Probabilistic numerics for deep learning

Presenter: Shijia Wang

Michael Osborne

Department of Engineering Science, University of Oxford

Deep Learning (DLSS) and Reinforcement Learning (RLSS) Summer School, Montreal 2017

Outline

1. Introduction
   - Probabilistic Numerics
2. Components
   - Probabilistic modeling of functions
   - Bayesian optimization as decision theory
   - Bayesian optimization for tuning hyperparameters
   - Bayesian stochastic optimization
   - Integration beats Optimization
3. Conclusion
   - Experiments

Outline

1. Introduction
   - Probabilistic Numerics
2. Components
   - Probabilistic modeling of functions
   - Bayesian optimization as decision theory
   - Bayesian optimization for tuning hyperparameters
   - Bayesian stochastic optimization
   - Integration beats Optimization
3. Conclusion
   - Experiments

Definition

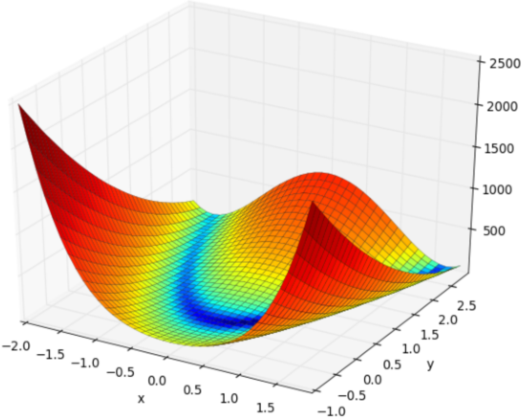
- Take the things were most interested in achieving and apply to computation
- Apply probability theory to numerics (computation cores)

Learning Algorithms

- Use numeric functions as learning algorithms
- Idea is to use Bayesian probability theories

Rosenbrock

$$f(x,y) = (1-x)^2 + 100(y-x^2)^2$$



Rosenbrock

- Easy to graph on a computer
- No easy way of finding its global optimum
- Reason: computational limits from the optimization problem

Uncertainty

- Epistemically uncertain about the function due to being unable to afford computation
- Probabilistically model function and use tools from decision theory to make optimal use of computation

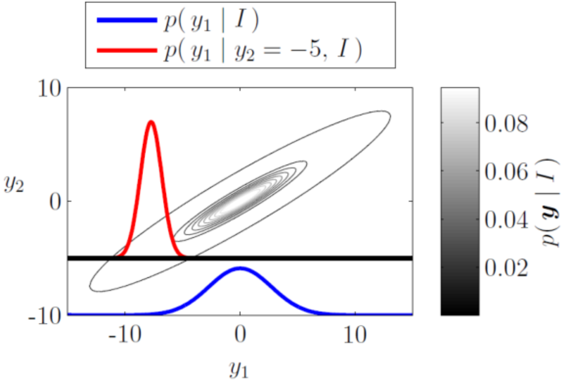
Outline

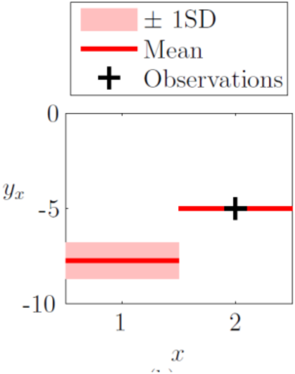
Probability

- Probability is an expression of confidence in a proposition
- Probability theory can quantify inverse of logic expression
- Depends on the agent's prior knowledge

Gaussian Distribution

Allows for distributions for variables conditioned on any other observed variables



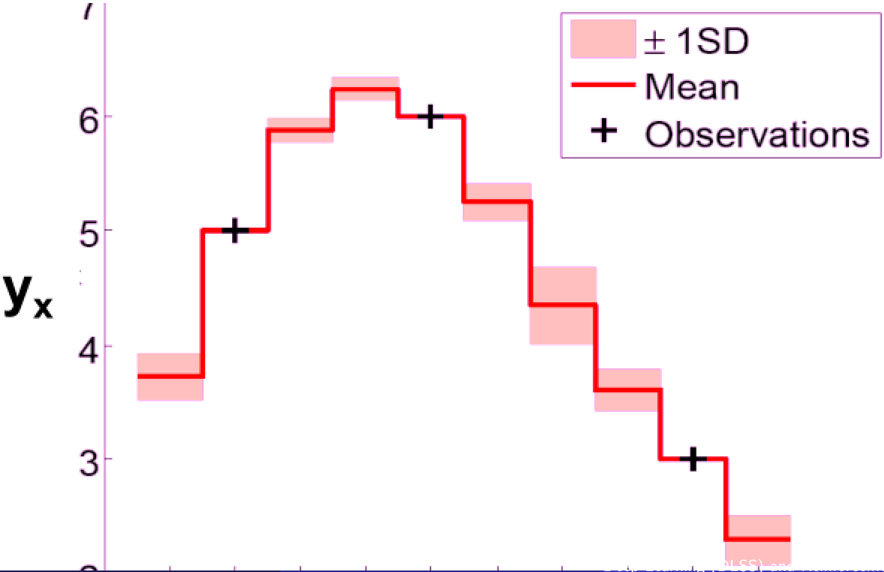


Gaussian Process

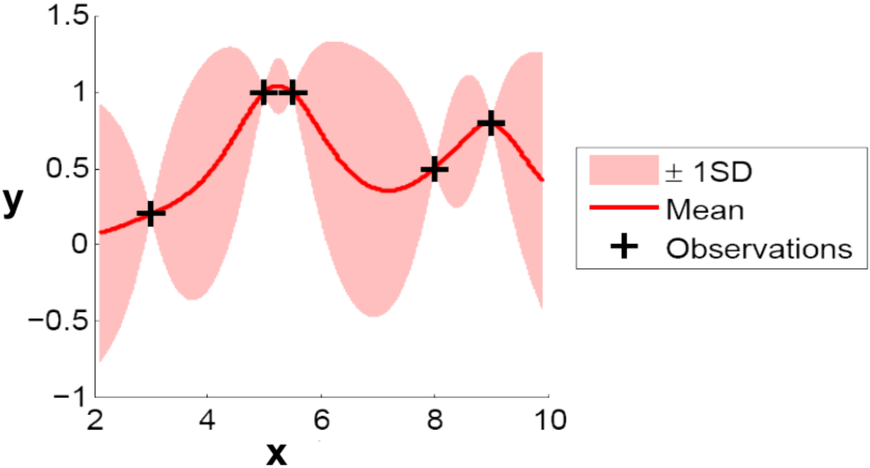
- A Gaussian process is the generalization of a multivariate Gaussian distribution to a potentially infinite number of variables
- Gives us the limit of potentially infinite number of variables infinitesimally closer together represented by an infinite-length dimension vector
- Provides non-parametric model for functions defined by mean and covariance

Gaussian Process

Infinite number of variables



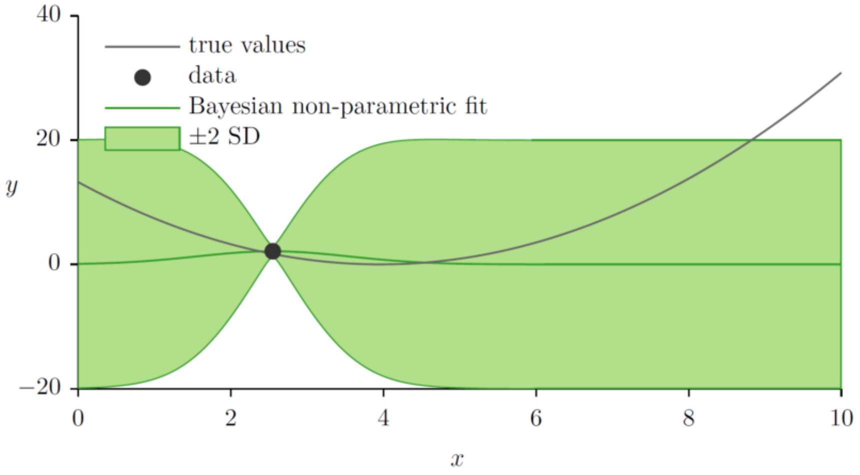
Non-parametric model



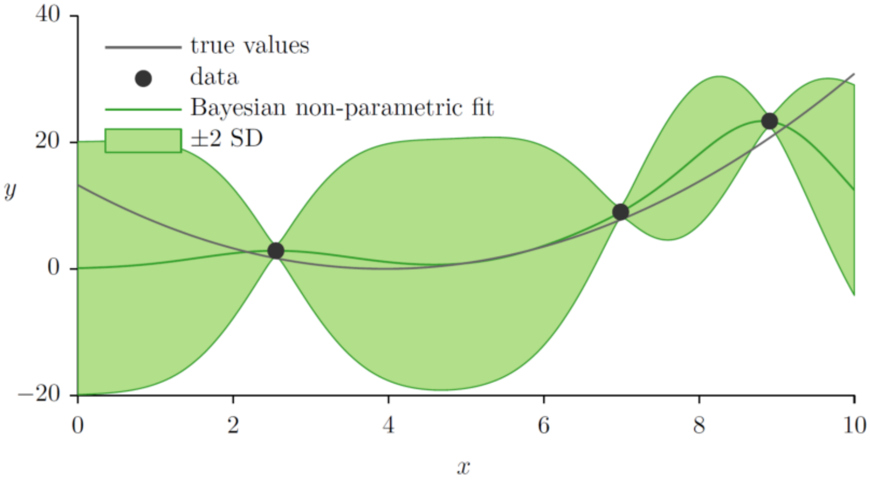
Gaussian Process

- Complexity that grows with data
- Robust to overfitting

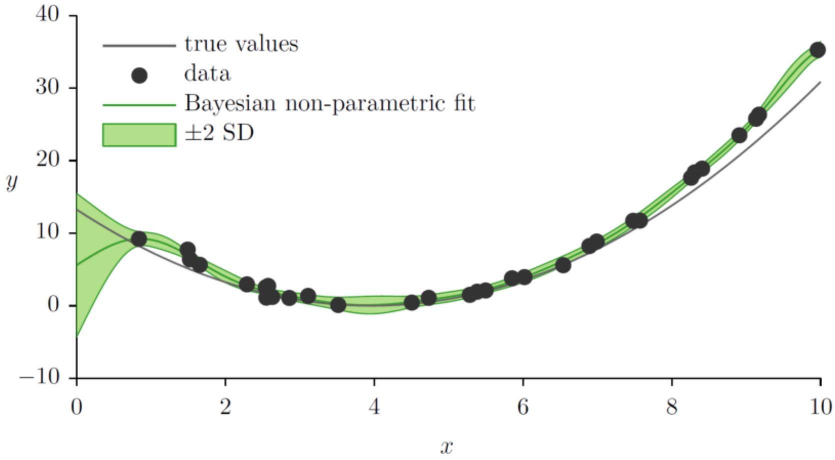
Gaussian Process



Gaussian Process



Gaussian Process



Outline

Bayesian Optimization

- Bayesian optimization is the approach of probabilistically modelling f(x,y) and using decision theory to make optimal use of computation
- by defining the costs of observation and uncertainty, we can select evaluations optimally by minimizing the expected loss with respect to a probability distribution
- Representing the core components: cost evaluation and degree of uncertainty

Loss

- loss function - lowest function value found after algorithm ends
- Take a myopic approximation and consider only the next evaluation
- The expected loss is the expected lowest value of the function we've evaluated after the next iteration

Myopic Loss

Consider only with one evaluation remaining, the loss of returning value y with current lowest value μ

$$\lambda(y) \triangleq \begin{cases} y; & y < \eta \\ \eta; & y \geq \eta \end{cases} .$$

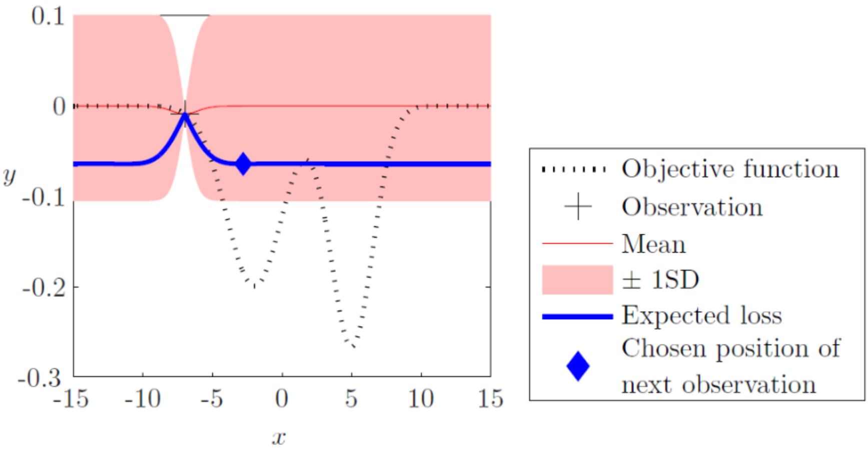
Expected Loss

Expected loss is the expected lowest value

$$\int \lambda(y)\, p(\, y \mid x,\, I_0\,)\ \mathrm{d}y$$

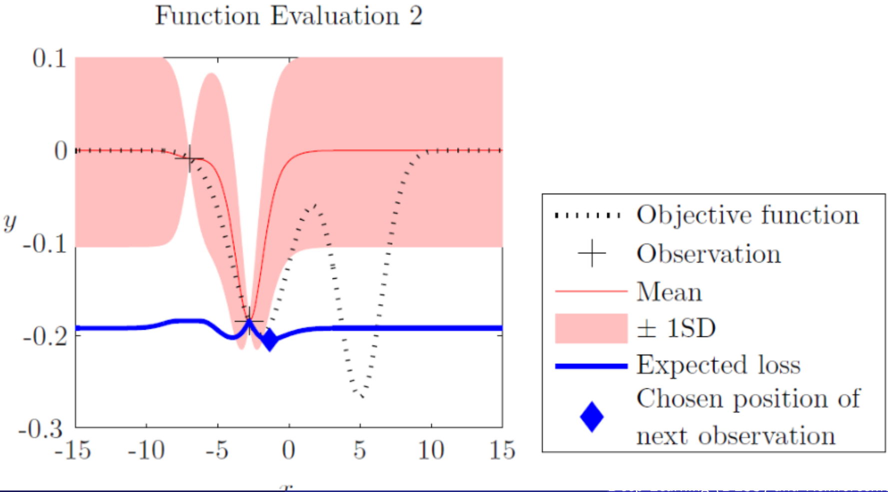
Expected Loss

Use a Gaussian process as the probability distribution for the objective function



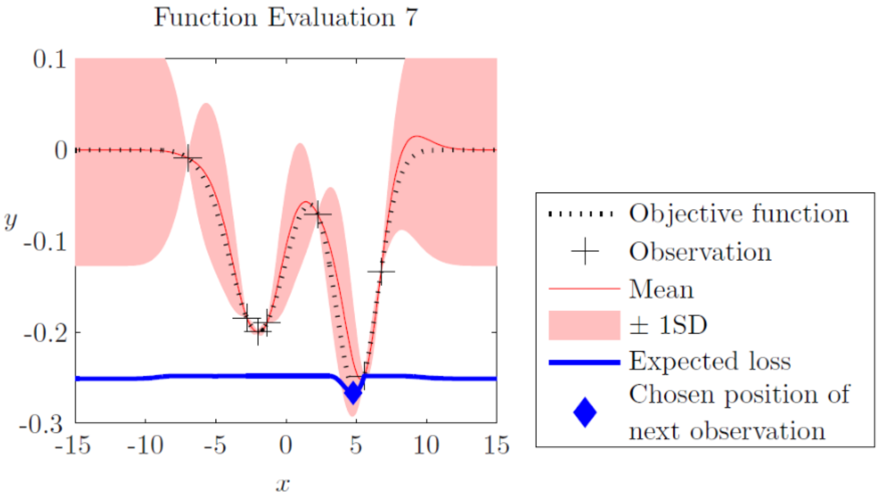
Expected Loss

Exploitative step



Expected Loss

Exploratory step



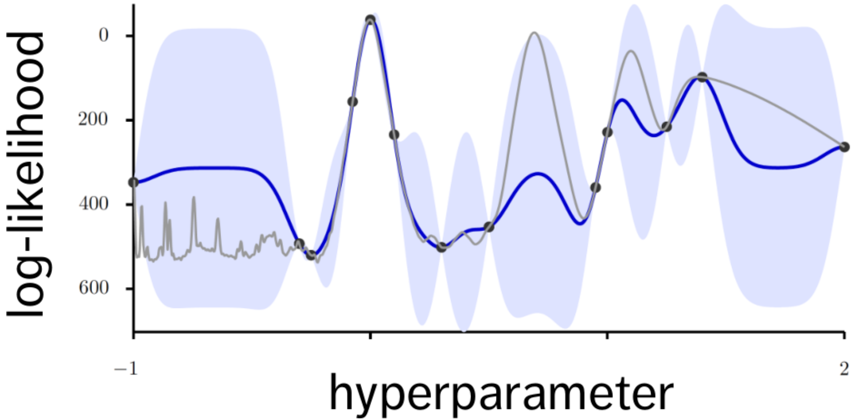
Outline

Tuning

- Tuning to cope with model parameters like periods
- Optimization gives a reasonable heuristic
- But Bayesian optimization better

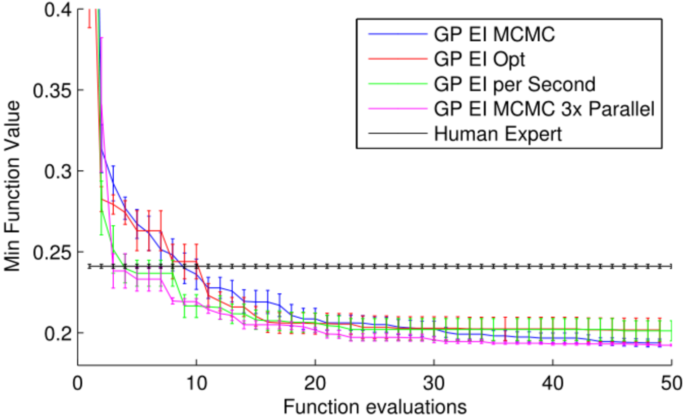
Bayesian Optimization

Better representation across hyperparameters

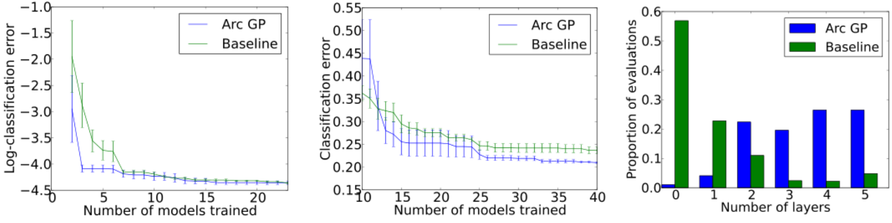


Bayesian Optimization

Tune convolutional neural networks
Allows defining the right prior information
Snoek, Larochelle and Adams (2012)



Automated structure learning
Swersky et al (2013)



(a) MNIST (b) CIFAR-10 (c) Architectures searched

Outline

Noise

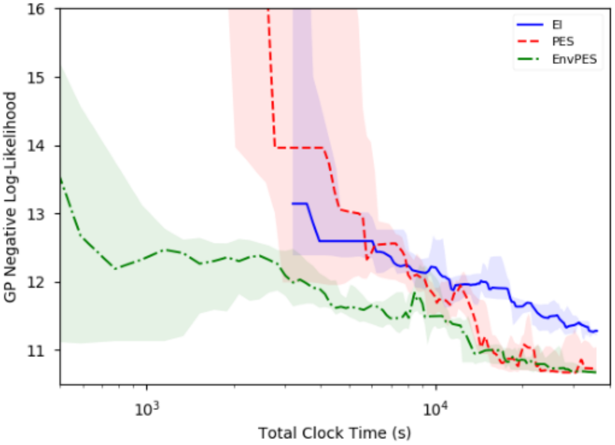
- Using only a subset of the data gives a noisy likelihood evaluation
- Use Bayesian optimization for stochastic learning

Noise

- Within Bayesian Optimization noise is not a problem
- If additional noise in the random variable we can just add a noise likelihood to complement model
- Encode that cost as a function of the number of data
- Intelligently choose the size of data that it needs at runtime to best optimization

Bayesian Optimization

Batch size
Klein, Falkner, Bartels, Hennig , Hutter (2017);
McLeod, Osborne Roberts (2017)



A Random Number

- is epistemic (personal particular to an agent) (computation is always conditional on prior knowledge)
- use useful to foil a malicious adversary (few in numerics)
- is never the minimizer of an expected loss (only when totally flat)

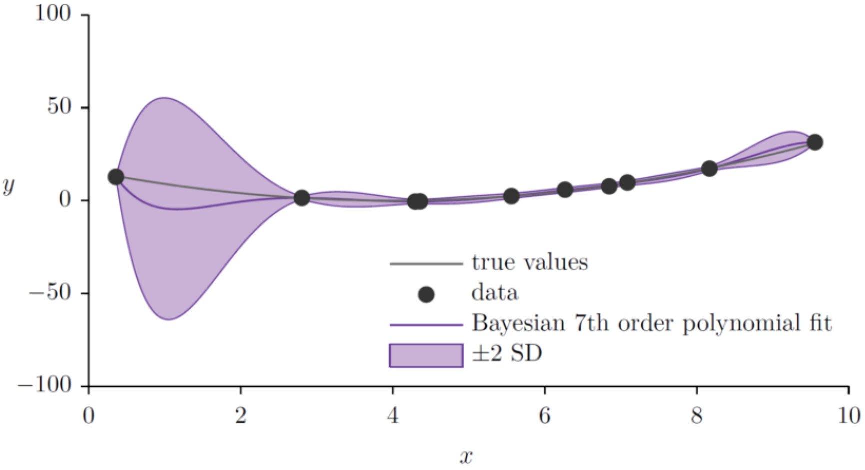
Outline

Optimization

- Naive fitting can lead to overfitting

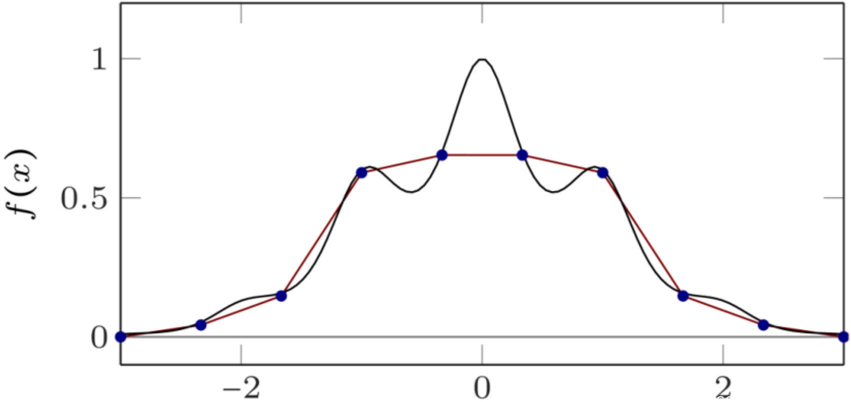
Integrating

Reduces overfitting and estimates uncertainty



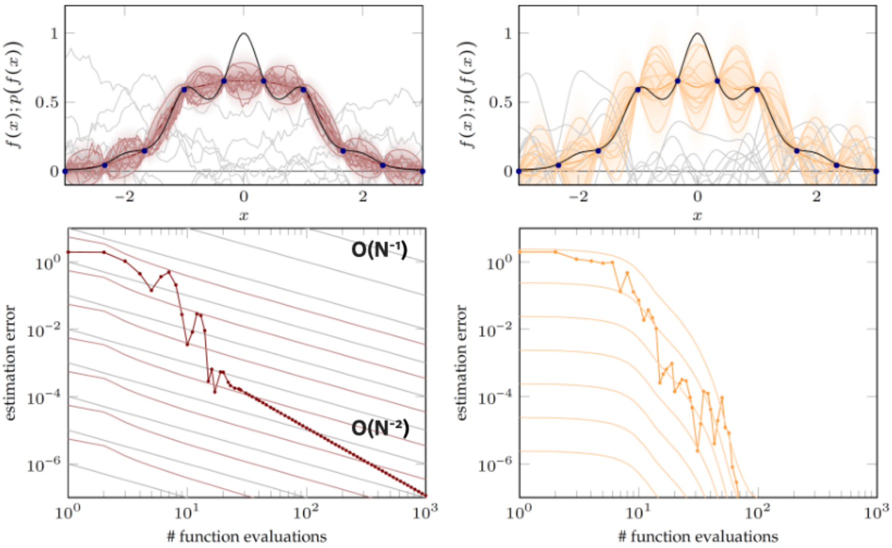
Don't average use quadrature

$$f(x) := \exp\left(-(\sin(3x))^2 - x^2\right)$$



Bayesian Quadrature

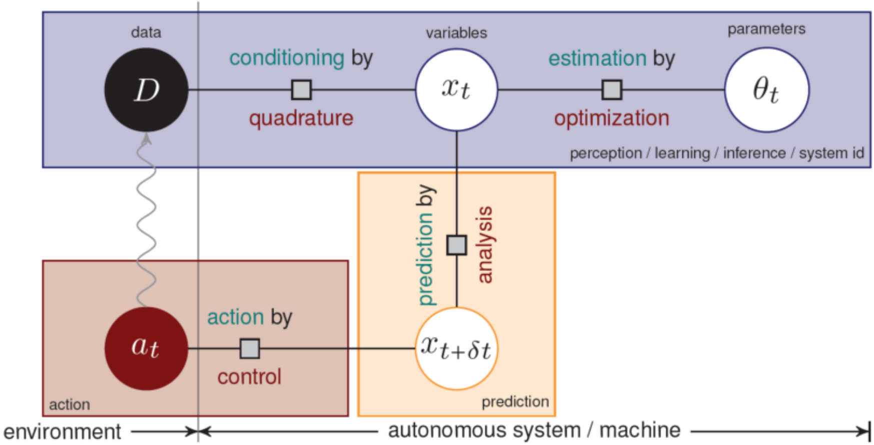
Trapezoid method



Outline

Model

Propagates uncertainty



Model

Converges