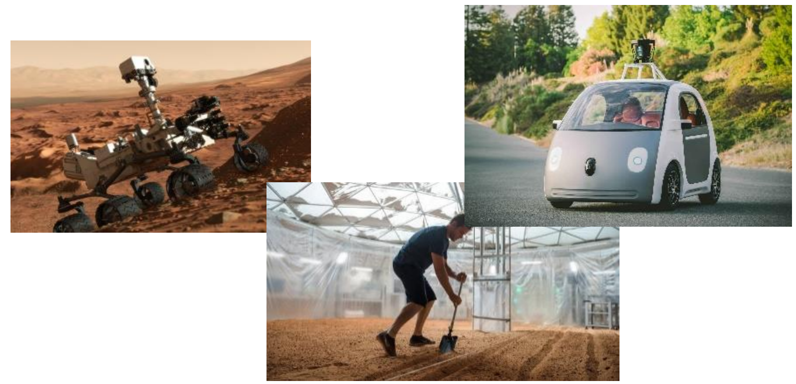
L0. Introduction

Motivation

- Machine Learning
- Artificial Intelligence
- Optimization
- Optimum Control
- Planning
- Markov Decision Process
- Influential Diagram
- Decision Tree
- Dynamic Control
- Game Theory
- Search
- Stochastic Programming
- Reinforcement Learning
- Bandit problem

⋮

Engineering is all about decision makings



What are the differences in these decision-making strategies

What are the common aspects in these decision-making strategies?

Data-driven decision making and control in engineering domain

Sensors (Hardware)
- Embedded computing
- Sensor-Actuator Network
- Wireless communication

Data analytics
- Feature extraction
- Machine learning
- Data mining

Decision making
- Mathematical programming
- Stochastic control
- Sequential decision making

Feedback

Target system

$x \rightarrow f(x; \theta_j) \rightarrow y$

input

output

θ_j

Environment

Data acquisition

$$\mathcal{D} = \{x^{1:n}, y^{1:n}, \theta^{1:n}\}$$

Learning

$$y = f(x; \theta)$$

Making decision

$$x^* | \theta = \underset{x}{\operatorname{argmax}} f(x; \theta)$$

Motivation

engineers’ creativity depends on the diversity of tools that he or she has

- Diversify tools for decision making
- Understand the usages of your tools
- Sharpen your tools
- Organize your tools

Core Ideas

What type of decision making framework will be used?

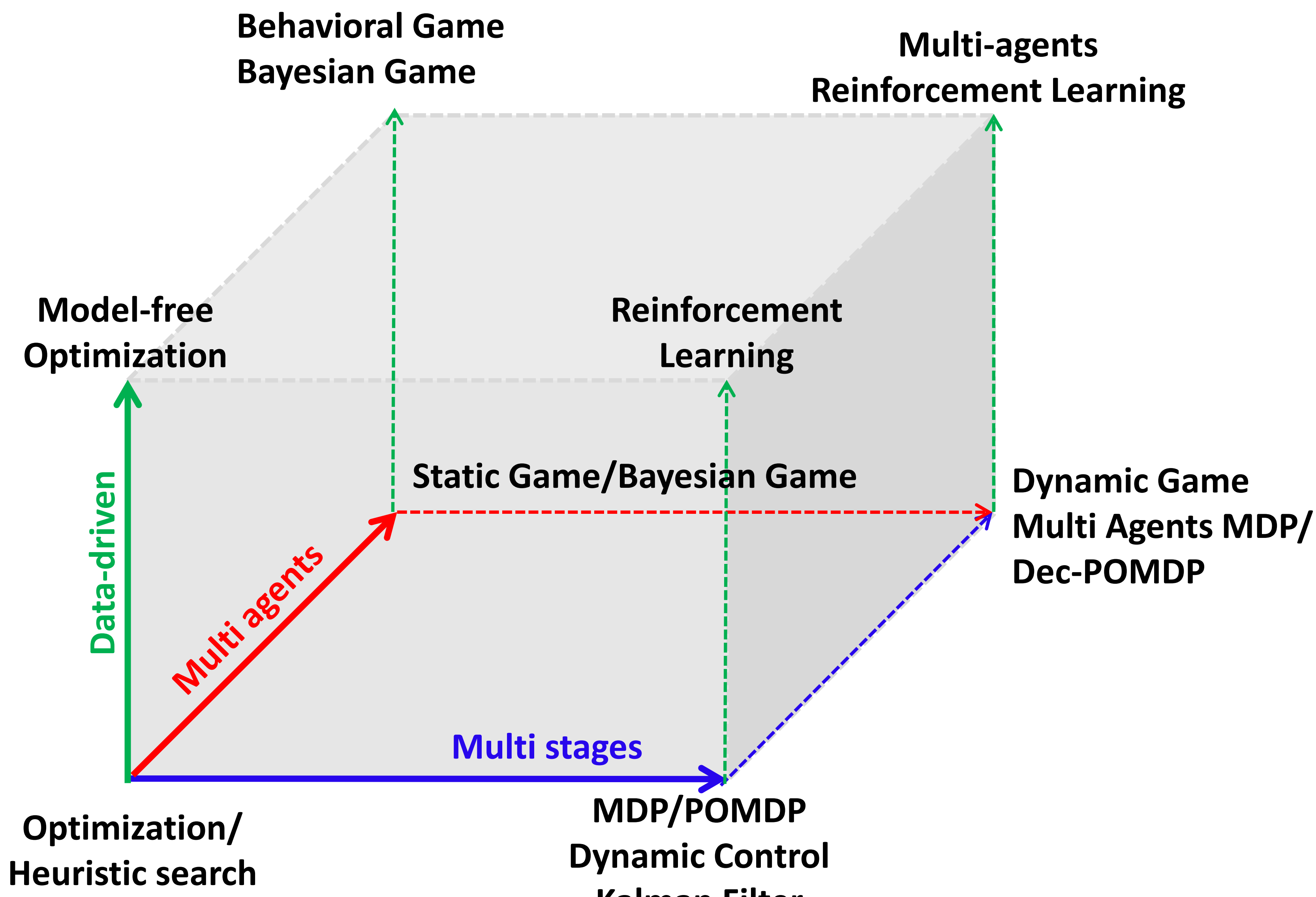
- Single stage or multi stages
- Single decision maker or many decision makers
- Model based or model-free

"Decision makings under uncertainties"

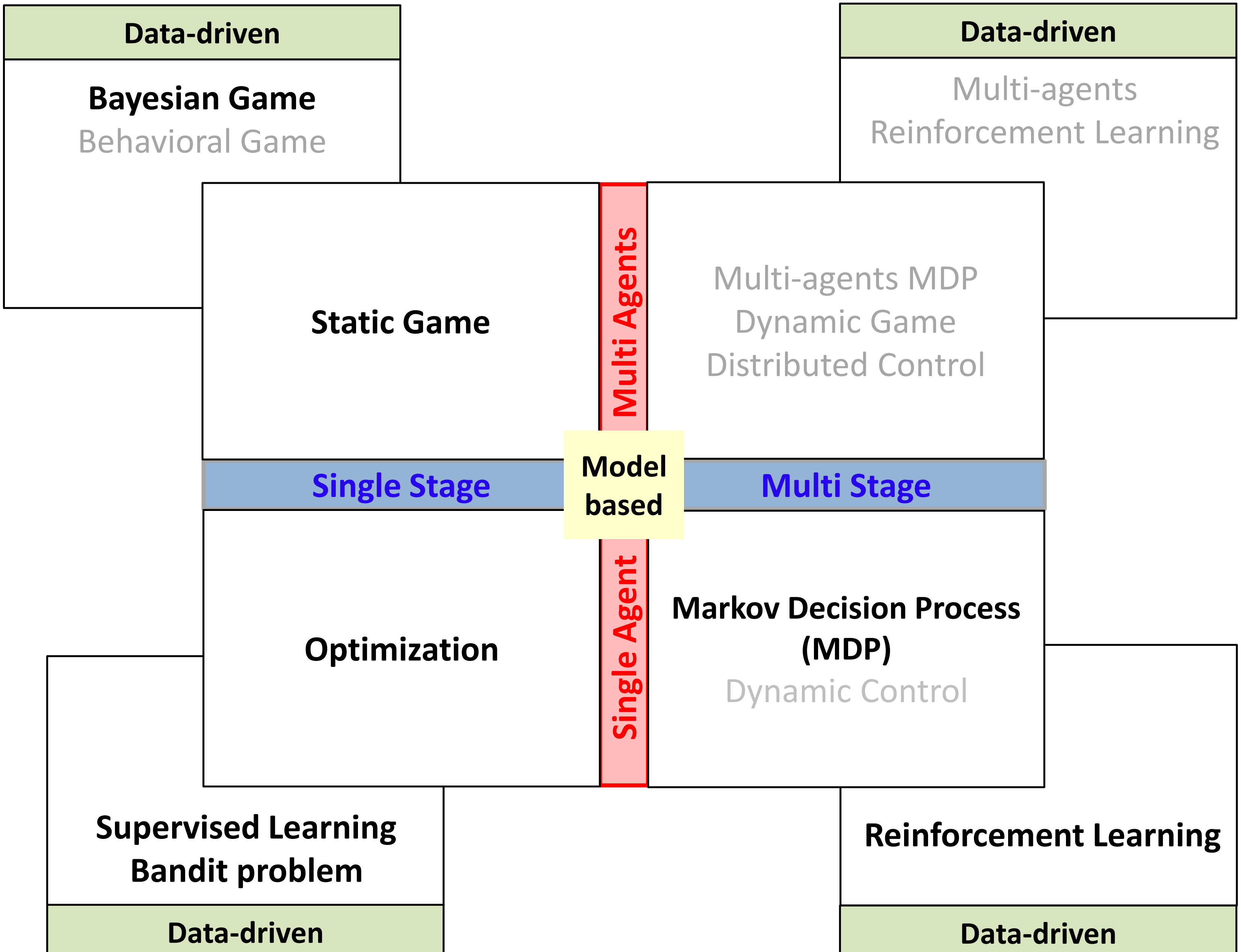
How to model uncertainties?

- **Epistemic Uncertainty** (systemic uncertainty) :
  Uncertainty arising through lack of knowledge
  - Model uncertainty
  - State uncertainty
- **Aleatoric uncertainty** (statistical uncertainty):
  Uncertainty arising through an underlying stochastic system

Frameworks of decision makings



Scope of this course

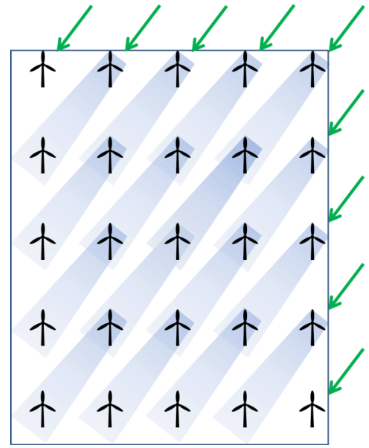


Problem solving

Real-world task

Modeling

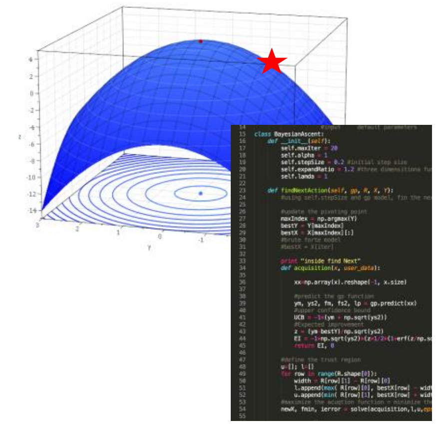
Formal task (model)



$$\underset{\boldsymbol{x}}{\text{maximize}} \sum_{i=1}^{N} P_i(\boldsymbol{x}; \theta, U)$$

Solving

Algorithm (program)



Validation

Are we building the right model?

Verification

Does the algorithm capture all the essential aspects of the model?

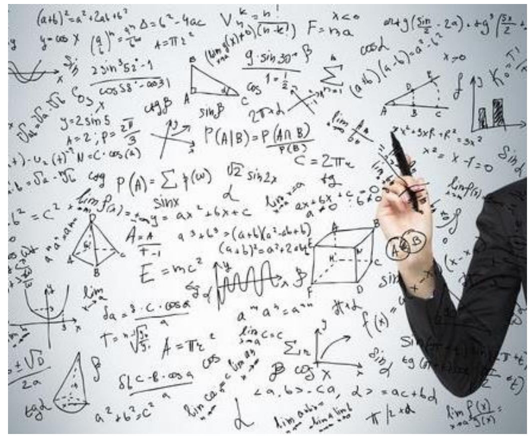
Is this solution is good for the target system?

Data can help model more realistically and derive more accurate solution!

Key elements of this course

Data analytics

- Bayesian Statistics
- Machine learning
- Bayesian Network



Modeling

- Optimization
- Markov Decision Process
- Game Theory

Decision Making

- Mathematical Programming
- Dynamic Programming
- Reinforcement Learning

Course Objectives

Upon successful completion of the course, you are able to

- *understand* various mathematical models describing decision making problems.
- *formulate* real-world decision making problems in a mathematical form.
- *implement* key algorithms and approaches to solving various decision making problems.
- *Interpret* the results of decision-making problems.

Course schedule

<u>1. Bayesian Modeling and Inference (3 weeks)</u>
- Probability distributions
- Prior, Likelihood, and Posterior
- Conjugate models
- Hierarchical Models

<u>2. Single-agent, single-stage decision-making (3 weeks)</u>
- Bayesian regression
- Bayesian classification
- Bayesian Network
- Influential Diagram

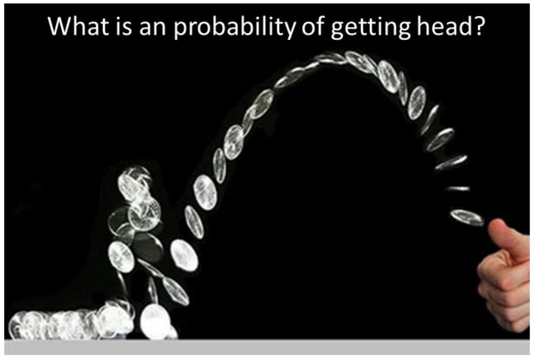
<u>3. Multi-stages decision-making (6 weeks)</u>
- Bandit problem
- Bayesian Optimization
- Markov Decision Process (MDP)
- Dynamic Programming
- Reinforcement Learning
  - Monte Carlo Methods
  - Temporal Difference Methods

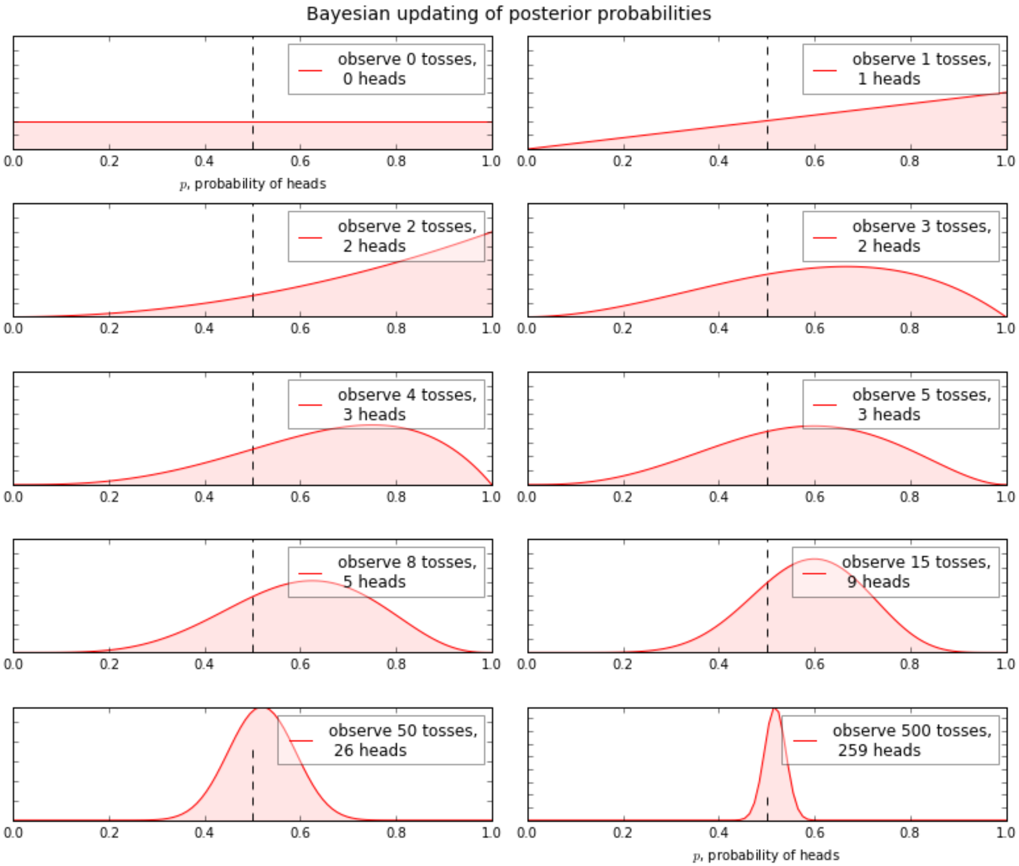
<u>4. Multi-agents decision-making (3 weeks)</u>
- Basics of game theory
- Bayesian Game (with uncertainty about other agents)

Main subject

1. Bayesian Modeling and Inference (3 weeks)

- Probability distributions
- **Prior, Likelihood, and Posterior**
- Conjugate models
- Hierarchical Models





$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

$$= \frac{p(y|\theta)p(\theta)}{\int_\theta p(y|\theta)p(\theta)d\theta}$$

$$\propto p(y|\theta)p(\theta)$$

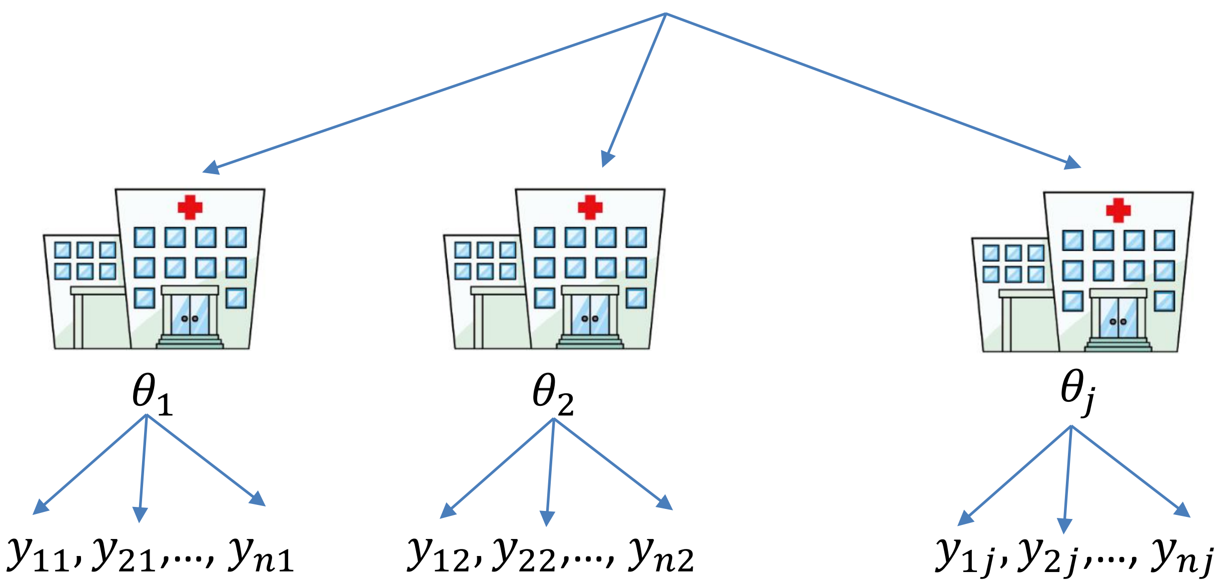
Posterior ∝ Likelihood X Prior

Main subject

<u>1. Bayesian Modeling and Inference (3 weeks)</u>

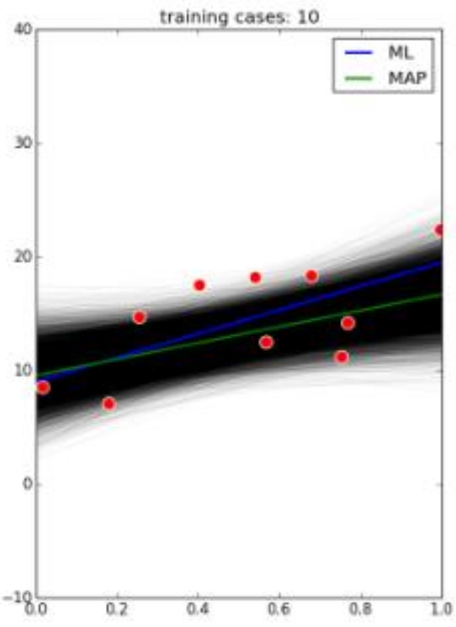
- Probability distributions
- Prior, Likelihood, and Posterior
- Conjugate models
- **Hierarchical Models**

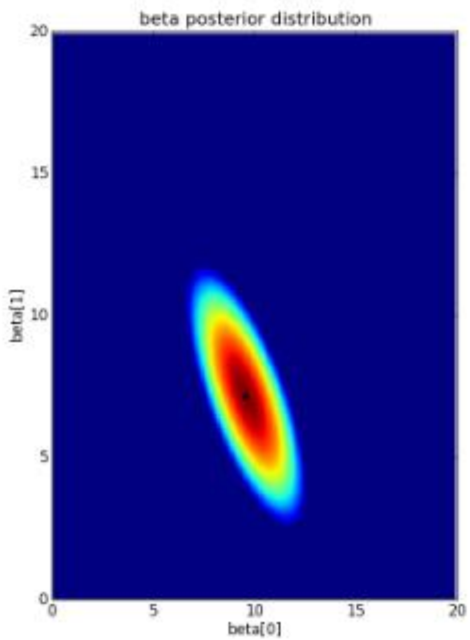
Survival probability of cardiac patients $\theta_j \sim$ population distribution

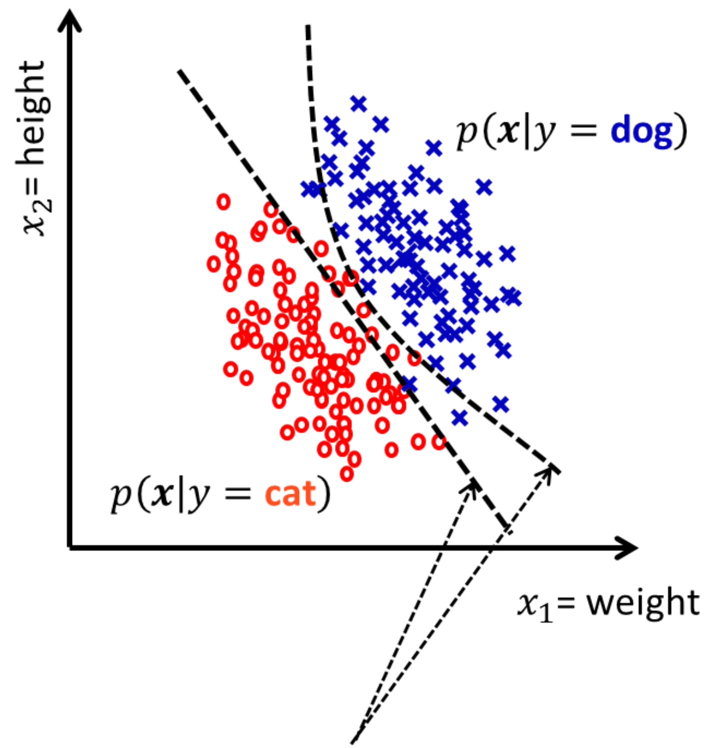


<u>2. Single-agent, single-stage decision-making (3 weeks)</u>

- **Bayesian regression**
- **Bayesian classification**
- Bayesian Network
- Influential Diagram







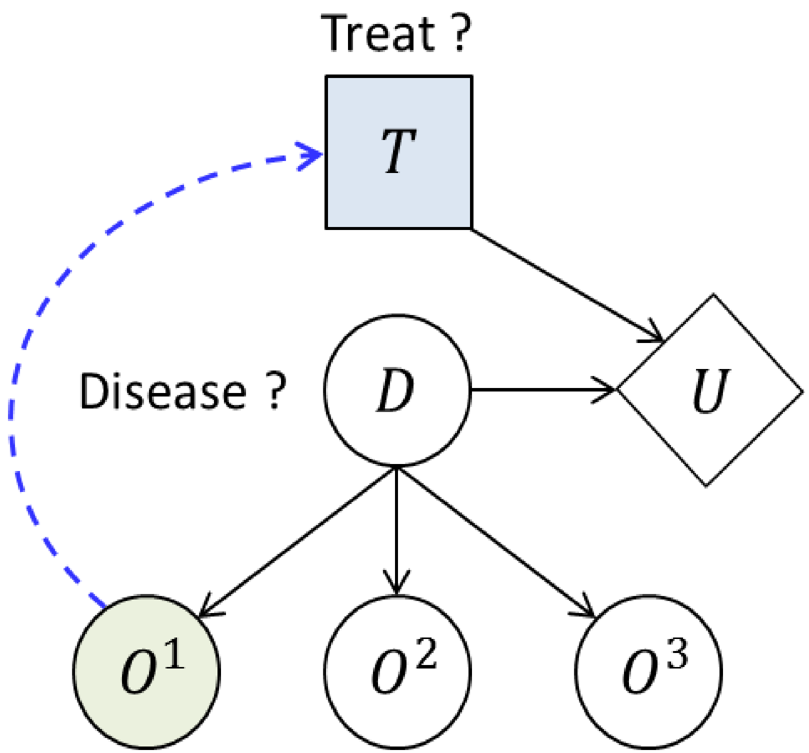
Decision boundary

$P(y = \text{dog}|\boldsymbol{x}) = P(y = \text{cat}|\boldsymbol{x})$

Main subject

<u>2. Single-agent, single-stage decision-making (3 weeks)</u>

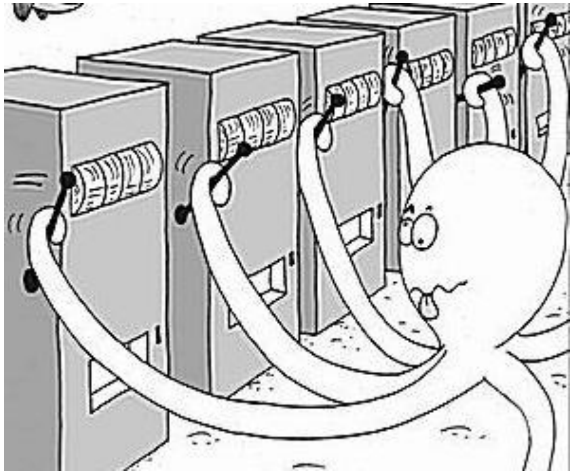
- Bayesian regression
- Bayesian classification
- **Bayesian Network**
- **Influential Diagram**



Main subject

3. Multi-stages decision-making (6 weeks)

- **Bandit problem**
- Bayesian Optimization
- Markov Decision Process (MDP)
- Dynamic Programming
- Reinforcement Learning
  - Monte Carlo Methods
  - Temporal Difference Methods



Facing with N slot machines with different payoff distribution,

→ **Devise a strategy to maximize payoff**

→ **Find the best bandit as quickly as possible**

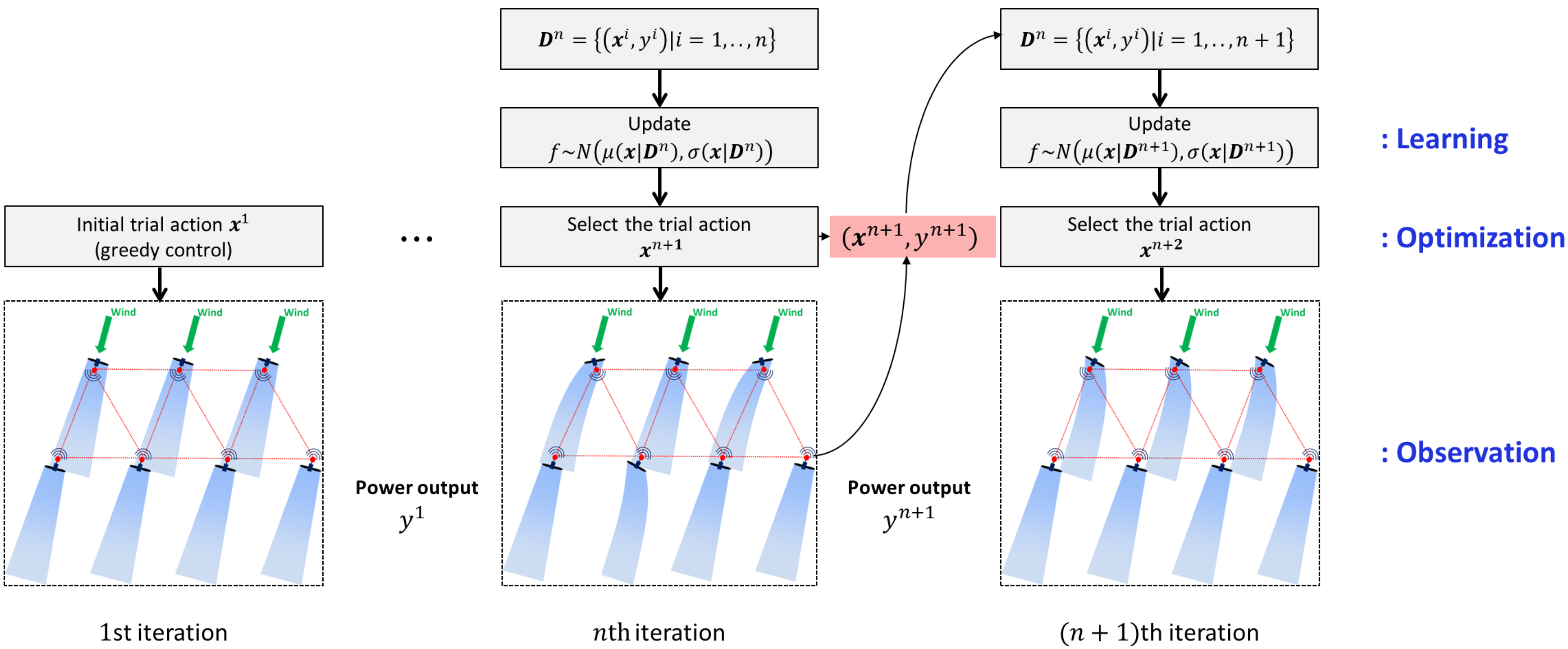
Acquiring new information ***(exploration)***

trade-off

capitalizing on the information available so far ***(exploitation)***

3. Multi-stages decision-making (6 weeks)

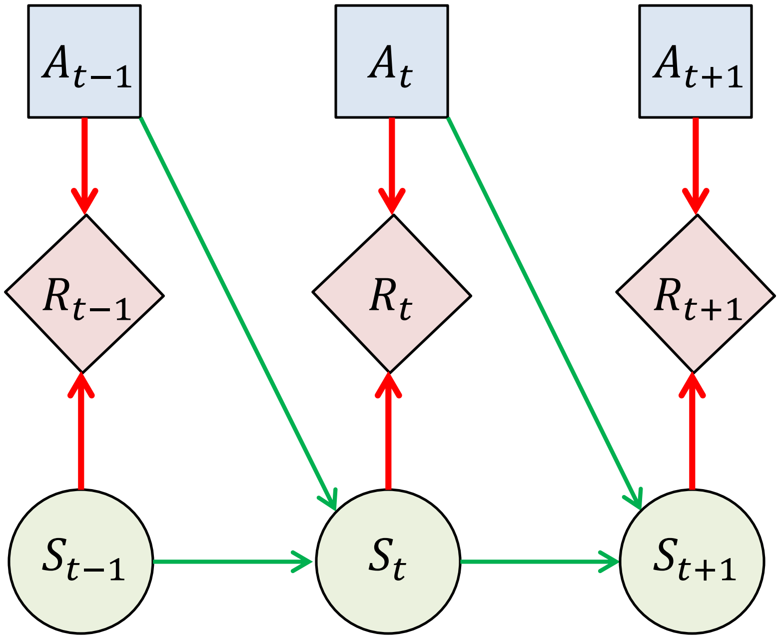
- Bandit problem
- **Bayesian Optimization**
- Markov Decision Process (MDP)
- Dynamic Programming
- Reinforcement Learning
  - Monte Carlo Methods
  - Temporal Difference Methods



Main subject

<u>3. Multi-stages decision-making (6 weeks)</u>

- Bandit problem
- Bayesian Optimization
- **Markov Decision Process (MDP)**
- Dynamic Programming
- Reinforcement Learning
  - Monte Carlo Methods
  - Temporal Difference Methods



A_t : action taken at time t

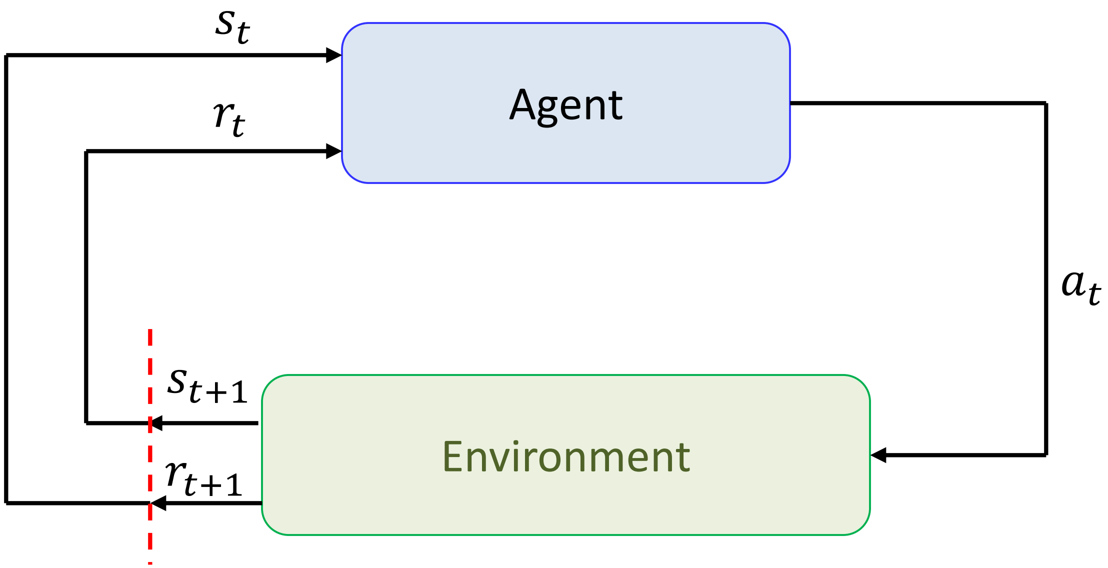
R_t : reward received at time t

S_t : state at time t

Main subject

<u>3. Multi-stages decision-making (6 weeks)</u>

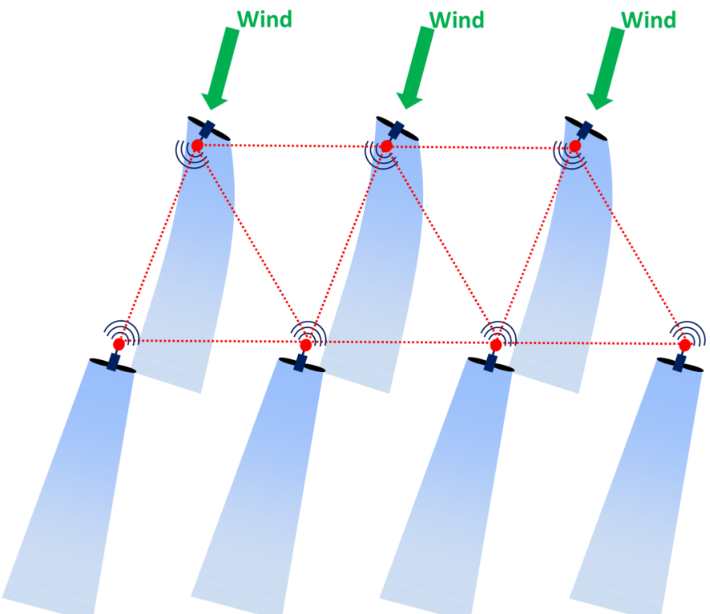
- Bandit problem
- Bayesian Optimization
- Markov Decision Process (MDP)
- Dynamic Programming
- **Reinforcement Learning**
  - Monte Carlo Methods
  - Temporal Difference Methods



Main subject

4. Multi-agents decision-making (3 weeks)

- **Basics of game theory**
- Bayesian Game (with uncertainty about other agents)



WT 1 \ WT 2	greedy (α_1^G, o_1^G)	cooper (α_2^C, o_2^C)
greedy (α_1^G, o_1^G)	$(70,70)$ $\sum = 140$	$(70,50)$ $\sum = 120$
cooper (α_1^C, o_1^C)	$(50,70)$ $\sum = 120$	$(60,90)$ $\sum = 150$

Computation tool

the Jupyter Notebook is a web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more.

http://jupyter.org/

Resource for learning:

http://ipython-books.github.io/cookbook

http://ipython.rossant.net/cookbook/

Computation tool

https://www.continuum.io/downloads

Download for Windows | Download for OSX | Download for Linux

Anaconda 4.1.1

For Windows

Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.

Changelog

1. Download the installer
2. Optional: Verify data integrity with MD5 or SHA-256
   More info
3. Double-click the **.exe** file to install Anaconda and follow the instructions on the screen

Behind a firewall? Use these zipped Windows installers

Python 3.5 version

64-BIT INSTALLER (351M)

32-BIT INSTALLER (292M)

Python 2.7 version

64-BIT INSTALLER (340M)

32-BIT INSTALLER (285M)

For older versions of Anaconda installers, see the Anaconda installer archive

For long-term support of the packages found in the Anaconda archives, please contact us.

http://jupyter.readthedocs.io/en/latest/install.html

Installing Jupyter using Anaconda and conda

For new users, we **highly recommend** installing Anaconda. Anaconda conveniently installs Python, the Jupyter Notebook, and other commonly used packages for scientific computing and data science.

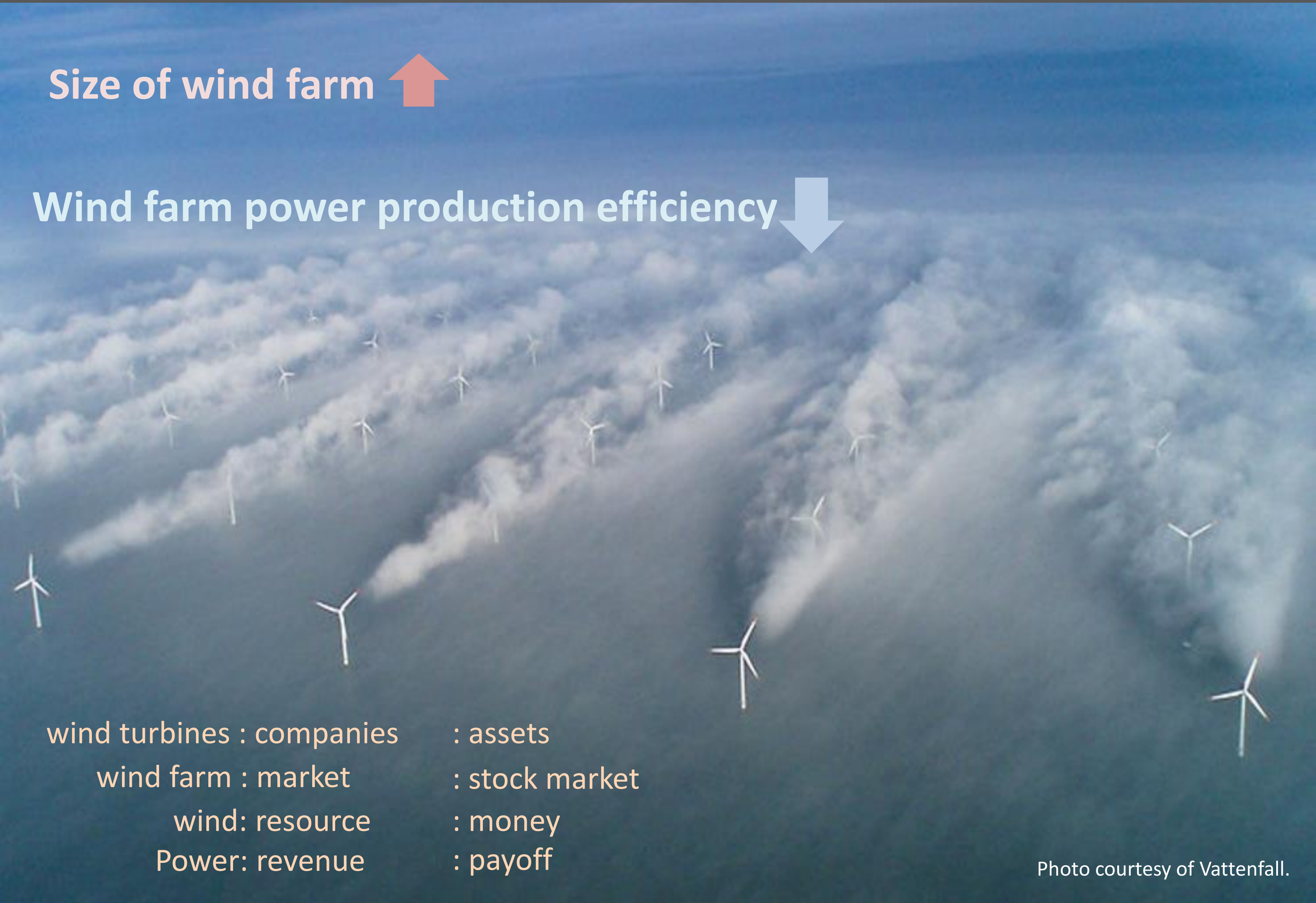
Use the following installation steps:

1. Download Anaconda. We recommend downloading Anaconda's latest Python 3 version (currently Python 3.5).
2. Install the version of Anaconda, which you downloaded.
3. Congratulations, you have installed Jupyter Notebook. To run the notebook:

```
jupyter notebook
```

See Running the Notebook for more details.

Example: Data-driven wind farm power maximization



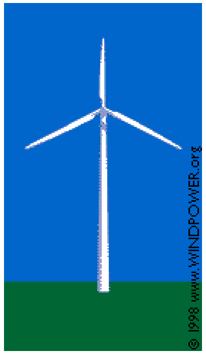
Size of wind farm ↑

Wind farm power production efficiency ↓

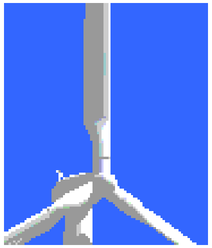
wind turbines : companies	: assets
wind farm : market	: stock market
wind: resource	: money
Power: revenue	: payoff

Photo courtesy of Vattenfall.

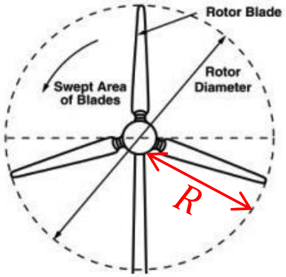
Example: Data-driven wind farm power maximization



<www.windpowr.org>



<www.windpowr.org>

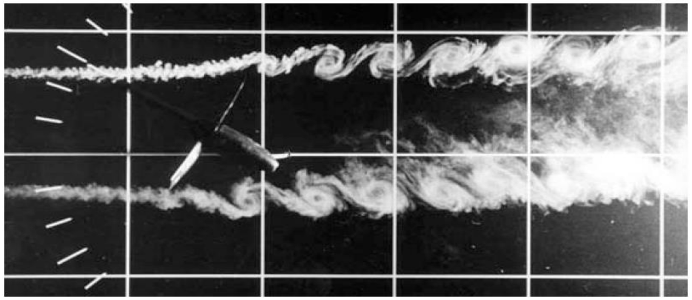


Yaw angle

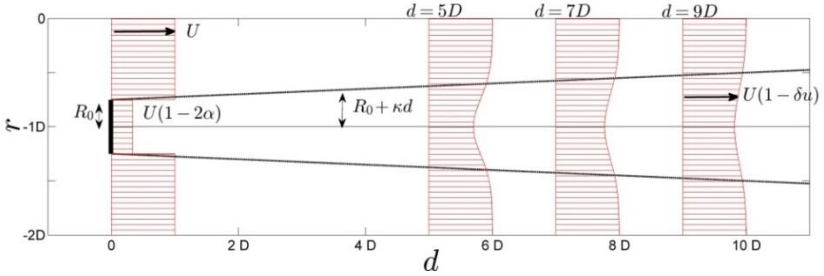
Blade pitch angle

Tip speed ratio

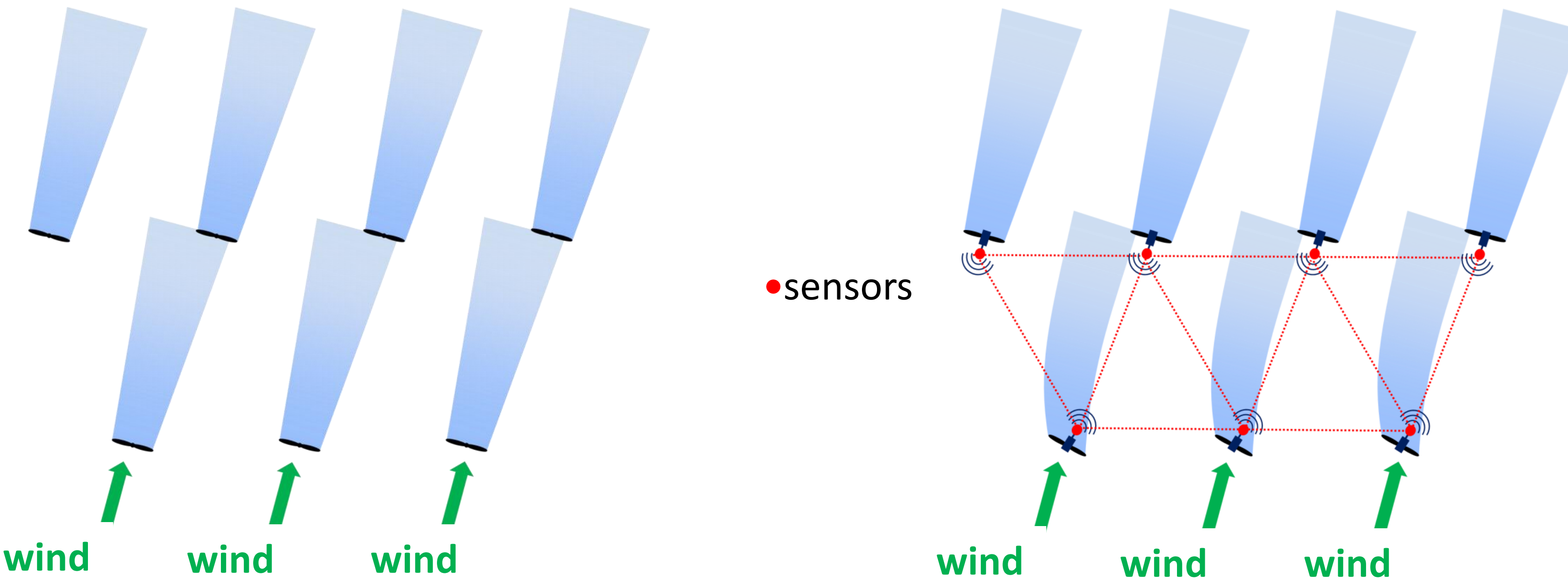
Control Inputs



Flow visualization of wake deflection. Conducted at the Royal institute of Technology at 1987, Dahlberg (2003)



Example: Data-driven wind farm power maximization



Greedy strategy:

$$\sum_{i=1}^{N} \underset{x_i}{\text{maximize}}\, P_i(\boldsymbol{x}; U, \theta)$$

$$\leq$$

Cooperative strategy:

$$\underset{\boldsymbol{x}}{\text{maximize}} \sum_{i=1}^{N} P_i(\boldsymbol{x}; U, \theta)$$

$P_i(\boldsymbol{x}; U, \theta)$:Power of wind turbine i

$\boldsymbol{x} = (x_1, \dots, x_N)$: Control inputs for N wind turbines

Example: Data-driven wind farm power maximization

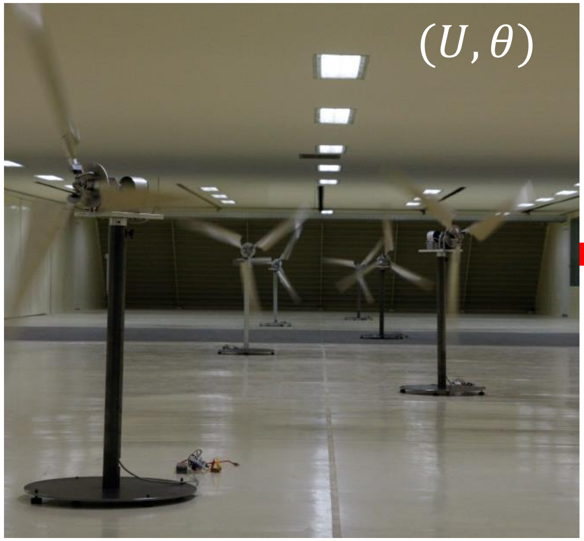
(U, θ) : **Fixed wind condition (Fixed context)**

Input (control actions)

$$\boldsymbol{x}^1 = (x_1^1, \ldots, x_N^1)$$

$$\boldsymbol{x}^2 = (x_1^2, \ldots, x_N^2)$$

$\vdots$



Output (power measurements)

$$y^1 = \sum_{i=1}^{N} P_i(\boldsymbol{x}^1; \theta, U) + \epsilon^1$$

$$y^2 = \sum_{i=1}^{N} P_i(\boldsymbol{x}^2; \theta, U) + \epsilon^2$$

$\vdots$

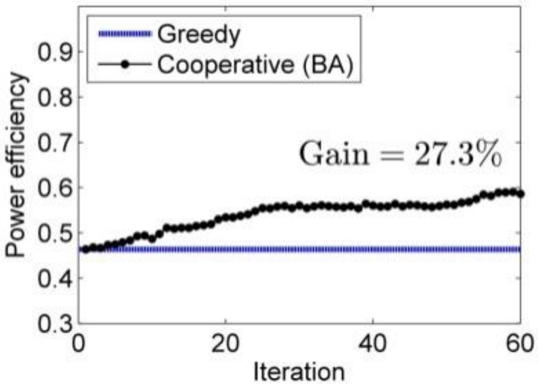
(y = total wind farm power)

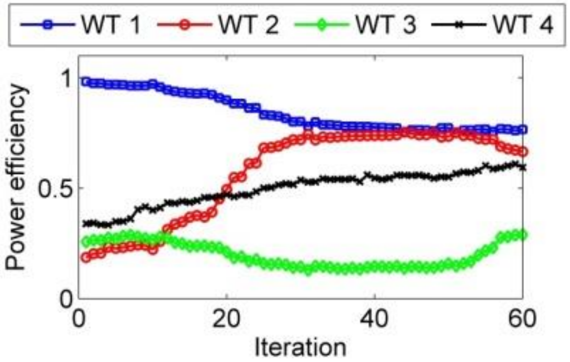
$$\underset{\boldsymbol{x}}{\text{maximize}} \;\; f(\boldsymbol{x}; U, \theta) = \sum_{i=1}^{N} P_i(\boldsymbol{x}; \theta, U)$$

$$\text{subject to} \;\; \boldsymbol{x}^l \le \boldsymbol{x} \le \boldsymbol{x}^u$$

Without knowing $f(\boldsymbol{x}; U, \theta)$

Example: Data-driven wind farm power maximization



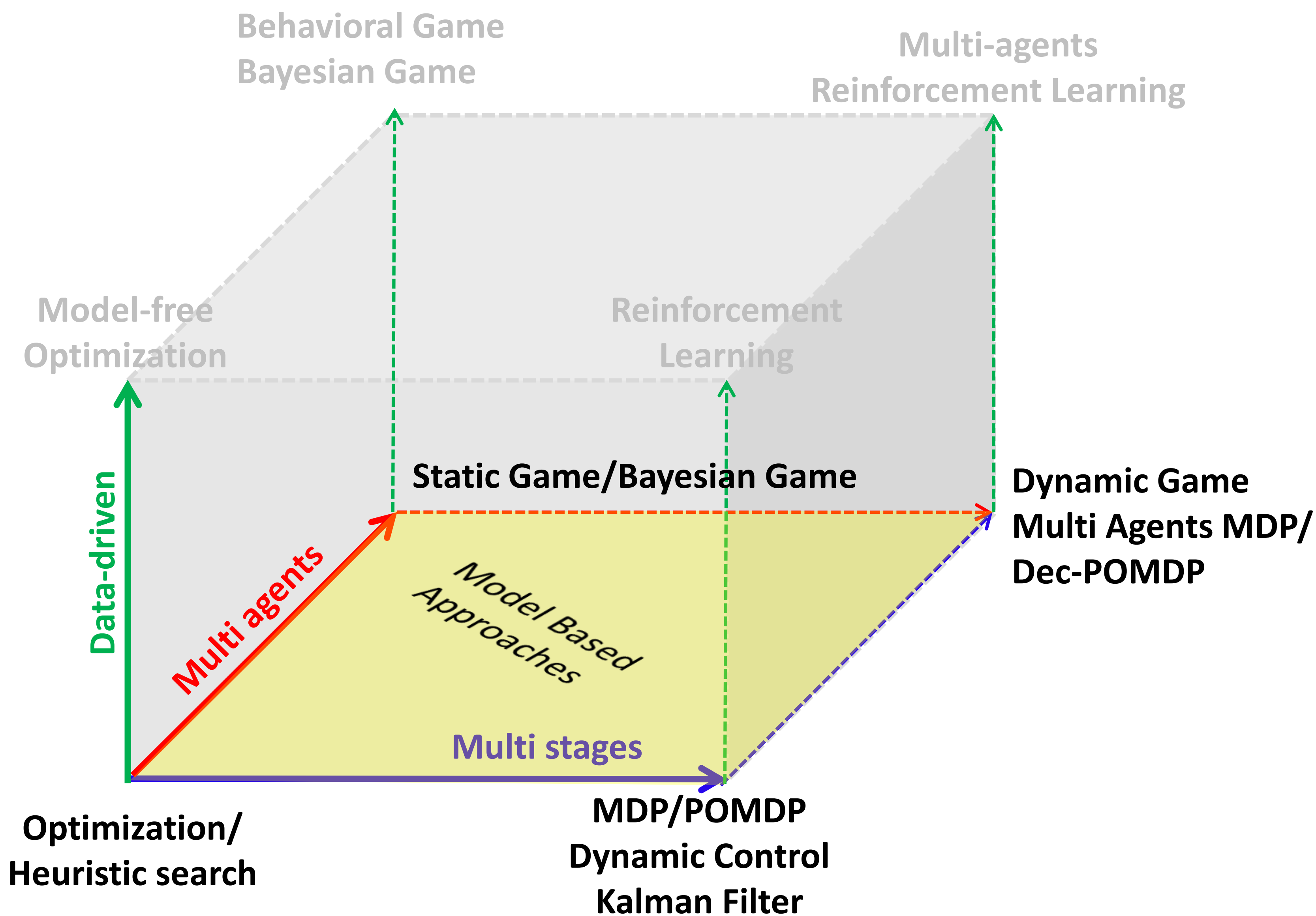


Greedy control (initial)

Cooperative control (after convergence)

Supplements

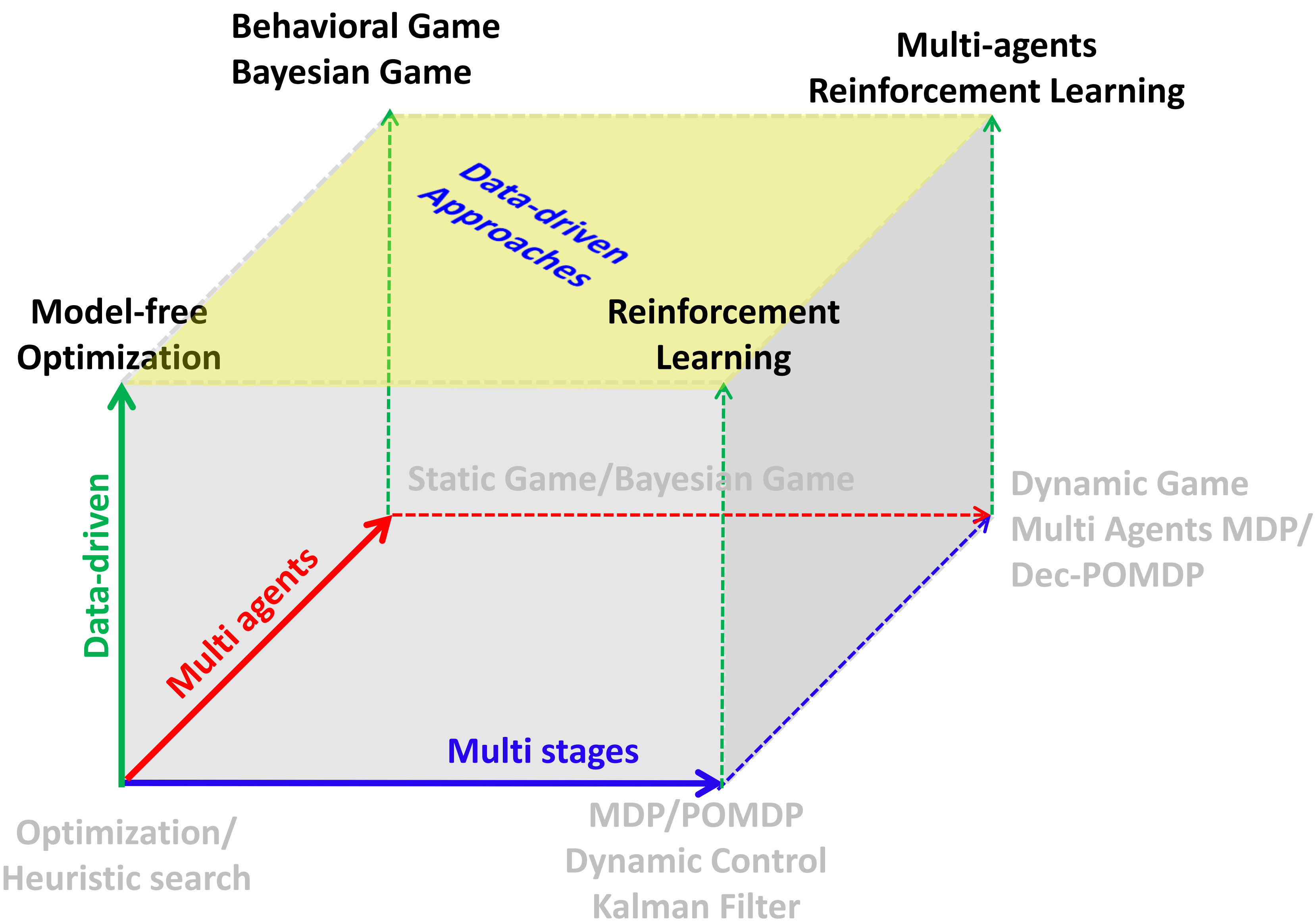
Frameworks of decision makings



The scope of this course

	Single stage	**Multi stages**
Single agent	Optimization	Markov Decision Process (MDP)
Multi agents	Static Game	Dynamic Game/ Multi agents MDP

Frameworks of decision makings



The scope of this course

	Single stage	**Multi stages**
Single agent	Supervised learning Bayesian network	Reinforcement Learning
	Bandit problem	
Multi agents	Bayesian Game	Multi-agent Reinforcement Learning