



**Student's Scientific Chapter
Computer Engineering Department
Amirkabir University of Technology(Tehran Polytechnic)**

Probabilistic Programming, Modeling the world through Uncertainty

Amirkabir Artificial Intelligence Student Summit Workshop

Ali Izadi

August 2021

1- Description	3
2- Probabilistic Programming	4
3- Uncertainty	5
3-1 Bayesian Framework	6
3-2 Introduction to Probabilistic Graphical models	7
4- Our First Probabilistic Program	8
5- Inference methods	8
5-1 markov chain monte carlo (mcmc)	9
5-2 variational inference	9
6- Introduction to Tensorflow Probability and PyMC3	10
7- How to enrich probability distributions (Advanced)	10
Main References	11

1- Description

Workshop Level: Intermediate

Prerequisites: Nothing but preferred to familiar with the probabilistic graphical models. If not, just being interested in Bayes theorem and Probability distributions is enough!

Syllabus:

You might hear about TensorFlow Probability. In this workshop, we discuss what it is for and in general what probabilistic programming is. To do so we model multiple problems that they need probabilistic methods. Problems that need to be solved in a way that we see their world through uncertainty. We also implement the solutions not only using TensorFlow Probability but using other packages like PyMC3.

The workshop includes:

- 1- Introduction to probabilistic programming.
- 2- Some concepts in probabilistic graphical models including variational inference and Monte-Carlo sampling.
- 3- Introduction to Tensorflow Probability and PyMC3
- 4- Model multiple probabilistic problems with these frameworks.

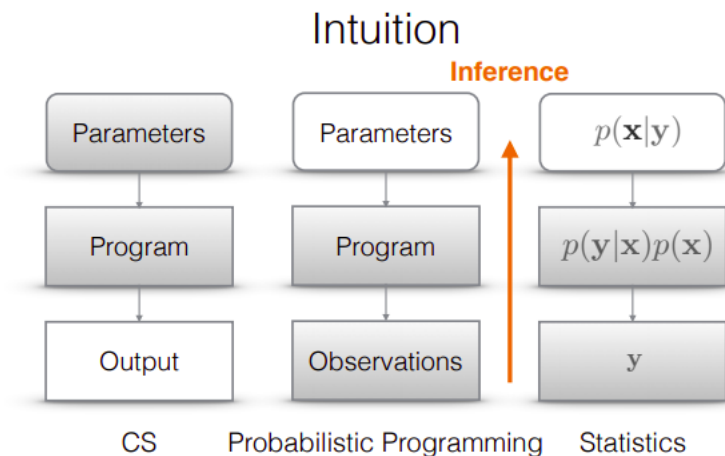
The Estimated Duration of your Workshop: 150 minutes

2- Probabilistic Programming

<https://www.youtube.com/watch?v=BrwKURU-wpk>

frameworks for probabilistic models definitions + automatic parameters estimation

Probabilistic Programming is a programming paradigm in which probabilistic models are specified and inferences for these models are performed automatically.



It can be used to create systems that help make decisions in the face of uncertainty.

```
import tensorflow as tf
import tensorflow_probability as tfp

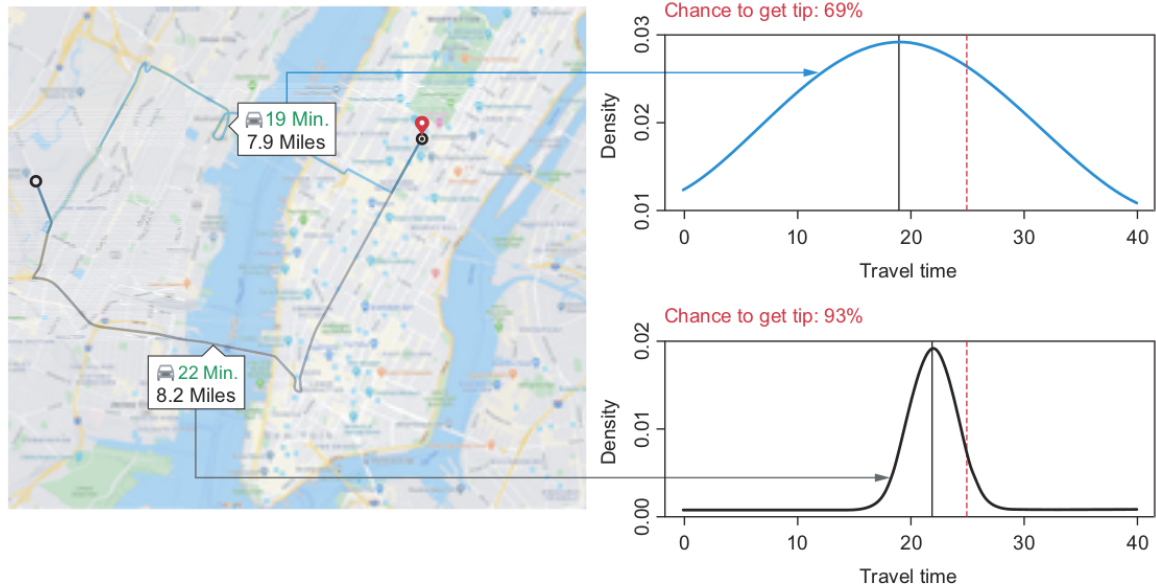
# Pretend to load synthetic data set.
features = tfp.distributions.Normal(loc=0., scale=1.).sample(int(100e3))
labels = tfp.distributions.Bernoulli(logits=1.618 * features).sample()

# Specify model.
model = tfp.glm.Bernoulli()

# Fit model given data.
coeffs, linear_response, is_converged, num_iter = tfp.glm.fit(
    model_matrix=features[:, tf.newaxis],
    response=tf.cast(labels, dtype=tf.float32),
    model=model)
```

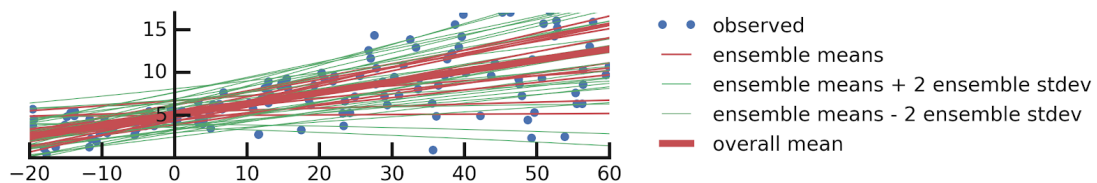
3- Uncertainty

- The necessity for modeling uncertainty:



Travel time prediction of the satnav. On the left side of the map, you see a deterministic version just a single number is reported. On the right side, you see the probability distributions for the travel time of the two routes.

- What is the difference between **probabilistic** vs **non-probabilistic** classification?
- Probabilistic regression**



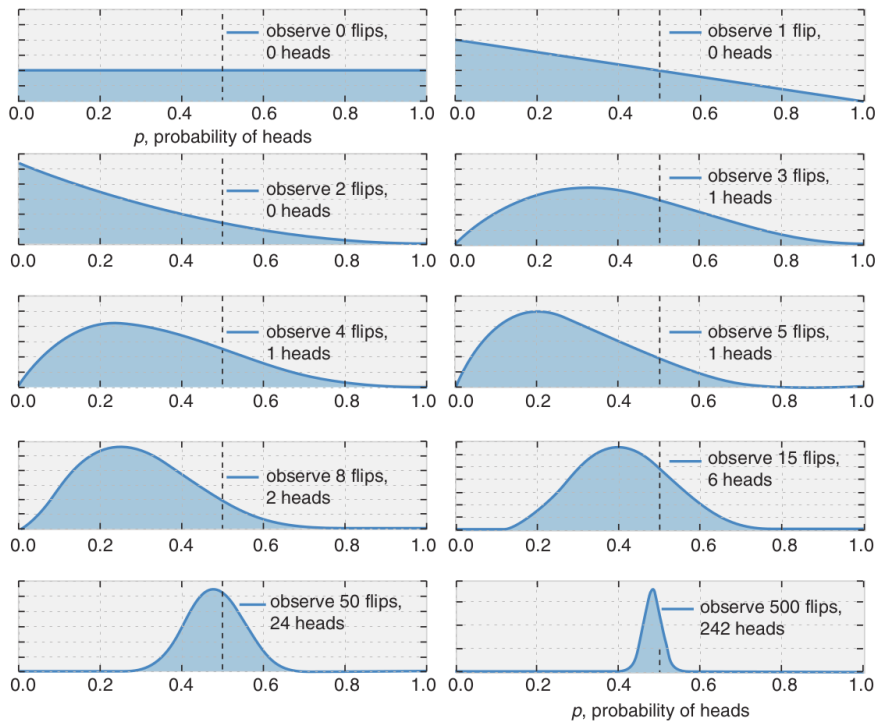
3-1 Bayesian Framework

$$P(A|X) = \frac{P(X|A)P(A)}{P(X)}$$
$$\propto P(X|A)P(A) \quad (\propto \text{ means "is proportional to"})$$

The main goal in this workshop is to compute the left part of this equation which is named **Posterior Probability**.

★ Coding

Implementation of coin-flip example to explain how to inference under the Bayesian framework.



https://github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/blob/master/Chapter1_Introduction/Ch1_Introduction_PyMC3.ipynb

3-2 Introduction to Probabilistic Graphical models

In this part, we will discuss probabilistic graphical models.

First How to represent PGM as graphs and then how to learn PGM after observing data.

- **Representation:** model distributions with graphs.

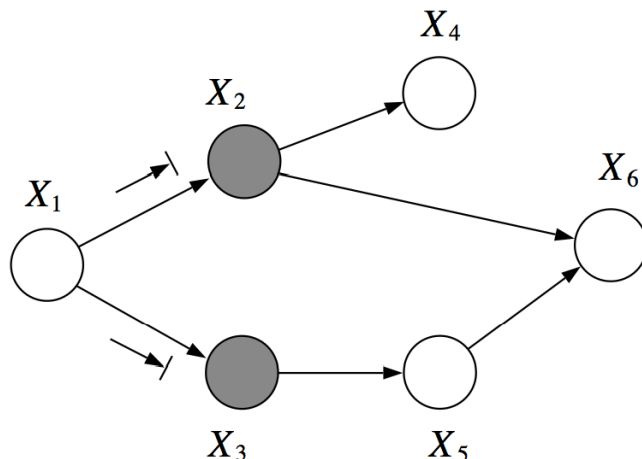
Joint distribution with chain rule:

$$p(x_1, x_2, \dots, x_n) = p(x_1)p(x_2 \mid x_1) \cdots p(x_n \mid x_{n-1}, \dots, x_2, x_1).$$

Conditional distribution from graph:

$$p(x_i \mid x_{i-1}, \dots, x_1) = p(x_i \mid x_{A_i}).$$

Graph Example:



- **Inference:** Answer questions from distribution.

For example, finding the most probable assignment from the distribution.

$$\operatorname{argmax}_{x_1, \dots, x_n} p(x_1, \dots, x_n, y = 1).$$

Or in general learning (fit the model to a dataset):

For inference, there are multiple ways including exact inference or approximation inference which is the most used approaches in probabilistic programming.

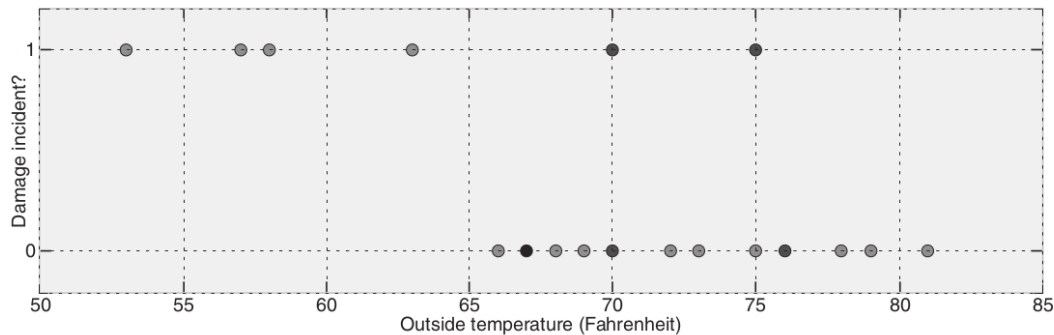
Two methods are **Markov chain Monte Carlo sampling(MCMC)** and

The variational inference which we will discuss later and use as codes in our problems.

4- Our First Probabilistic Program

Space shuttle disaster:

Detect incident by temperature according to uncertainty in the data.



★ Coding (PyMC3)

We will model this problem using PyMC3 and MCMC methods.

<https://blog.tensorflow.org/2018/12/an-introduction-to-probabilistic.html>

https://github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/blob/master/Chapter2_MorePyMC/Ch2_MorePyMC_PyMC3.ipynb

5- Inference methods

Probabilistic methods need computing posterior probability after observing data. MCMC and variational inference are the two most common approaches. We will explain more about these two methods and compare them to understand how to choose them based on the problem.

http://pyro.ai/examples/intro_part_ii.html

<https://ermongroup.github.io/cs228-notes/>

https://docs.pymc.io/pymc-examples/examples/variational_inference/bayesian_neural_network_advi.html

https://probflow.readthedocs.io/en/latest/user_guide/inference.html

5-1 markov chain monte carlo (mcmc)

Class of algorithms for sampling from a probability distribution. The more samples included from the distribution, the more closely the distribution of samples matches the actual desired distribution.

5-2 variational inference

Variational inference casts approximate Bayesian inference as an optimization problem trying to fit a simpler distribution to the desired posteriors.

https://colab.research.google.com/github/tensorflow/probability/blob/main/tensorflow_probability/examples/jupyter_notebooks/Variational_Inference_and_Joint_Distributions.ipynb
<https://arxiv.org/pdf/1907.08956.pdf>

★ Coding (Tensorflow Probability)

In this part, we implement probabilistic regression as an alternative to classic regression. The final modeling considers uncertainty and is solved by variational inference methods using TensorFlow probability.

<https://blog.tensorflow.org/2019/03/regression-with-probabilistic-layers-in.html>
https://docs.pymc.io/pymc-examples/examples/variational_inference/bayesian_neural_network_advi.html
https://www.tensorflow.org/probability/api_docs/python/tfp/layers/DenseVariational
https://www.tensorflow.org/probability/examples/Probabilistic_PCA
https://www.tensorflow.org/probability/api_docs/python/tfp/vi/fit_surrogate_posterior

6- Introduction to Tensorflow Probability and PyMC3

In this part, we revisit Tensorflow Probability and PyMC3 and review different parts in these frameworks including:

1. How to define probability distributions?
2. How to define priors?
3. How to compute posterior probability?

★ Coding

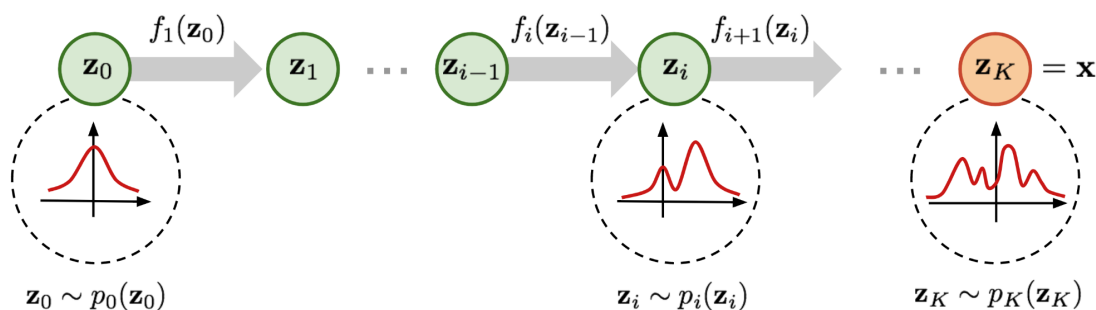
https://docs.pymc.io/pymc-examples/examples/getting_started.html

https://www.tensorflow.org/probability/examples/A_Tour_of_TensorFlow_Probability

7- How to enrich probability distributions (Advanced)

As we discussed before there are famous probability distributions including normal distribution. Are this distribution can always model the data generation process, or can variational inference model complex distributions with just normal distribution? The answer is No!

Normalizing flow is one the most common methods in probability literature to convert simpler distributions to more complex ones. Having this capability we enrich variational inference as a more accurate approximation method.



If we have time:

★ Coding (Tensorflow Probability)

<https://towardsdatascience.com/variational-bayesian-inference-with-normalizing-flows-a-simple-example-1db109d91062>

https://www.tensorflow.org/probability/examples/Variational_Inference_and_Joint_Distributions

<https://gowrishankar.info/blog/normalizing-flows-a-practical-guide-using-tensorflow-probability/>

Main References

- [1] Davidson-Pilon, C., 2015. **Bayesian methods for hackers: probabilistic programming and Bayesian inference**. Addison-Wesley Professional.
- [2] Duerr, O., Sick, B. and Murina, E., 2020. **Probabilistic Deep Learning: With Python, Keras and TensorFlow Probability**. Manning Publications.
- [3] Dillon, J.V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M. and Saurous, R.A., 2017. **Tensorflow distributions**. *arXiv preprint arXiv:1711.10604*.
- [4] Salvatier, J., Wiecki, T.V. and Fonnesbeck, C., 2016. **Probabilistic programming in Python using PyMC3**. *PeerJ Computer Science*, 2, p.e55.
- [5] <https://github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers>
- [6] <https://www.tensorflow.org/probability>
- [7] <https://docs.pymc.io/>
- [8] <https://ermongroup.github.io/cs228-notes/>

