Introduction

Francisco Pereira

Filipe Rodrigues

DTU Management Engineering
Department of Management Engineering

Outline

- Motivation
- A practical example
- Probabilistic Programming
- Structure
  - Syllabus
  - Method
  - Requirements
  - Evaluation

Motivation - focus on the problem!

- Traditional machine learning
    - A very wide range of algorithms

Motivation - focus on the problem!

- Traditional machine learning
  - A very wide range of algorithms
  - When faced with a new problem, map it onto one of the existing machine learning methods

Motivation - focus on the problem!

- Traditional machine learning
  - A very wide range of algorithms
  - When faced with a new problem, map it onto one of the existing machine learning methods
  - Focus is on algorithms

Motivation - focus on the problem!

- Traditional machine learning
  - A very wide range of algorithms
  - When faced with a new problem, map it onto one of the existing machine learning methods
  - Focus is on algorithms
- Probabilistic Graphical Models
  - No need to learn about the huge range of traditional methods

Motivation - focus on the problem!

- Traditional machine learning
  - A very wide range of algorithms
  - When faced with a new problem, map it onto one of the existing machine learning methods
  - Focus is on algorithms
- Probabilistic Graphical Models
  - No need to learn about the huge range of traditional methods
  - When faced with a new problem, think about
    - the number and types of variables in the problem domain
    - which variables affect each other
    - what the effect of changing one variable is on another variable
    - encode that knowledge in the form of a model

Motivation - focus on the problem!

- Traditional machine learning
  - A very wide range of algorithms
  - When faced with a new problem, map it onto one of the existing machine learning methods
  - Focus is on algorithms
- Probabilistic Graphical Models
  - No need to learn about the huge range of traditional methods
  - When faced with a new problem, think about
    - the number and types of variables in the problem domain
    - which variables affect each other
    - what the effect of changing one variable is on another variable
    - encode that knowledge in the form of a model
  - Focus is on the problem
    - includes all the assumptions about the problem domain
    - highly tailored models for specific scenarios

MBML in a nutshell

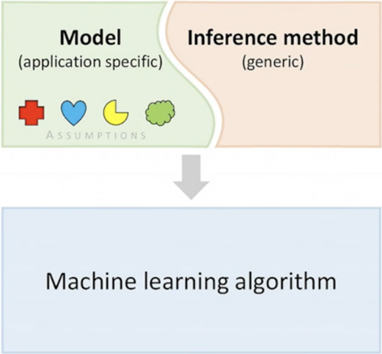


Figure: MBML big picture[1]

[1] In Christopher Bishop's Keynote, "Model-Based Machine Learning"
https://www.microsoft.com/en-us/research/video/keynote-model-based-machine-learning

Motivation - Why probabilistic models?

- **Uncertainty** in the real world
    - partial knowledge of state of the world
    - noisy observations
    - inherent stochasticity

Motivation - Why probabilistic models?

- **Uncertainty** in the real world
    - partial knowledge of state of the world
    - noisy observations
    - inherent stochasticity
- It is essential to account for uncertainty when building models of reality!
- **Probability theory** provides us with a consistent framework for quantifying and manipulating uncertainty

Motivation - A recent example...

starting to cross

crossing

stopping

bending-in

Motivation - A recent example...

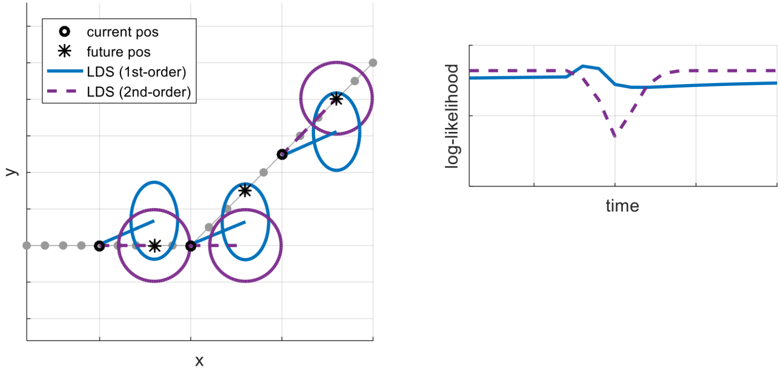
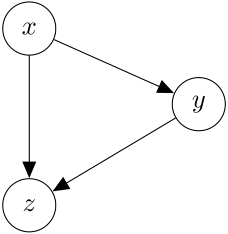


Figure: From Julian Kooij, TUDelft

Motivation - A recent example...

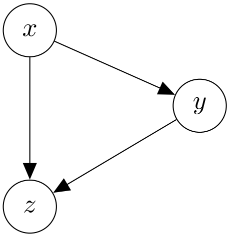
- Video

Motivation - Why probabilistic graphical models (PGMs)?



- **Nodes** - Represent random variables
- **Arrows** - Represent causal relationships

Motivation - Why probabilistic graphical models (PGMs)?



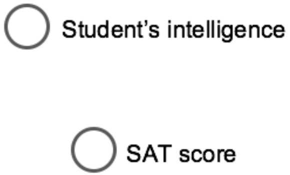
- Provide an **intuitive** and **compact** way of representing the structure of a probabilistic model
  - gives us insights about the properties of the model (e.g. relationships between variables, conditional independencies)
  - helps us communicate and design new models

A practical example

- A recruiter looking for the most intelligent students

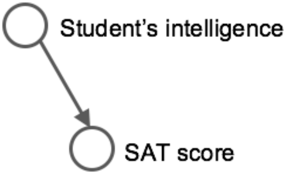
A practical example

- A recruiter looking for the most intelligent students



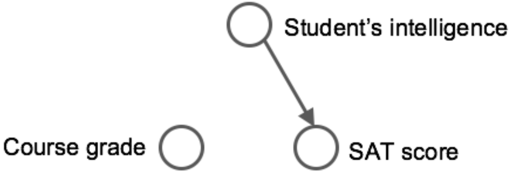
A practical example

- A recruiter looking for the most intelligent students



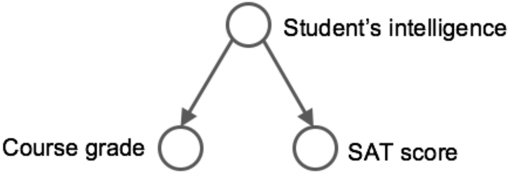
A practical example

- A recruiter looking for the most intelligent students



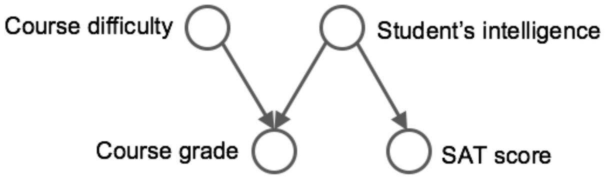
A practical example

- A recruiter looking for the most intelligent students



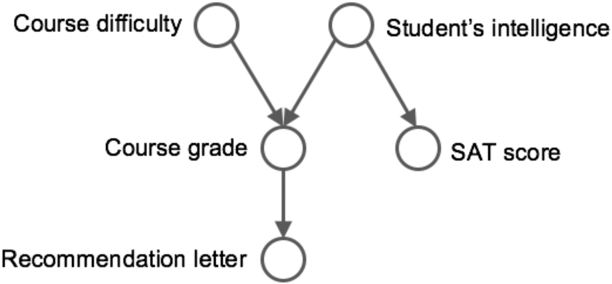
A practical example

- A recruiter looking for the most intelligent students



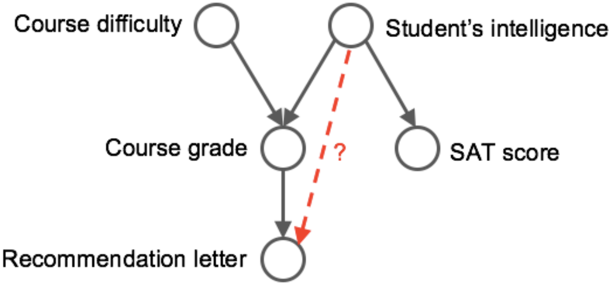
A practical example

- A recruiter looking for the most intelligent students



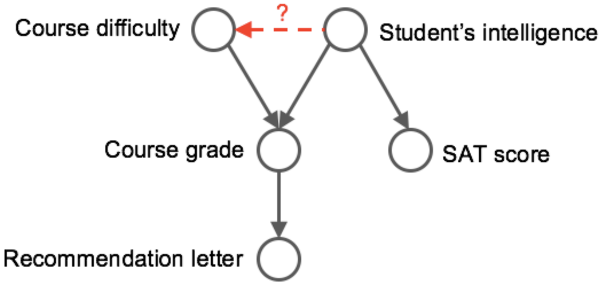
A practical example

- A recruiter looking for the most intelligent students



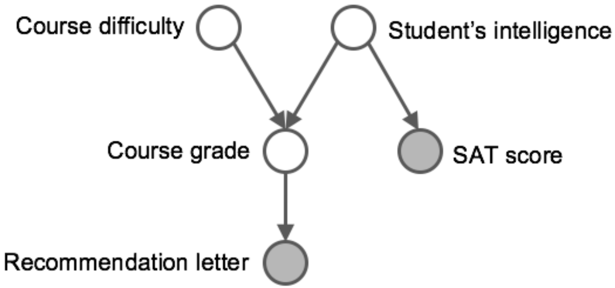
A practical example

- A recruiter looking for the most intelligent students



A practical example

- A recruiter looking for the most intelligent students



Playtime!

- Your first exercise! :-)
- In pairs, build a graphical model that you think best represents this problem involving the following variables:
  - Lung cancer
  - Smoking
  - Allergy
  - Cough
  - Type of work
  - Bloody sputum
- Duration: 5 min.

Playtime!

- Your first exercise! :-)
- In pairs, build a graphical model that you think best represents this problem involving the following variables:
  - Lung cancer
  - Smoking
  - Allergy
  - Cough
  - Type of work
  - Bloody sputum
- Duration: 5 min.
- Exchange your graphical models between groups (and check if they are similar...) ;-)

A (familiar) example - Linear regression

- y - target (a.k.a. response or dependent variable)
- $\mathbf{x}$ - vector of explanatory variables (a.k.a. predictor or independent variables)
- Target y is assumed to be a function of $\mathbf{x}$

$$y = f(\mathbf{x})$$

A (familiar) example - Linear regression

- y - target (a.k.a. response or dependent variable)
- $\mathbf{x}$ - vector of explanatory variables
  (a.k.a. predictor or independent variables)
- Target y is assumed to be a function of $\mathbf{x}$

$$y = f(\mathbf{x})$$

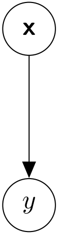
- Linear regression assumes f to be a **linear function** of $\mathbf{x}$

$$y = \boldsymbol{\beta}^T \mathbf{x} + \epsilon$$

- $\boldsymbol{\beta}$ is a vector of coefficients
- $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is an error term (accounts for observation noise)

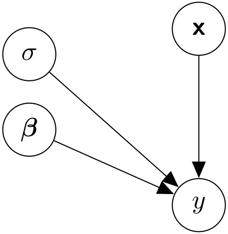
Linear regression

- Linear regression as a graphical model
  - We have a set of N observations of the targets y_n which depend on their corresponding explanatory variables $\mathbf{x}_n$



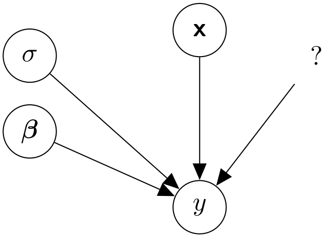
Linear regression

- Linear regression as a graphical model
  - We have a set of N observations of the targets y_n which depend on their corresponding explanatory variables $\mathbf{x}_n$
- We need to include parameters



Linear regression

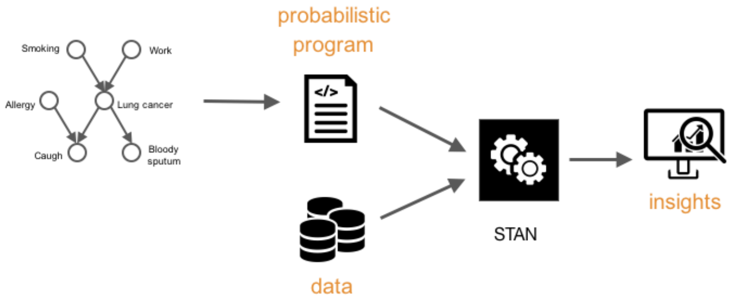
- Linear regression as a graphical model
  - We have a set of N observations of the targets y_n which depend on their corresponding explanatory variables $\mathbf{x}_n$



- In fact, we can even include other models (within our model)!

$$y = \boldsymbol{\beta_{lr}}^T \mathbf{x} + \boldsymbol{\beta_k} \textit{kernel\_model} + \boldsymbol{\beta_{dl}} \textit{deep\_learning\_model} \ldots$$

Probabilistic programming



About STAN

- STAN is a state-of-the-art platform for statistical modeling and high-performance statistical computation.
- It belongs to the family of “Probabilistic Programming Languages"
- It allows for specification of a graphical model
- We will use STAN quite a bit in our course - you’ll see it’s pretty easy! :-)
- Check http://mc-stan.org for your installation

Example with Linear Regression

```
lr_STAN="""
data {
int<lower=0>N; //Data points
vector[N] x; //independent variable
vector[N] y; //dependent variable
}
parameters{
real beta1;
real beta0;
real sigma;
}

model{
y ~ normal(beta0+beta1*x, sigma);
}
"""
```

Figure: Linear Regression with STAN

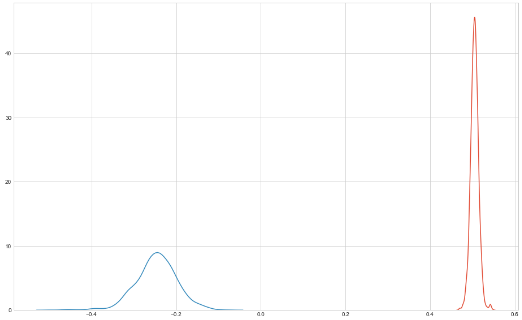
Example with Linear Regression

```
N=1000
sigma=0.1
beta0=-0.2
beta1=0.5

x_=np.random.uniform(-10, 10, size=N)
y_=np.random.normal(beta0+x_*beta1, sigma)

fit = pystan.stan(model_code=lr_STAN, data={'N':N, 'x':x_, 'y':y_}, algorithm="NUTS", seed=0, iter=1000)

fit.traceplot()
```

Figure: Python code calling STAN

Example with Linear Regression

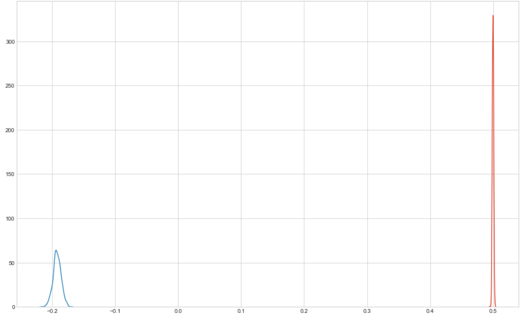
- Running inference... (N=10)



	mean
beta1	0.51
beta0	-0.25
sigma	0.1

Example with Linear Regression

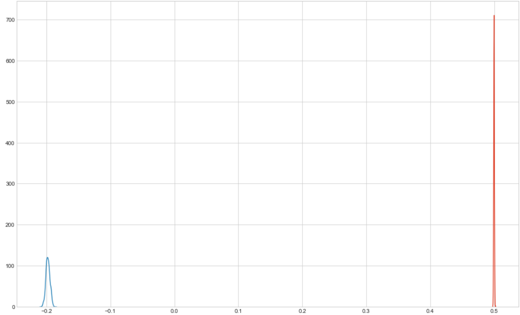
- Running inference... (N=200)



	mean
beta1	0.5
beta0	-0.19
sigma	0.1

Example with Linear Regression

- Running inference... (N=1000)



	mean
beta1	0.5
beta0	-0.2
sigma	0.1

Syllabus: Lectures

1. Probability and statistics review (1 lecture, F. Pereira)
2. PGM foundations (2 lectures, F. Pereira, F. Rodrigues)
3. Frequentist vs Bayesian and Probabilistic Programming (1 Lecture, F. Rodrigues)
4. Regression, classification, time series and topic models (4 lectures, F. Pereira, F. Rodrigues)
5. Bayesian Inference (3 lectures, F. Pereira, F. Rodrigues)
6. Gaussian Processes (1 lecture, F. Rodrigues)
7. Project Support (1 lecture)

Method

- Python (or R, MATLAB, Julia) and Stan
- Jupyter notebooks (in class and homework)

Requirements

- Programming is important (Python, MATLAB or R, preferably)
- Confident knowledge about basic statistics and probability
- Creativity, domain knowledge

Evaluation

- 2 Tests, 25% each (March 22, May 10)
- Project, 50% (deadline: May 10)