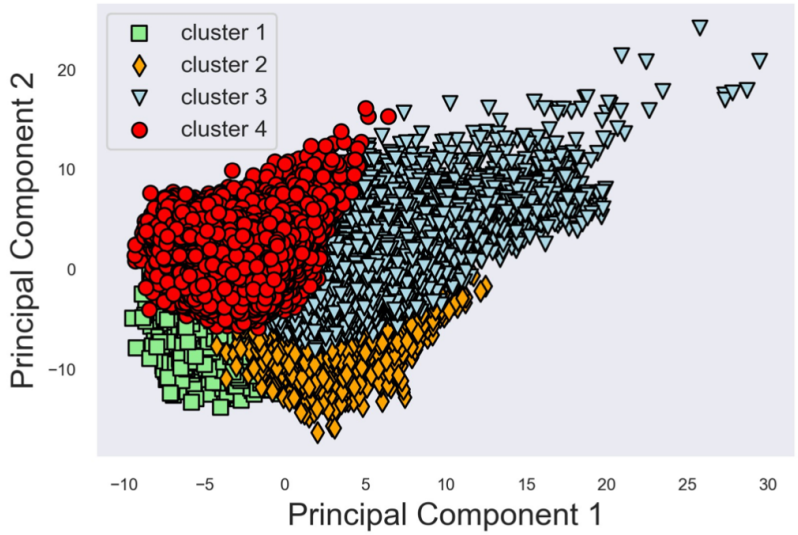


Figure 31: Four clusters of majority class against two first PCA components.

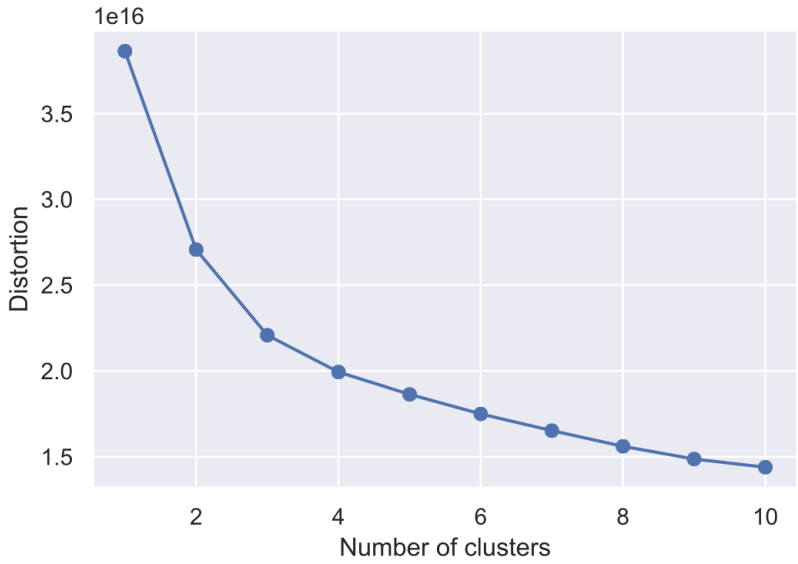


Figure 32: Distortion score after K-Means clustering for different values of K

11 Additional Projects and Entrepreneurial Ventures

In my spare time, I have explored various side projects and small businesses, gaining valuable experience in the world of entrepreneurship. While I have been involved in numerous ventures, I would like to highlight a few key projects that demonstrate the competencies I have developed, such as:

- Market analysis and benchmarking
- Navigating regulatory environments
- Developing marketing strategies
- Digital marketing (Facebook/Google Ads) and KPI monitoring
- Business acumen
- Customer service

11.1 E-Commerce: Women's Clothing Brand

I successfully managed an end-to-end women's clothing brand Beldora, overseeing both administrative and operational aspects, which included:

1. Conducting market research to identify optimal fabric types
2. Developing a comprehensive business plan, accounting for costs and production estimates
3. Acquiring industry-specific knowledge and partnering with a local tailor
4. Overseeing the production process
5. Designing and launching a Shopify website
6. Implementing digital marketing strategies
7. Collaborating with influencers and managing Facebook/Google ad campaigns
8. Coordinating with delivery companies for streamlined logistics

11.2 AI Freelance and Consulting Projects

I established a niche AI-focused freelance agency and engaged in marketing efforts such as delivering talks within AI communities. Some of my key projects include:

Stock Market Forecasting using Temporal Fusion Transformers I implemented various forecasting models, such as LSTMs and Temporal Fusion Transformers (TFT), a state-of-the-art research model with a wide array of configuration options. Key features of TFT include:

- Handling missing data
- Supporting forecasts using present and past variables
- Incorporating global input variables, such as market sentiment or seasonality

DeepEcho Computer Vision Consulting I collaborated with the innovative startup DeepEcho[11], advising the team on advanced deep learning techniques to enhance their computer vision model's performance.

Public Highways Functionality using Computer Vision I conducted research and prepared a report on advanced machine learning techniques for fraud detection in public highway systems, offering recommendations for improvement.

[11] Recognized as one of the top 30 Healthcare Innovators in Africa