From Science to Data Science

Stephane SENECAL @ Orange Labs
stephane.senecal@orange.com
CRI Data Science Club, 31/03/2017

"From Science to Data Science" Overview

- Academic Background and Activities (17 slides)
- Professional Background and Activities (33 slides + 2 videos!)
- (Data Science…) Projects (6 slides)
- Working @ Orange Labs (3 slides)
- Machine Learning/Data Science… (2 slides)
- Take Away Messages (1 slide)
- Bonus: Appendix (44 slides)

Academic Background (1/3): Masters in Mathematics & Signal Processing

- MS in mathematics:
  - obtained in 1998 @ Joseph Fourier University, Grenoble, France
  - thesis on holomorphic functions of several complex variables
- MEng/MS in signal processing:
  - obtained in 1999 @ Grenoble INP, France
  - thesis on curvilinear component analysis for model order estimation

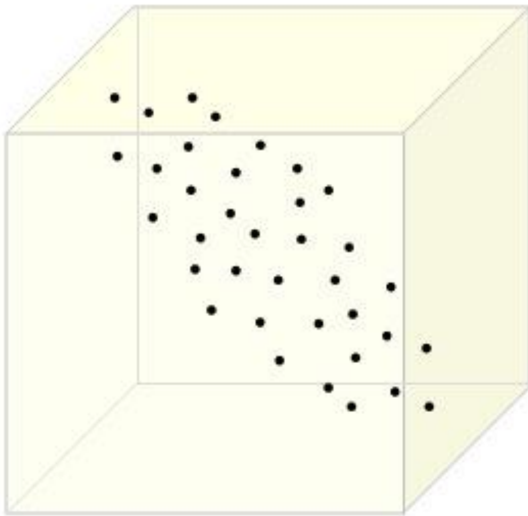
Data Dimension Reduction

Input: dataset composed of N samples each of dimension n ($n >> 1$)

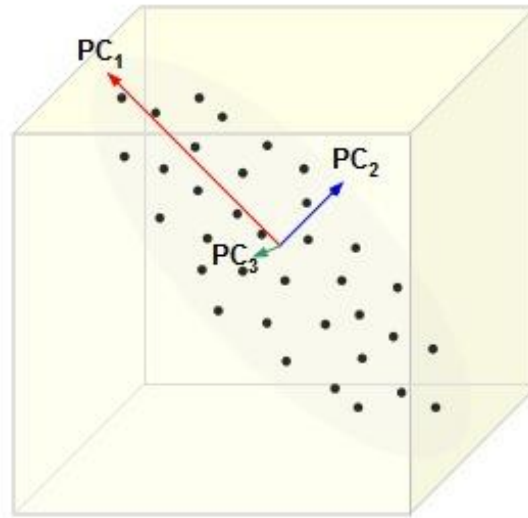
Output: dataset composed of N samples each of dimension p ($p << n$)

Various techniques:
Principal Component Analysis and its extensions,
Kohonen/self-organizing maps,
Multi-Dimensional Scaling, …

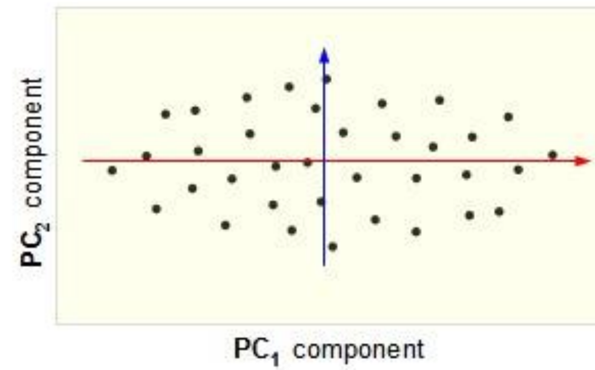
Principal Component Analysis

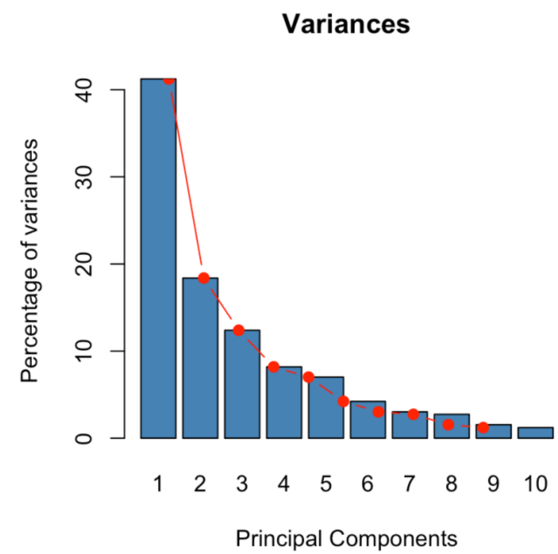


a



b





Computation of an output representation basis
(orthogonal axes)
from the covariance matrix of the input dataset
(eigenvectors obtained from diagonalization)

Academic Background (2/3): PhD in Applied Statistics for Signal Processing

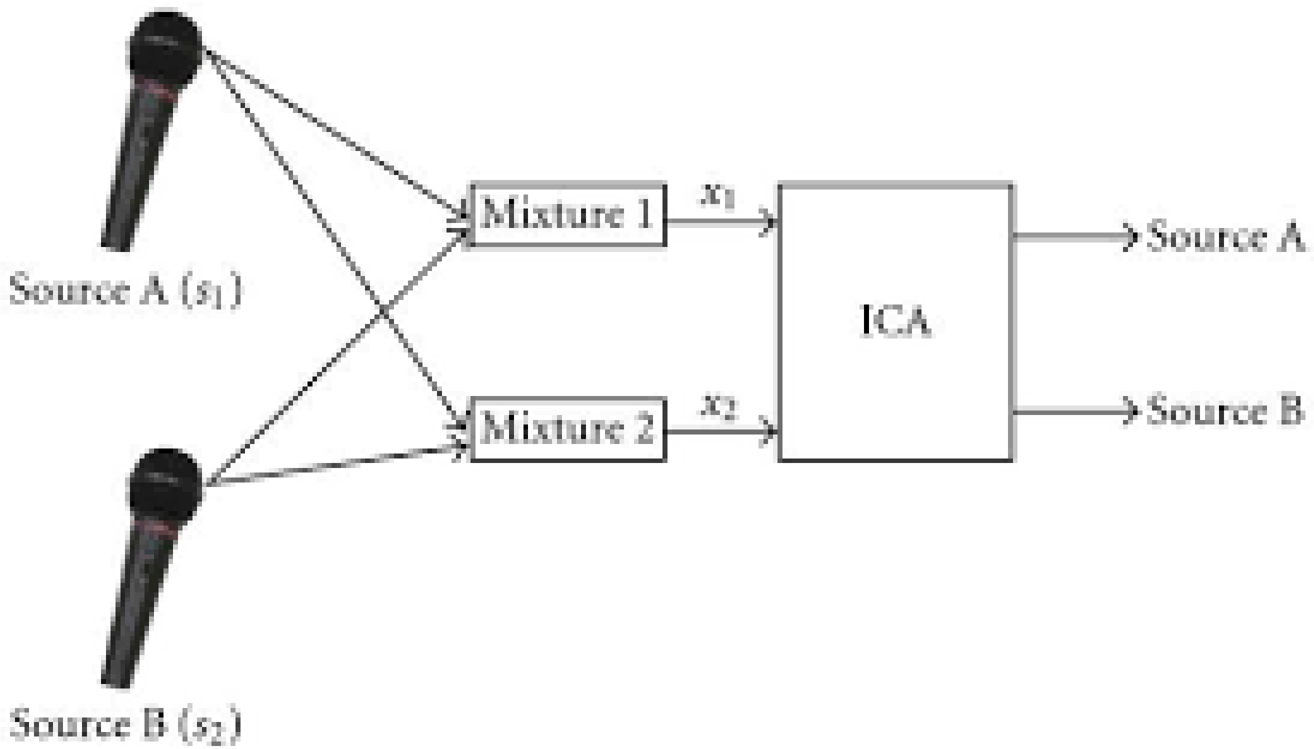
- PhD in statistical signal processsing for telecommunications:

- obtained from Grenoble INP, France
- conducted from 1999 to 2002 @ GIPSA-Lab
- with MESR fellowship support

- → **Statistical simulation methods:**
  - Markov Chain Monte Carlo (MCMC): Hastings-Metropolis, Gibbs sampling, reversible jumps MCMC
  - Sequential Monte Carlo/Particle Filtering

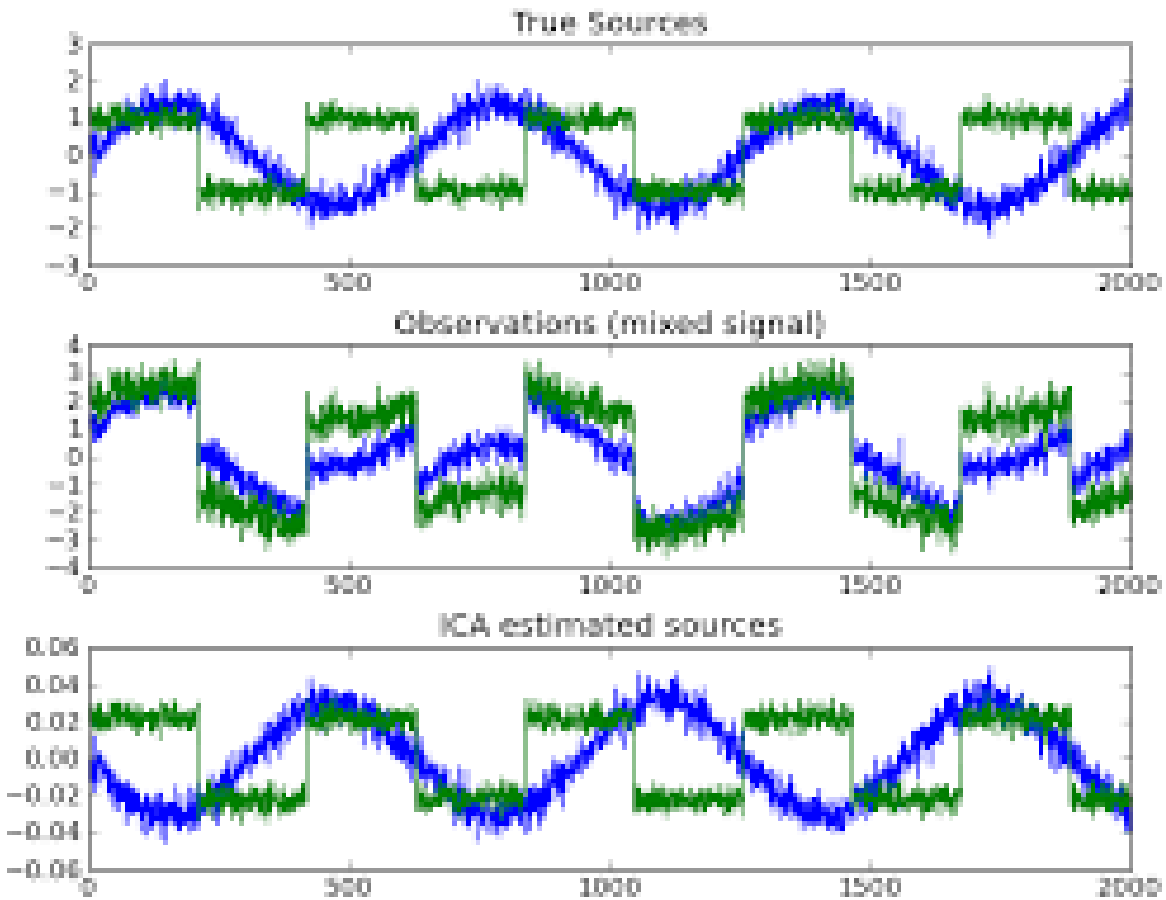
- → Application to **Bayesian estimation problems** for:
  - Independent Component Analysis/Blind Source Separation
  - Equalization of nonlinear system for satellite communications

Independent Component Analysis/Source Separation → The Cocktail Party Problem



ill-posed inverse problem → x = Ms + noise
→ Recover/estimate source signals s from noisy mixtures x

Independent Component Analysis/Source Separation → Example



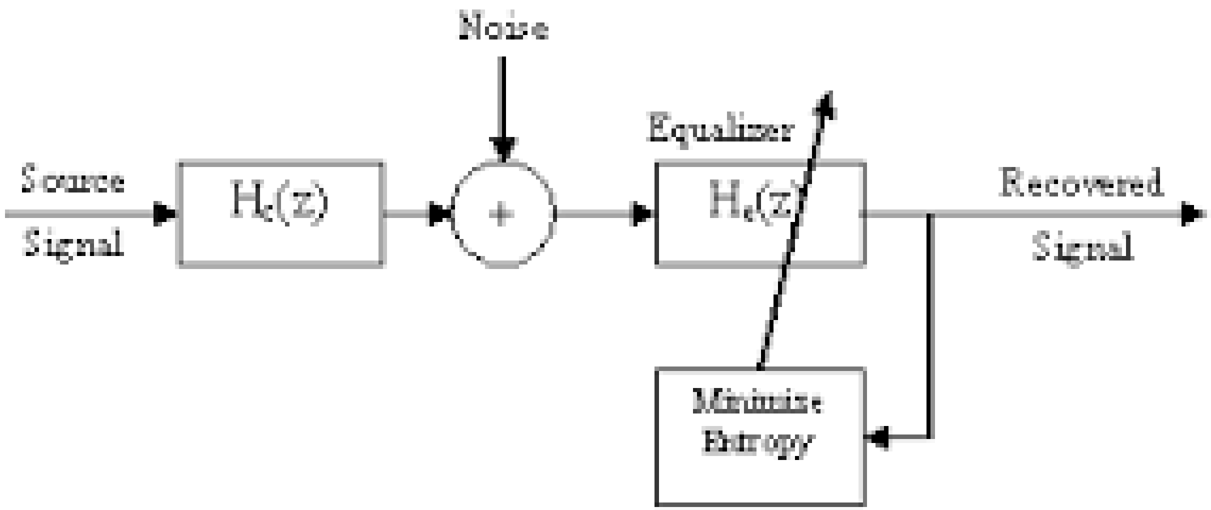
source signals:
s1, s2

observed/measured
mixture signals:
x1, x2

estimated/recovered
source signals

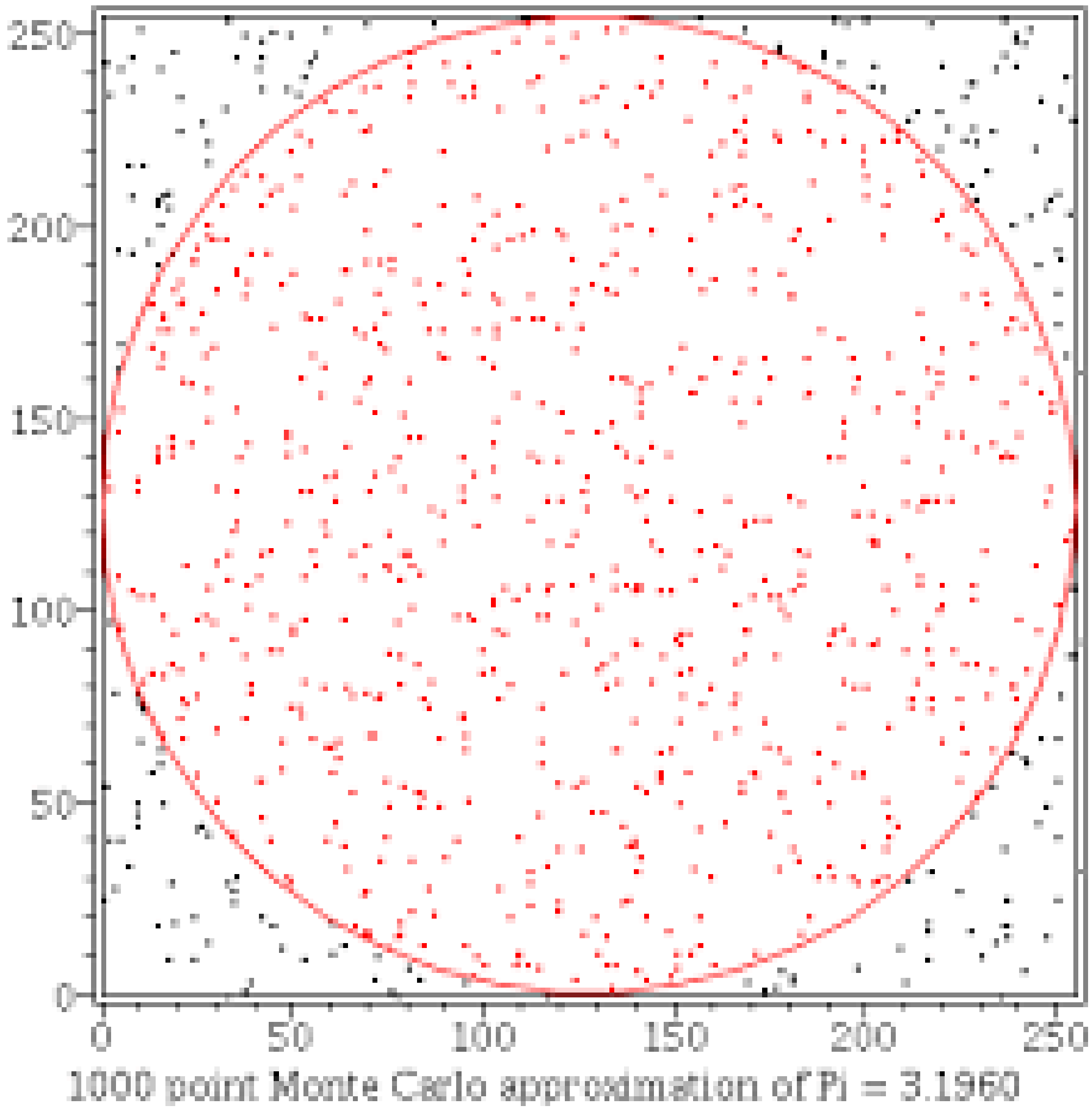
x = Ms + noise

Blind Equalization



Blind Equalization Setup

Monte Carlo Approach (1/2)



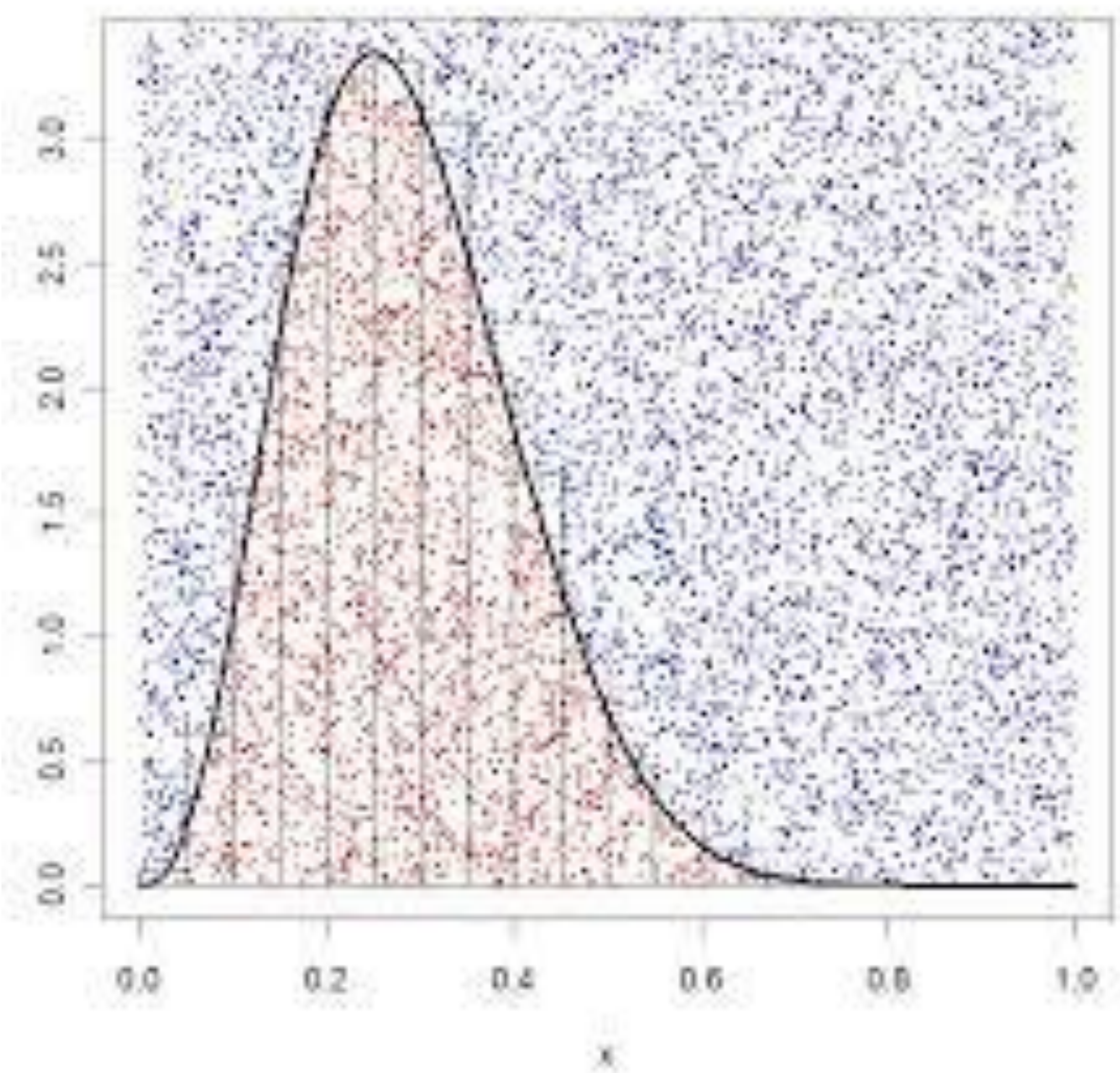
Approximation of π:

disc surface
/
square surface
=
$\pi R^2 / (2R)^2 = \pi/4$

≈

dots inside circle
/
dots inside square

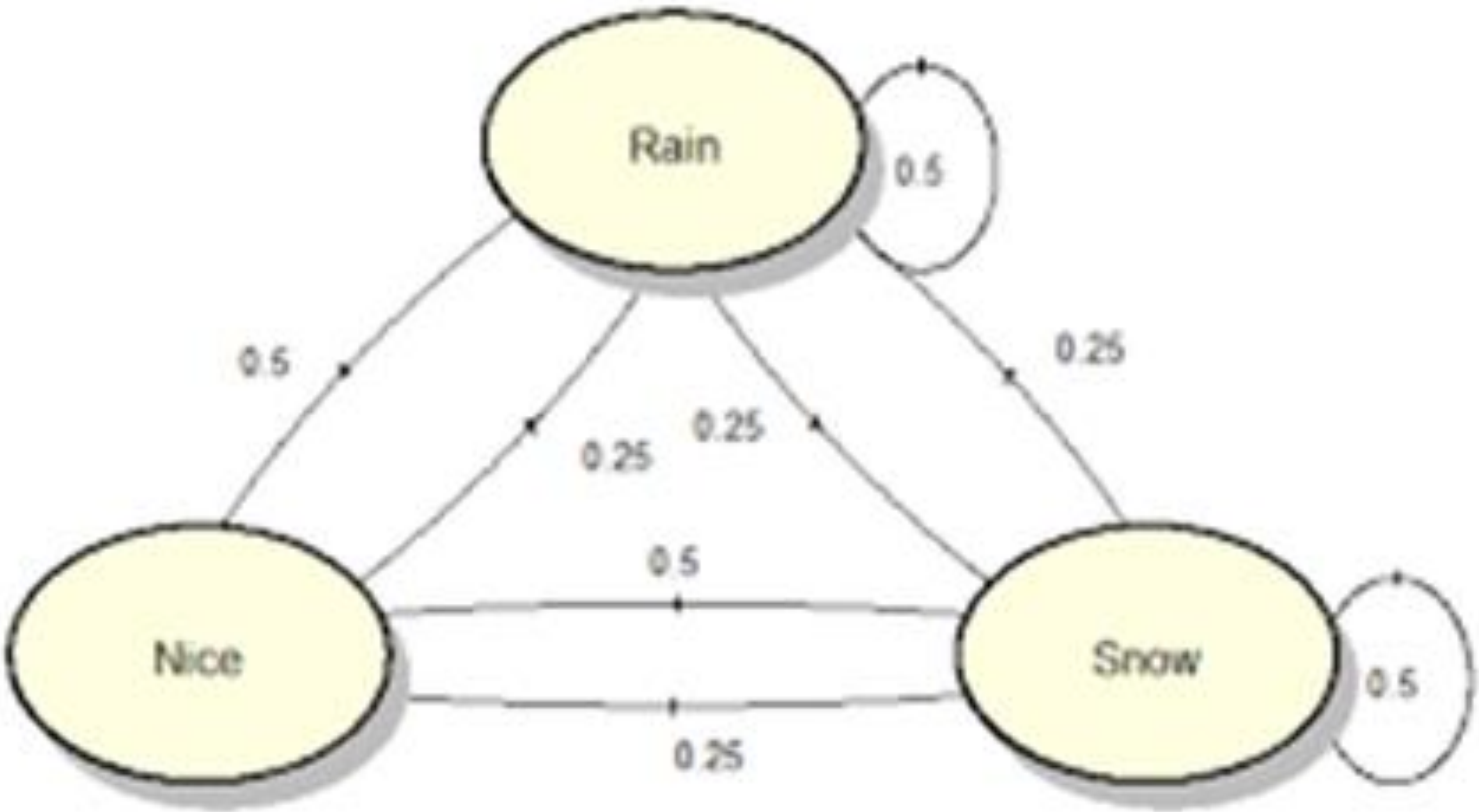
Monte Carlo Approach (2/2)



Numerical approximations of expectations (integrals) of functions under/for non-standard distributions

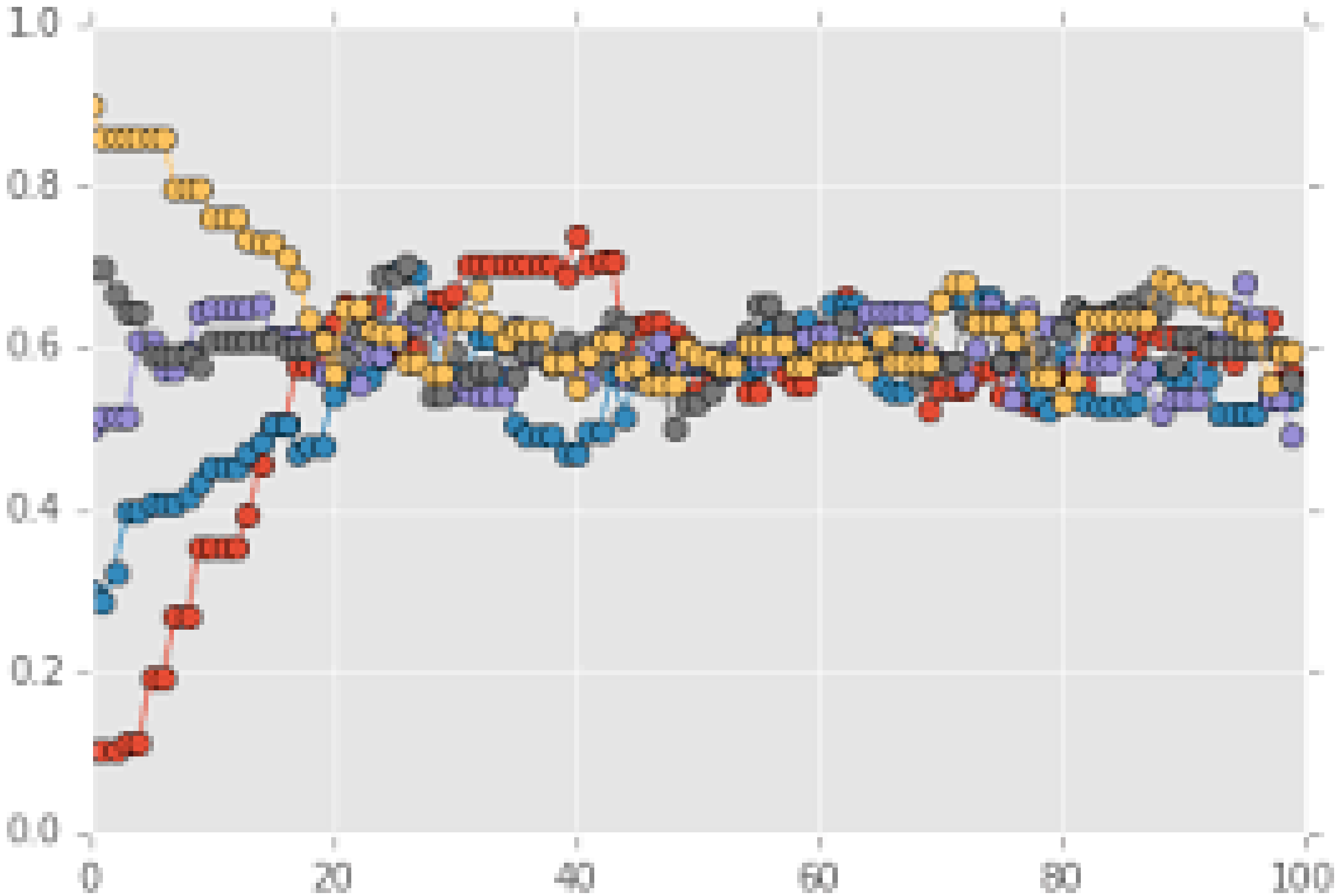
→ How to sample from non-standard distributions?
→Markov chains!

Markov Chain → Example: Weather
→ Discrete-valued Markov Chain (3 states)



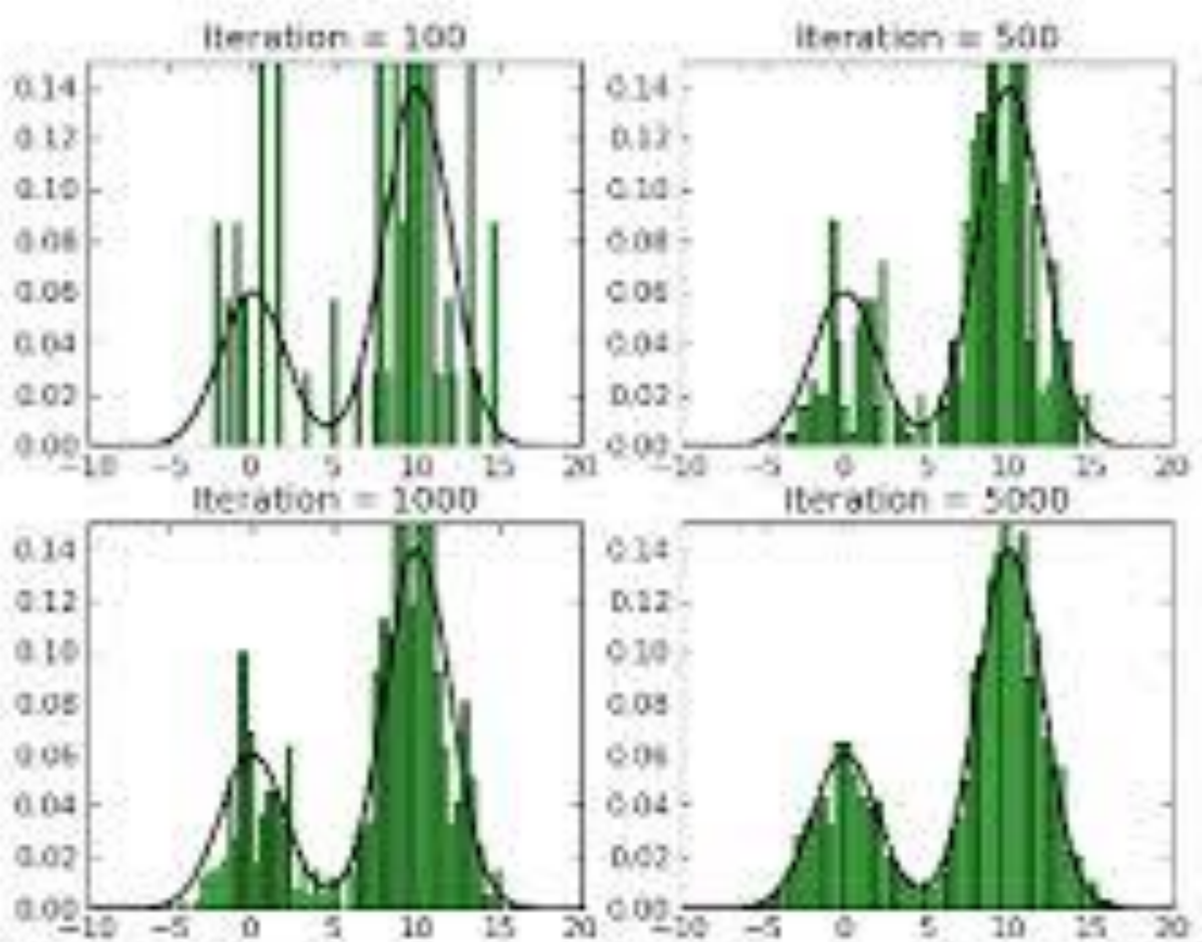
Transition Probabilities Matrix/Operator → Target Distribution = Fixed Point

Markov Chain Monte Carlo (1/2) → Convergence after Burn-in



5 realizations of a Markov chain (≠ colors), 100 iterations

Markov Chain Monte Carlo (2/2)
→ Convergence in Distribution



Bayesian Estimation Problems → Solved by MCMC

- Independent Component Analysis/Source Separation:

  x = Ms + noise   noise drawn from Gaussian(0,sigma^2)

- → Bayesian posterior distribution p(s, M, sigma | x)

  p(s, M, sigma | x) α p(x | s, M, sigma)p(s, M, sigma)

- → Generate samples from p(s, M, sigma | x) via a Markov chain

- → Compute Monte Carlo estimates of (s, M, sigma) from the samples

Academic Background (3/3): Post-Doc in Computational Statistics

- Post-PhD: Post-Doc in Computational Statistics

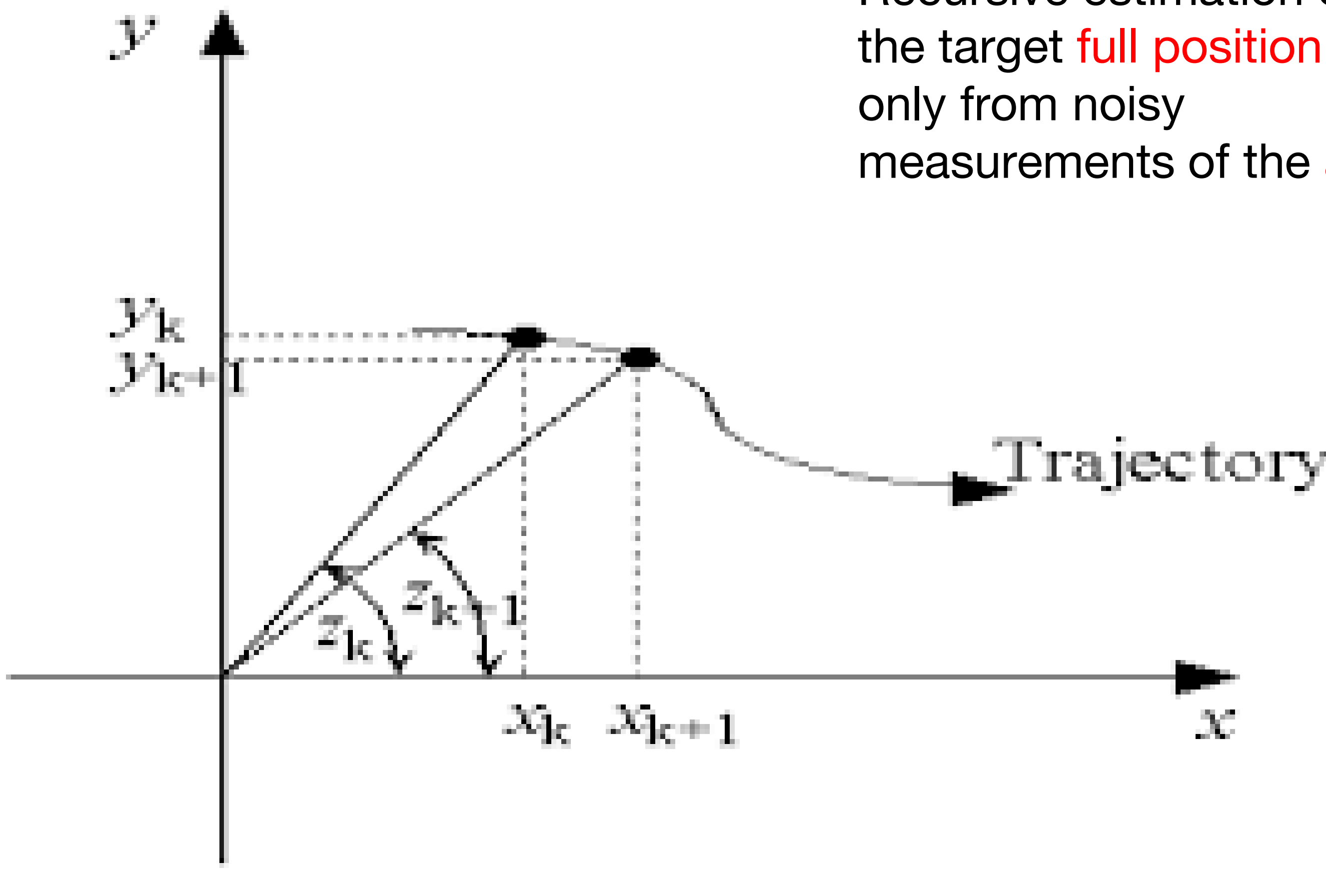
- conducted @ Institute of Statistical Mathematics, Research Organization of Information and Systems (Tokyo, Japan)
- conducted in 2003-2004
- thanks to a JSPS fellowship support

- **Design** of statistical **simulation** algorithms/methods/**techniques**:

  - Block/fixed-lag sampling strategies for Sequential Monte Carlo methods, applications in:
    - optimal filtering for bearing-only target tracking in radar
    - estimation/prediction of stochastic volatility in econometrics

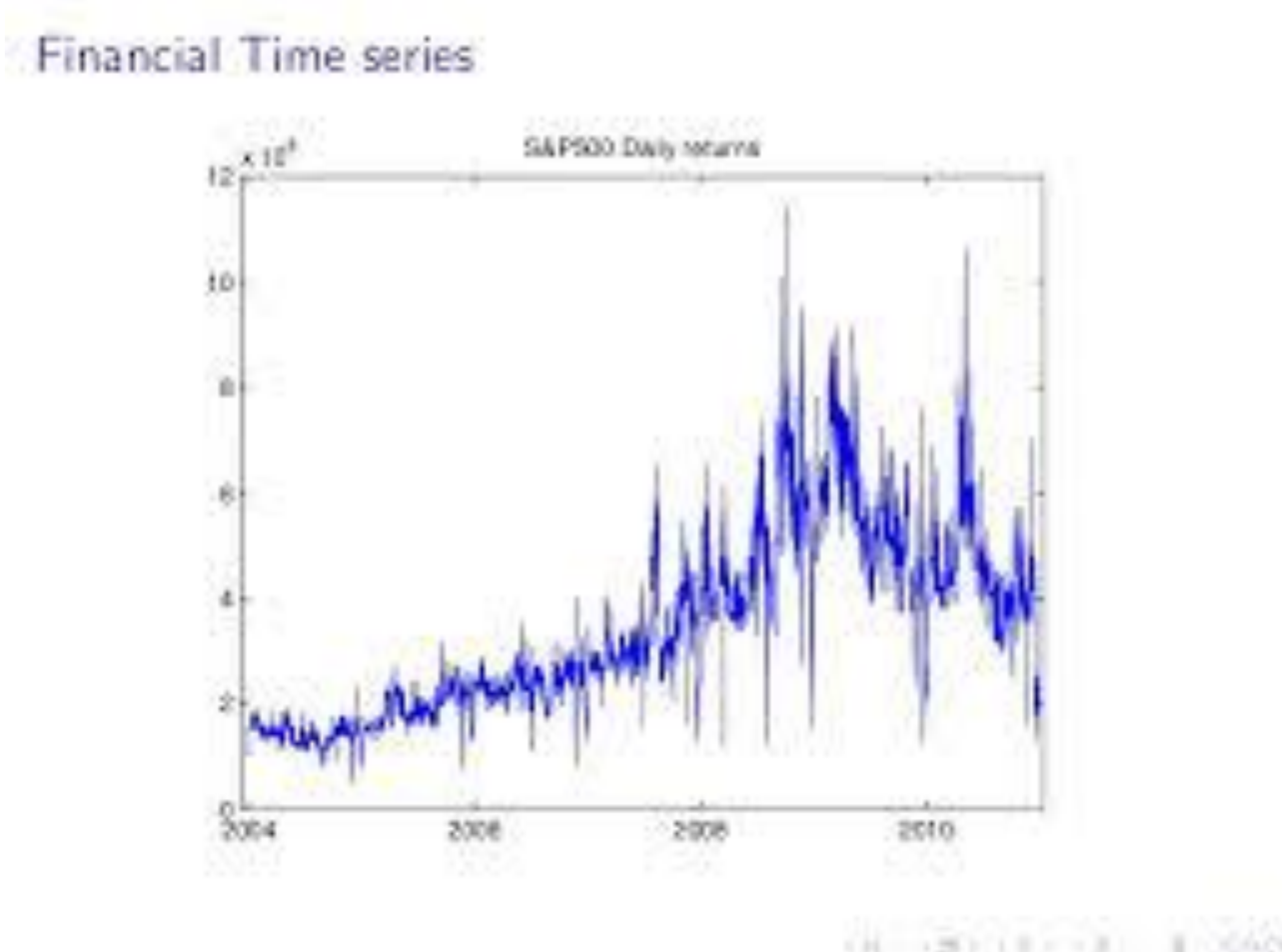
  - Space alternating data augmentation techniques, application to the estimation of finite mixtures of Gaussian distributions for speaker recognition

Bearing-Only Tracking (Radar)
Estimation of Nonlinear State Space Models



Recursive estimation of the target full position only from noisy measurements of the angle!

Stochastic Volatility (Econometrics)
Nonlinear Time Series Prediction/Forecasting



Speaker Recognition
Bayesian Modeling and Estimation

Speakers voices:

→ Gaussian mixture models (GMM)

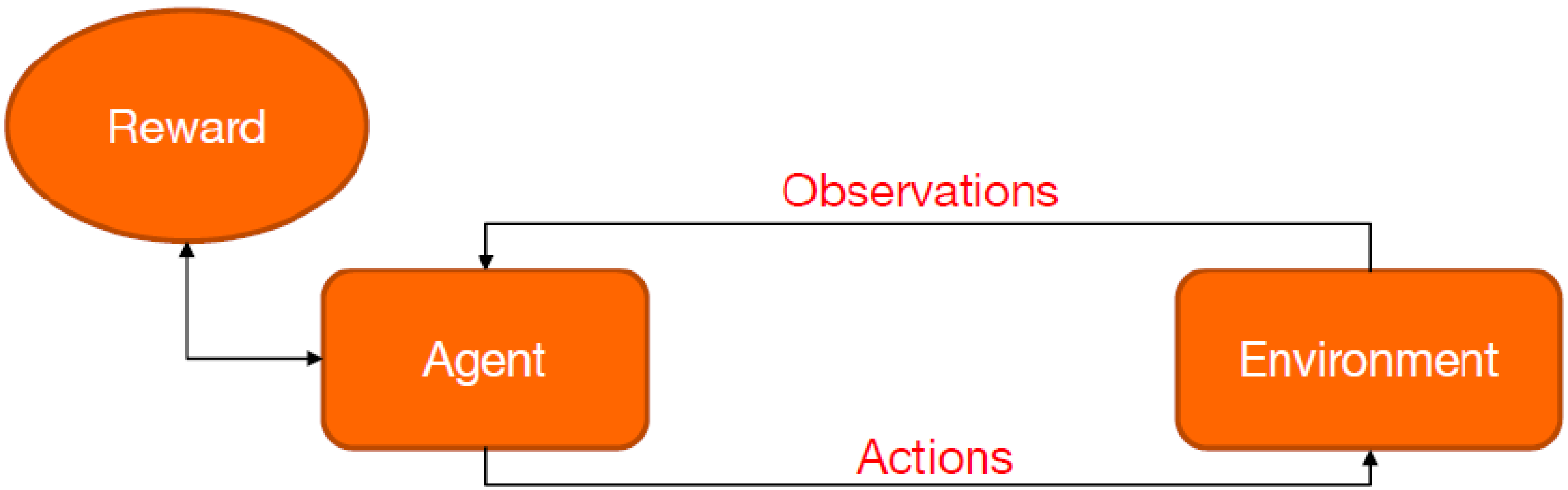
→ estimation of the models via
EM-type/Gibbs sampling algorithms

Professional Background and Activities (1/7): Machine Learning/Data Science

- Research Engineer/Scientist @ Orange Labs since 2005:
  - 2005-2006: Orange Labs Tokyo (Japan)
  - since 2007: Orange Labs Paris (now in Châtillon (92))
- Tackling problems and models/techniques/algorithms/methods in **Machine Learning** (statistical learning):
  - reinforcement learning → optimization/control of dynamic systems
  - supervised learning → input/output systems modeling and prediction
  - unsupervised learning → clustering, data dimensionality reduction
- Applications to telecommunications:
  - design of fixed and mobile networks optimization systems
  - design of traffic data processing systems

Reinforcement Learning Core Idea (1/2)

Reinforcement Learning Core Idea (2/2)



Reinforcement learning goal: **optimize** rewards by choosing adequately actions for given observations ⇒ from **policies**

Professional Background and Activities (2/7): Orange Labs Tokyo, 2005-2006

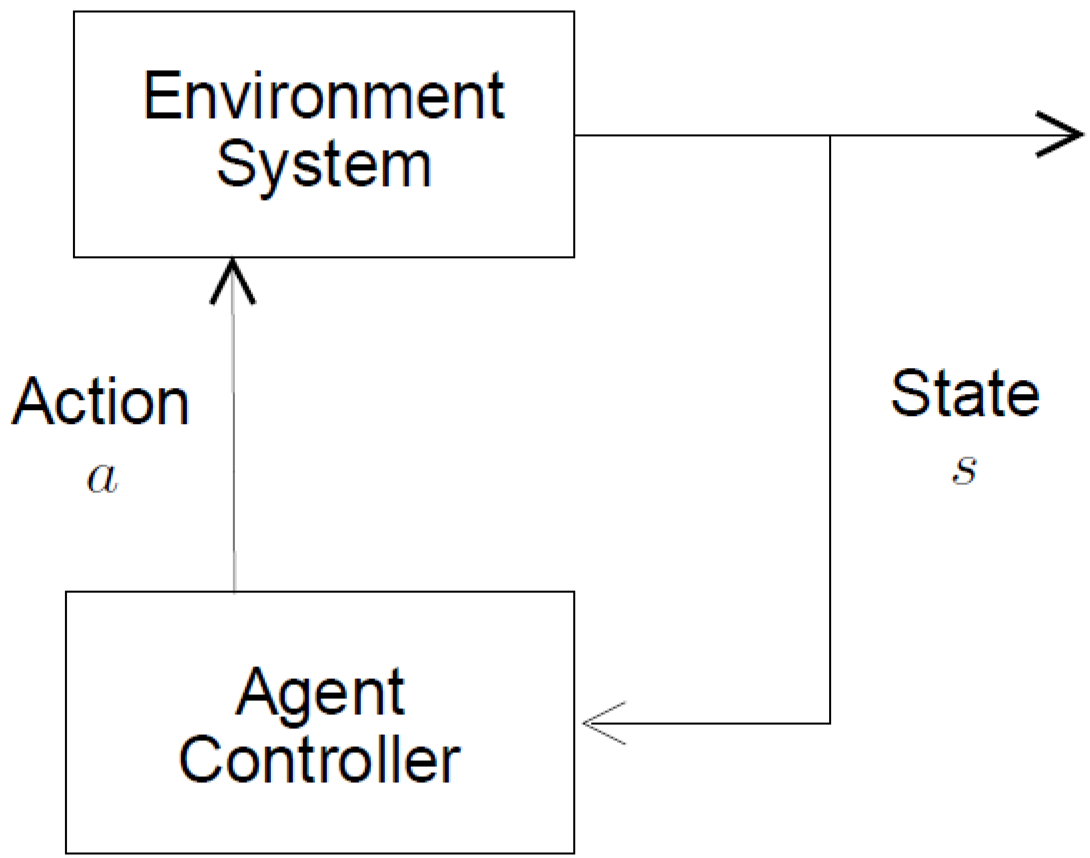
- Markov Decision Processes (MDP) models and Reinforcement Learning techniques:
  - Dynamic programming
  - Temporal Differences (TD-lambda)
  - Q-Learning algorithm and its extensions (SARSA, eligibility traces)
  - Parametric approximation techniques (Policy Gradient, Least Squares Policy Iteration)

- Support Vector Machines (SVM) techniques and related extensions for regression and classification (→ input/output systems modeling and prediction)

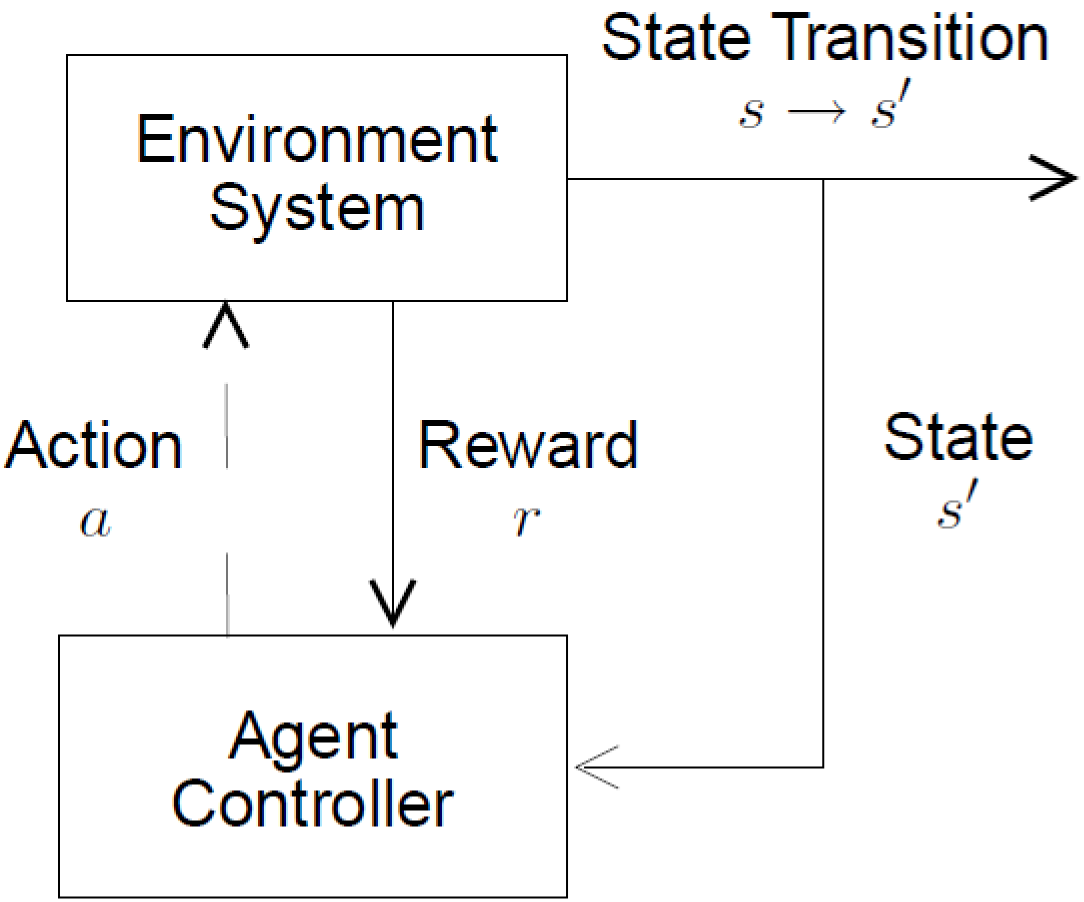
Professional Background and Activities (3/7): Orange Labs Tokyo, 2005-2006

- Applications for telecommunication systems:
  - Radio Resource Management for mobile networks (via Reinforcement Learning techniques)
  - Automated selection of radio access networks (via Support Vector Machines techniques)

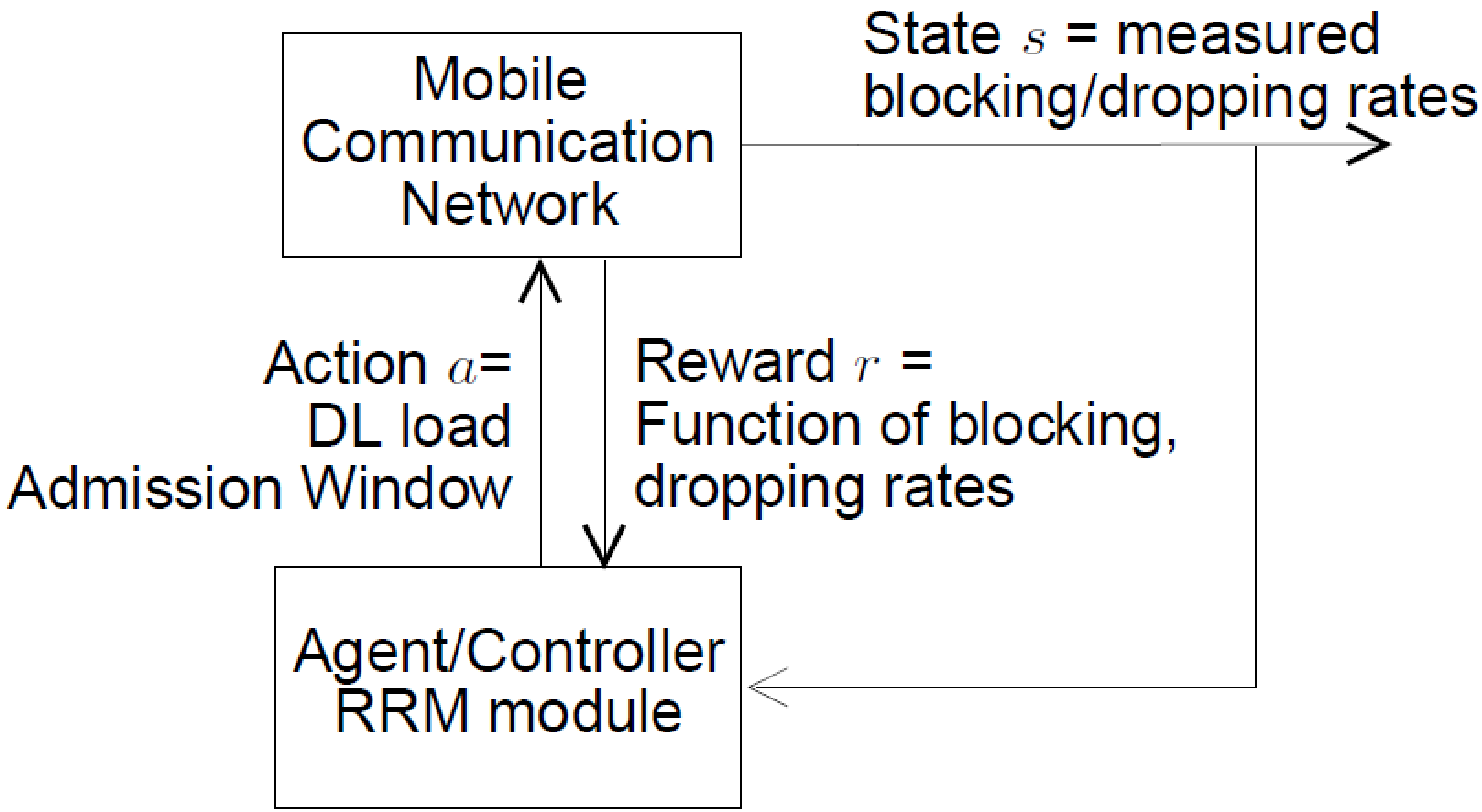
Learning & Control (1/3): Reinforcement Learning Framework



Learning & Control (2/3): Reinforcement Learning Framework



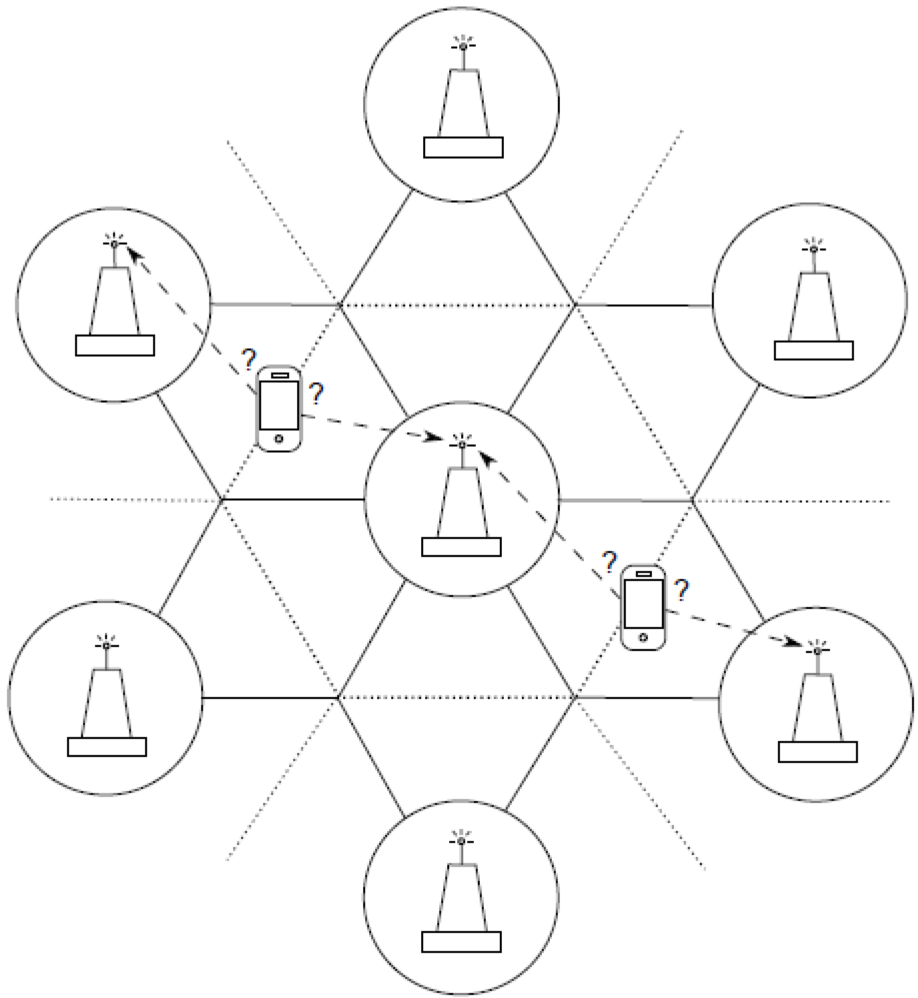
Learning & Control (3/3): Radio Resource Management for Mobile Networks



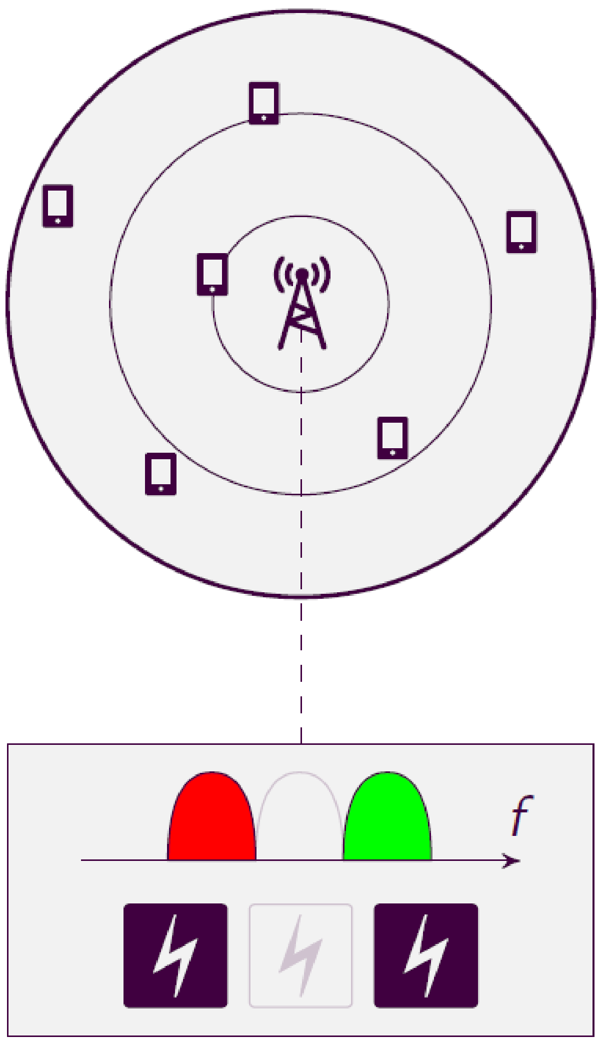
Professional Background and Activities (4/7): Orange Labs, Since 2007

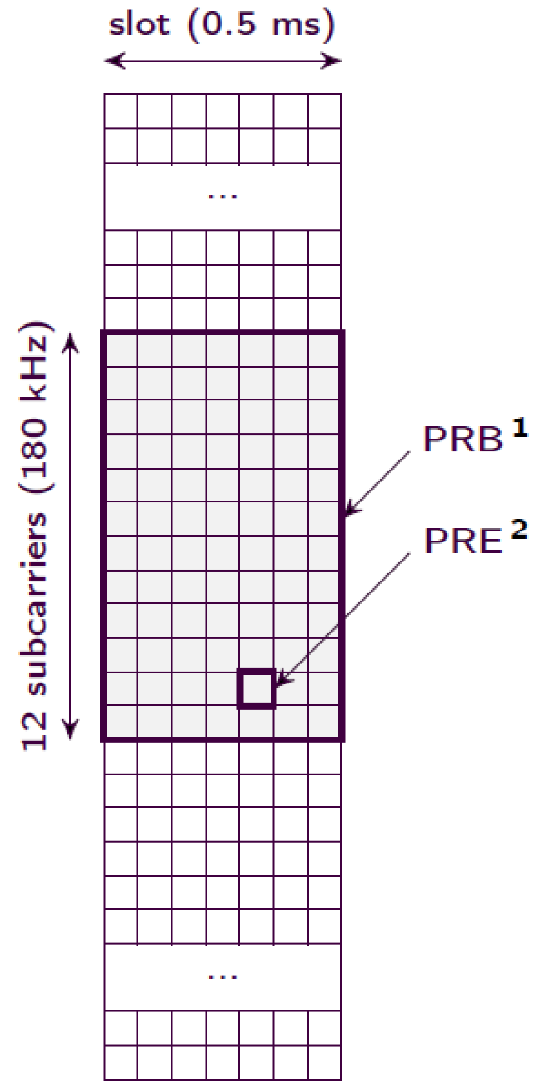
- **Reinforcement Learning:**
  - Partially observed models (POMDP) and dedicated learning techniques (Belief States → Monte Carlo POMDP, 2007-2009)
  - Policy Gradient (2011-2013):
    - Application to solution implementation for association problem of users to base stations for a mobile communication network
    - Design of variance reduction algorithms for Policy Gradient type estimation techniques in Reinforcement Learning
  - Dynamic Programming (2015):
    - Application to joint QoS and energy consumption control for mobile communication networks

Users Association Problem for Mobile Networks



Joint QoS and Energy Consumption Control





1: Physical Resource Block
2: Physical Resource Element

Reinforcement Learning Applications

- Dynamic channel allocation for mobile communications
- Job-shop scheduling
- Robotics: self-localization and mapping (SLAM)
- Learn helicopter-drones to perform loopings ☺
- Computer games:
  - Backgammon…
  - Reinforcement learning combined with deep learning architectures (AI agents designed by Google DeepMind):
    - Video Games: Space Invaders, Breakout, …
    - Computer Go: AlphaGo system

Professional Background and Activities (5/7): Orange Labs, 2010-2012

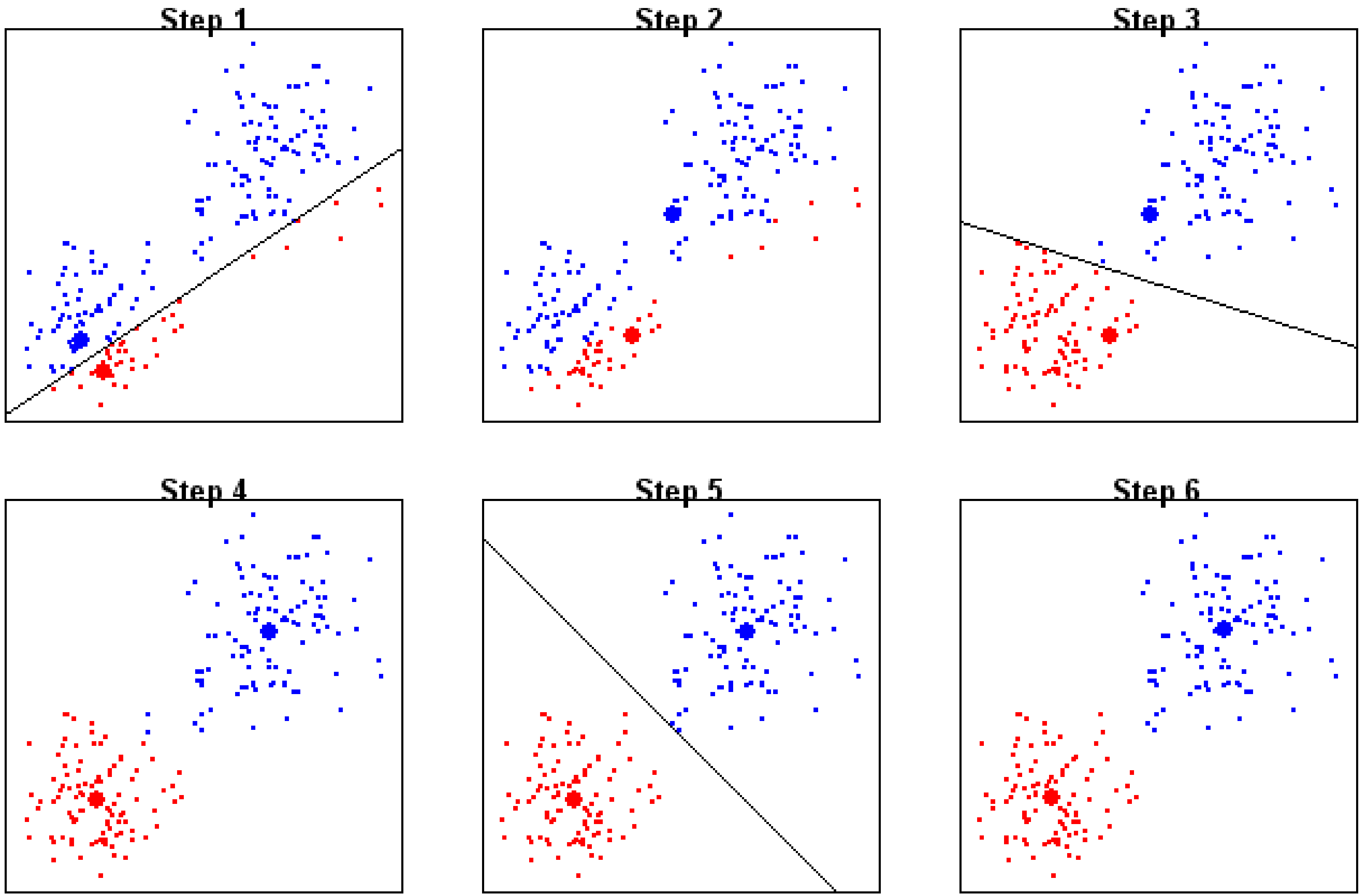
- **Unsupervised learning:**
  - Clustering: K-Means algorithm and its extensions
  - Application to computer networks platforms optimization for handling, managing, processing Internet traffic
  - Domain Name System (DNS), Internet traffic load balancing, DNSSEC implementation

Data Segmentation/Clustering

Input dataset → Output dataset composed of K « homogeneous » subsets

Various techniques:
combinatorial algorithms, Gaussian mixture models,
vector quantization, hierarchical clustering (dendograms),
K-Means and its extensions…

K-Means Clustering Algorithm Example

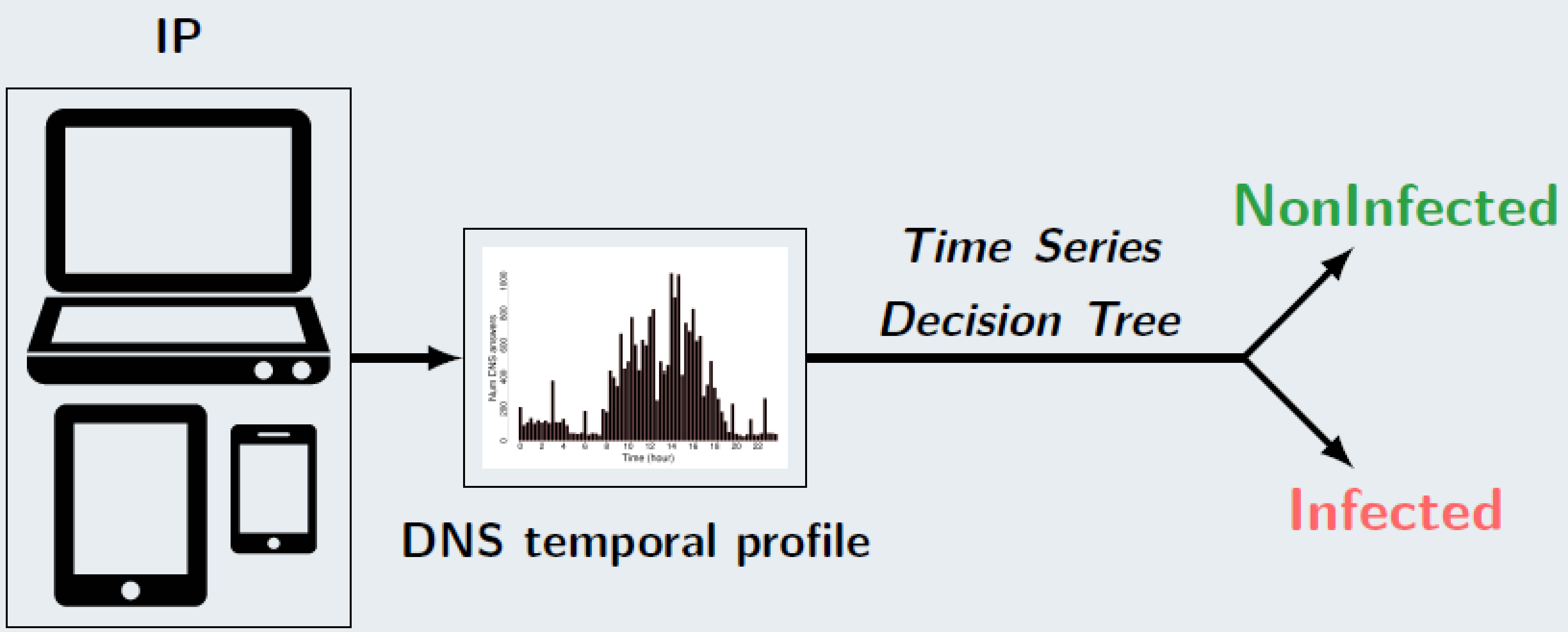


Professional Background and Activities (6/7): Orange Labs, 2011-2015

- **Supervised Learning:**
  - Models and methods for implementing Internet/DNS traffic traces classification:
    - Kernel Learning (Multiple Kernel Learning, Support Vector Data Description, 2011-2014)
    - Artificial Neural Networks (Extreme Learning Machines, 2013-2014)
    - Decision Trees (with time series as inputs, 2014-2015)
  - Application to Internet traffic analysis for network security
    → **Botnets/malwares detection**

Botnets/Malwares Detection from @IP Traffic Data

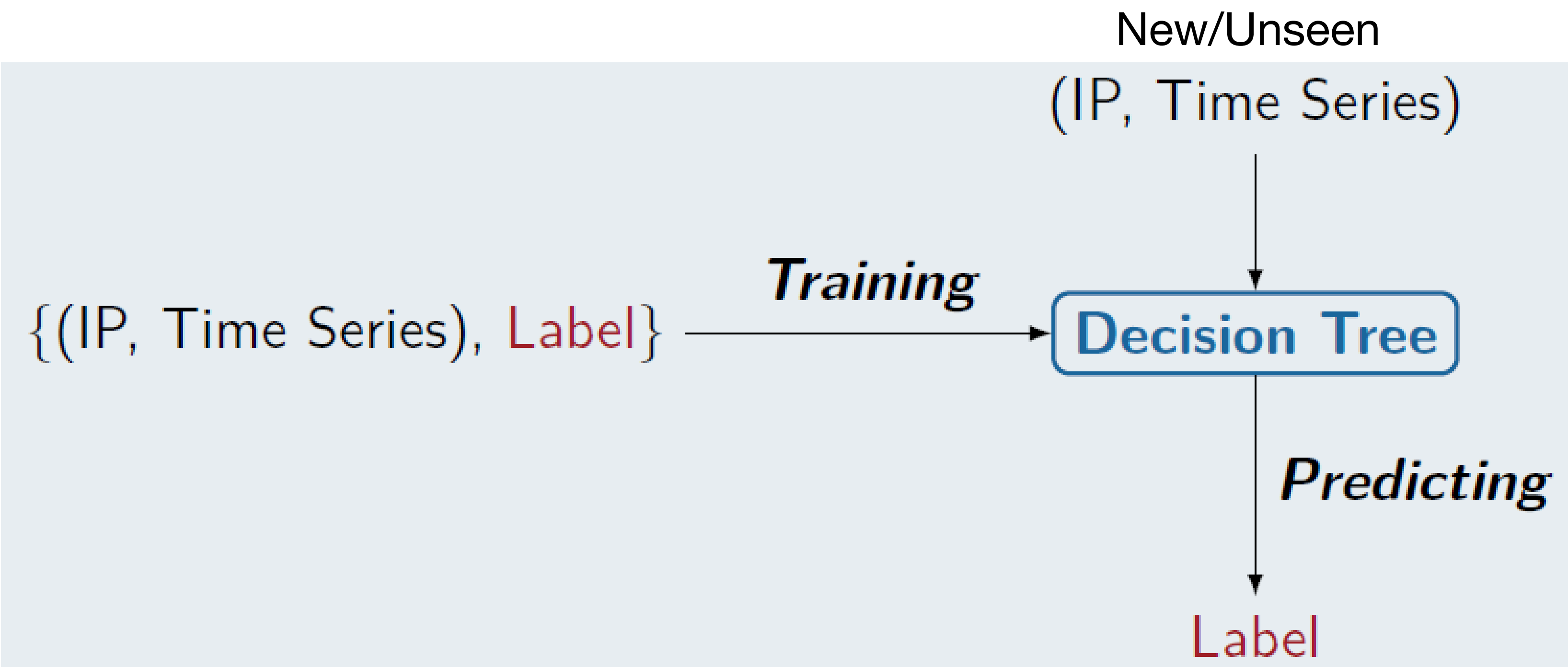
Detecting Infected IPs



Supervised Learning (1/2): Training/Learning Phase

$$\{(\text{IP, Time Series}), \text{Label}\} \xrightarrow{\textit{Training}} \boxed{\text{Decision Tree}}$$

Supervised Learning (2/2): Prediction/Testing Phase



Decision Trees (1/4): Training Data Example

Id	Outlook (O)	Temperature (T)	Humidity (H)	Windy (W)	*Play*
a	overcast	$83°F$	86%	false	yes
b	overcast	$64°F$	65%	true	yes
c	overcast	$72°F$	90%	true	yes
d	overcast	$81°F$	75%	false	yes
e	rainy	$70°F$	96%	false	yes
f	rainy	$68°F$	80%	false	yes
g	rainy	$65°F$	70%	true	no
h	rainy	$75°F$	80%	false	yes
i	rainy	$71°F$	91%	true	no
j	sunny	$85°F$	85%	false	no
k	sunny	$80°F$	90%	true	no
l	sunny	$72°F$	95%	false	no
m	sunny	$69°F$	70%	false	yes
n	sunny	$75°F$	70%	true	yes

Decision Trees (2/4): Building a Tree Model

$\mathtt{H} \geq 82.5\%$

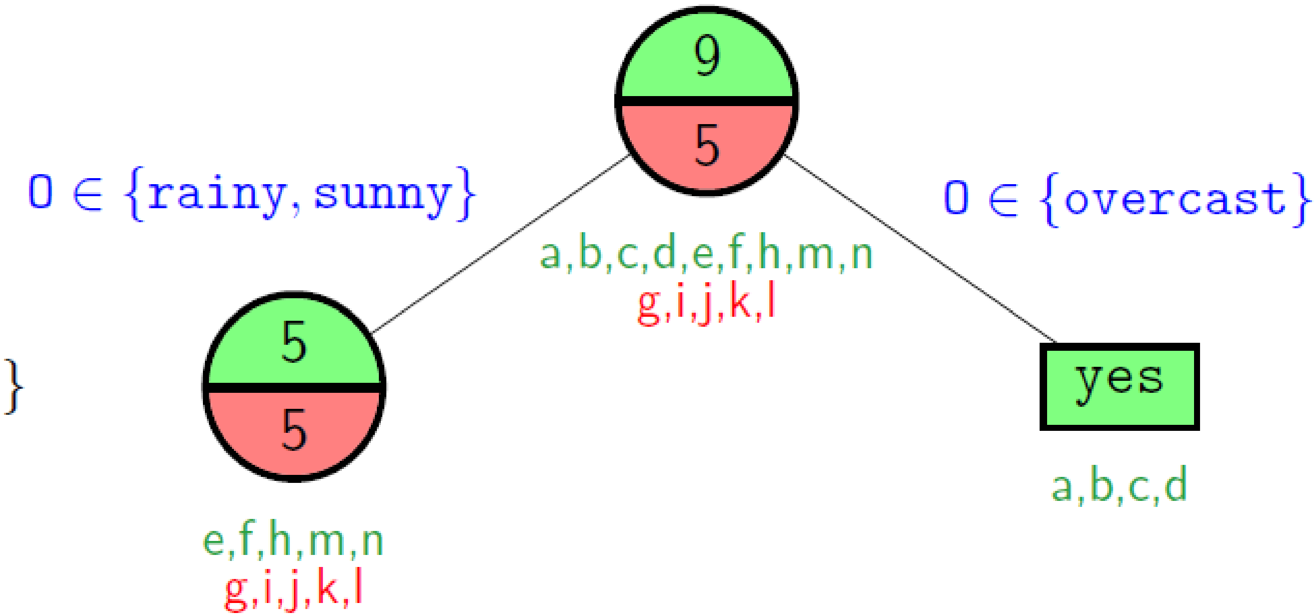
$\mathtt{H} \geq 75.5\%$

$\mathtt{O} \in \{\mathtt{rainy}, \mathtt{sunny}\}$

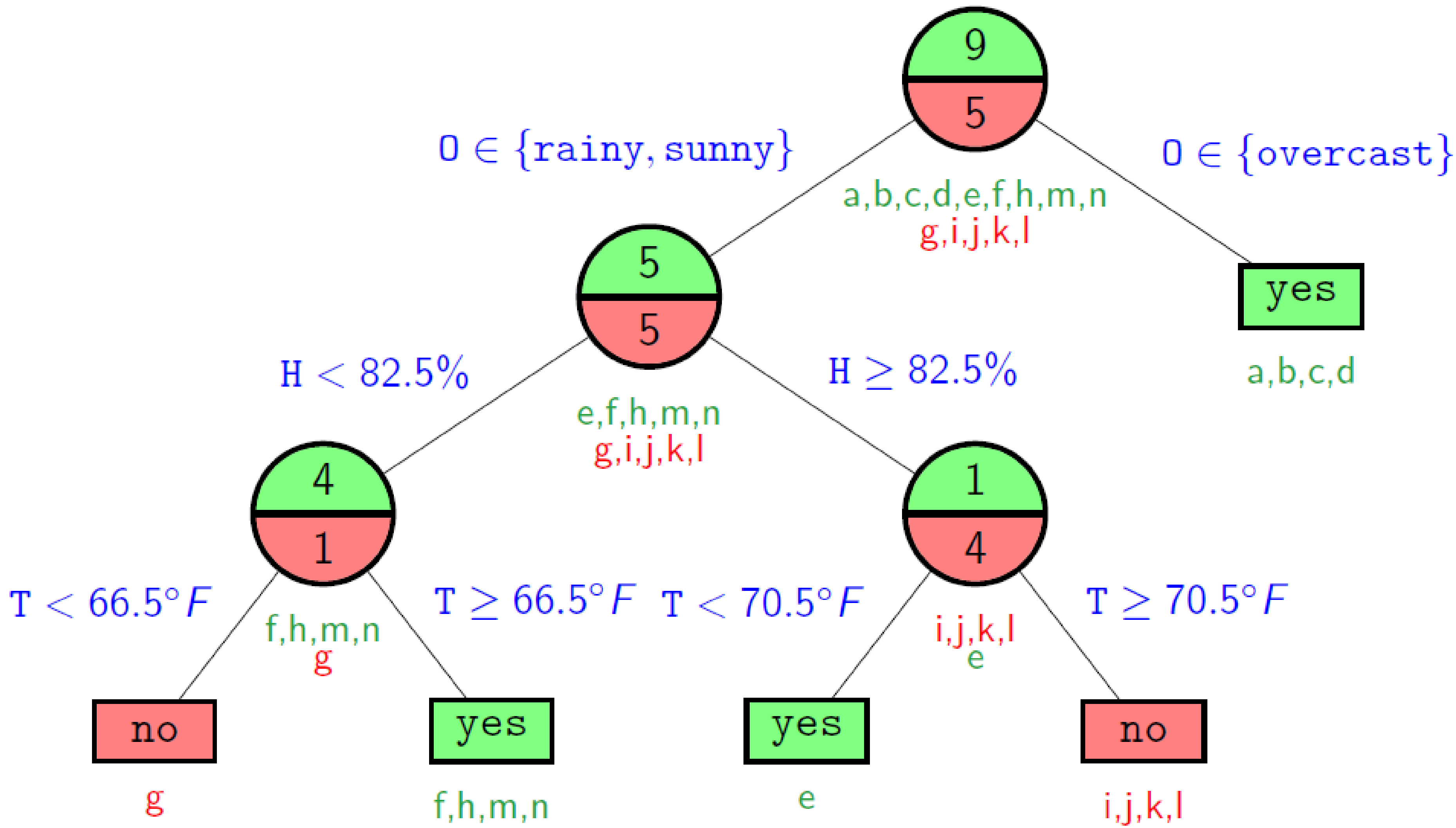
$\ldots$

$\mathtt{O} \in \{\mathtt{rainy}, \mathtt{overcast}\}$

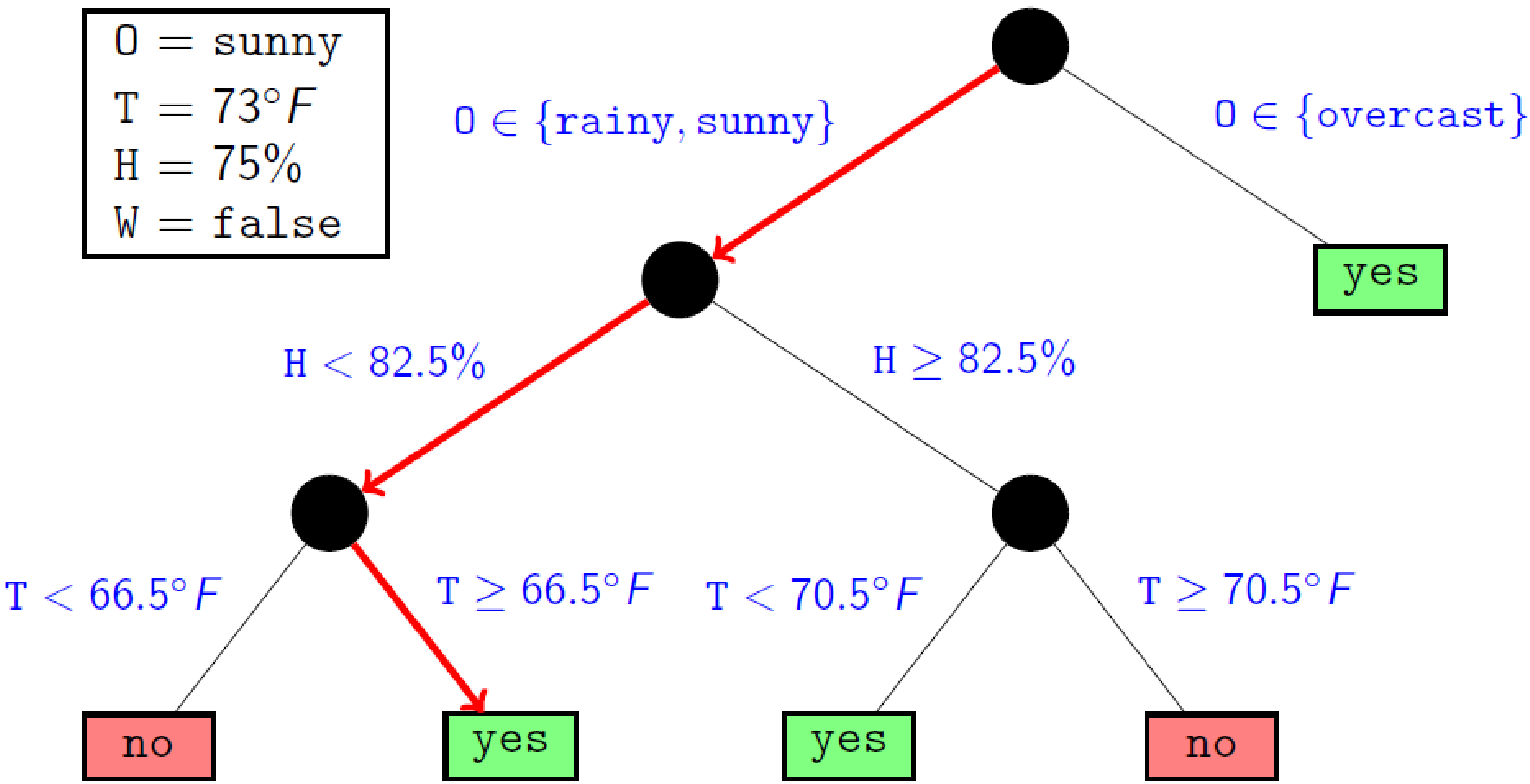
$\mathtt{W} = \mathtt{true}$



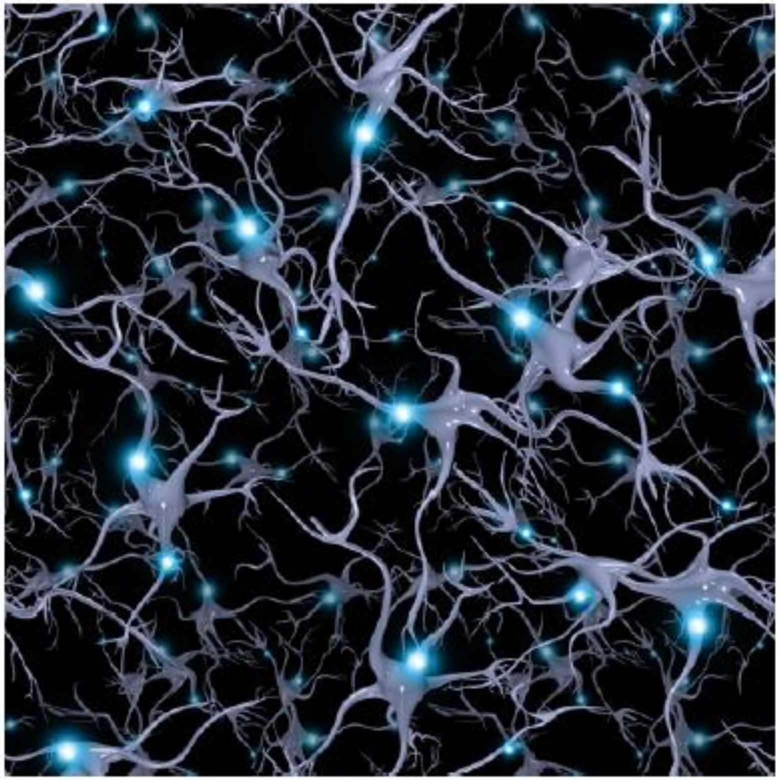
Decision Trees (3/4): Building a Tree Model

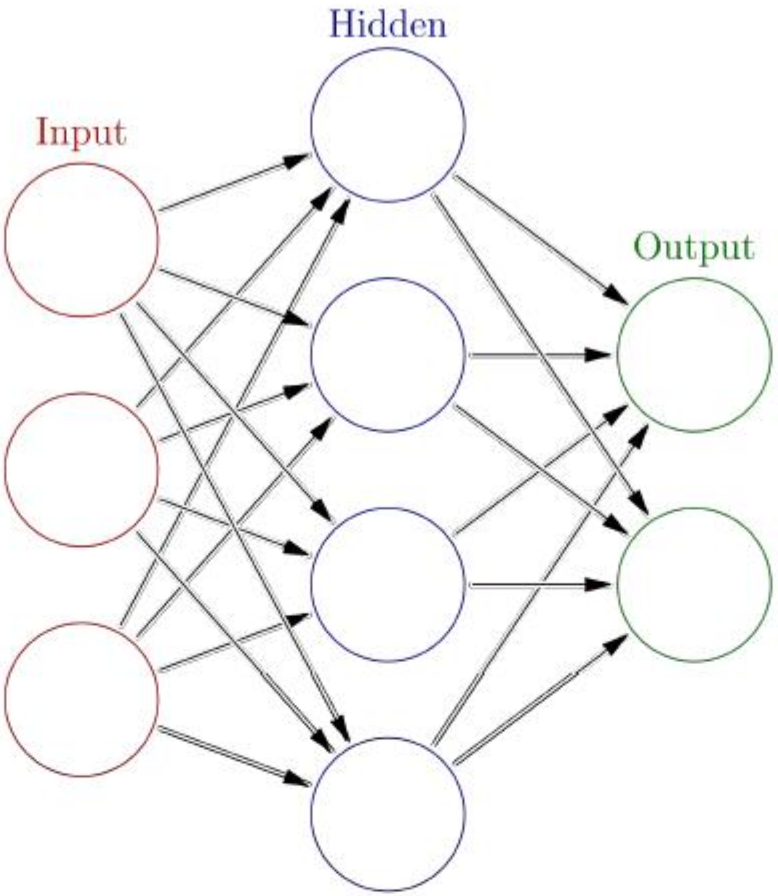


Decision Trees (4/4): Prediction from a Tree Model

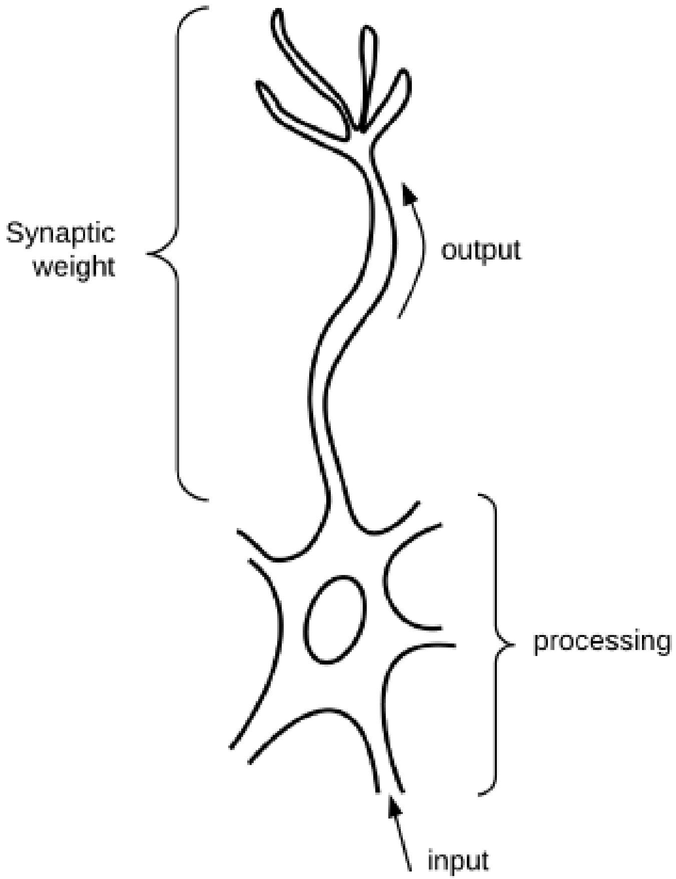


Artificial Neural Networks Models (1/4)

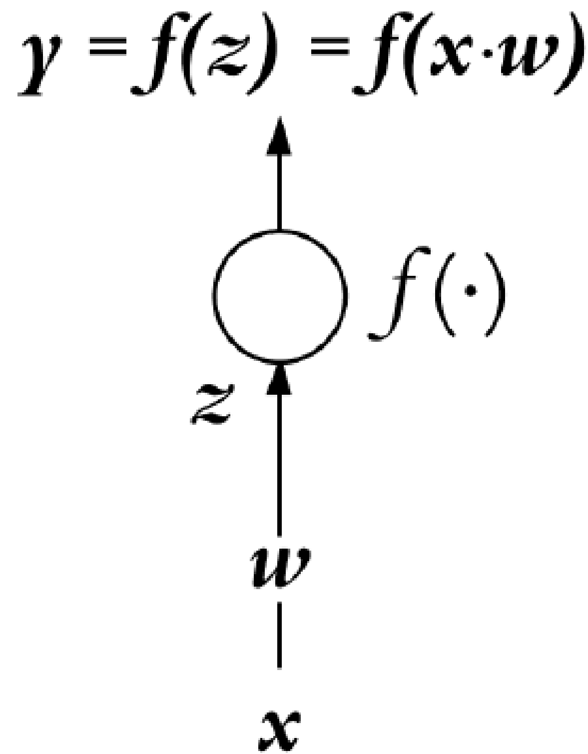




Artificial Neural Networks Models (2/4)

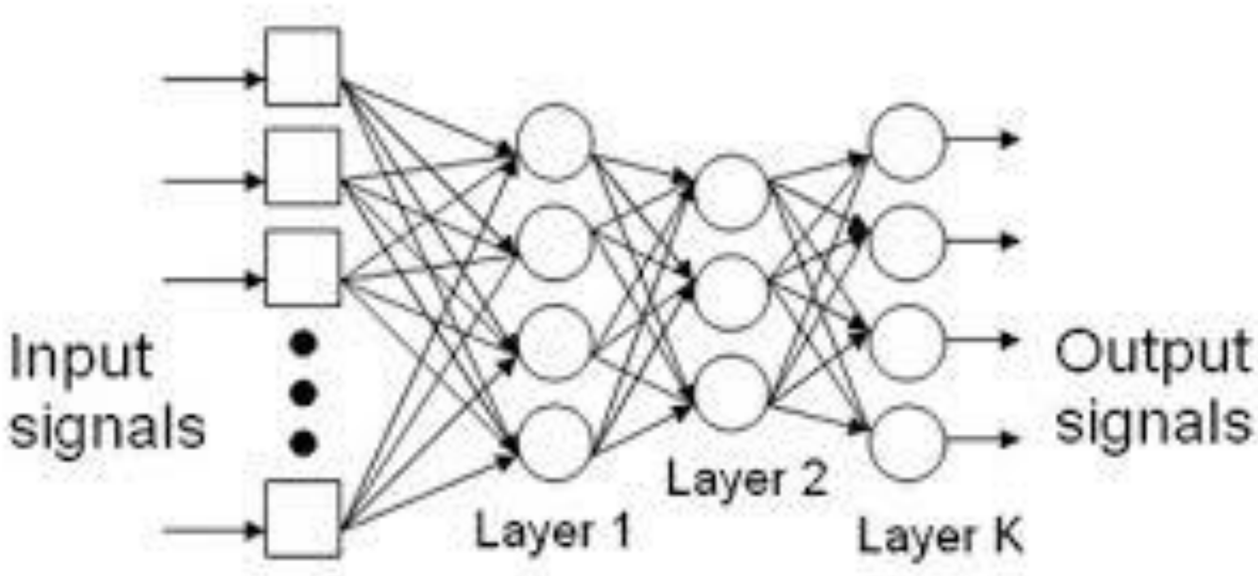


(a) Biological neuron.



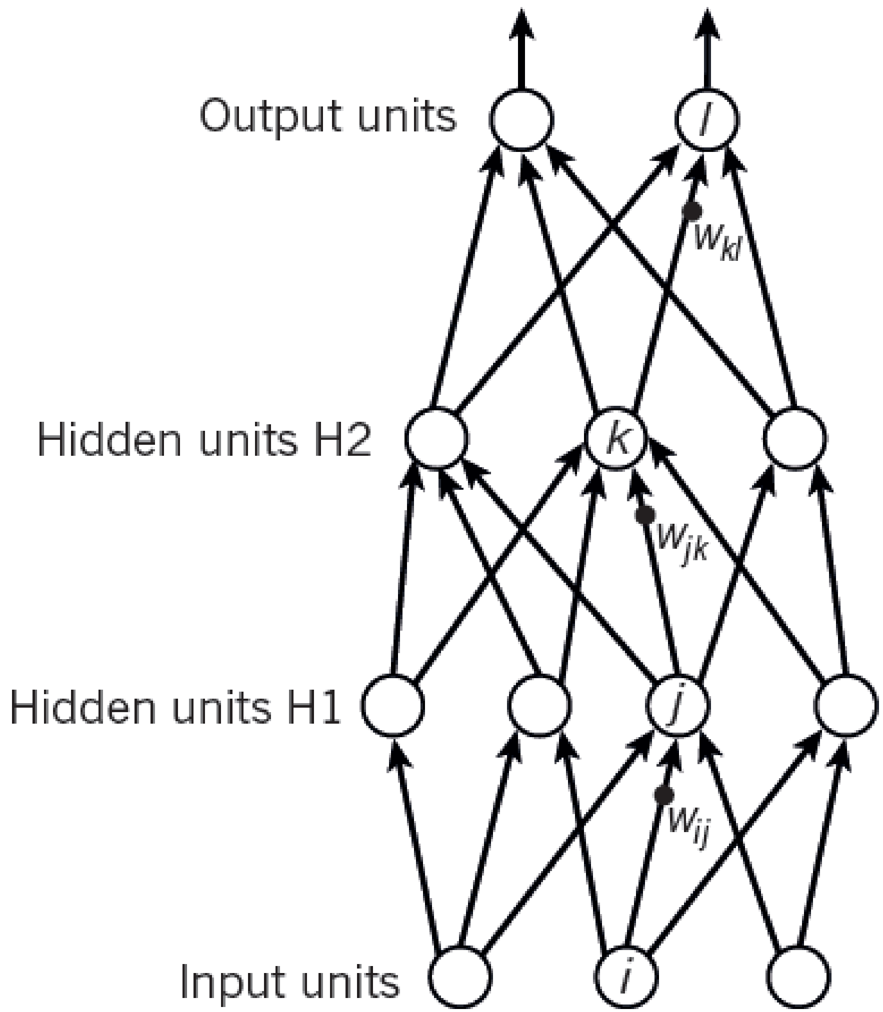
(b) Artificial neuron.

Artificial Neural Networks Models (3/4)



« Multi-Layer Perceptron » (MLP) architecture

Artificial Neural Networks Models (4/4)



$$y_l = f(z_l)$$

$$z_l = \sum_{k \,\varepsilon\, H2} w_{kl}\, y_k$$

$$y_k = f(z_k)$$

$$z_k = \sum_{j \,\varepsilon\, H1} w_{jk}\, y_j$$

$$y_j = f(z_j)$$

$$z_j = \sum_{i \,\varepsilon\, \text{Input}} w_{ij}\, x_i$$

Model Learning/Training with Backpropagation of the Error

Professional Background and Activities (7/7): Orange Labs, Currently: 2016-2017

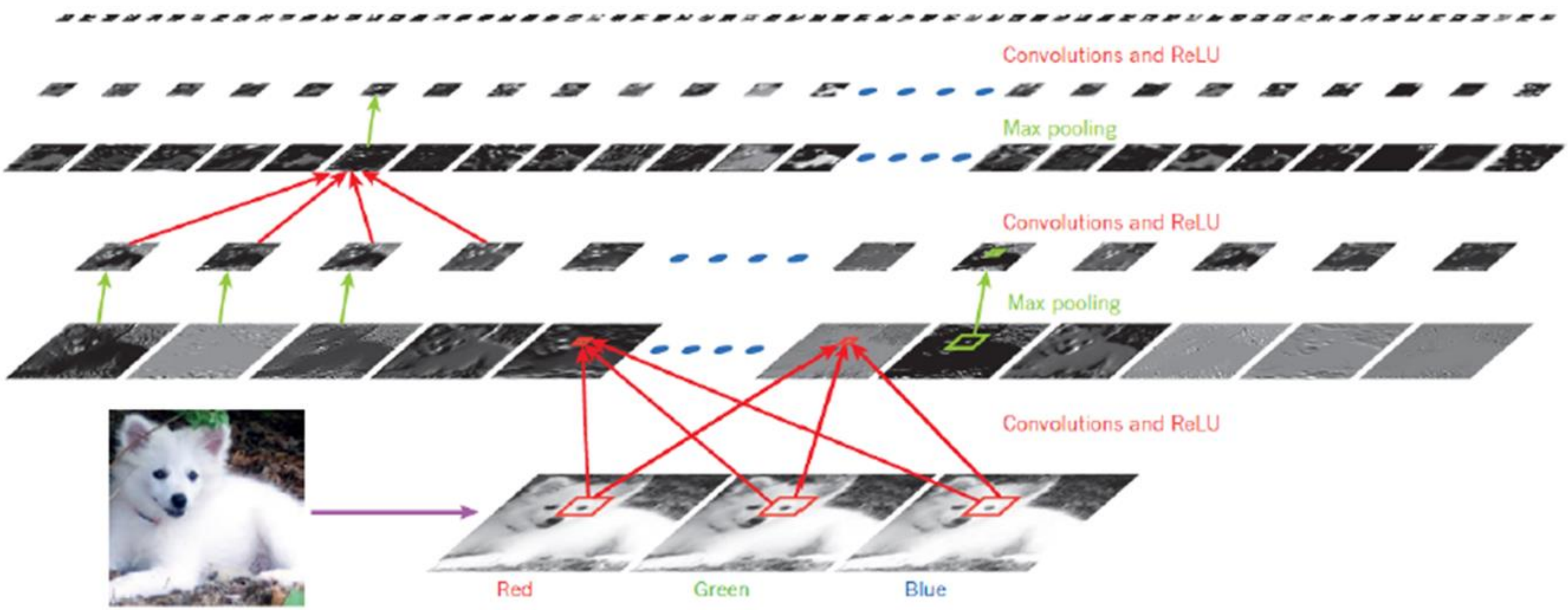
- **Deep Reinforcement Learning techniques:**
  - Understanding of Google DeepMind AlphaGo system, 2016
  - Application to resource allocation for mobile networks, 2017
- **Markov Chain Monte Carlo (MCMC) simulation methods:**
  - Event-Chain based Monte Carlo techniques
  - → Nonreversible Markov chains
- Networks metrics **data prediction benchmark** for autonomic network management:
  - Many Machine Learning models and techniques
  - Essentially supervised learning (regression)

AlphaGo?

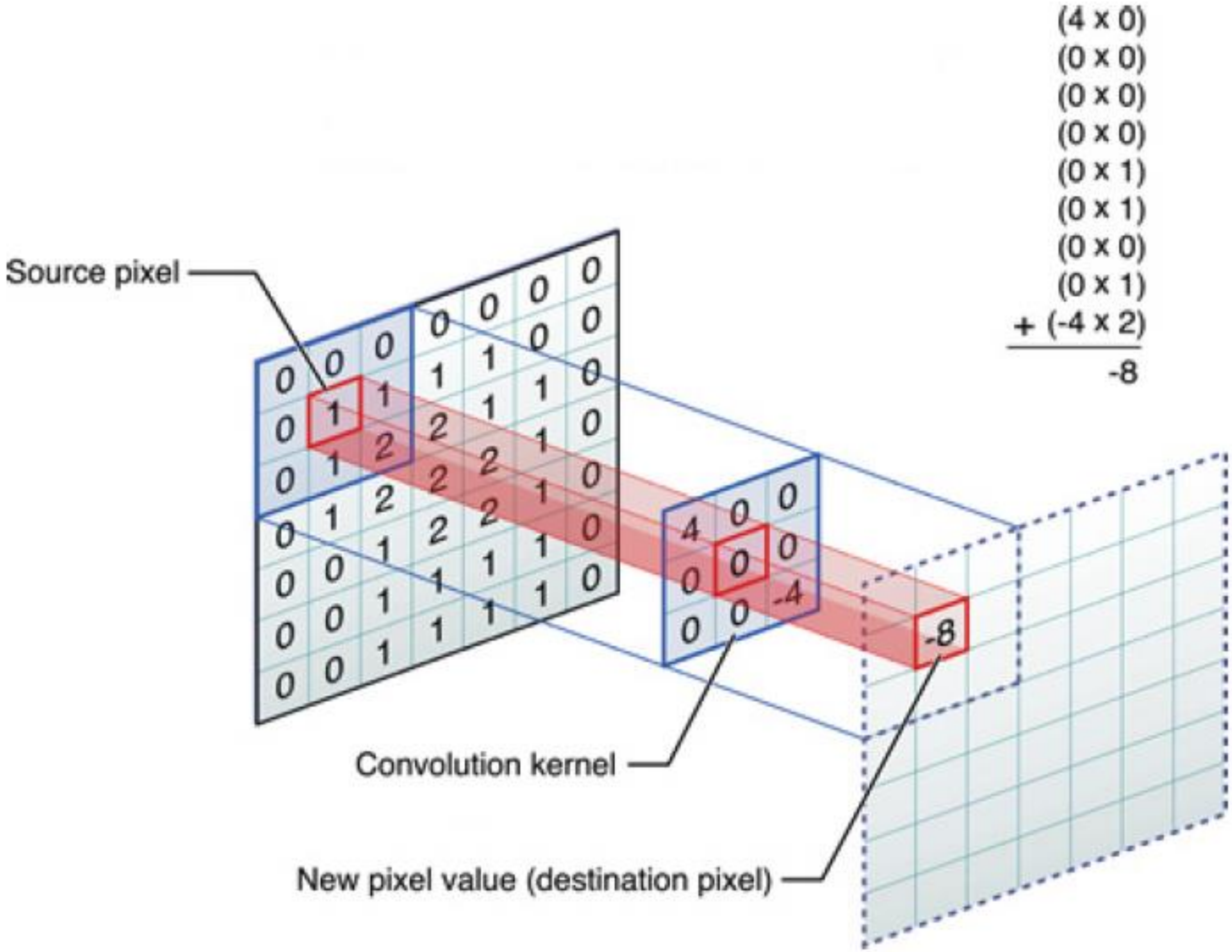
- Computer program, designed by Google DeepMind, which plays the game of Go

Video: AlphaGo masters the game of Go!

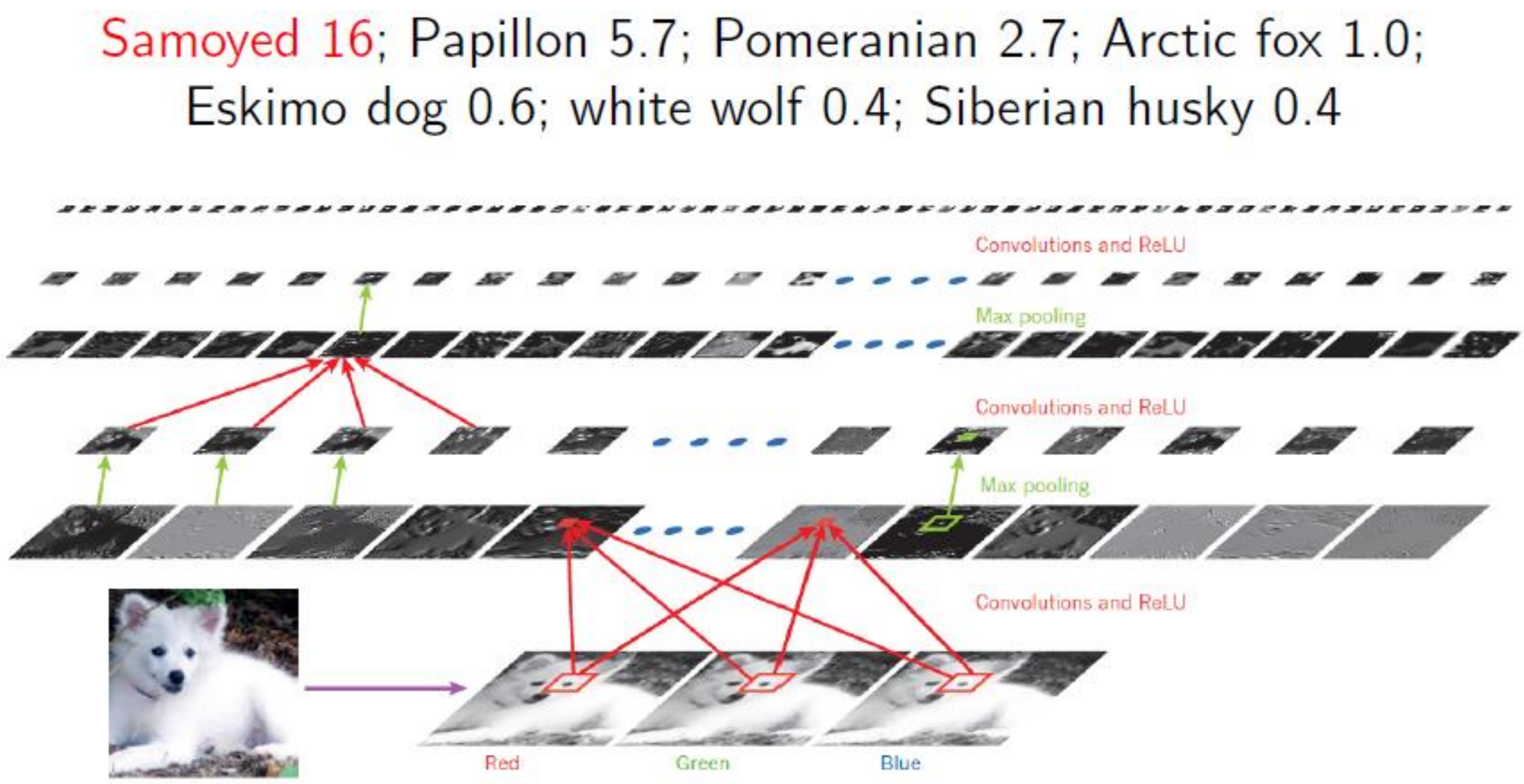
Convolutional Neural Networks (1/3): Modeling and Training/Learning Phases



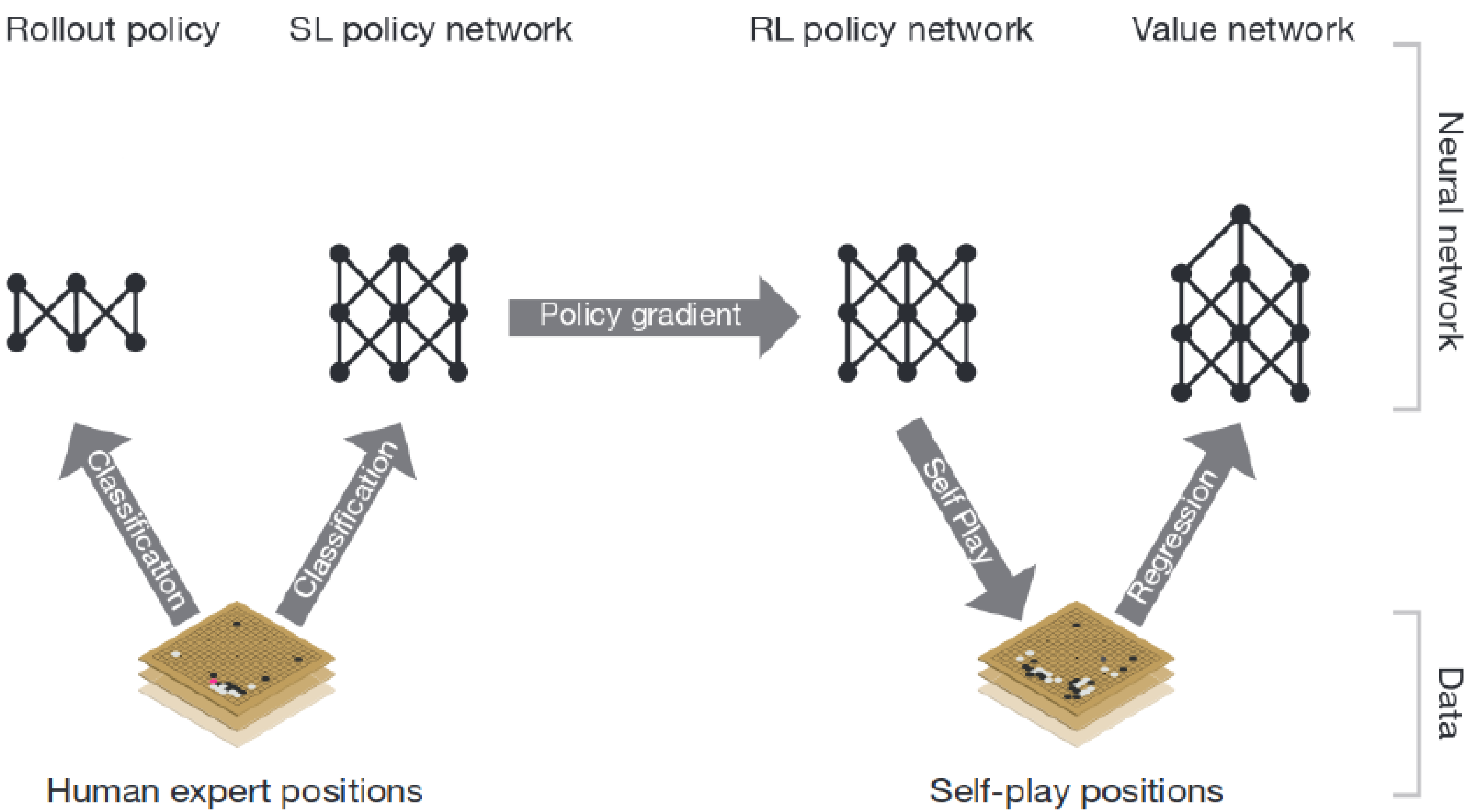
Convolutional Neural Networks (2/3): Example of a Convolution Kernel



Convolutional Neural Networks (3/3): Testing/Prediction Phase



AlphaGo System Training/Learning Global Pipeline



“From Science to Data Science”
Overview → Break! ☺

- Academic Background and Activities (17 slides)
- Professional Background and Activities (33 slides + 2 videos!)
- → Projects (in Data Science and others…) (6 slides)
- Working @ Orange Labs (3 slides)
- Machine Learning/Data Science… (2 slides)
- Take Away Messages (1 slide)
- Bonus: Appendix (44 slides)

Machine Learning/Data Science Activities → Projects

- Daily work organization in **Projects**, e.g. hosted by the "Applied Maths and Computer Science" Research Group @ Orange Labs:
  - Internal/Orange projects
  - Bilateral projects with Orange ("external research contracts")
  - Collaborative projects:
    - ANR "ECOSCELLS" 2009-2012
    - EU FP7 STREP "HARP" 2012-2015
    - ANR INFRA "NETLEARN" 2013-2017
    - EU H2020 5G-PPP "COGNET" 2015-2017

Projects: Practical Aspects (1/4) → Organization

- Working on a research theme in a “fixed-term” mode:
  - Work schedule: Project Management Plan (PMP), including Gantt charts, elaborated and submitted for validation before the launch of the project (“Kick-Off”)
  - Costs management:
    - In human resources: People*Day, People*Month or People*Year with monthly follow-up/reporting of consumed resources
    - Financial: elaboration of an initial budget, then management of missions and material costs, with on-the-fly reporting

Projects: Practical Aspects (2/4)
→ Organization

- Organization of the works in "Work Packages" (WP) with specific Tasks with tasks and WPs leaders + 1coordinator (Project Head) and 1 technical coordinator/leader

- Working meetings (Face-to-Face, conf calls) and milestones meetings (plenary = for all partners, Face-to-Face)

- Scientific and technical skills indeed, but also good communication and relational skills, patience, resilience and a certain sense of humor! ☺

Projects: Practical Aspects (3/4) → Deliverables

- Valorization of project works → "**Deliverables**":
  - Internal valorization:
    - Elaboration of technical reports and presentations
    - Reporting:
      - towards Orange: on the fly + semestrial official meetings with the hierarchy and with project entities (Project Head, Research Group Head)
      - towards the sponsor (ANR, EU) officially on a trimestrial, semestrial or annual basis
  - External valorization for research projects:
    - Publications in scientific and technical conferences (oral and poster presentations) and in scientific and technical journals/magazines: IEEE, ACM…
    - Organization of Workshops/Seminars in conferences or by ourselves

Projects: Practical Aspects (4/4) → Deliverables

- Valorization of project works → "Deliverables":
  - Patent filling
  - Normalization/standardization activities: 3GPP, IETF, ETSI, ITU…
  - Development of technical solutions and industrialization:
    - internally: simulators and prototypes → development transfer → inclusion in the Information System and/or transfer towards technical and operational directions, even towards Business Units (BU) sometimes…
    - externally: Open Source…

Research Works Valorization Example → Publications

- Conferences:
  - 27 papers published (3 invited papers)
  - 1 submitted
  - 2 in preparation
- Journals:
  - 6 papers published
  - 2 in preparation
- Books:
  - 1 chapter in “Data Mining Applications with R”, Elsevier, 2013

Working @ Orange Labs (1/2)

- “Department” (Team) “**Modeling and Statistical Analysis**”
- Activities on networks and traffic modeling for fixed (ADSL and Fiber Internet) and mobile (2G/3G/4G → 5G) communications
- Currently 22 people, including:
  - 1 intern
  - 1 apprentice
  - 5 CIFRE PhD students
  - 1 post-doc

Working @ Orange Labs (2/2)

- "**Modeling and Statistical Analysis**" Department/Team Goals:
  - Come up with a better understanding of communication trafic, related to terminals and usage evolution, regarding the very important increase of this trafic
  - Estimate and optimize networks infrastructure costs by geographical zones regarding the strategic choices for deploying new generation technologies (optical fiber, 4G mobile networks) with new architectures
  - Improve the Quality of Service (QoS) and performance for mobile networks
  - Provide analytical models for equipments energy consumption in order to estimate and predict the networks energy consumption

Information about Working @ Orange

- Opportunities for internships for MS/MEng students:
  - 4 to 6 months duration
  - Schedule: application from November for the forthcoming year
- Opportunities for apprenticeship, PhD programs (CIFRE) and post-docs
- Opportunities for permanent positions: e.g. @ R&D/Orange Labs, Research Engineer/Scientist positions
- → Check and apply on https://orange.jobs/site/en-home/
- Contact: Stephane SENECAL
  - email: stephane.senecal@orange.com
  - LinkedIn: https://www.linkedin.com/in/stephanesenecal/

More Information (1/2):
→ Machine Learning/Data Science

- Informal meetings and discussion groups (meet ups) in Paris/IDF:
  - → Paris Machine Learning Applications Group:
  - https://www.meetup.com/fr-FR/Paris-Machine-learning-applications-group/
  - 1+ meeting(s) per month
  - + Groups on LinkedIn, Facebook and Google+, Twitter account, Nuit Blanche blog (including the meet ups archive)…

  - → Deep Learning Paris:
  - https://www.meetup.com/fr-FR/Deep-Learning-Paris-Meetup/
  - + Workshops…

More Information (2/2):
→ Machine Learning/Data Science

- Academic seminar on Machine Learning in Paris:
  – → Statistical Machine Learning “SMILE in Paris”
  – https://sites.google.com/site/smileinparis/
  – Organized by ENS and Mines-ParisTech

- Academic group on Data Science in France:
  – → GdR MaDICS: Masses de Données, Informations et Connaissances en Science
  – http://www.madics.fr/
  – Organized by CNRS

- Internet Group/Forum (worldwide audience → in English☺):
  – → Google Group: Machine Learning News
  – CFPs, job offers, …
  – https://groups.google.com/forum/#!forum/ML-news

“From Science to Data Science” → Key/Take Away Messages!

- Even with a different background, you can make it in Machine Learning/Data Science! ☺
- Machine Learning/Data Science is a broad field → choose 1 or 2 specific topics first… and study/work hard! ☺
- Many information sources and resources in Data Science…
  → finding relevant and appropriate/useful information is essential
- Try to be kept up-to-date for recent advances in the field, e.g. for Machine Learning: follow-up of ICML and NIPS conferences, JMLR journal and papers on ArXiv repository…
- → Always be curious in science and in technology! ☺

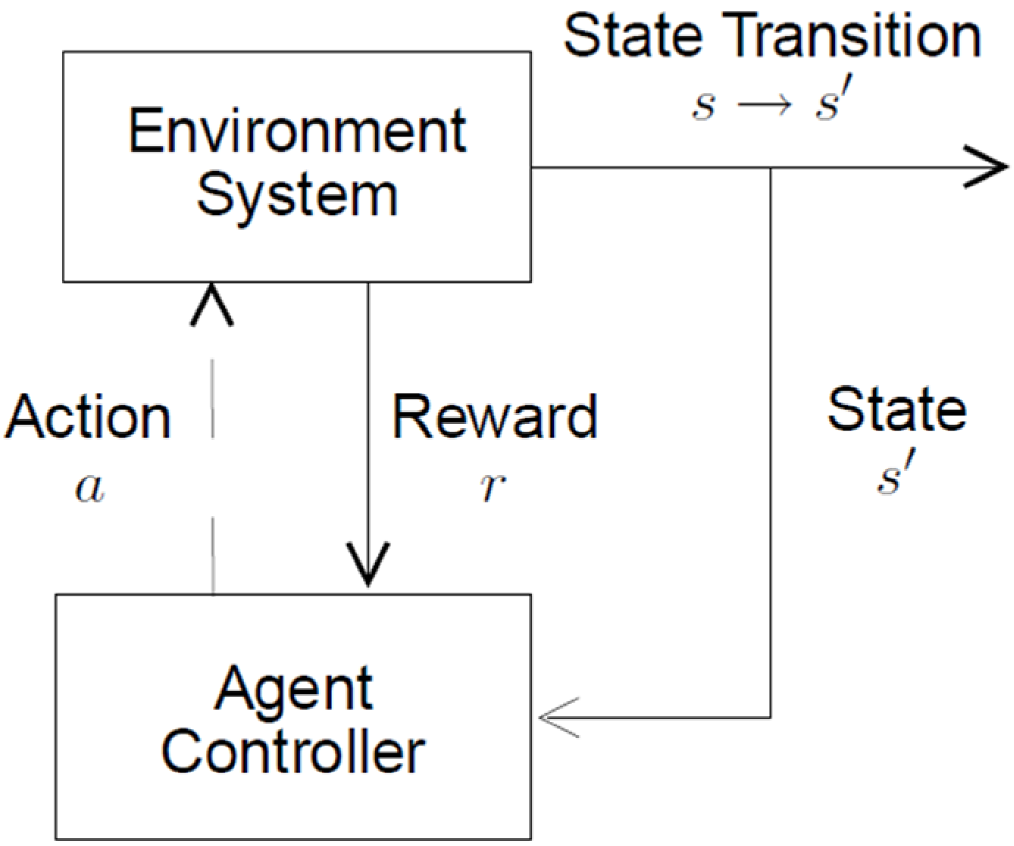
Thank You!☺

Questions?

Appendix

Reinforcement Learning
Policy Evaluation, Policy Iteration

Learning and Control Framework



➜ Optimization: find policy $\pi : s \in \mathcal{S} \mapsto \pi(a|s)$ to maximize objective/target function $f_\pi(\mathcal{R}(s,a))$

Policy Evaluation → Value Function

- Policy evaluation ? $\rightarrow$ state-action value function for delayed rewards under policy π

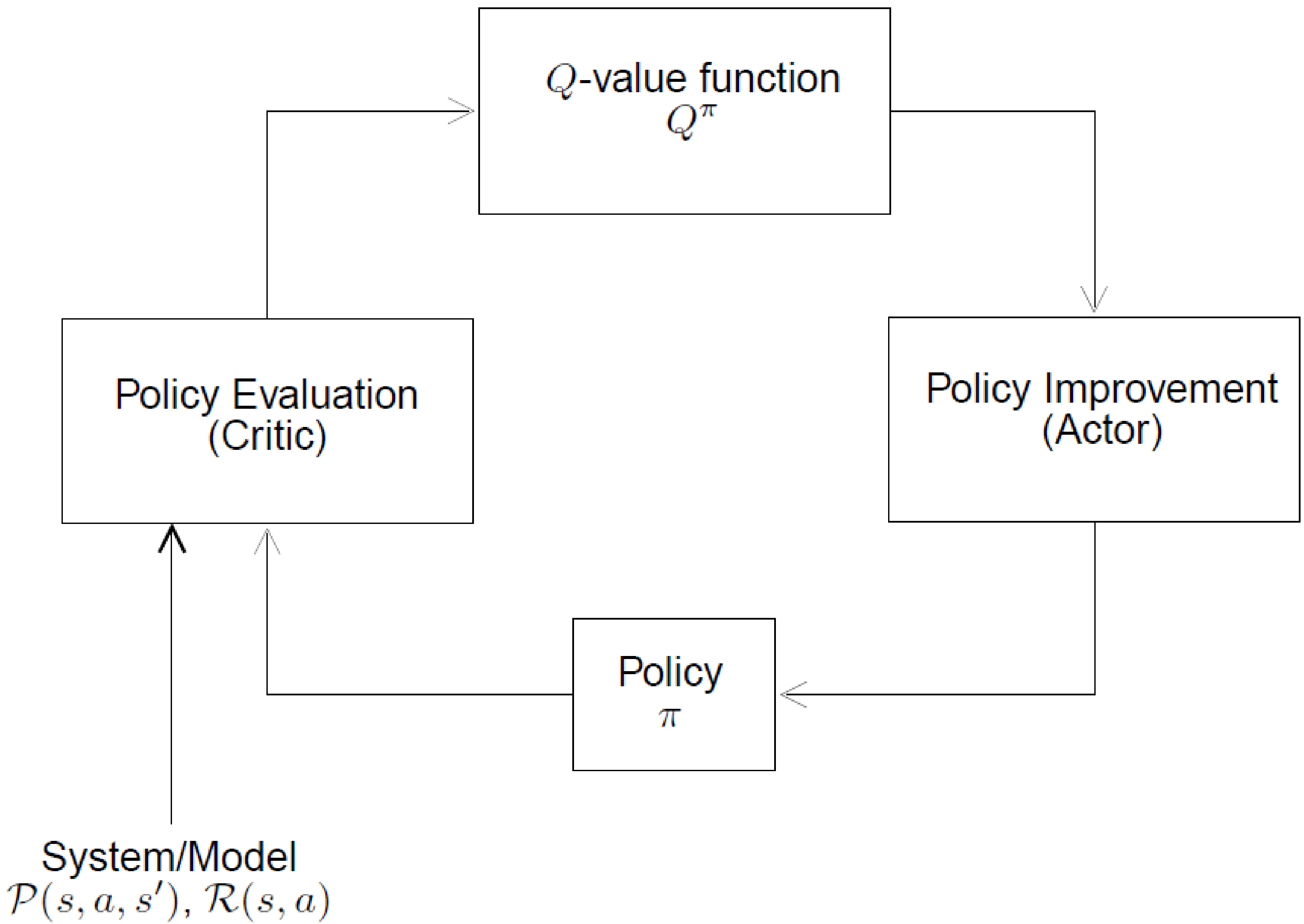
$$Q^{\pi}(s,a) = E_{(s_t \sim \mathcal{P}, a_t \sim \pi)} \left\{ \sum_{t=0}^{+\infty} \gamma^t r_t \,|s_0 = s, a_0 = a \right\}$$

γ = discount factor $0 < \gamma < 1$

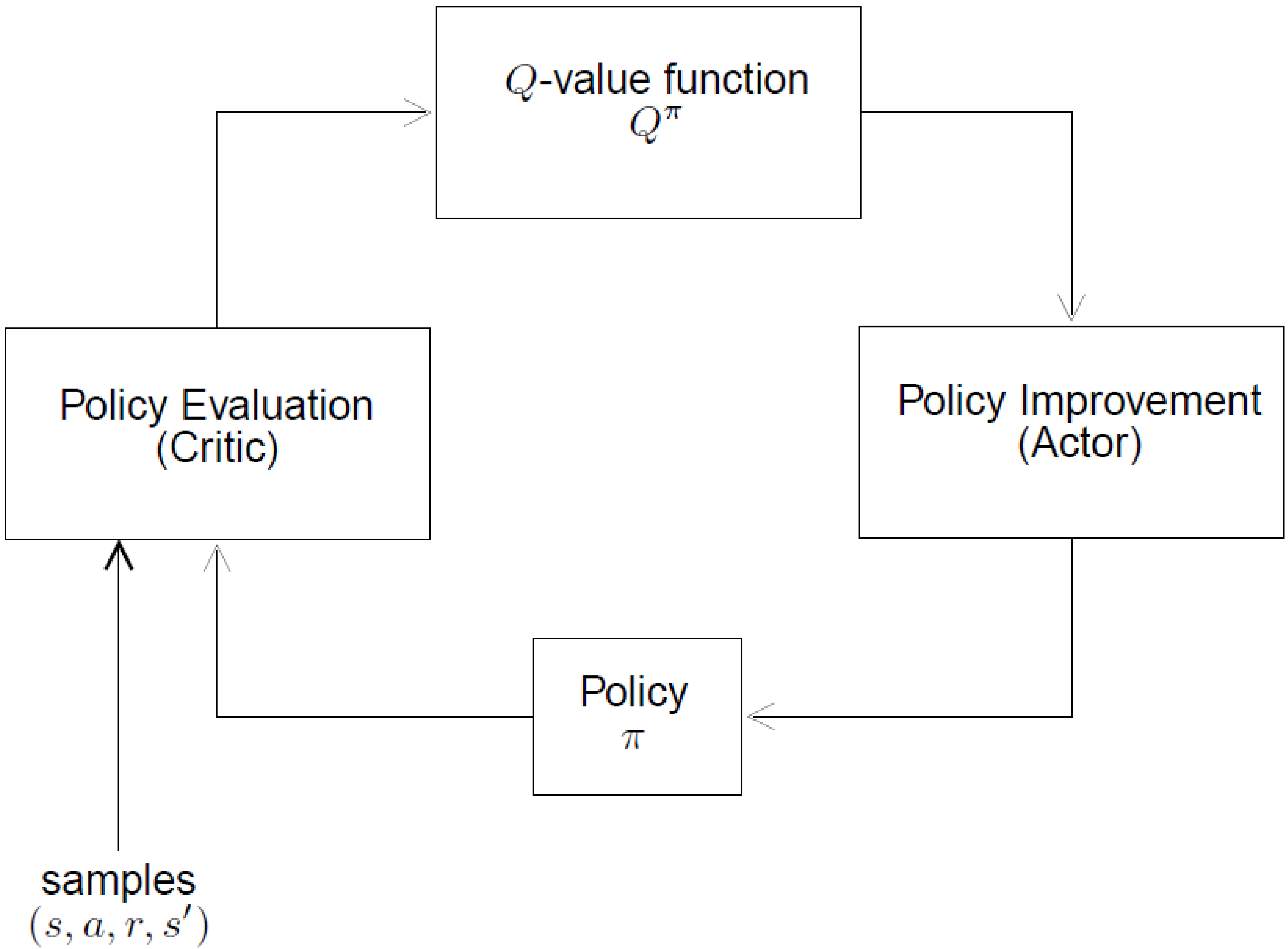
- Find the optimal policy

$$\pi^\star = \arg\max_{\pi} Q^{\pi}$$

Policy Iteration → Dynamic Programming



Policy Iteration → Reinforcement Learning



Supervised Learning
Support Vector Machines

Supervised Learning: Classification

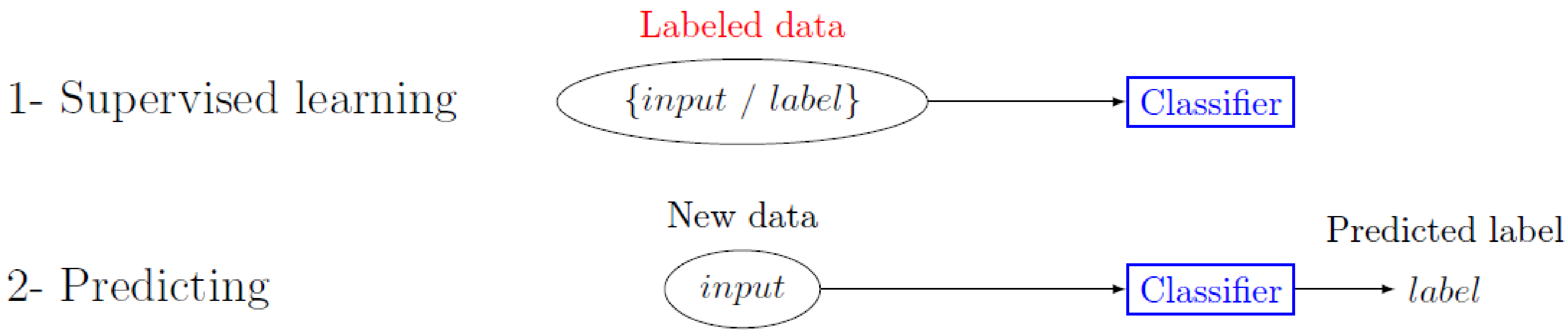
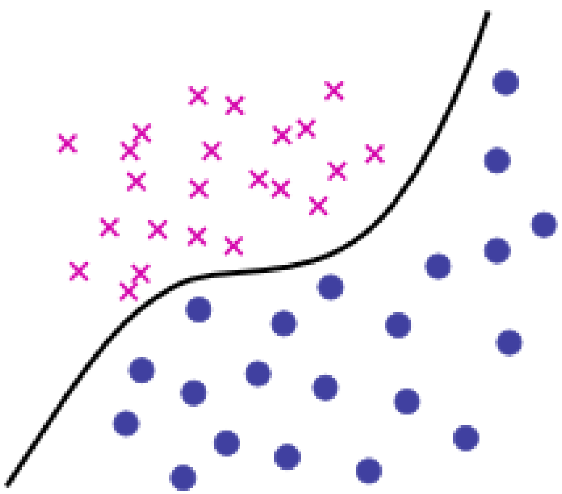
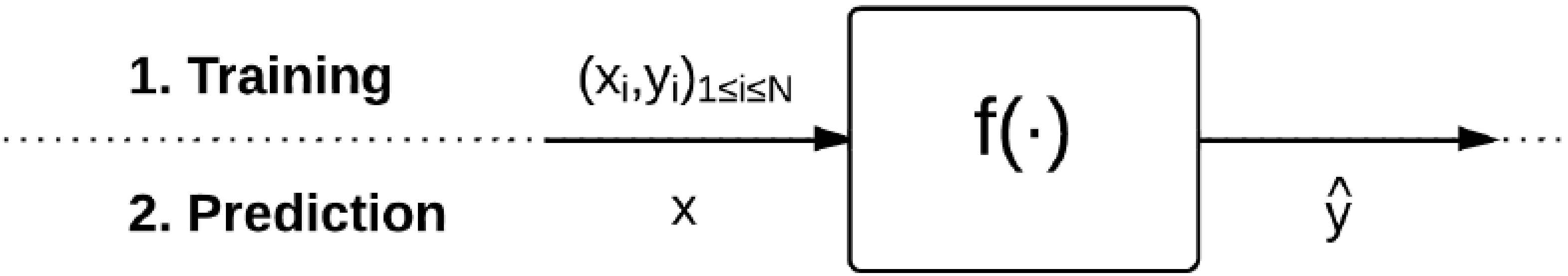


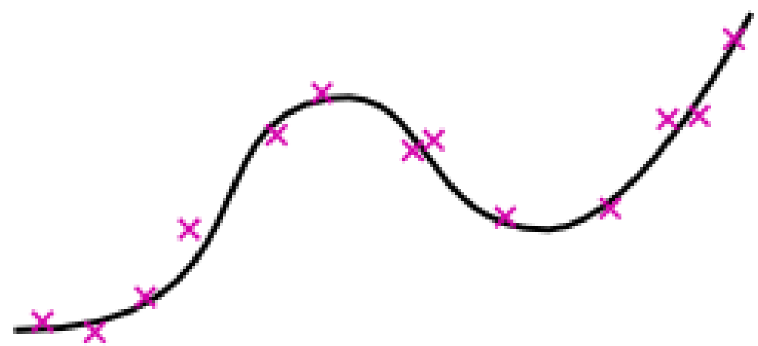
Figure 11: Classification process



Supervised Learning: Regression



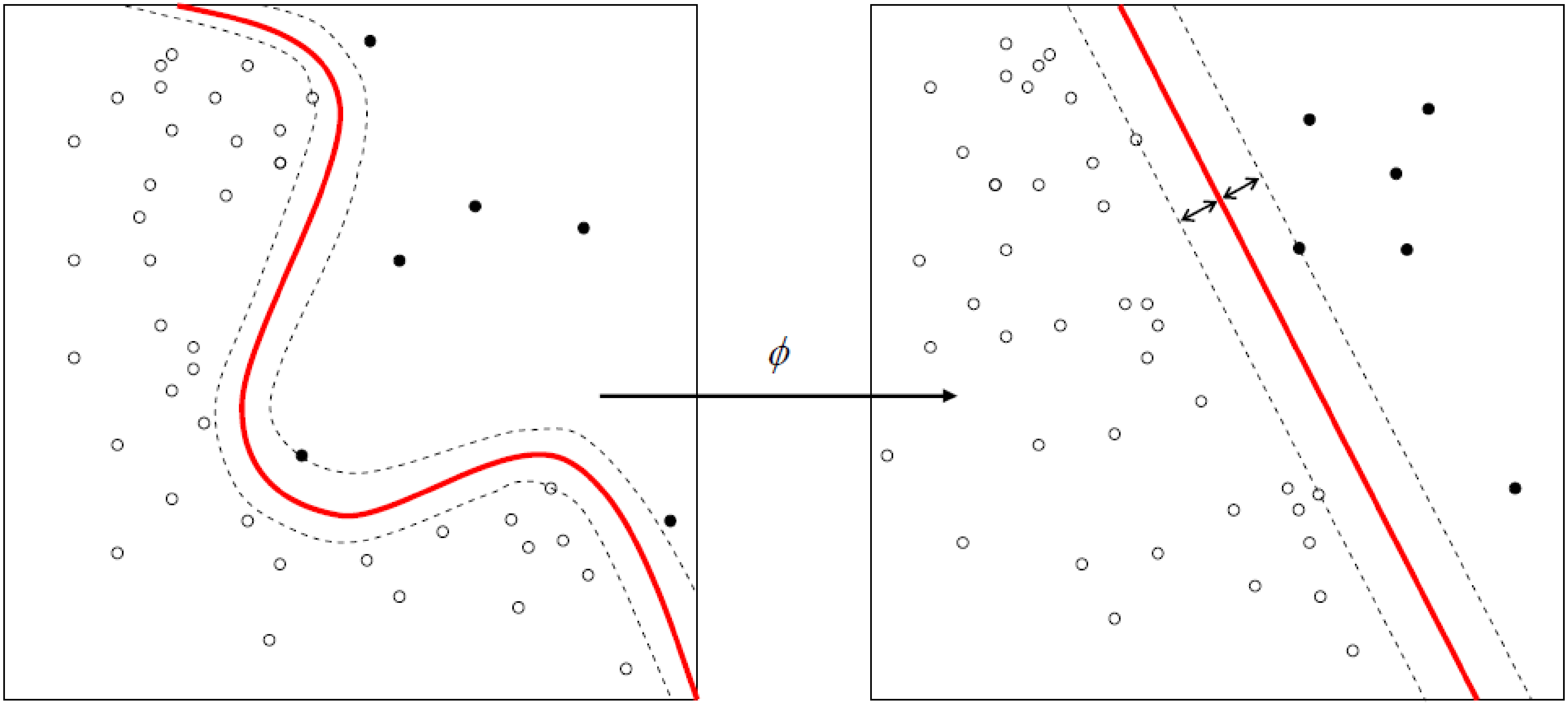
y is a numerical or vector variable
(classification → regression)



Supervised Learning: Models

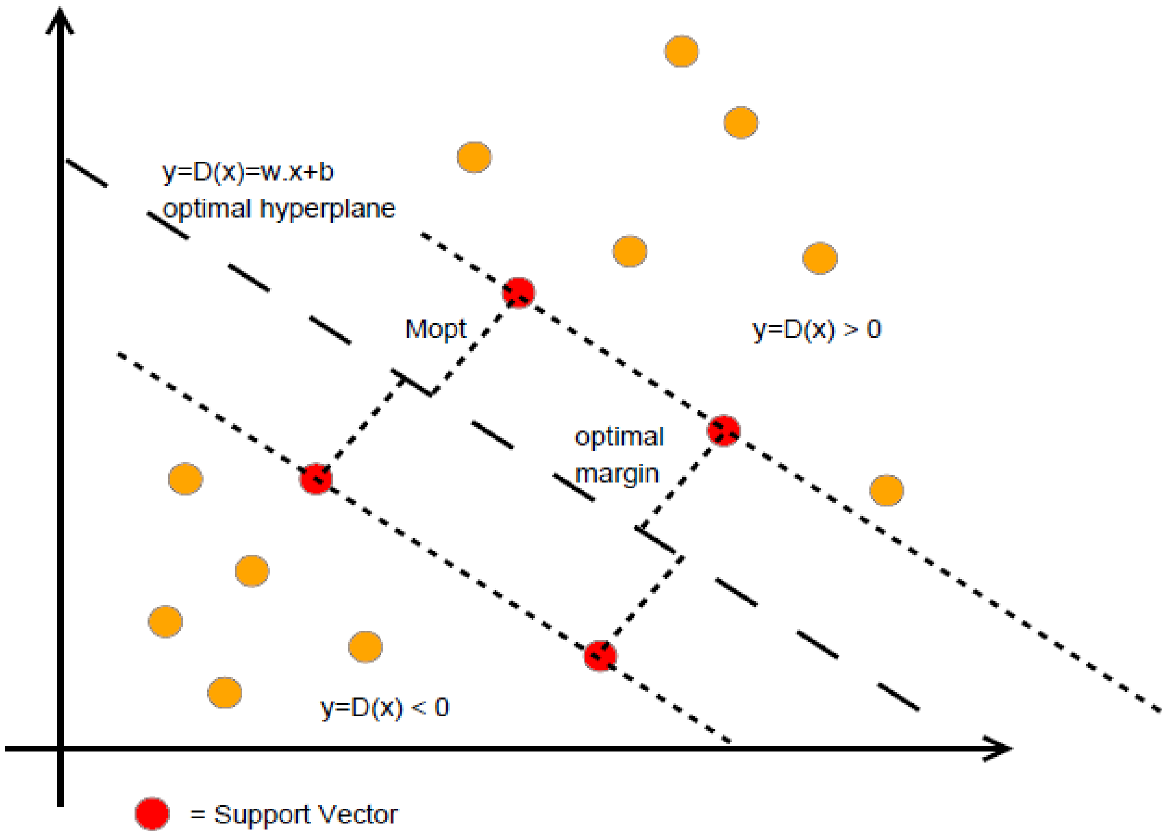
- Linear
- Logistic, sigmoid
- Bayesian classifiers (naive and general)
- → Decision trees
- → Neural networks
- → Support vector machines
- Kernel learning
- Relevance vector machines
- …

Support Vector Machines (1/2)



Embedding of the data set in a representation (« feature ») space
→ computation of a linear separating hyperplane

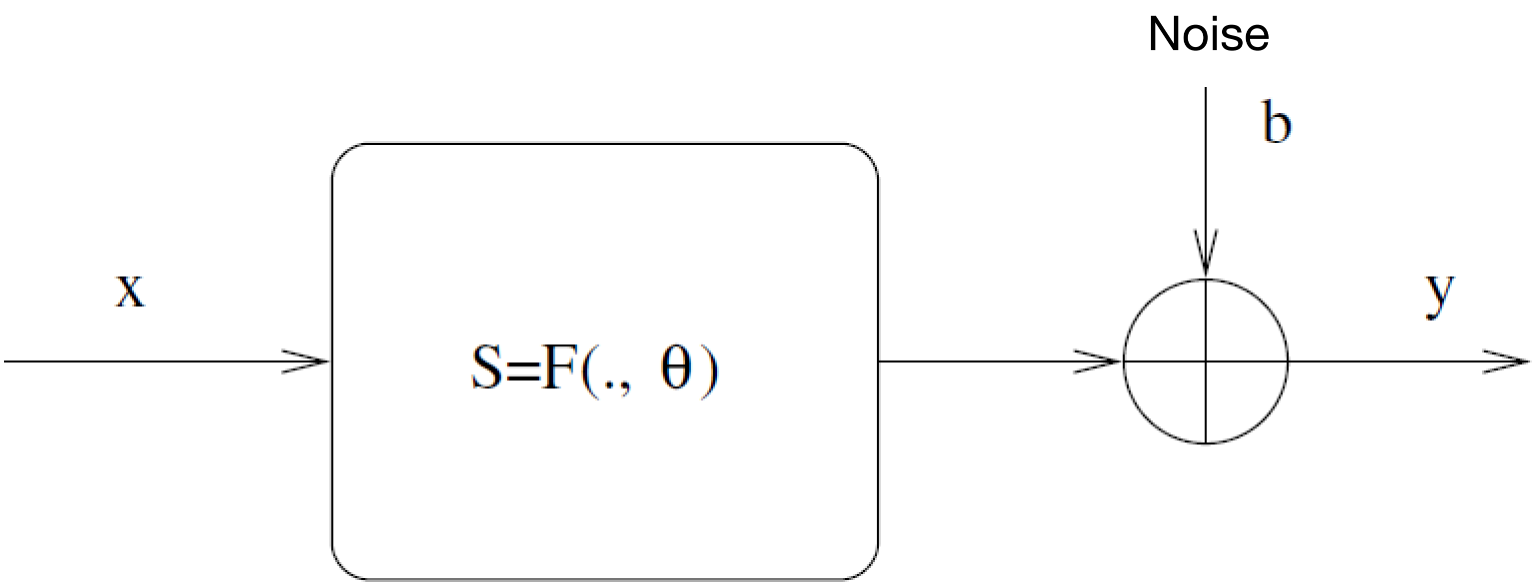
Support Vector Machines (2/2)



→ computation of the optimal separating hyperplane with maximum margin
→ typically via quadratic programming techniques

Markov Chain Monte Carlo
Sequential Monte Carlo

Bayesian Estimation



Information on (x, θ): distribution of probability

$$p(x, \theta | y, F, prior) \propto p(y | x, \theta, F, prior) \times p(x, \theta | prior)$$

$$\Rightarrow \text{ Estimates } (\widehat{x}, \widehat{\theta})$$

Bayesian Estimates

- *Maximum a posteriori* (MAP)

$$(\widehat{x}, \widehat{\theta}) = \arg\max_{x,\theta} p(x, \theta | y, prior)$$

- Expectation: *posterior* mean $\mathrm{E}\{x, \theta | y, prior\}$

$$\mathrm{E}_{p(.|y,prior)}\{f(x,\theta)\} = \int f(x,\theta) p(x,\theta|y, prior) d(x,\theta)$$

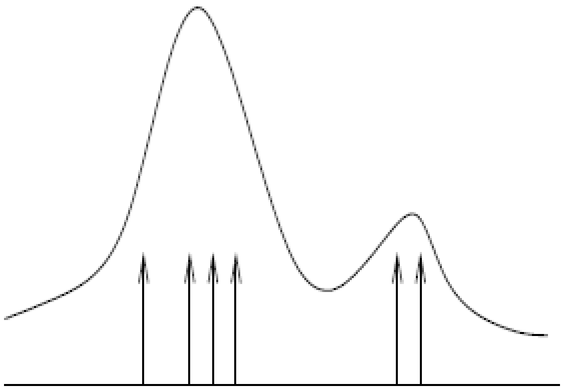
Computation : asymptotic, numerical, stochastic methods

$\Rightarrow$ Monte Carlo simulation methods

Monte Carlo Estimates

$$x_1, \ldots, x_N \sim \pi$$

$$\Rightarrow \widehat{\pi}_N = \frac{1}{N} \sum_{n=1}^{N} \delta_{x_n}$$



$$\widehat{S}_N(f) = \frac{1}{N} \sum_{n=1}^{N} f(x_n) \longrightarrow \int f(x)\pi(x)dx = \mathrm{E}_\pi\{f\}$$

$$\widehat{x}_{max} = \arg\max_{x_n} \widehat{\pi}_N \text{ approximates } x_{max} = \arg\max_{x} \pi(x)$$

$$\Rightarrow \text{generate samples } x_\ell \sim \pi \text{ ?}$$

$\rightarrow$ Markov chain and sequential Monte Carlo

Simulation Techniques

- Classical distributions : cumulated density function
  $\longrightarrow$ transformation of uniform random variable

- Non-standard distributions, $\mathbb{R}^n$, known up to a normalizing constant $\longrightarrow$ usage of instrumental distribution:

  Accept-reject, importance sampling $\longrightarrow$ sequential/recursive
  $\Rightarrow$ SMC aka particle filtering, condensation algorithm

  $\Rightarrow$ MCMC : distribution = fixed point of an operator

$$\pi = K\pi$$

$\longrightarrow$ simulation schemes with Markov chain: Hastings-Metropolis, Gibbs sampling

Markov Chain

Definition:

$$X_n | X_{n-1}, X_{n-2}, \ldots, X_0 \stackrel{d}{=} X_n | X_{n-1}$$

$$\text{homogeneity} : X_n | X_{n-1} \text{ independent of } n$$

Realization:

$$X_0 \sim \pi_0(x_0)$$

$$p.d.f. \text{ of } X_n | X_{n-1} = \text{transition kernel } K(x_n | x_{n-1})$$

Simulation of a Markov Chain

Convergence: $X_n \sim \pi$ asymptotically ?

$$\pi\text{-invariance} : \pi(.) = K\pi(.)$$

$$\int_A \pi(x)dx = \int_{y \in A} \int K(y|x)\pi(x)dxdy$$

$$\Leftarrow \pi\text{-reversibility} : Pr(A \to B) = Pr(B \to A)$$

$$\int_{y \in B} \int_{x \in A} K(y|x)\pi(x)dxdy = \int_{y \in A} \int_{x \in B} K(y|x)\pi(x)dxdy$$

Construct kernels $K(.|.)$ such that the chain is π-invariant

- Hastings-Metropolis algorithm
- Gibbs sampling

Hastings-Metropolis algorithm (1/2): scheme

Draw $\boldsymbol{x}$ from $\pi(.)$

1. `initialize` $\boldsymbol{x}_0 \sim \pi_0(\boldsymbol{x})$
2. `Iteration` ℓ
   - `propose candidate` $\boldsymbol{x}^\star$ `for` $\boldsymbol{x}_{\ell+1} \quad \rightarrow \boldsymbol{x}^\star \sim q(\boldsymbol{x}|\boldsymbol{x}_\ell)$
   - `accept it with prob` $\alpha = \min\{1, r\}$
3. $\ell \leftarrow \ell + 1$ `and go to (2)`

$$r = \frac{\pi(\boldsymbol{x}^\star)q(\boldsymbol{x}_\ell|\boldsymbol{x}^\star)}{q(\boldsymbol{x}^\star|\boldsymbol{x}_\ell)\pi(\boldsymbol{x}_\ell)} \rightarrow \pi(x)K(y|x) = \pi(y)K(x|y)$$

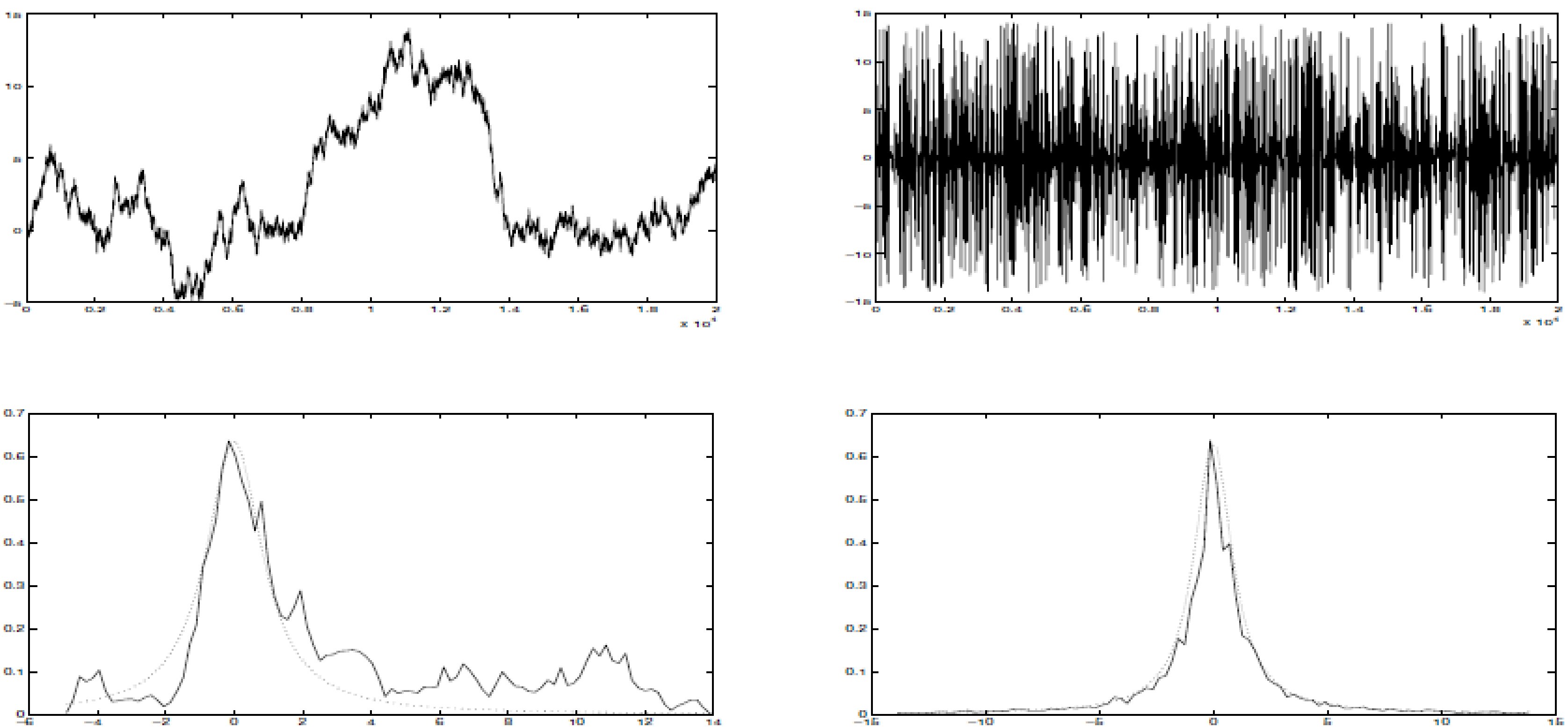
$$\pi(x)q(y|x)\min\left\{1, \frac{\pi(y)q(x|y)}{q(y|x)\pi(x)}\right\} = \min\left\{\pi(x)q(y|x), \pi(y)q(x|y)\right\}$$

$$q(\boldsymbol{x}^\star|\boldsymbol{x}_\ell) = q(\boldsymbol{x}^\star) \quad q(\boldsymbol{x}^\star|\boldsymbol{x}_\ell) = q(|\boldsymbol{x}^\star - \boldsymbol{x}_\ell|)$$

Hastings-Metropolis algorithm (2/2): example

sample $x \sim p(x) \propto \frac{1}{1+x^2}$ 20,000 iterations

$x^\star \sim \mathcal{N}(x_\ell, 0.1^2)$ $x^\star \sim \mathcal{U}_{[a,b]}$



acc. rate = 97% acc. rate = 26%

Gibbs Sampling algorithm (1/2): scheme

Sample $\boldsymbol{x} = (x_1, ...x_p) \sim \pi(x_1, ...x_p)$

1. initialize $\boldsymbol{x}^{(0)} \sim \pi_0(\boldsymbol{x})$, $\ell = 0$

2. iteration ℓ : Sample

$$
\begin{aligned}
x_1^{(\ell+1)} &\sim \pi_1(x_1 | x_2^{(\ell)}, \ldots, x_p^{(\ell)}) \\
x_2^{(\ell+1)} &\sim \pi_2(x_2 | x_1^{(\ell+1)}, x_3^{(\ell)}, \ldots, x_p^{(\ell)}) \\
&\vdots \\
x_p^{(\ell+1)} &\sim \pi_p(x_p | x_1^{(\ell+1)}, \ldots, x_{p-1}^{(\ell+1)})
\end{aligned}
$$

3. $\ell \leftarrow \ell + 1$ and go to (2)

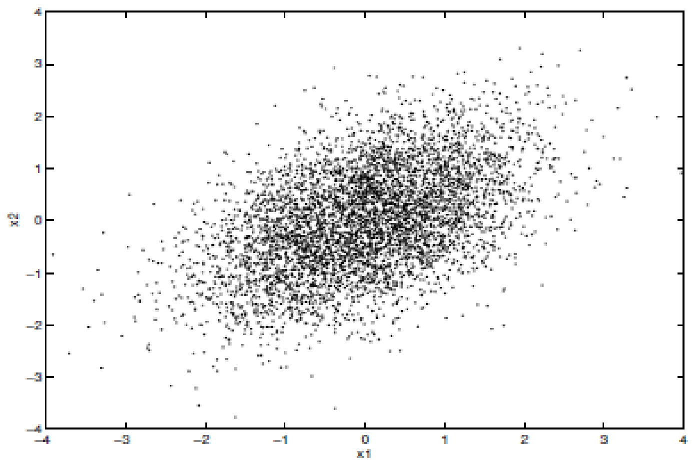
$\rightarrow$ no rejection, reversible kernel

Gibbs Sampling algorithm (2/2): example

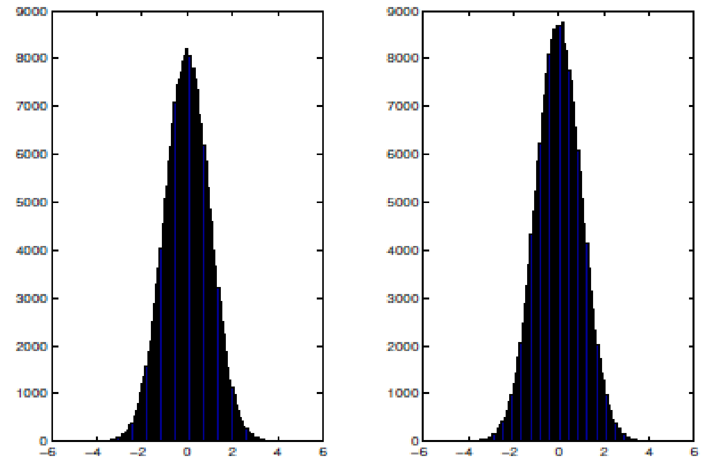
$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$$

$$x_1^{(\ell+1)} | x_2^{(\ell)} \sim \mathcal{N}\left(\rho x_2^{(\ell)}, 1-\rho^2\right)$$

$$x_2^{(\ell+1)} | x_1^{(\ell+1)} \sim \mathcal{N}\left(\rho x_1^{(\ell+1)}, 1-\rho^2\right)$$



5,000 samples, ρ=0.5



histograms (x_1^ℓ, x_2^ℓ)

Improving convergence of simulation techniques

How to obtain fast converging simulation scheme ?

$\rightarrow$ **Missing Data, Data Augmentation, Latent Variables**

Idea : extend sampling space $x \rightarrow (x, z)$ and distribution $\pi(x) \rightarrow \widetilde{\pi}(x, z)$ with constraint

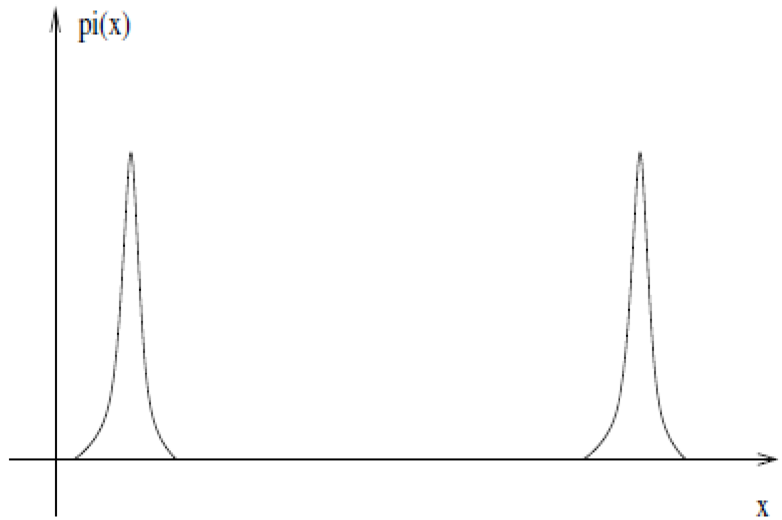
$$\int \widetilde{\pi}(x, z)dz = \pi(x)$$

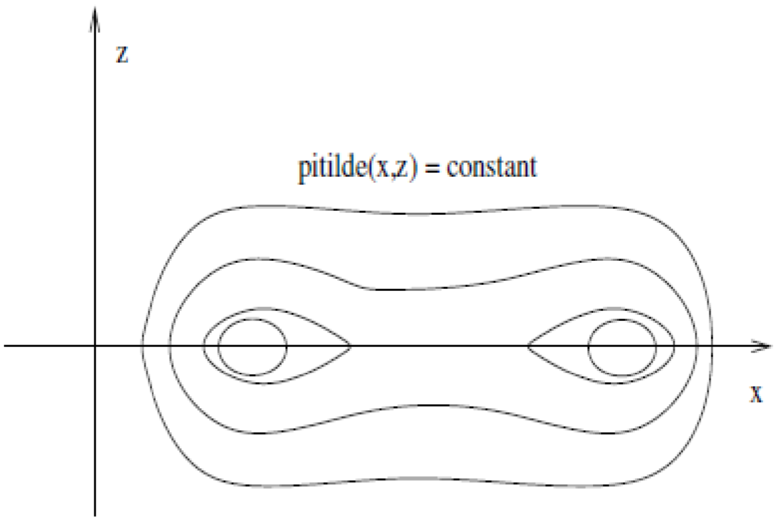
such that Markov chain $(x^{(i)}, z^{(i)}) \sim \widetilde{\pi}$ faster

- Optimization : Expectation-Maximization (EM) algorithm
- Simulation : Data Augmentation, Gibbs sampling

Efficient Data Augmentation Schemes

Idea: construct missing data space as less informative as possible





$x \sim \pi(x)$ $\qquad$ $(x, z) \sim \widetilde{\pi}(x, z)$

Information introduced in missing data $\searrow$: convergence $\nearrow$

Estimation of State Space Models

$$x_t = f_t(x_{t-1}, u_t) \qquad y_t = g_t(x_t, v_t)$$

$$p(x_{0:t}|y_{1:t}) \quad \rightarrow \quad p(x_t|y_{1:t}) = \int p(x_{0:t}|y_{1:t}) dx_{0:t-1}$$

distribution of $x_{0:t} \Rightarrow$ computation of estimate $\widehat{x}_{0:t}$:

$$\widehat{x}_{0:t} = \int x_{0:t} p(x_{0:t}|y_{1:t}) dx_{0:t} \rightarrow \mathrm{E}_{p(.|y_{1:t})}\{f(x_{0:t})\}$$

$$\widehat{x}_{0:t} = \arg\max_{x_{0:t}} p(x_{0:t}|y_{1:t})$$

Computation of the estimates

$p(x_{0:t}|y_{1:t}) \Rightarrow$ multidimensionnal, non-standard distributions:

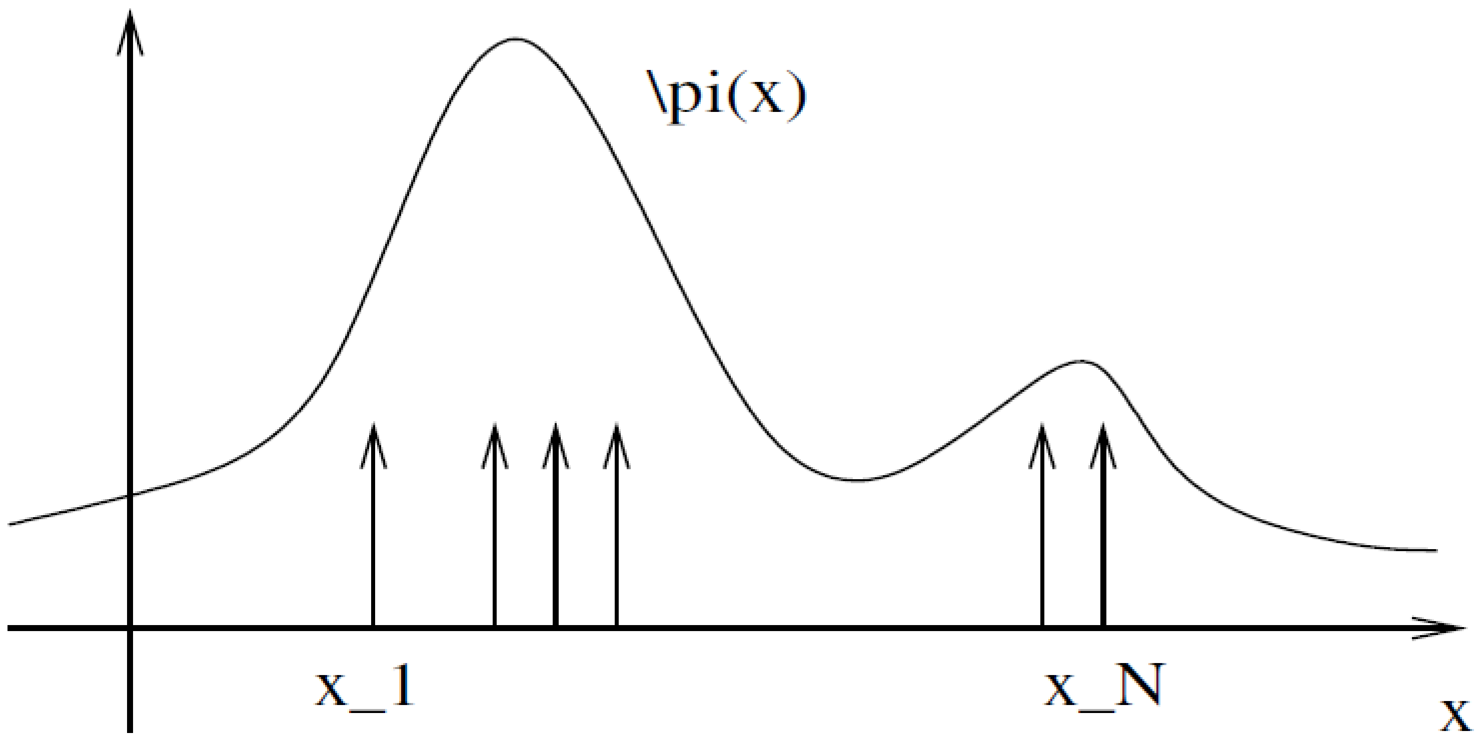
$\longrightarrow$ analytical, numerical approximations

$\longrightarrow$ integration, optimisation methods

$\Rightarrow$ Monte Carlo techniques

Monte Carlo Approach

compute estimates for distribution $\pi(.) \rightarrow$ samples $x_1, \ldots, x_N \sim \pi$



$\Rightarrow$ distribution $\widehat{\pi}_N = \frac{1}{N} \sum_{i=1}^{N} \delta_{x_i}$ approximates $\pi(.)$

Monte Carlo Estimates

$$\widehat{S}_N(f) = \frac{1}{N}\sum_{i=1}^{N} f(x_i) \longrightarrow \int f(x)\pi(x)dx = \mathrm{E}_\pi\{f(x)\}$$

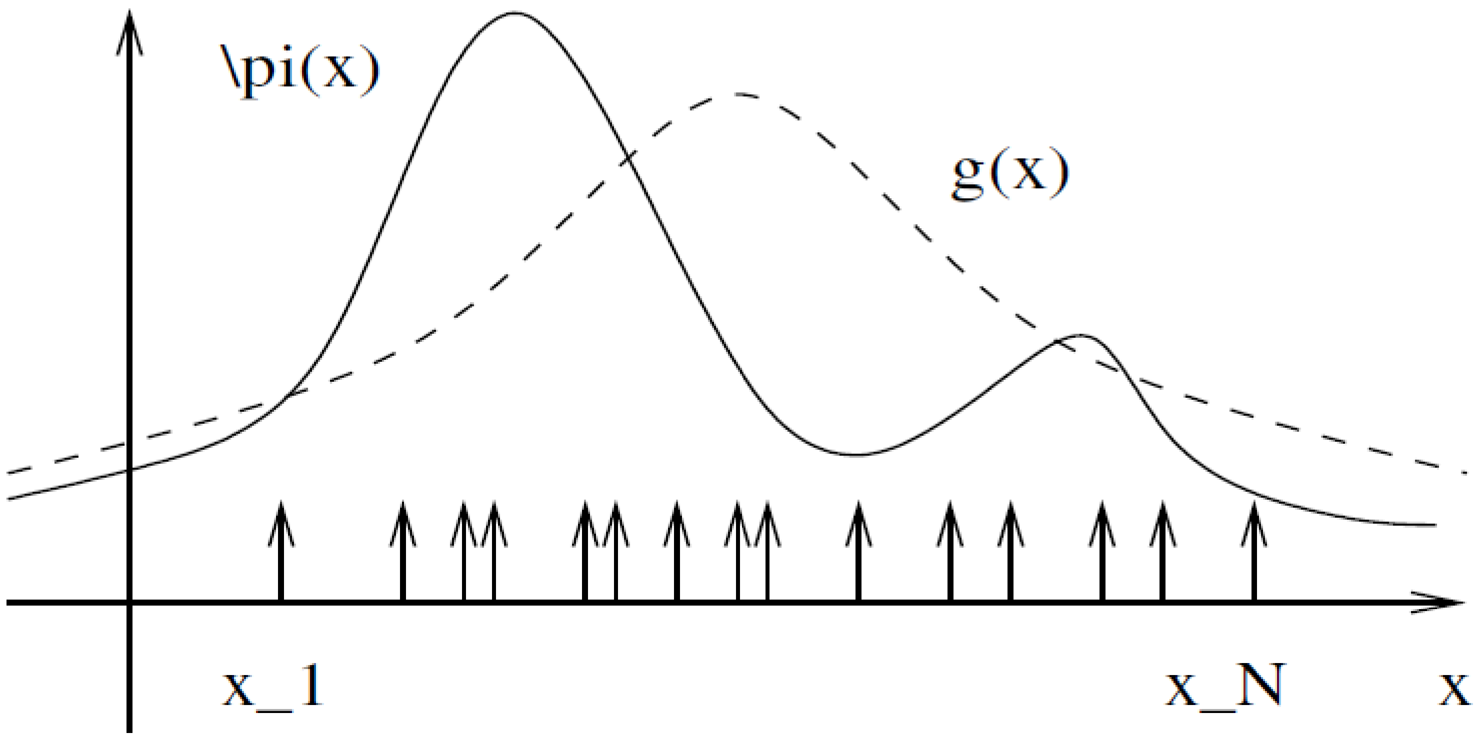
$\arg\max_{(x_i)_{1\leq i\leq N}} \widehat{\pi}_N(x_i)$ approximates $\arg\max_x \pi(x)$

$\Rightarrow$ sampling $x_i \sim \pi$ difficult

$\longrightarrow$ importance sampling techniques

Importance Sampling (1/2)

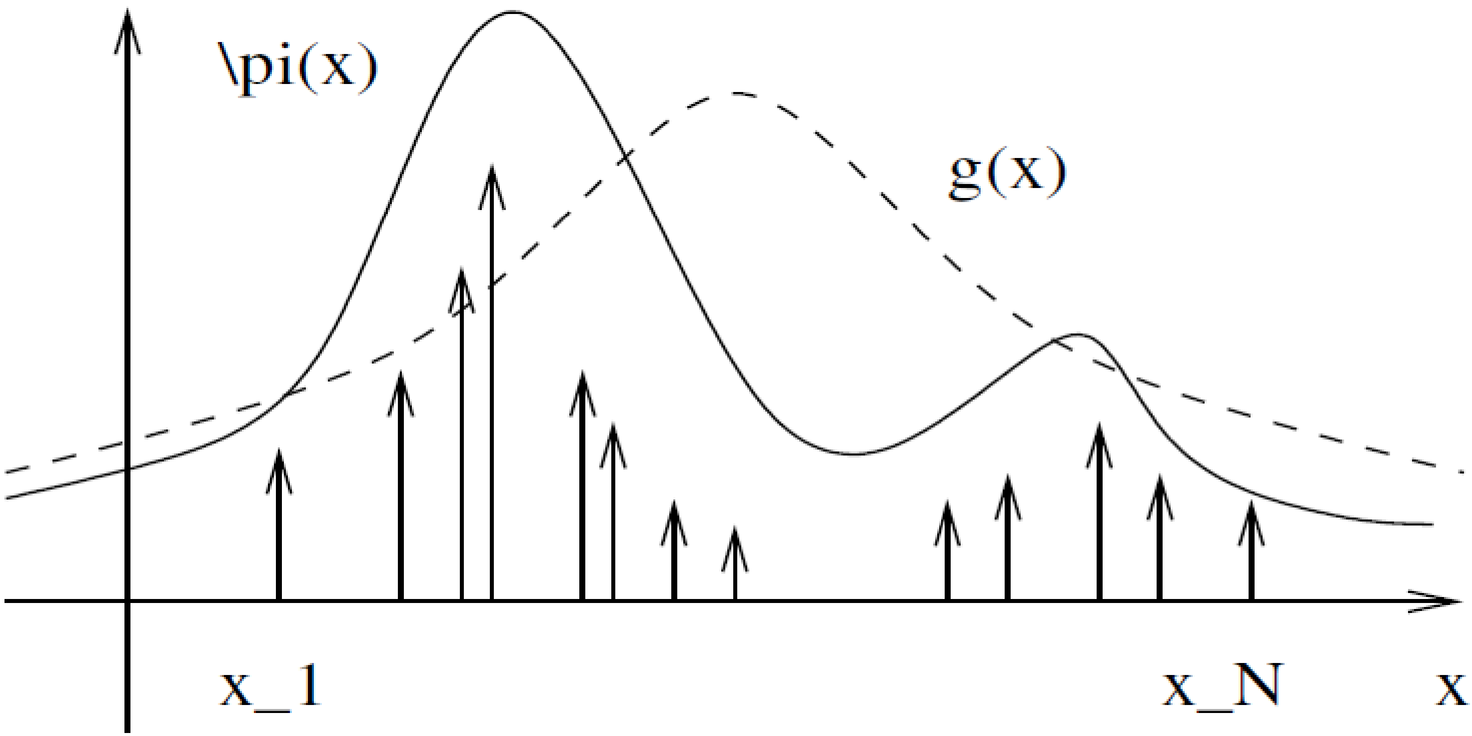
$x_i \sim \pi \rightarrow$ candidate/proposal distribution $x_i \sim g$



Importance Sampling (2/2)

$$x_i \sim g \neq \pi \rightarrow (x_i, w_i) \text{ weighted sample}$$

$$\Rightarrow \text{ weight } w_i = \frac{\pi(x_i)}{g(x_i)}$$



Estimation

importance sampling $\rightarrow$ computation of Monte Carlo estimates

e. g. expectations $\mathrm{E}_\pi\{f(x)\}$:

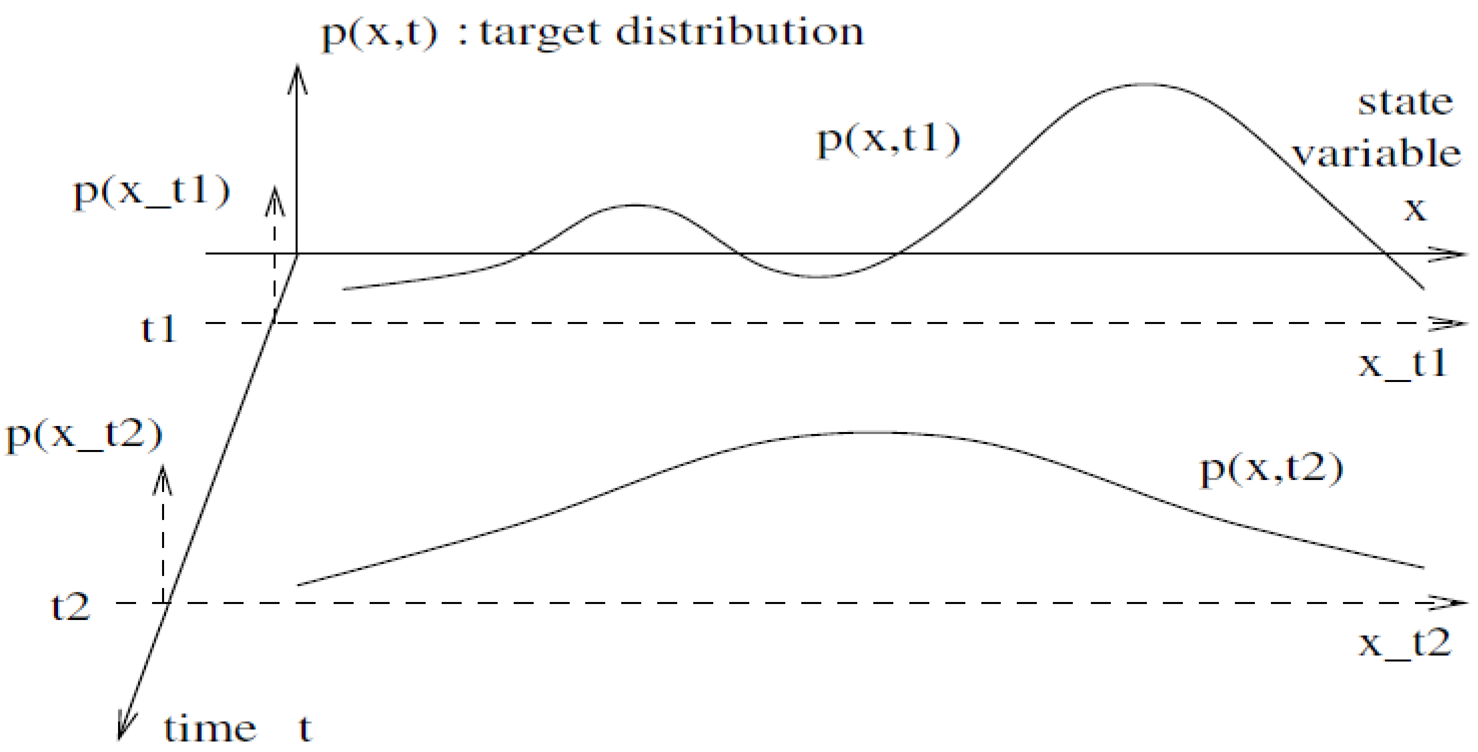
$$\begin{aligned} \int f(x)\frac{\pi(x)}{g(x)}g(x)dx &= \int f(x)\pi(x)dx \\ \sum_{i=1}^{N} w_i f(x_i) &\rightarrow \int f(x)\pi(x)dx = \mathrm{E}_\pi\{f(x)\} \end{aligned}$$

dynamic model $(x_t, y_t) \Rightarrow$ recursive estimation $\widehat{x}_{0:t-1} \rightarrow \widehat{x}_{0:t}$

Monte Carlo techniques $\Rightarrow$ sampling sequences $x^{(i)}_{0:t-1} \rightarrow x^{(i)}_{0:t}$

Sequential Simulation

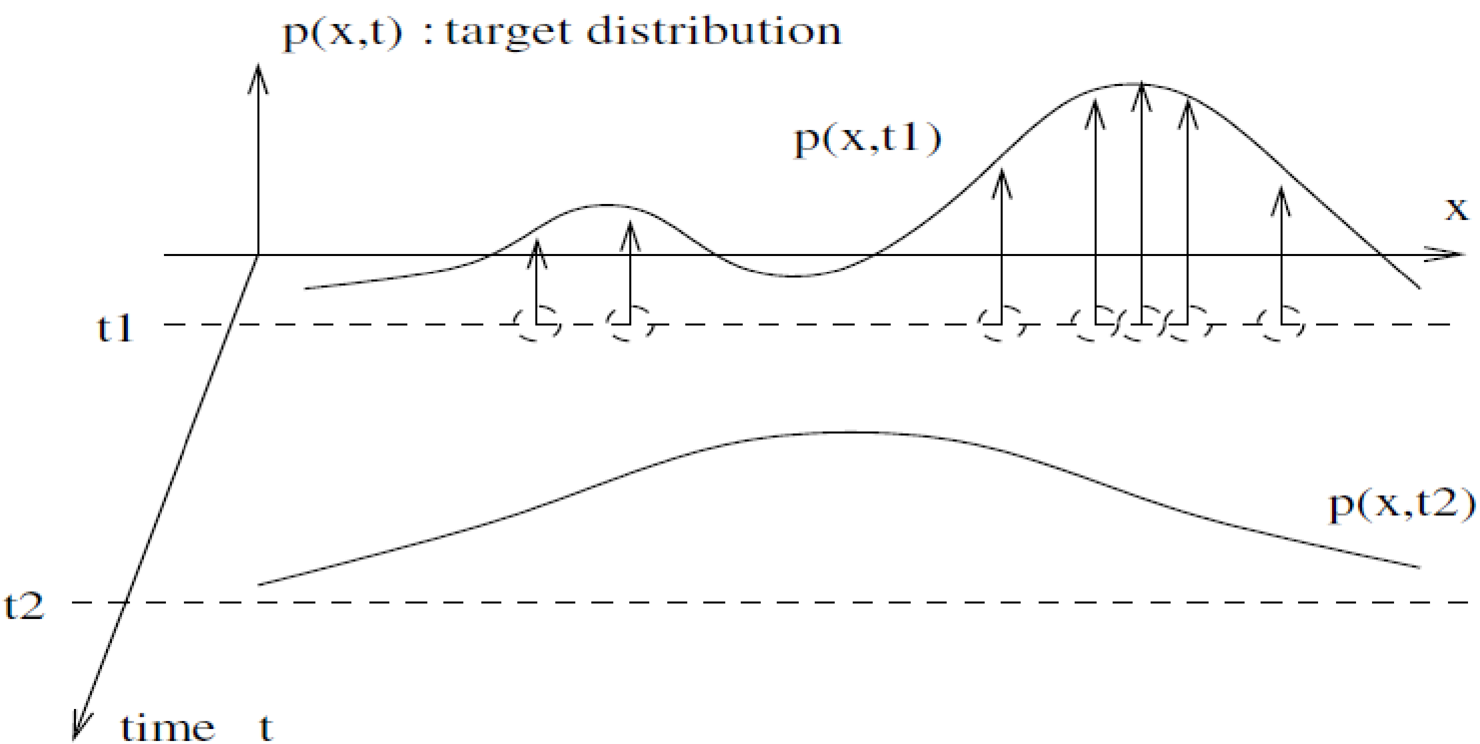
sampling sequences $x_{0:t}^{(i)} \sim \pi_t(x_{0:t})$ recursively:



Sequential Simulation: Importance Sampling

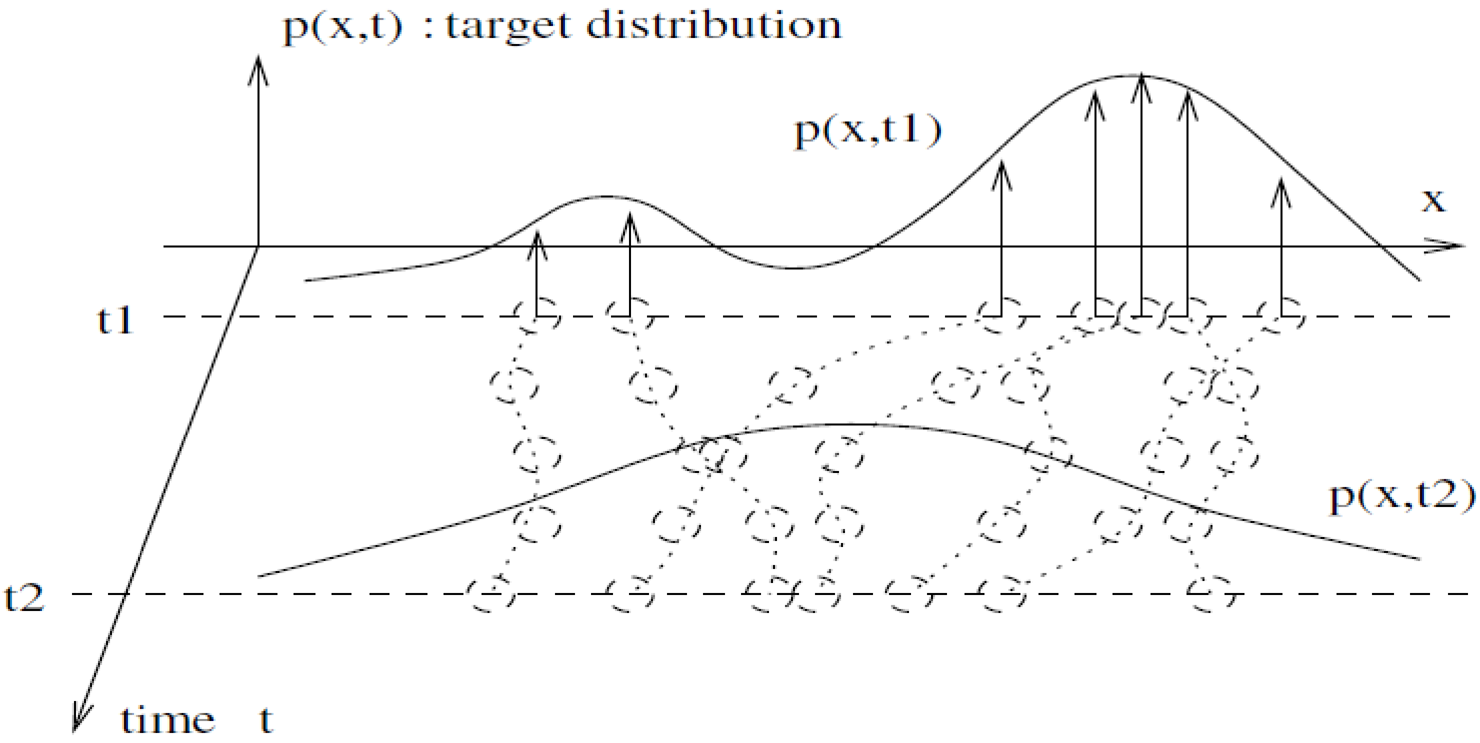
samples $x_{0:t}^{(i)} \sim \pi_t(x_{0:t})$ approximated by weighted particles

$$(x_{0:t}^{(i)}, w_t^{(i)})_{1 \leq i \leq N}$$



Sequential Importance Sampling (1/2)

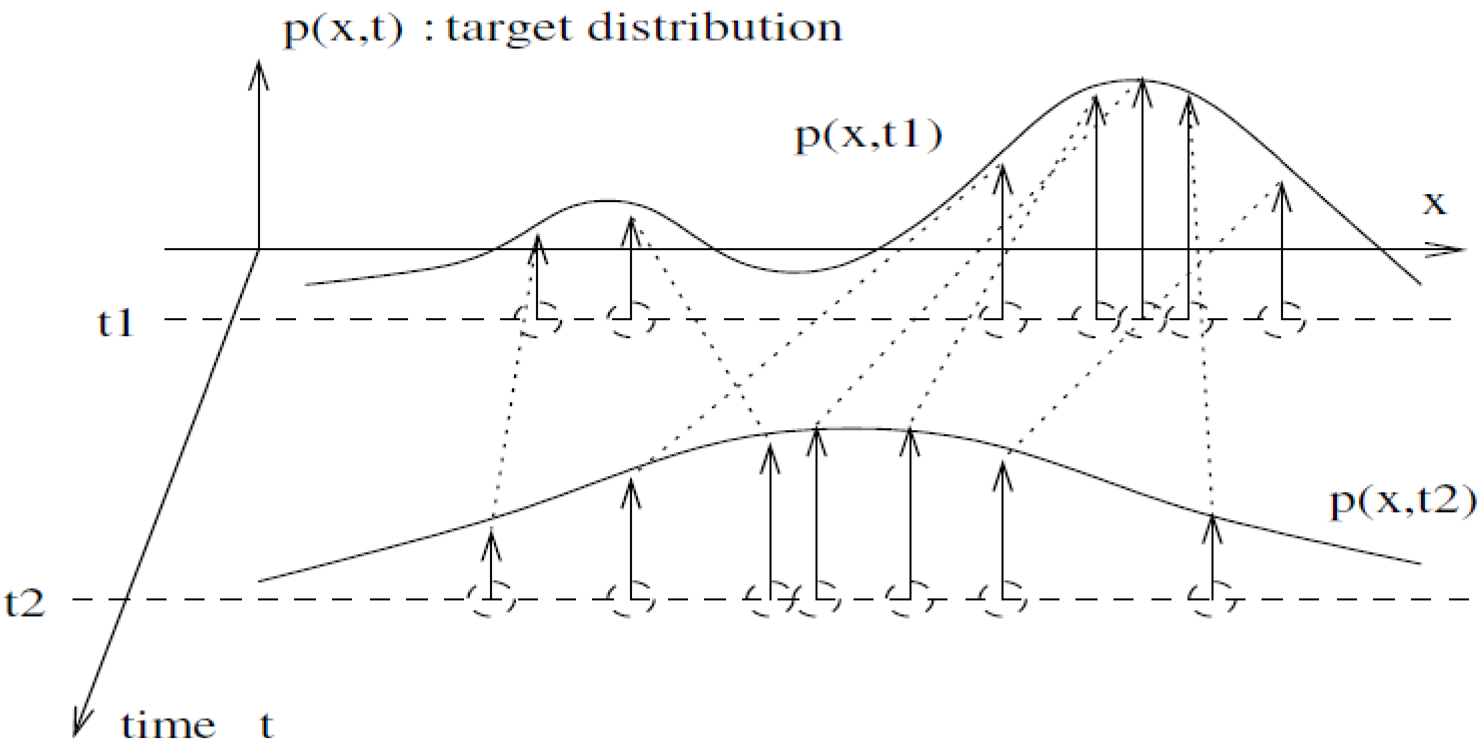
diffusing particles $x_{0:t_1}^{(i)} \rightarrow x_{0:t_2}^{(i)}$



$\Rightarrow$ sampling scheme $x_{0:t-1}^{(i)} \rightarrow x_{0:t}^{(i)}$

Sequential Importance Sampling (2/2)

updating weights $w_{t_1}^{(i)} \rightarrow w_{t_2}^{(i)}$



$$\Rightarrow \text{ updating rule } w_{t-1}^{(i)} \rightarrow w_t^{(i)}$$

Sequential Importance Sampling Scheme

$$x_{0:t} \sim \pi_t(x_{0:t}) \Rightarrow (x_{0:t}^{(i)}, w_t^{(i)})_{1 \le i \le N}$$

Simulation scheme $t-1 \rightarrow t$:

- Sampling step $x_t^{(i)} \sim q_t(x_t | x_{0:t-1}^{(i)})$

- Updating weights

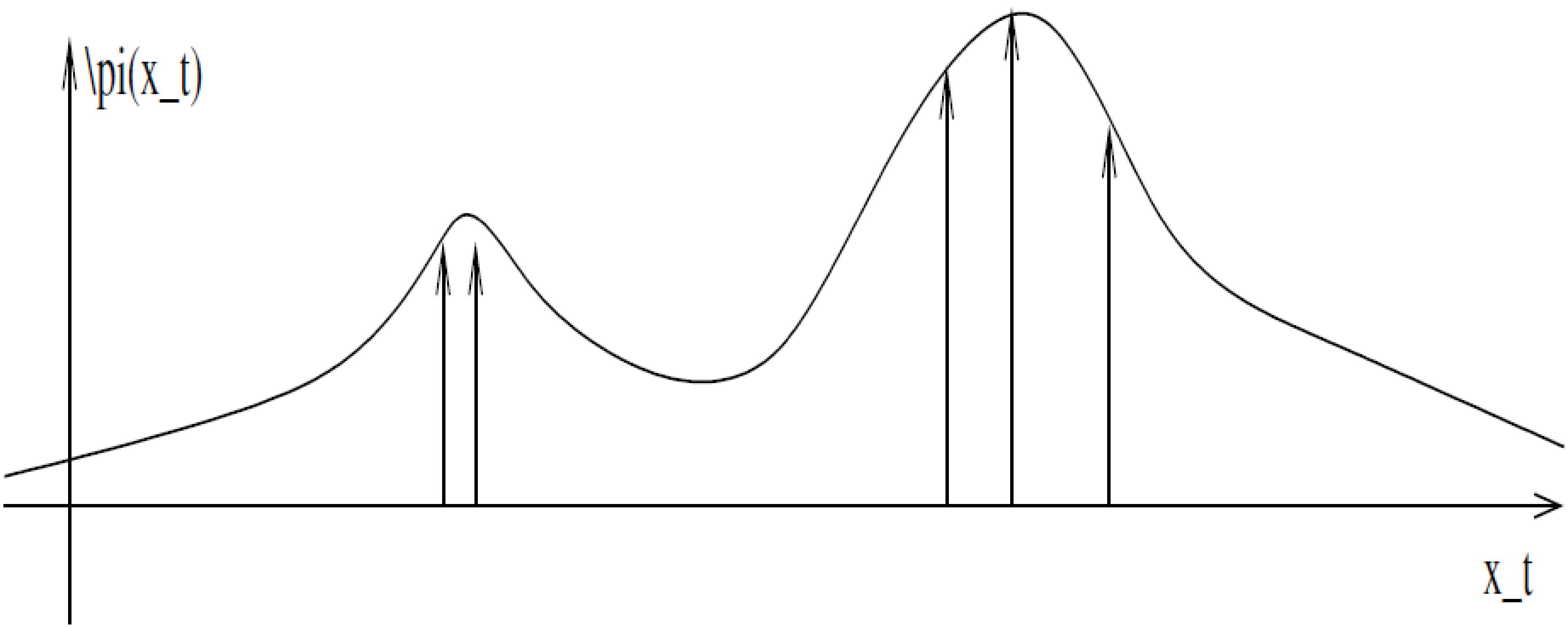
$$w_t^{(i)} \propto w_{t-1}^{(i)} \times \underbrace{\frac{\pi_t(x_{0:t-1}^{(i)}, x_t^{(i)})}{\pi_{t-1}(x_{0:t-1}^{(i)}) q_t(x_t^{(i)} | x_{0:t-1}^{(i)})}}_{\text{incremental weight (iw)}}$$

normalizing $\sum_{i=1}^{N} w_t^{(i)} = 1$

Sequential Importance Sampling Issue (1/2)

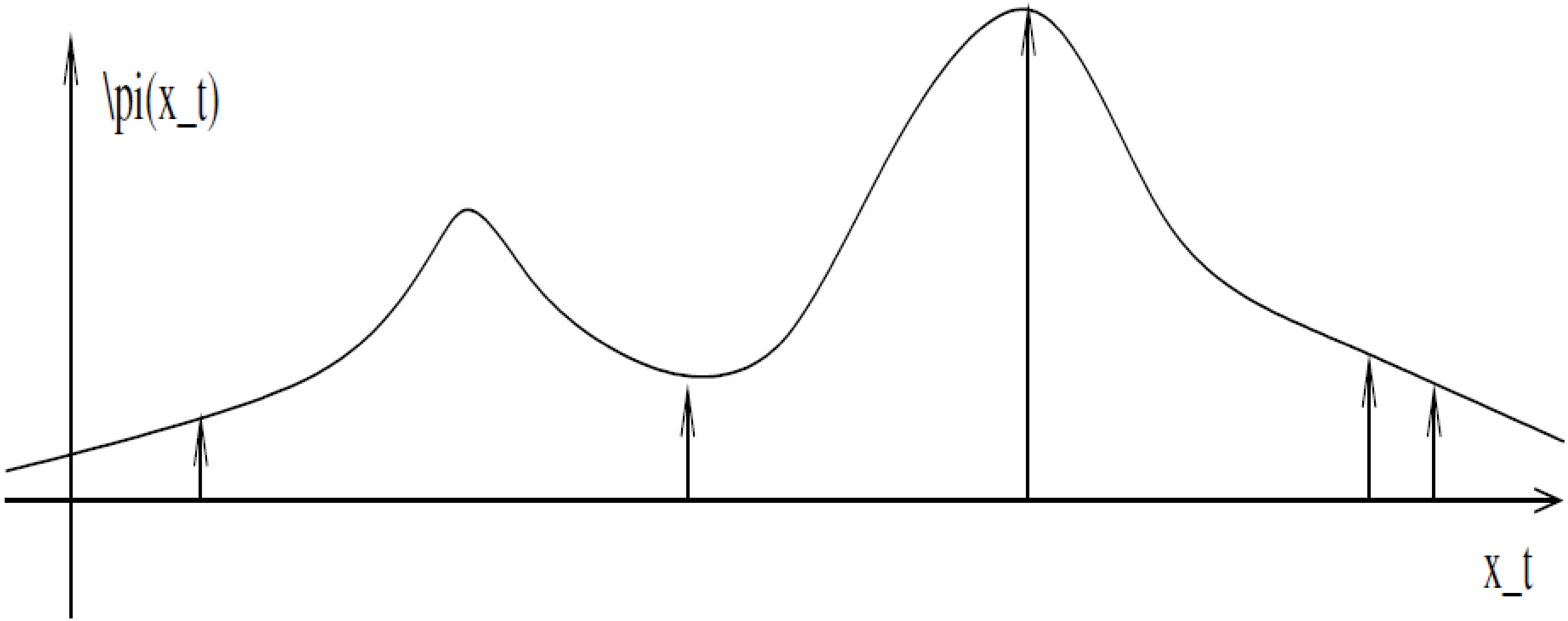
$$x_{0:t} \sim \pi_t(x_{0:t}) \Rightarrow (x_{0:t}^{(i)}, w_t^{(i)})_{1 \leq i \leq N}$$

$$\text{proposal} + \text{reweighting} \rightarrow$$



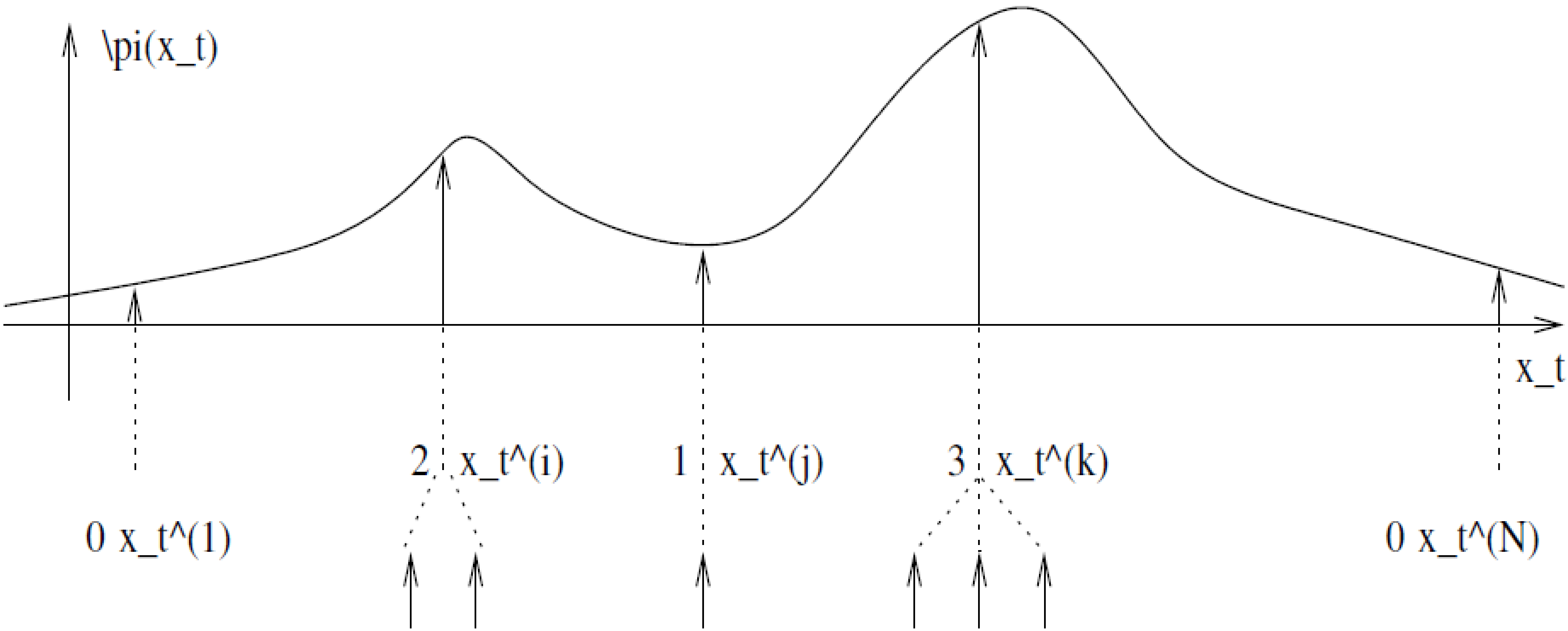
Sequential Importance Sampling Issue (2/2)

$$\text{proposal} + \text{reweighting} \rightarrow \text{var}\{(w_t^{(i)})_{1\leq i\leq N}\} \nearrow \text{with } t$$



$$\rightarrow w_t^{(i)} \approx 0 \text{ for all } i \text{ except one}$$

→Resampling



$$\rightarrow \text{draw } N \text{ particles paths from the set } (x_{0:t}^{(i)})_{1\leq i\leq N}$$
$$\text{with probability } (w_t^{(i)})_{1\leq i\leq N}$$

Sequential Importance Sampling/Resampling Scheme

Simulation scheme $t-1 \to t$:

- Sampling step $x_t'^{(i)} \sim q_t(x_t' | x_{0:t-1}^{(i)})$

- Updating weights $w_t^{(i)} \propto w_{t-1}^{(i)} \times \frac{\pi_t(x_{0:t-1}^{(i)}, x_t'^{(i)})}{\pi_{t-1}(x_{0:t-1}^{(i)}) q_t(x_t'^{(i)} | x_{0:t-1}^{(i)})}$

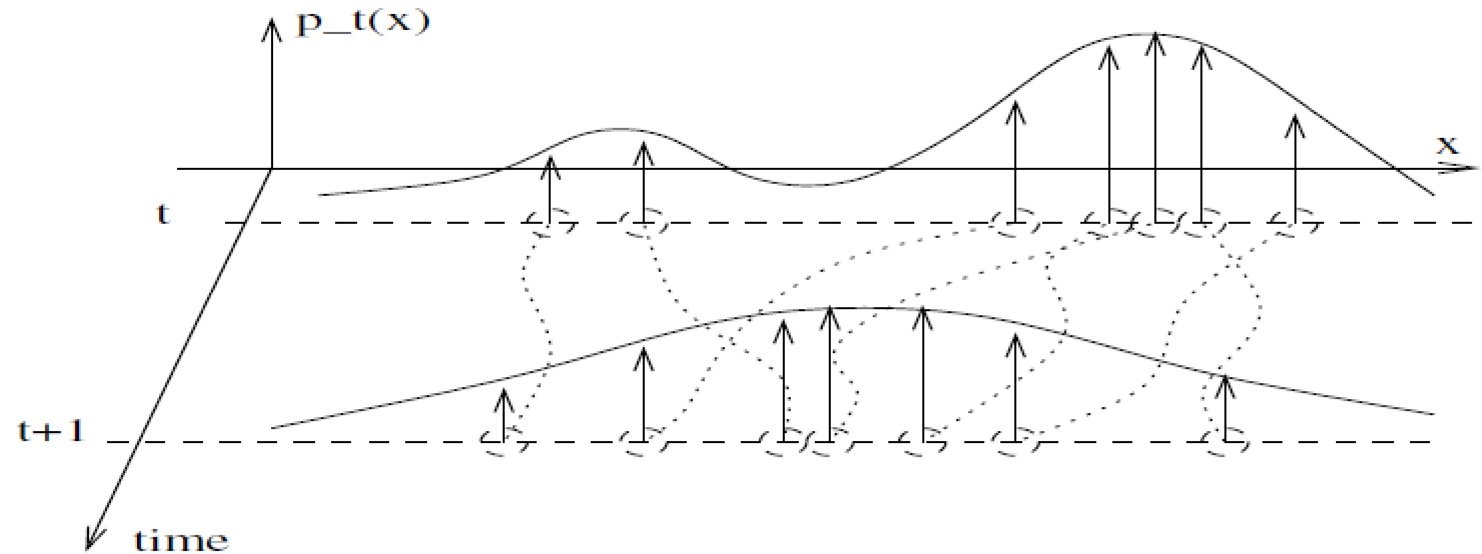
  $\longrightarrow$ parallel computing

- $\Rightarrow$ **Resampling step**: sample N paths from $(x_{0:t-1}^{(i)}, x_t'^{(i)})_{1 \leq i \leq N}$
  $\longrightarrow$ particles interacting : computation at least $O(N)$

Sequential simulation: SISR

Recursive estimation of state space models.

Approximation with particles, importance sampling.



Bootstrap, particle filtering

Gordon et al. 1993, Kitagawa 1996, Doucet et al. 2001

$\longrightarrow$ time series, tracking.

Sequential Importance Sampling Resampling (SISR)

Samples $x_{0:t}^{(i)} \sim \pi_t(x_{0:t})$ approximated by
weighted particles $(x_{0:t}^{(i)}, w_t^{(i)})_{1 \leq i \leq N}$

Simulation scheme $t-1 \to t$:

- Sampling step $x_t'^{(i)} \sim q_t(x_t' | x_{0:t-1}^{(i)})$

- Updating weights $w_t^{(i)} \propto w_{t-1}^{(i)} \times \underbrace{\dfrac{\pi_t(x_{0:t-1}^{(i)}, x_t'^{(i)})}{\pi_{t-1}(x_{0:t-1}^{(i)}) q_t(x_t'^{(i)} | x_{0:t-1}^{(i)})}}_{\text{incremental weight (iw)}}$

- Resampling step: sample N paths from $(x_{0:t-1}^{(i)}, x_t'^{(i)})_{1 \leq i \leq N}$

SISR for Recursive Estimation of State Space Models

$$x_t = f_t(x_{t-1}, u_t) \quad \longrightarrow \quad p(x_t|x_{t-1})$$
$$y_t = g_t(x_t, v_t) \quad \longrightarrow \quad p(y_t|x_t)$$

Usual SISR: Bootstrap filter (Gordon et al. 93, Kitagawa 96):

- Sampling step $x_t^{(i)} \sim p(x_t|x_{t-1}^{(i)})$
- Updating weights : incremental weight $w_t^{(i)} \propto w_{t-1}^{(i)} \times iw$

$$iw \propto p(y_t|x_t^{(i)})$$

- Stratified/Deterministic resampling

efficient, easy, fast for a wide class of models

tracking, time series $\longrightarrow$ nonlinear non-Gaussian state spaces

Improving Simulation

Optimal proposal distribution $q_t(x_t|x_{0:t-1}^{(i)})$

$\rightarrow$ mimimizing variance of incremental weight $(w_t^{(i)} \propto w_{t-1}^{(i)} \times iw)$

$$iw = \frac{\pi_t(x_{0:t-1}^{(i)}, x_t^{(i)})}{\pi_{t-1}(x_{0:t-1}^{(i)})q_t(x_t^{(i)}|x_{0:t-1}^{(i)})}$$

$\Rightarrow$ 1-step ahead predictive:

$$\pi_t(x_t|x_{0:t-1}) = p(x_t|x_{t-1}, y_t)$$

$\Rightarrow$ incremental weight:

$$iw \rightarrow \frac{\pi_t(x_{0:t-1})}{\pi_{t-1}(x_{0:t-1})} = \frac{p(x_{0:t-1}|y_{1:t})}{p(x_{0:t-1}|y_{1:t-1})}$$

$$\propto p(y_t|x_{t-1}) = \int p(y_t|x_t)p(x_t|x_{t-1})dx_t$$