

# PGMs session 1

# Factor Operations

This session we focus on getting our infrastructure going, and exploring the manipulation of discrete probability factors (on their own, i.e. without combining them in some graphical structure).

## 1 Preparation: Downloads and infrastructure

- Download the Barber textbook (or get a physical copy) – see the info sheet for details.
- Register on Coursera and locate Koller’s probabilistic graphical model (PGM) lectures – see the info sheet for details.
- Install `emdw` on you computer (we will help you during the first practical session if you struggle).

## 2 Preparation: Probability review

- Review basic probability theory: Consult a standard probability theory textbook (such as PZ Peebles or RD Yates and DJ Goodman) and make sure that you are comfortable with the following:
  - **Random variables** – both discrete and continuous (RVs)
  - **Probability distributions** – both probability mass functions (PMF) and probability density functions (PDF)
  - **Joint distributions** and how it describes the behaviour of multiple random variables
  - **Conditional distributions** and how you obtain them from joint distributions (chain or product rule)
  - **Marginal distributions** and how you obtain them from joint distributions (sum rule)
  - **Bayes rule** – which will get a life of its own in this work
  - **Statistical independence** – both the definition and its interpretation
- Read the following sections of Barber Chapter 1: 1.1 Probability Refresher, 1.2 Probabilistic Reasoning, 1.3 Prior, Likelihood and Posterior, and 1.4 Summary.

## 3 Preparation: Watch some Koller videos

Watch the following videos in the Coursera course Probabilistic Graphical Models 1: Representation, Week 1:

- Watch the *Introduction and Overview: Motivation and Overview (19:17)* video to get an idea of what probabilistic graphical models are about. We will mostly, but not exclusively, cover the same material as that discussed in this video.
- Watch the *Introduction and Overview: Distributions (4:56)* and *Introduction and Overview: Factors (6:40)* videos on factor operations:
  - Note how a factor product is determined and how this can be used to build a joint distribution.
  - Note how conditional and marginal distributions are implemented via factors.
  - These examples are for discrete variables. Muse a bit about what might be involved in defining their continuous counterparts.

## 4 Preparation: Get familiar with some C++ concepts

We will be using a C++ PGM library, `emdw`, for the coding exercises. In preparation for this, try to build up a **basic** understanding of the following concepts (there are loads of tutorials, forums and documentation websites online; the last few chapters of *C How to Program* by Paul and Harvey Deitel is also a good starting point):

- C++ syntax
- Programme structure, header and source files, including libraries, compiling and executing a programme
- Using the C++ Standard Library output stream to write to the console
- Pointers, memory management, smart pointers (shared and unique pointers)
- Classes, objects, inheritance, polymorphism
- C++ Standard Library containers, specifically `std::vector` and `std::map`
- Templates (just the idea)

Please note that we do not expect you to be so fluent in C++ at this point that you can write your own applications from scratch, or understand all these concepts in great depth. Your main goal should be to understand the code examples that we give you and make relatively small adaptations (see e.g. Question 6(b)-(d) below). We will also aim to document the code examples well, and we will of course be available to help you to understand and use `emdw`.

## 5 Practical: Do a simple paper exercise

This one is simple enough that the answers are nearly self-evident, you can easily check against your common sense (please do this!). The objective here is to confirm that you can systematically get to those self-evident answers by using the factor operations you saw in the videos (products, marginalisations and conditions/evidence).

Take two balanced coins. For both of them, mark one side with the number '0' and the other with the number '1'. Now consider flipping them and denote their outcomes by letting the random variables  $X$  and  $Y$  take on the values on the facing sides of the coins. Let  $Z = X + Y$ . Make use of the factor operations you encountered earlier on and do the following on paper:

- (a) Define the three factors,  $p(X)$ ,  $p(Y)$  and  $p(Z|X, Y)$ , as tables.
- (b) Show symbolically that  $p(X, Y, Z) = p(Z|X, Y)p(X)p(Y)$  by using the chain rule and the fact that  $X \perp\!\!\!\perp Y$ .
- (c) Determine the joint distribution  $p(X, Y, Z)$  as a table.
- (d) Determine the marginal distribution  $p(X, Z)$  as a table.
- (e) Determine the marginal distribution  $p(Z)$  as a table (i.e., what is the distribution over the sum  $Z$  given no evidence, also known as the prior belief over  $Z$ ?).
- (f) Determine the conditional distribution  $p(Z|X = 1)$  as a table (i.e., what is the distribution over the sum  $Z$  given the observation that the first coin flip resulted in '1'?).
- (g) Determine the conditional distribution  $p(X|Z = 2)$  as a table (i.e., what is the belief over  $X$  given the observation that the sum  $Z$  is 2?).
- (h) Convince yourself that the above really only is a nicely crystallised version of the classical operations you reviewed earlier on.

## 6 Practical: Do a simple coding exercise

We will use the `emdw DiscreteTable` class to represent the basic discrete factor (see Table 1. This class embodies a basic discrete factor (a generalisation of a probability distribution) and its associated factor operators. You will be using this class extensively throughout this course. It provides the basic machinery to automate probabilistic calculations.

Table 1: Discrete factors (also known as a discrete potential functions / probability mass functions / conditional PMFs) are represented in `emdw` by the `DiscreteTable` class. In this assignment you will use the following member functions – consult `emdw/docs/userguide.pdf` for more details.

Description	<code>emdw</code>
Multiply two distributions to get their joint	<code>absorb</code>
Remove a previously multiplied in distribution	<code>cancel</code>
Condition on evidence; i.e., observe some RVs	<code>observeAndReduce</code>
Marginalise; i.e., remove certain RVs from a joint distribution	<code>marginalize</code> ( <code>spec. keep</code> )
Max-marginalize	<code>marginalize via</code> <code>MaxMarginalize</code> operator ( <code>spec. keep</code> )
Renormalize a probability table	<code>normalize</code>
Make a clone of a factor, using different RVs	<code>copy</code>

We will for now specifically avoid using the higher-level magic also available in `emdw`. We do this to to develop a clear understanding of the basic underlying principles and ideas on which these higher-level constructs are built.

- Read through the EMDW User Guide (`emdw/docs/userguide.pdf`) Sections 2, 3, and 4.1.
- Look through `emdw/src/bin/example.cc` and read Section 6.1 in the EMDW User Guide.
- Create your own source file by copying `emdw/src/bin/example.cc` as explained in Section 1.2 of the EMDW User Guide.
- Now edit this source file to code Question 5(a) and (c)-(g) of the coin flipping problem. Carefully compare the results with what you have obtained in Question 5.