Bayesian optimization of expensive cost functions with application in machine learning

Sanjan Gupta

(Reed Group)

Systems Seminar

7th Apr, 2017

Machine Learning (ML)

- Algorithms that learn iteratively from data

- Supervised ML:

  Given $(\underline{x}_1, y_1), (\underline{x}_2, y_2), \ldots (\underline{x}_t, y_t)$

  Learn $h: \underline{x} \rightarrow y$

  Predict $h(\underline{x}_{t+1})$

discrete y's ➔ classification

continuous y's ➔ regression

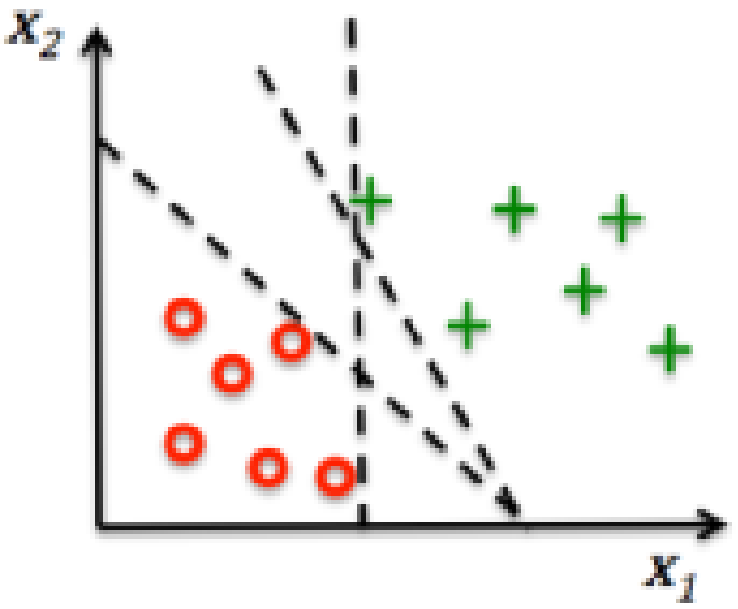
Strengths of ML:

- Ability to infer patterns in data
- Difficult to build first principles based models
- Adaptive in nature

Use of machine learning in a wide range of applications

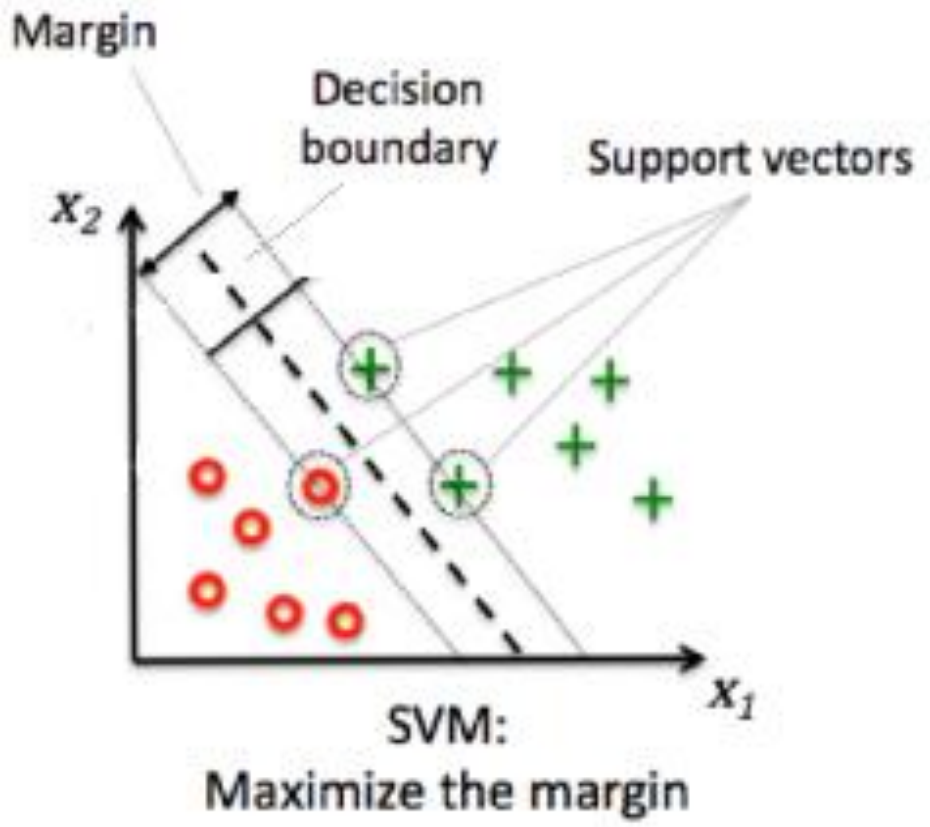
Support Vector Machine (SVM)

- To classify a given query as positive or negative example
- Multiple possible hyper-planes



Support Vector Machine (SVM)

- To classify a given query as positive or negative example
- Multiple possible hyper-planes
- Choose the one that yields maximum margin of separation

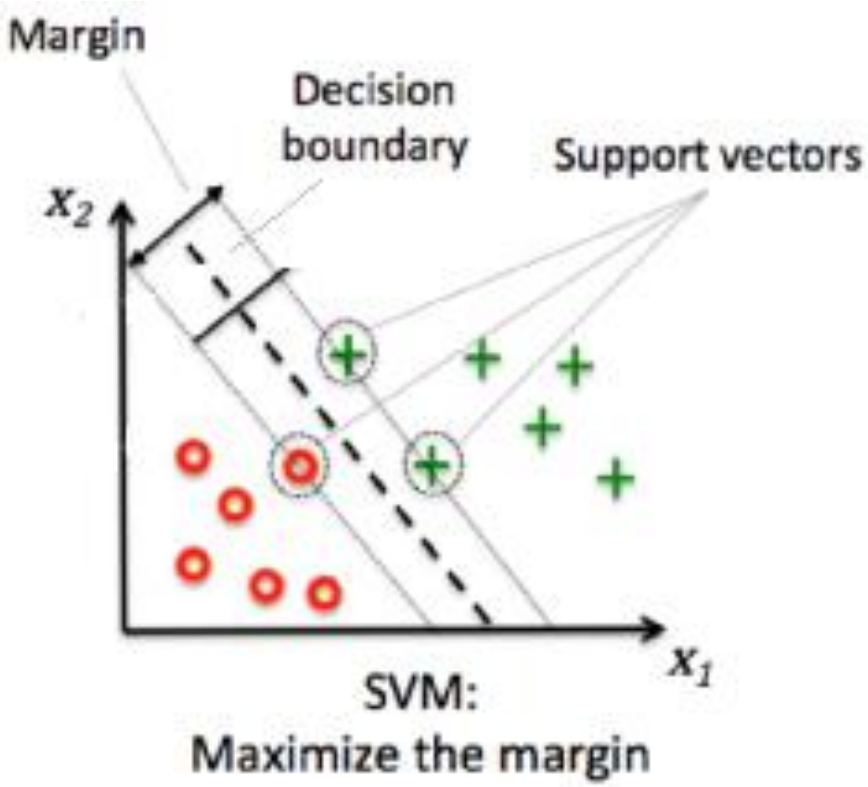


SVM formulation

Margin = $\frac{1}{2}\hat{w}^{\mathsf{T}}(x_+ - x_-) = \frac{1}{\|w\|_2}$

where x_+, x_- - closest positive & negative instance resp.

w – vector of weights defining the hyper-plane



$$\underset{w,b}{\text{minimize}} \quad \frac{1}{2}\| w \|_2^2$$

subject to constraints : $y^{(i)}(w^{\mathsf{T}}x^{(i)} + b) \geq 1$

for $i = 1,\ldots, m$

where m – # of instances

y – class label

b – bias (or const. term)

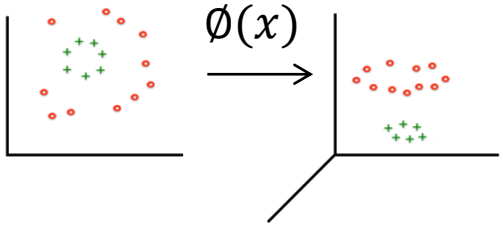
SVM contd. ...

Soft margin

$$\min_{w,\,b} \frac{1}{2}||w||^2 + C\left(\sum_{i=1}^{m} \varepsilon_i\right)$$

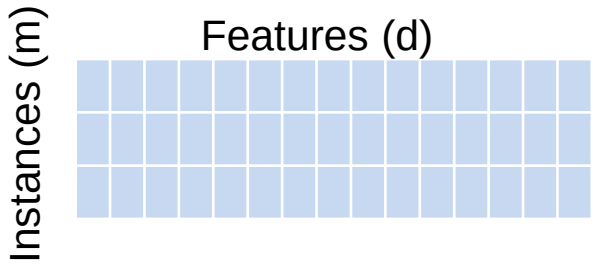
$$s.t. \quad y_i\,(w^T \emptyset(x_i) + b) \geq 1 - \varepsilon_i \qquad \forall\, i \in [1; m]$$

Transformation of data



Some typical challenges

- Tuning hyper-parameters
- Expensive black-box function
- High dimensional dataset (m << d)



" Need automatic approaches that can optimize performance of any ML algorithm in min. no. of evaluations"

➔ **Bayesian Optimization**

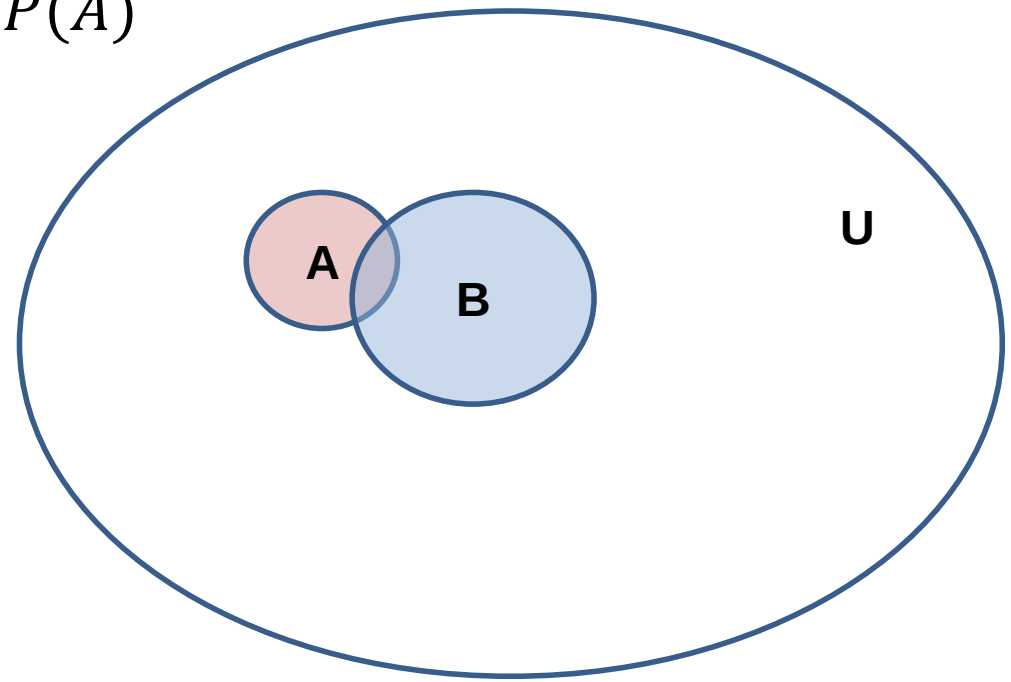
Prob 101 !

- $P(A|B) = \frac{P(A,B)}{P(B)}$

$\propto P(A,B)$

$= P(B|A)P(A)$

$[\because \mathrm{P(B)} = \sum_a P(A,B)\ ]$

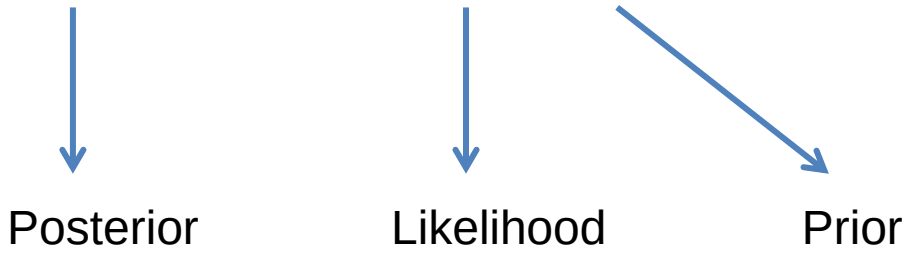


Problem statement

$$\max_x f(x)$$

where f – machine learning problem's objective function

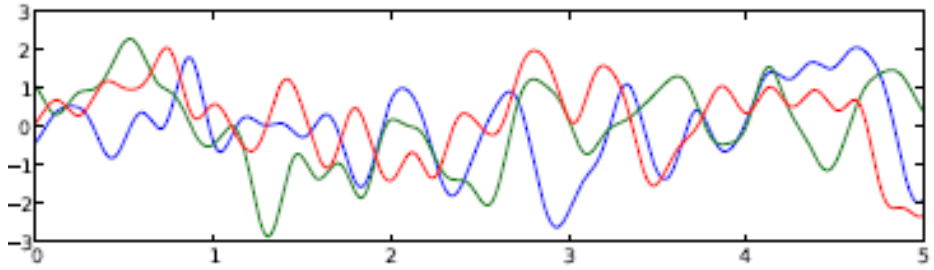
- Given: Set of 't' observations, $D_{1:t} = \{ x_{1:t} , f(x_{1:t}) \}$
- $P(f \mid D_{1:t}) \propto P(D_{1:t} \mid f)\, P(f)$



[follows from Bayes thm: $P(A|B) \propto P(B| A)\, P(A)$]

Defining the prior

- Say, f(x) ~ GP (m(x), k(x, .))



$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$$

Functions sampled from Gaussian Process with squared exponential kernel

- Kernel matrix corresponding to ‘t’ observations:

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \dots & k(\mathbf{x}_1, \mathbf{x}_t) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_t, \mathbf{x}_1) & \dots & k(\mathbf{x}_t, \mathbf{x}_t) \end{bmatrix}$$

Deriving the posterior

$$\begin{bmatrix} f_{1:t} \\ f_{t+1} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \boldsymbol{K} & k \\ k^T & k(x_{t+1}, x_{t+1}) \end{bmatrix}\right)$$

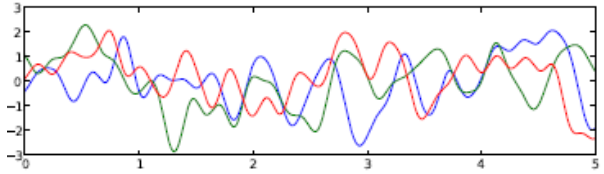
$$where \qquad k = [k(x_{t+1}, x_1)k(x_{t+1}, x_2) \dots k(x_{t+1}, x_t)]$$

- The desired conditional density would be given by

$$P(f_{t+1}|D_{1:t}, x_{t+1}) \sim \mathcal{N}(\mu_t(x_{t+1}), \sigma^2{}_t(x_{t+1}))$$

$$where \qquad \mu_t(x_{t+1}) = k^T \boldsymbol{K}^{-1} f_{1:t}$$
$$\sigma^2{}_t(x_{t+1}) = k(x_{t+1}, x_{t+1}) - k^T \boldsymbol{K}^{-1} k$$

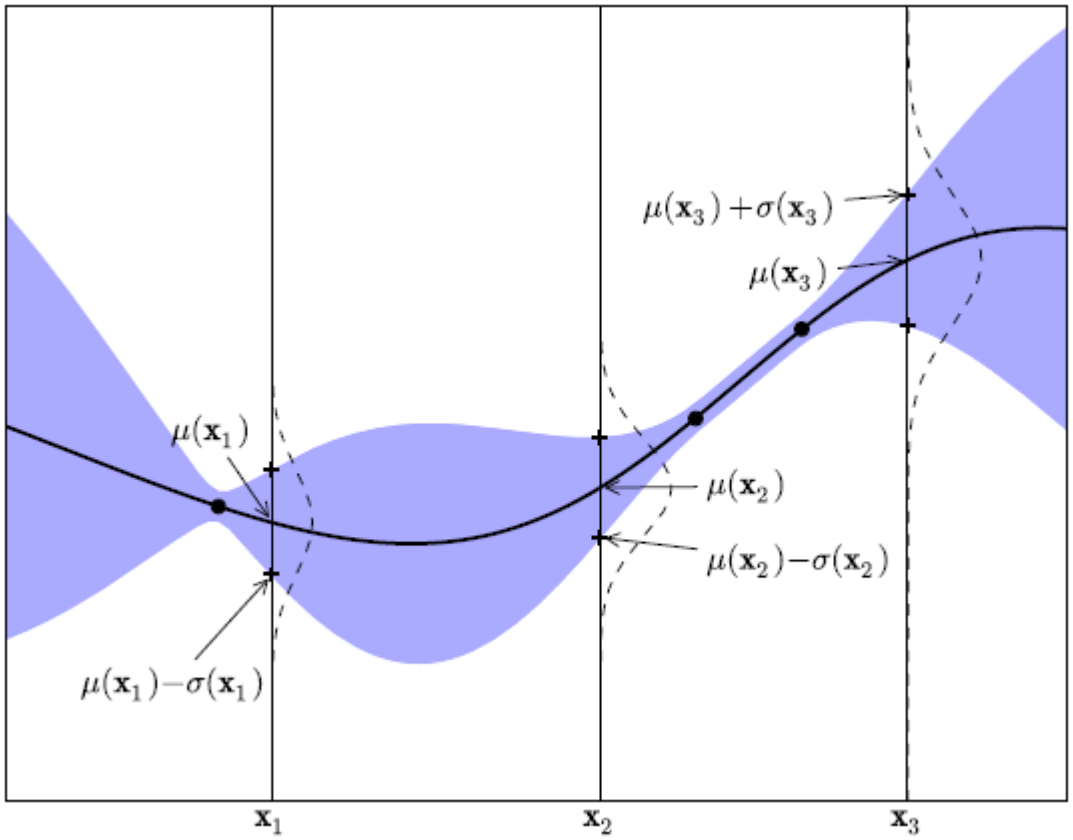
Posterior



$$P(f_{t+1}|\mathcal{D}_{1:t}, \mathbf{x}_{t+1}) = \mathcal{N}\left(\mu_t(\mathbf{x}_{t+1}), \sigma_t^2(\mathbf{x}_{t+1})\right)$$

where

$$\begin{aligned} \mu_t(\mathbf{x}_{t+1}) &= \mathbf{k}^T\mathbf{K}^{-1}\mathbf{f}_{1:t} \\ \sigma_t^2(\mathbf{x}_{t+1}) &= k(\mathbf{x}_{t+1}, \mathbf{x}_{t+1}) - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k}. \end{aligned}$$

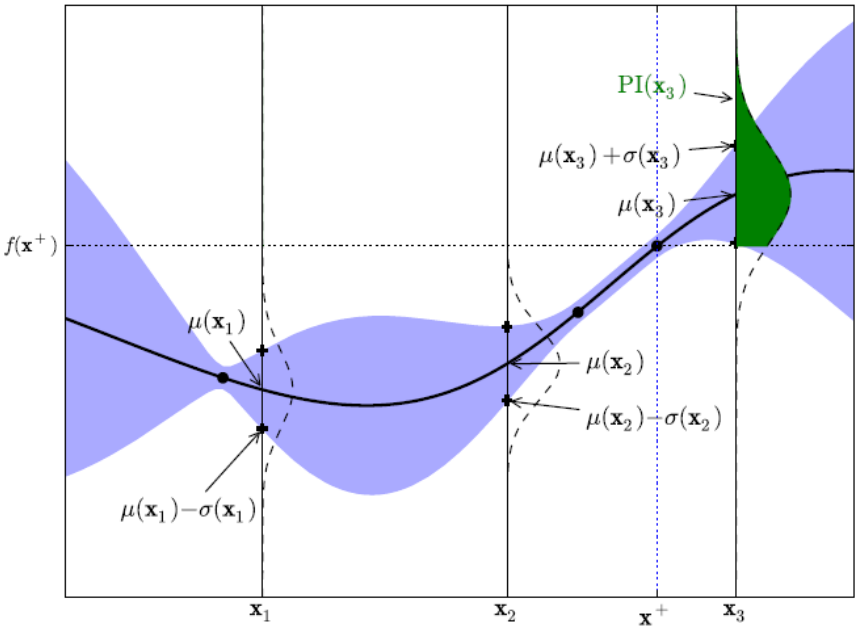


1D Gaussian Process with three observations (black dots)

Acquisition functions

- Improvement-based acquisition function:

$$PI(x) = P\big(f(x) \geq f(x^+)\big)$$
$$where\ x^+ = argmax_{x_i \in x_{1:t}} f(x_i)$$

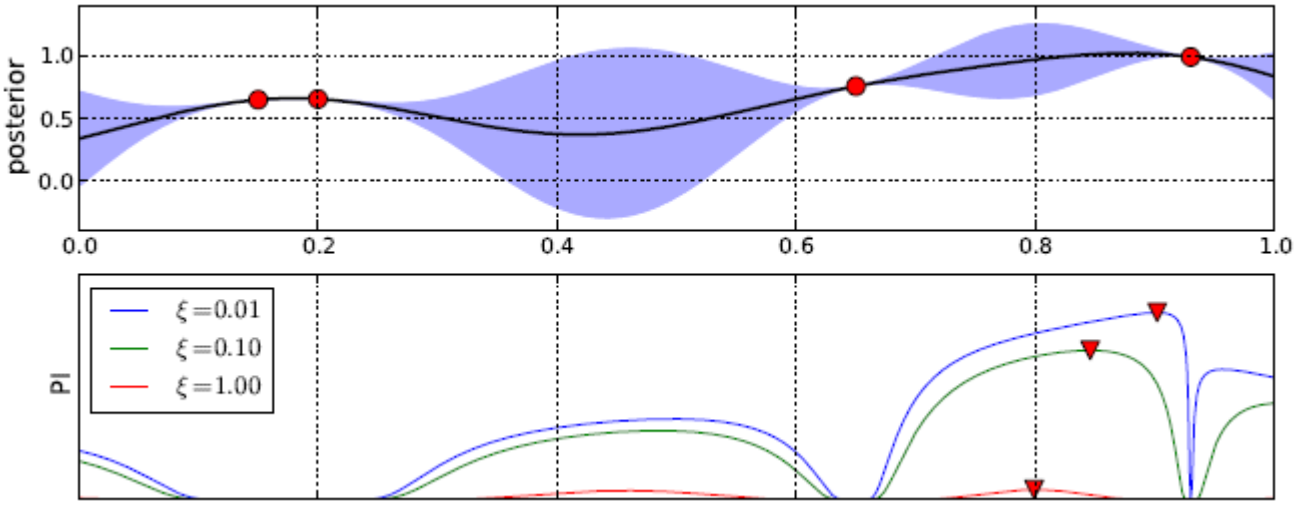


- Introducing a trade-off parameter, $\varepsilon > 0$

$$PI(x) = P(f(x) \geq f(x^+) + \varepsilon)$$

$$= \emptyset\left(\frac{\mu(x) - f(x^+) - \varepsilon}{\sigma(x)}\right)$$

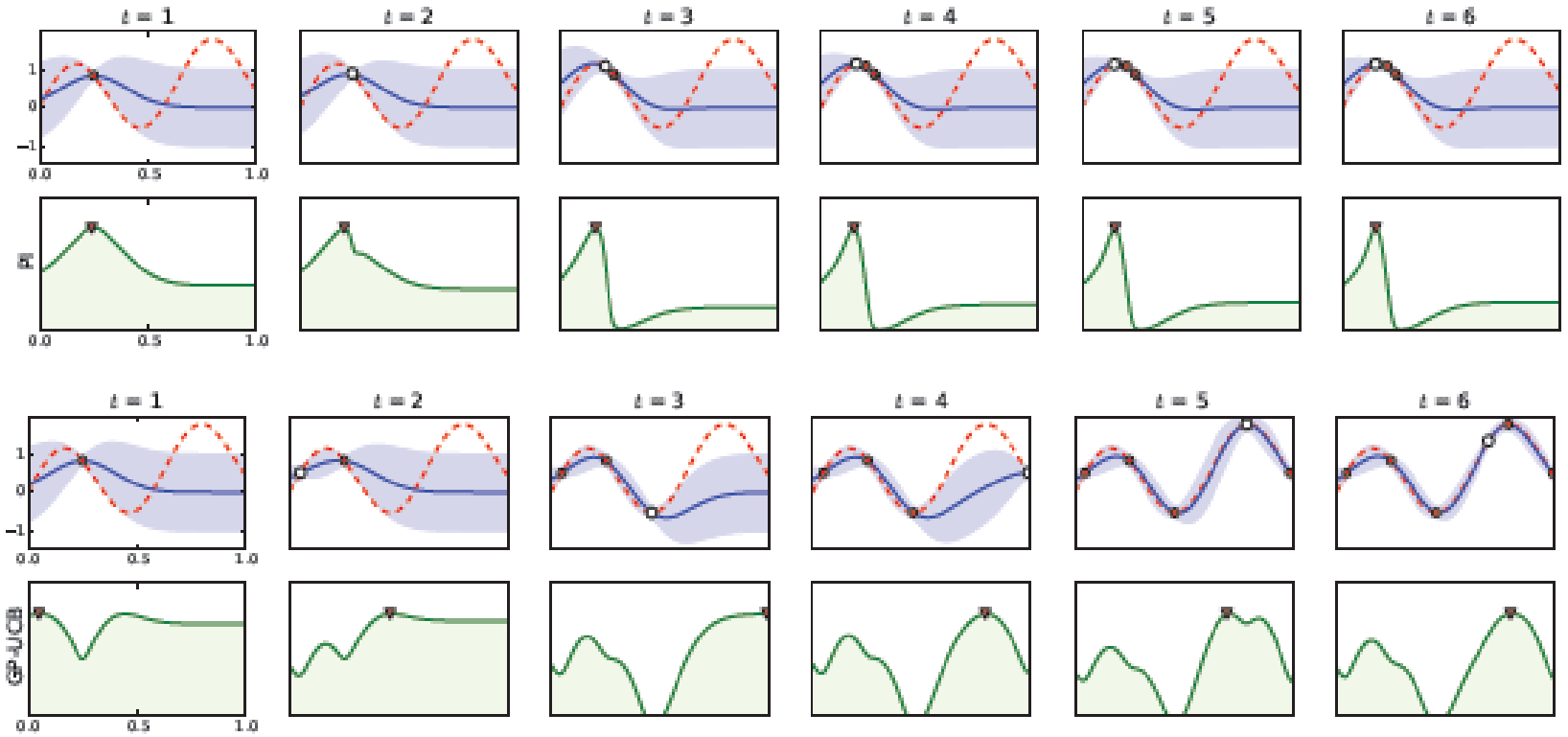
where $\emptyset - Normal\ CDF$



Example of acquisition function: Probability of Improvement (PI)

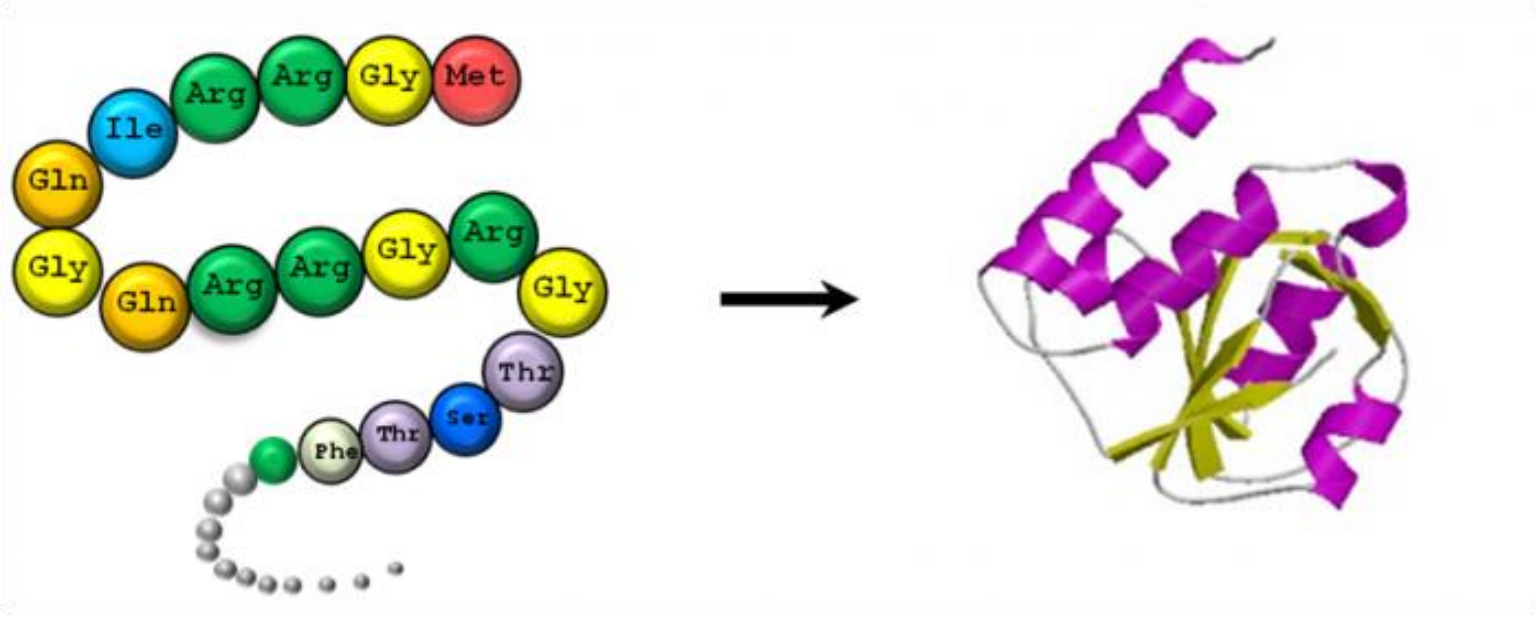
- Confidence bound criteria:

$$UCB(x) = \mu(x) + K\,\sigma(x)$$



Comparison of the two acquisition functions: Probability of Improvement (PI) and Upper Confidence Bound (UCB).

Protein engineering



- Search space for 100mer – 20^{100}
- Complex design criteria (high activity, better stability & folding kinetics, high expression ...)

[Image Source: https://www.ebi.ac.uk/training/online/course/introduction-protein-classification-ebi/protein-classification]

Summary

- Pros

  - derivative free global optimizer

  - works for expensive cost functions & high dimensional datasets

  - powerful tool for active learning – pts. to be labelled

- Cons

  - Choice of prior is critical

  - Myopic nature of acquisition functions

References

- Snoek J, Larochelle H, Adams R (2012) Practical bayesian optimization of machine learning algorithms. *NIPS*

- Brochu E, Cora VM, Freitas N (2010) A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning.

Thank You

http://xkcd.com/605/