Machine Learning in Materials and Chemicals Design

Dr Tom Whitehead

19th November 2025

intellegens

Introducing Intellegens

Applied machine learning

Unique ML algorithm

Easy-to-use apps

Expertise from 100s of successful projects

Our vision

"Machine learning will drive innovation and deliver value wherever data is used in R&D"

Value for our customers

Optimize products and processes

50-80% fewer experiments

Deep insights into R&D data

Chemicals

Materials

FMCG

Life Sci

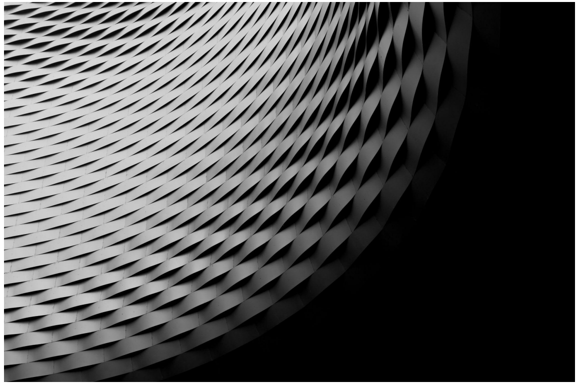
Manufacturing

Agenda

- How do we design new materials and chemicals?
- How does machine learning help?
- Practical considerations and workflows

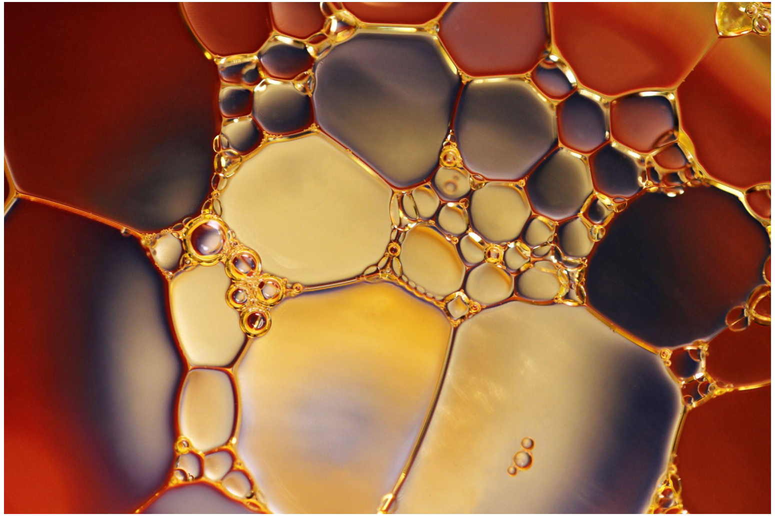
New materials and chemicals

- Where physics meets application
    - Energy transition, sustainability, advanced manufacturing
- Enormous design spaces
    - Billions of possible molecules, mixtures, or microstructures
- Data-limited regime
    - Each experiment is costly, so space is sparsely sampled
- Physical theories do not capture true complexity



New materials and chemicals

- Goal is *not* to understand materials better
- Goal is to design a material that solves business problem
- How do we find the right material for the application?



How do you solve a problem like experimental design?

Try every possible formulation

Guaranteed to find the best formulation

- May be infinitely many possibilities
- Budgets / timescales are finite

How do you solve a problem like experimental design?

Try every possible formulation

Guaranteed to find the best formulation

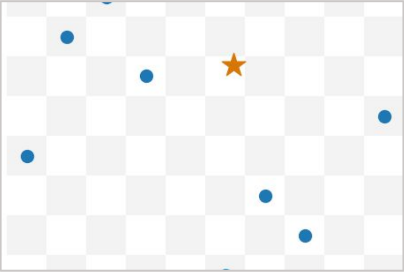
- May be infinitely many possibilities
- Budgets / timescales are finite

Ask an expert

Uses knowledge from past projects

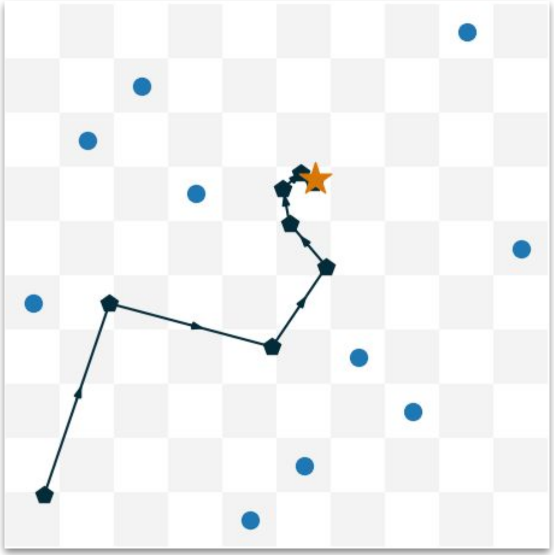
- Expensive resource
- Limited time available

How do you solve a problem like experimental design?

	**Try every possible formulation**	Guaranteed to find the best formulation	• May be infinitely many possibilities • Budgets / timescales are finite
	**Ask an expert**	Uses knowledge from past projects	• Expensive resource • Limited time available
	**Structured design / DoE**	Efficiently covers design space	• May require a large number of experiments • Requires statistical knowledge

Adaptive Experimental Design

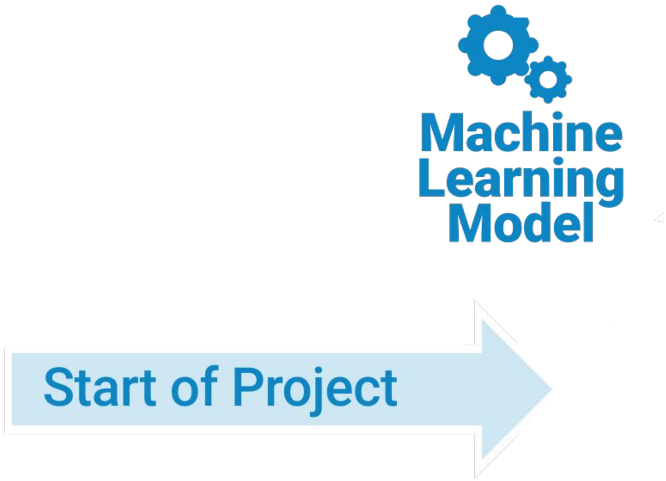
- Instead of static experimental designs, in Adaptive Experimental Design machine learning is used to iteratively update experimental suggestions as more information becomes available
- Also known as Bayesian Optimization



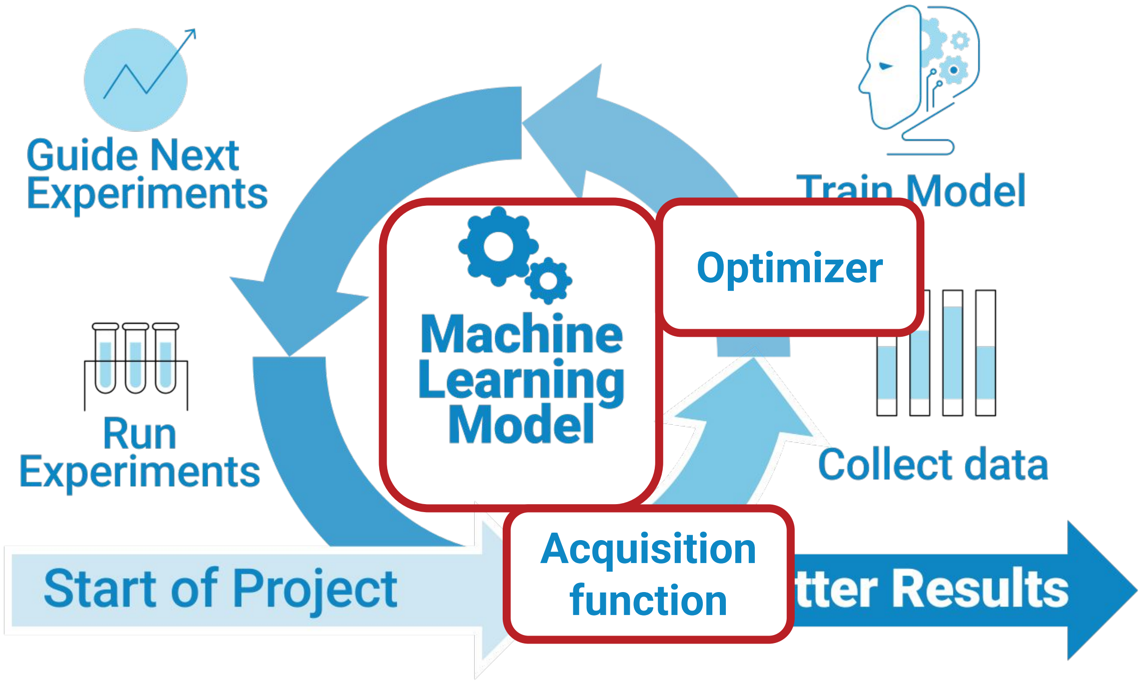
Why Adaptive Experimental Design?

- Iterative adjustments based on emerging data
- Reduced number of experiments
  - Reducing time
  - Reducing cost
- Learning-driven approach
- Less statistical background required to utilize than DoE

Adaptive Experimental Design



Adaptive Experimental Design



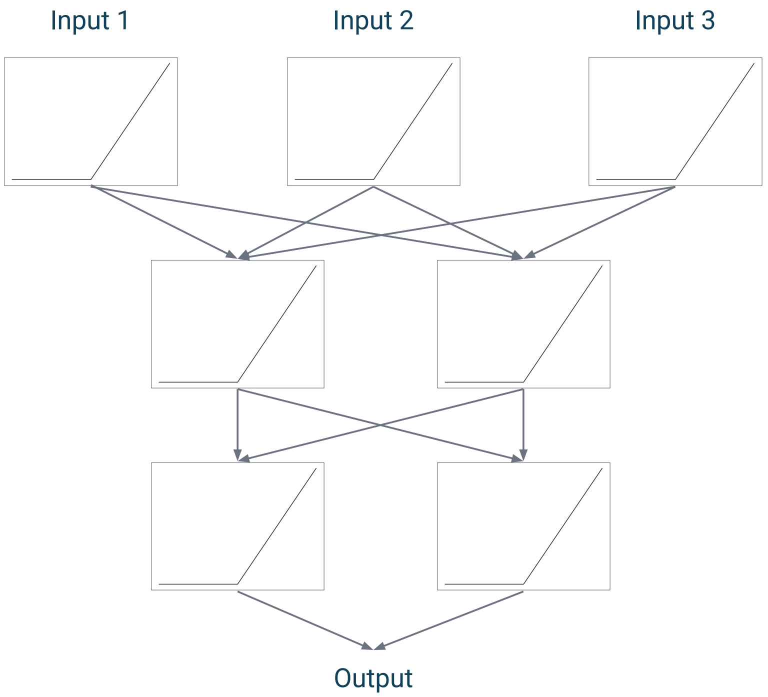
Machine Learning surrogate models

Machine Learning

- Statistics is complicated: get the computer to do it
- Multiple different ML algorithms, with different assumptions, strengths, and weaknesses
- Key questions for adaptive experimental design:
    - Will it work with the amount of data I have?
    - Does it provide uncertainty quantification?
    - How explainable is it?

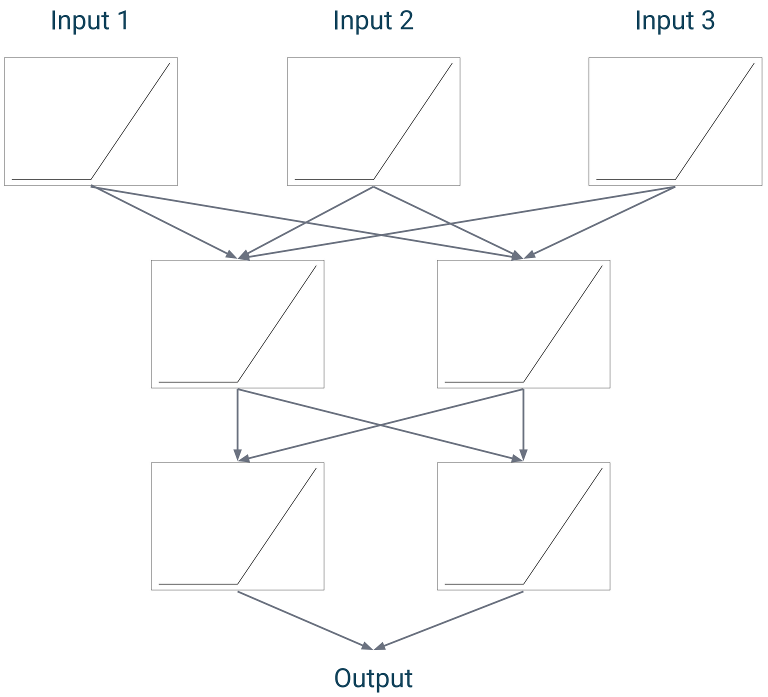
Neural Networks

- What most people think of when you say 'AI'
- With enough 'neurons' you can fit any function
    - It can need a lot of neurons



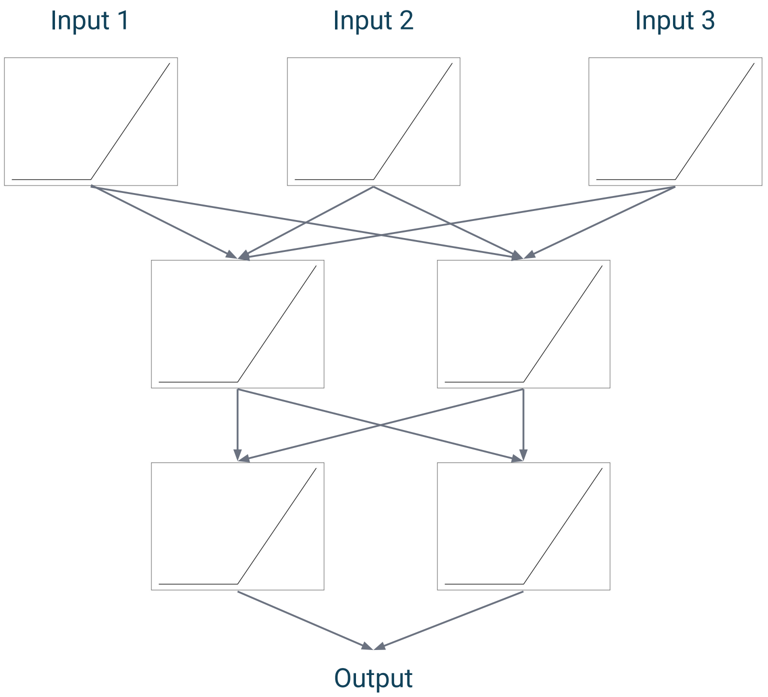
Neural Networks

- What most people think of when you say 'AI'
- With enough 'neurons' you can fit any function
  - It can need a lot of neurons
- *Will it work with the amount of data I have?*
  - Big data: ✅
  - Small data: ❌



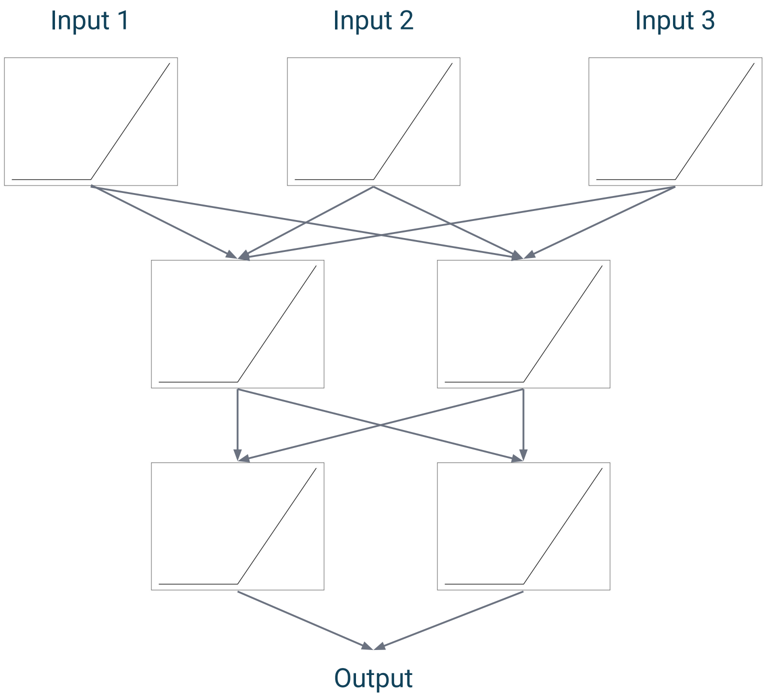
Neural Networks

- What most people think of when you say 'AI'
- With enough 'neurons' you can fit any function
    - It can need a lot of neurons
- *Will it work with the amount of data I have?*
    - Big data ✅
    - Small data ❌
- *Does it provide uncertainty quantification?*
    - Sometimes ✅



Neural Networks

- What most people think of when you say 'AI'
- With enough 'neurons' you can fit any function
    - It can need a lot of neurons
- *Will it work with the amount of data I have?*
    - Big data ✓
    - Small data ✗
- *Does it provide uncertainty quantification?*
    - Sometimes ✓
- *How explainable is it?*
    - Awful ✗



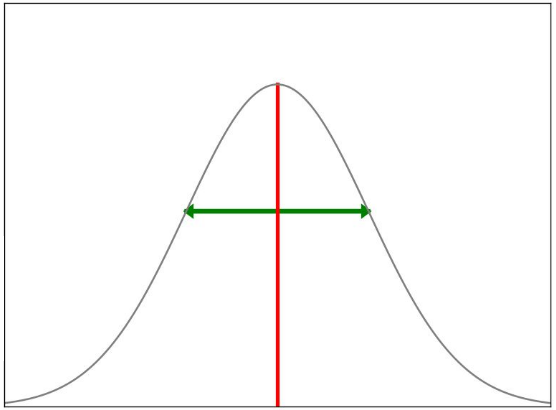
Gaussian Process Regression

- Assume all your data is multi-normally distributed, with one dimension for each data point

$$g(\mathbf{y}; \boldsymbol{\Sigma}, \boldsymbol{\mu}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right)$$

- Find conditional distribution of test data given training data: neat formula because Gaussians

$$p(\mathbf{t}|\mathbf{y}) = \frac{1}{(2\pi)^{d_t/2}|\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{\mathbf{yy}}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{t}-\boldsymbol{\Sigma}_{\mathbf{yt}}\boldsymbol{\Sigma}_{\mathbf{yy}}^{-1}\mathbf{y})^{\top}(\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{\mathbf{yy}})^{-1}(\mathbf{t}-\boldsymbol{\Sigma}_{\mathbf{yt}}\boldsymbol{\Sigma}_{\mathbf{yy}}^{-1}\mathbf{y})\right)$$



- *Will it work with the amount of data I have?*
  - Big data: ✗
  - Small data: ✓

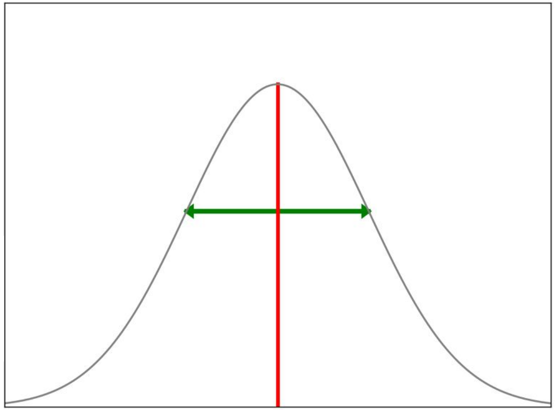
Gaussian Process Regression

- Assume all your data is multi-normally distributed, with one dimension for each data point

$$g(\mathbf{y}; \boldsymbol{\Sigma}, \boldsymbol{\mu}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right)$$

- Find conditional distribution of test data given training data: neat formula because Gaussians

$$p(\mathbf{t}|\mathbf{y}) = \frac{1}{(2\pi)^{d_t/2}|\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{\mathbf{yy}}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{t}-\boldsymbol{\Sigma}_{\mathbf{yt}}\boldsymbol{\Sigma}_{\mathbf{yy}}^{-1}\mathbf{y})^\top (\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{\mathbf{yy}})^{-1}(\mathbf{t}-\boldsymbol{\Sigma}_{\mathbf{yt}}\boldsymbol{\Sigma}_{\mathbf{yy}}^{-1}\mathbf{y})\right)$$



- *Will it work with the amount of data I have?*
  - Big data:
  - Small data:
- *Does it provide uncertainty quantification?*

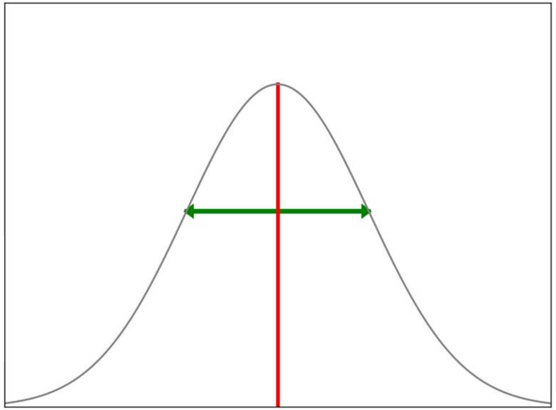
Gaussian Process Regression

- Assume all your data is multi-normally distributed, with one dimension for each data point

$$g(\mathbf{y}; \boldsymbol{\Sigma}, \boldsymbol{\mu}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right)$$

- Find conditional distribution of test data given training data: neat formula because Gaussians

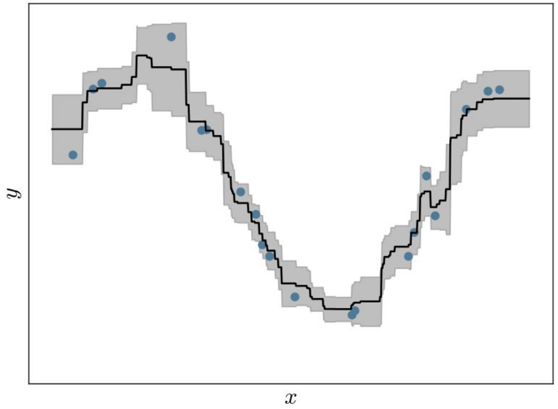
$$p(\mathbf{t}|\mathbf{y}) = \frac{1}{(2\pi)^{d_t/2}|\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{\mathbf{yy}}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{t}-\boldsymbol{\Sigma}_{\mathbf{yt}}\boldsymbol{\Sigma}_{\mathbf{yy}}^{-1}\mathbf{y})^\top (\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{\mathbf{yy}})^{-1}(\mathbf{t}-\boldsymbol{\Sigma}_{\mathbf{yt}}\boldsymbol{\Sigma}_{\mathbf{yy}}^{-1}\mathbf{y})\right)$$



- *Will it work with the amount of data I have?*
  - Big data:
  - Small data:
- *Does it provide uncertainty quantification?*
- *How explainable is it?*

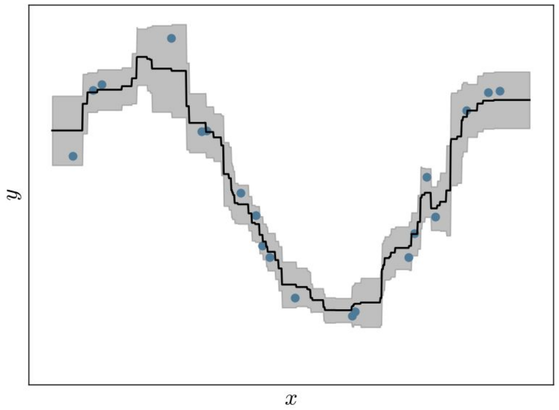
Random Forests

- Train decision trees on bootstrap samples of data
- Average over trees to reduce variance in predictions without increasing bias (much)



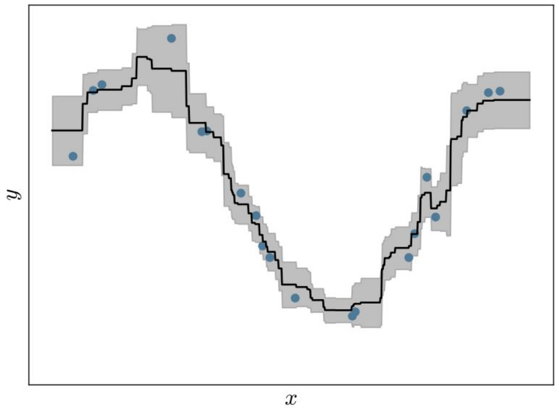
Random Forests

- Train decision trees on bootstrap samples of data
- Average over trees to reduce variance in predictions without increasing bias (much)
- *Will it work with the amount of data I have?*
  - Big data:
  - Small data:



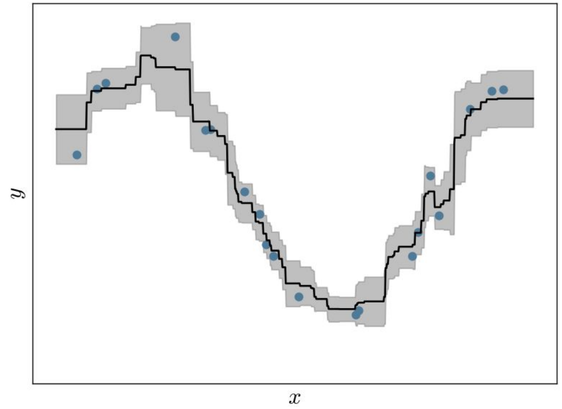
Random Forests

- Train decision trees on bootstrap samples of data
- Average over trees to reduce variance in predictions without increasing bias (much)
- *Will it work with the amount of data I have?*
  - Big data: ✓
  - Small data: ✓
- *Does it provide uncertainty quantification?* ✓



Random Forests

- Train decision trees on bootstrap samples of data
- Average over trees to reduce variance in predictions without increasing bias (much)
- *Will it work with the amount of data I have?*
  - Big data: ✓
  - Small data: ✓
- *Does it provide uncertainty quantification?* ✓
- *How explainable is it?* ✓



Machine Learning surrogate models

	Neural Networks	Gaussian Process Regression	Random Forests
Works with big data?	✓	✗	✓
Works with small data?	✗	✓	✓
Uncertainty quantification?	✓	✓	✓
Explainable?	✗	✓	✓

Acquisition Functions

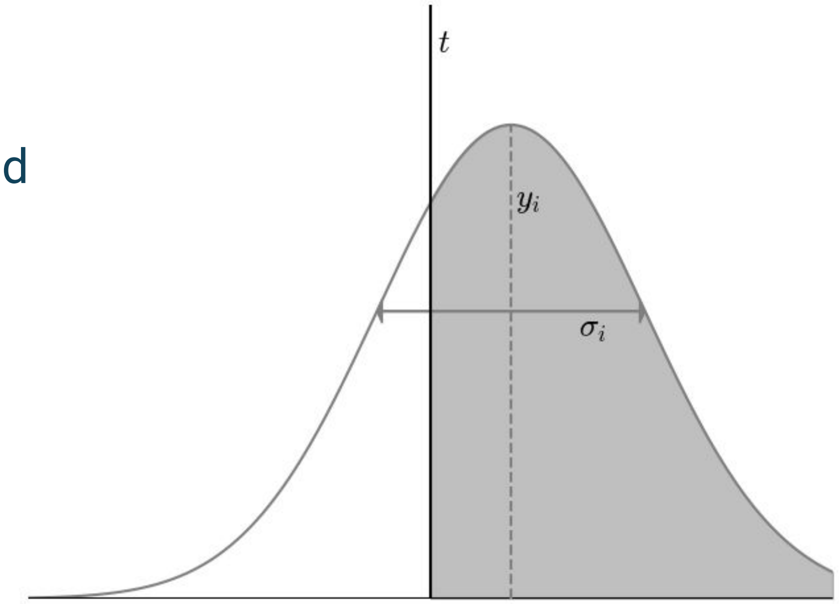
Acquisition Functions: what do we want to achieve?

- Remember we don't want to find the *best* material
- We want something 'good enough' to achieve business objectives

Probability of Improvement

- What is the probability that a suggestion achieves our target?
- Assume prediction is normally distributed

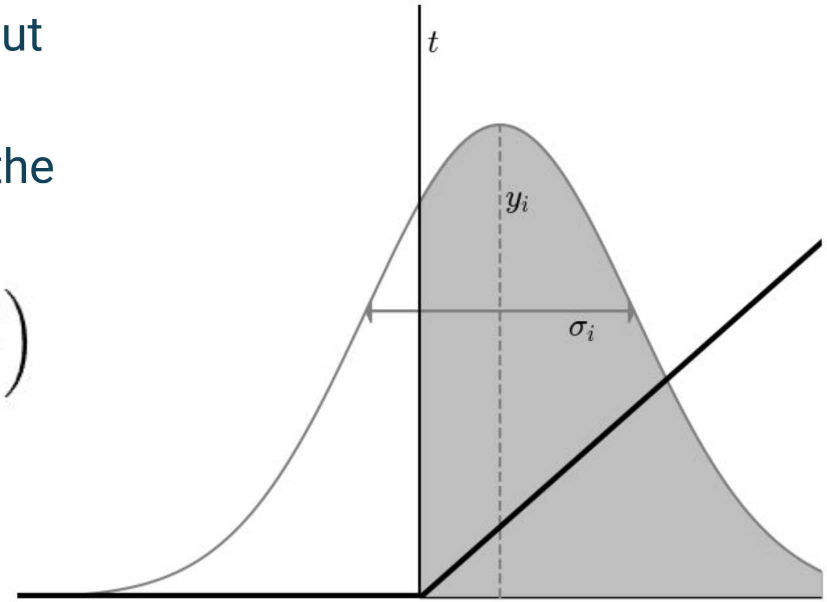
$$\mathrm{PI}(t; y_i, \sigma_i) = \Phi\left(\frac{y_i - t}{\sigma_i}\right)$$



Expected Improvement

- Objective is to achieve specified target: but overachieving is even better!
- What is the expected improvement over the target value?

$$\mathrm{EI}(t; y_i, \sigma_i) = (y_i - t)\Phi\left(\frac{y_i - t}{\sigma_i}\right) + \sigma_i \phi\left(\frac{y_i - t}{\sigma_i}\right)$$

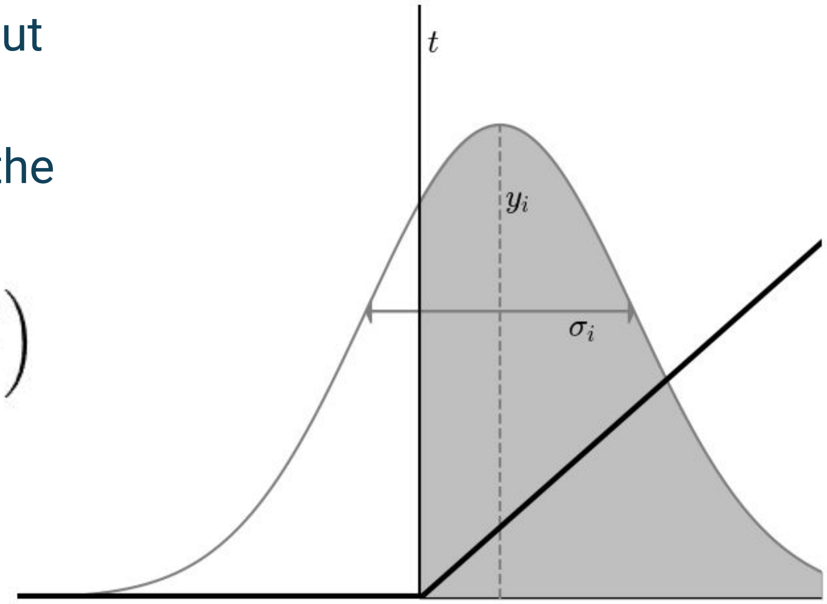


Expected Improvement

- Objective is to achieve specified target: but overachieving is even better!
- What is the expected improvement over the target value?

$$\mathrm{EI}(t; y_i, \sigma_i) = \boxed{(y_i - t)\Phi\left(\frac{y_i - t}{\sigma_i}\right)} + \sigma_i \phi\left(\frac{y_i - t}{\sigma_i}\right)$$

"Exploitation"

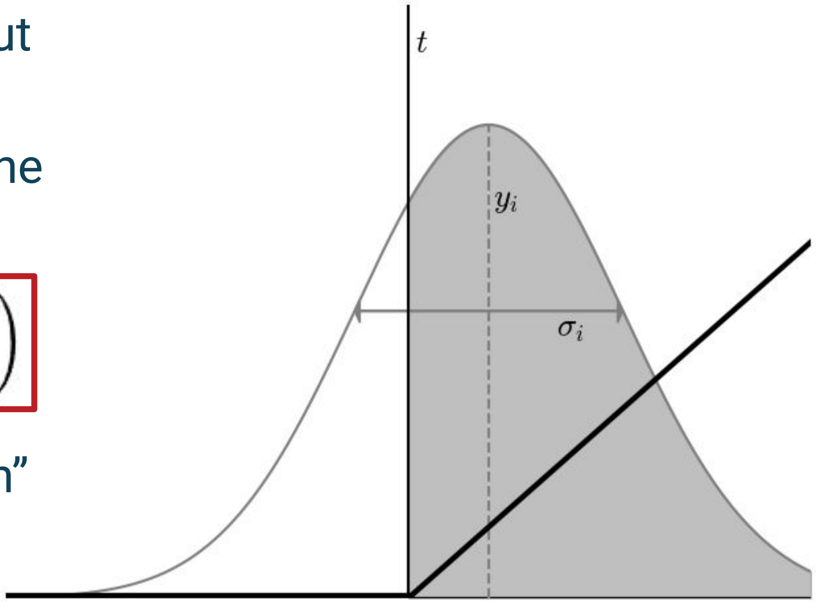


Expected Improvement

- Objective is to achieve specified target: but overachieving is even better!
- What is the expected improvement over the target value?

$$\mathrm{EI}(t; y_i, \sigma_i) = (y_i - t)\Phi\left(\frac{y_i - t}{\sigma_i}\right) + \boxed{\sigma_i \phi\left(\frac{y_i - t}{\sigma_i}\right)}$$

"Exploration"



Gotchas

- Expected Improvement is widely used in Adaptive Experimental Design

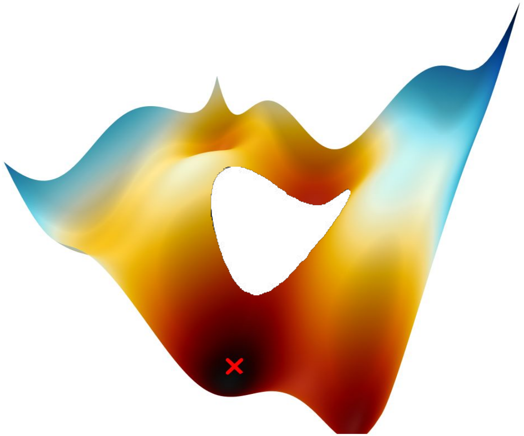
$$\mathrm{EI}(t; y_i, \sigma_i) = (y_i - t)\Phi\left(\frac{y_i - t}{\sigma_i}\right) + \sigma_i \phi\left(\frac{y_i - t}{\sigma_i}\right)$$

- Why this particular exploration/exploitation tradeoff?
- How confident are we in our uncertainty estimates?

Optimizers

How do we optimize the cost function?

- Gradient-based optimizers (gradient descent, SGD, Adam, etc) can be used if surrogate model gives gradient information
  - Neural Networks and Gaussian Processes OK - Random Forests are not smooth
  - Some types of data (e.g. categories) not really differentiable
- Gradient-based methods can struggle with non-convex constraints

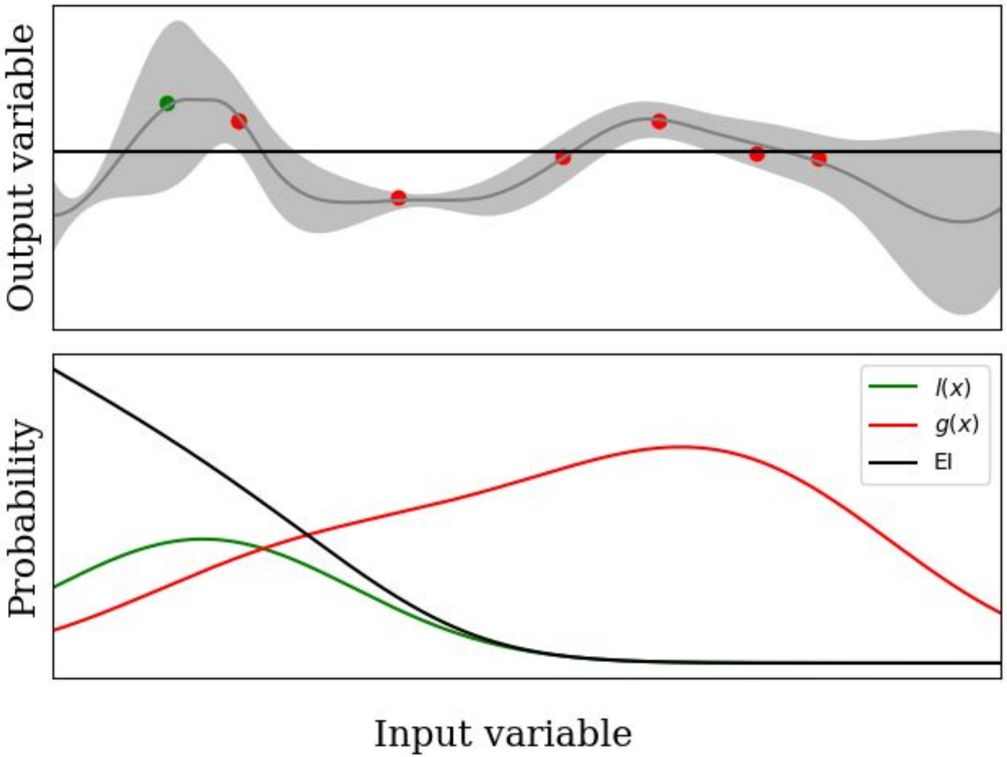


Bayesian Optimizers

- Putting the Bayesian into Bayesian Optimization
- Handle complicated, non-smooth, non-convex landscapes, at the cost of speed
- Tree-structured Parzen Estimators (TPE)
  - Select quantile $\gamma = P(y > t)$ (exploration/exploitation trade-off!)
  - Parametrize $P(x|y) = \begin{cases} l(x) & \text{if } y > t \\ g(x) & \text{if } y \leq t \end{cases}$
  - $l(x)$ and $g(x)$ constructed by including each data point as a Gaussian peak
  - Calculate EI $$\mathrm{EI}_t(x) = \int_t^\infty (y-t)P(y|x)\mathrm{d}y = \int_t^\infty (y-t)\frac{P(x|y)P(y)}{P(x)}\mathrm{d}y \propto \left(\gamma + \frac{g(x)}{l(x)}(1-\gamma)\right)^{-1}$$
  - So to maximise EI we want to find inputs where $l(x)$ is large and $g(x)$ is small

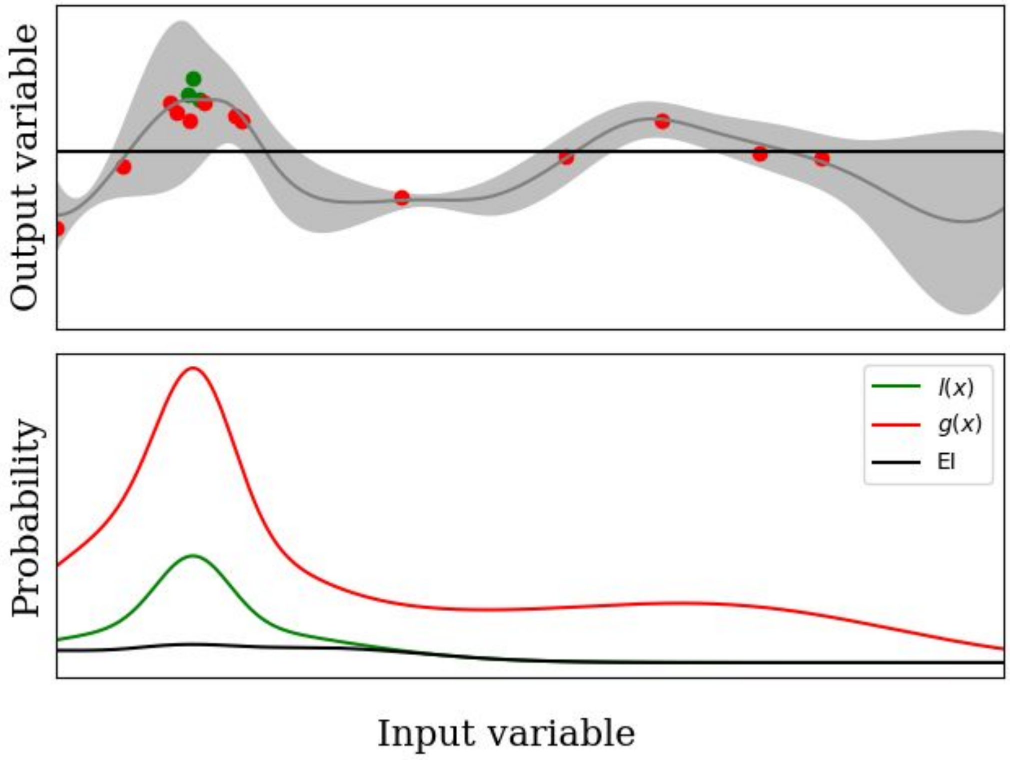
Tree-structured Parzen Estimators (TPE)

- γ = 0.2 (exploitation)
- Initial data



Tree-structured Parzen Estimators (TPE)

- γ = 0.2 (exploitation)
- After 10 new suggestions
- Note most new suggestions tightly grouped



Honorable mention: random search

- In high dimensional systems, searching randomly is very effective
- Many other optimization algorithms start with random searching to ‘seed’ the algorithm
- Particularly effective if multiple optima
- Very quick
- Not generally quite as accurate as other algorithms

Adaptive Experimental Design

How to choose a setup

- Surrogate models, acquisition functions, and optimizers should all be selected together
- Common choice is Gaussian Process surrogate model, EI acquisition function, and gradient-based optimizer
- BUT this struggles with realistic constraints, categorical options, high dimensionality

Case Study: Heat Exchanger at NASA

- Objective: design more efficient heat exchanger
- Design space:
    - Material composition, height, width, splay, etc (continuous)
    - Base shape, configuration (categorical)
- Objectives:
    - Minimize base area, thermal resistance

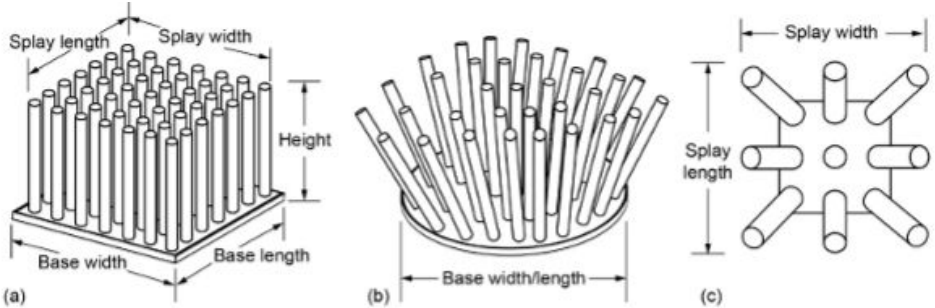
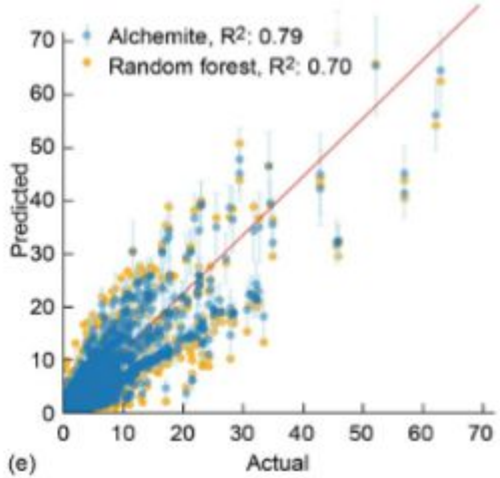


Figure 1.—Illustration of heat exchangers and size variables.

Case Study: Heat Exchanger at NASA

- Data scraped from heat exchanger vendor
- Intellegens' Alchemite™ ML surrogate model outperformed Random Forest
- Focus on exploitation, Probability of Improvement acquisition function
- TPE optimizer as non-smooth design space



Case Study: Heat Exchanger at NASA

- Suggested design: to maximize airflow, include an integrated fan!
  - Permitted in the design space
  - But NASA actually wanted no energy input: needed to adjust design space
  - https://ntrs.nasa.gov/citations/20220008637
- Across dozens of projects we typically see 50-80% reduction in number of experiments needed

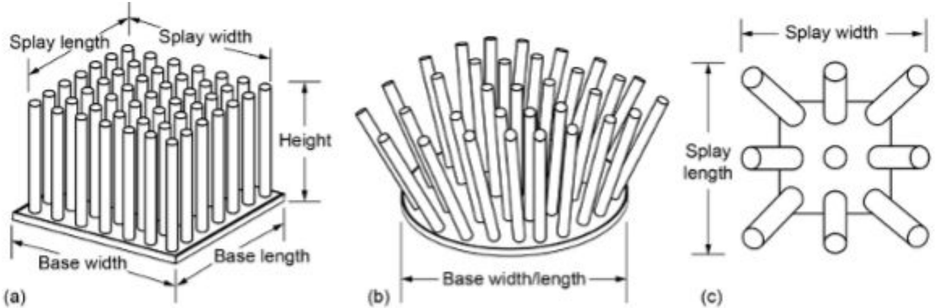


Figure 1.—Illustration of heat exchangers and size variables.

Summary

Summary

- Adaptive Experimental Design uses Machine Learning to accelerate materials and chemicals design
- Multiple tools in the toolkit: consider mathematical properties of tools and problems to select between them
- Accelerate R&D using machine learning

Questions?

tom@intellegens.com

intellegens.com

/in/tom-whitehead-33a319131/

/company/intellegensai

SUBSCRIBE for monthly newsletter
and webinar alerts

intellegens.com/subscribe