Dual Probabilistic Programming

Alessio Benavoli
Dalle Molle Institute for Artificial intelligence (IDSIA), Manno (CH)
alessio@idsia.ch

Bayesian inference often involve complex and intractable integrals; this has restricted the application of these models to experts. Probabilistic Programming languages (PPls) address this issue by letting users express a probabilistic model as a program. The program specifies how to generate output data by sampling latent probability distributions. Then a compiler checks this program for type errors and translates it to a form suitable for an inference procedure, which uses observed output data to fit the latent distributions. Probabilistic models are very effective because they explicitly model uncertainty and, therefore, lead to explainable machine learning.

A central theme in PP is to address the problem of knowledge representation and, in particular, reverse-engineering human intelligence. How can we model expert knowledge in the most faithful way?

In this article, we present Dual PP (DPP) a (mathematical) dual formulation of PP for knowledge (belief) representation and inference. This belief representation model is rooted in the axiomatisation of Bayesian theory of probability and, the work of de Finetti and, in particular, Williams and Walley.

To understand this framework, we consider an experiment whose outcome ω belongs to a certain space of possibilities Ω (e.g., Head or Tail). We can model a subject's (we call the subject Alice) beliefs about ω by asking her whether she accepts engaging in certain risky transactions, called *gambles*, whose outcome depends on the actual outcome of the experiment. Mathematically, a gamble is a bounded real-valued function on Ω, $g : \Omega \to \mathbb{R}$, which is interpreted as an uncertain reward in a linear utility scale. If Alice accepts a gamble g, this means that she commits herself to receive $g(\omega)$ *utiles*[1] if the outcome of the experiment eventually happens to be the event $\omega \in \Omega$. Since $g(\omega)$ can be negative, Alice can also lose utiles. Therefore Alice's acceptability of a gamble depends on her knowledge about the experiment.

The set of gambles that the subject accepts is called her set of *desirable* (or *acceptable*) *gambles*. One such set, denoted as $\mathcal{K}$, is said to be *coherent* when it satisfies the following rationality criteria [4]:[2]

A.1 If $\inf g > 0$ then $g \in \mathcal{K}$ (Accepting Sure Gains);
A.2 If $g \in \mathcal{K}$ then $\sup g \geq 0$ (Avoiding Sure Loss);
A.3 If $g \in \mathcal{K}$ then $\lambda g \in \mathcal{K}$ for every $\lambda > 0$ (Positive Scaling);
A.4 If $g, h \in \mathcal{K}$ then $g + h \in \mathcal{K}$ (Additivity);
A.5 If $g + \epsilon \in \mathcal{K}$ for every $\epsilon > 0$ then $g \in \mathcal{K}$ (Continuity).

Note how these five axioms express some truly minimal requirements: the first means that Alice likes to increase her wealth; the second that she does not like to decrease it; the third and fourth together simply rephrase the assumption that Alice's utility scale is linear; the last one imposes some regularity (continuity) requirement.

In spite of the simple character of these requirements, these axioms alone define a very general theory of probability. De Finetti's (Bayesian) theory is the particular case obtained by additionally imposing completeness, that is, the idea that a subject should always be capable of comparing options [4, 5]. Note that, from a geometric point of view, Alice's set of desirable gambles $\mathcal{K}$ mathematically is a pointed and salient closed convex-cone that includes the set of non-negative gambles (this follows from the above properties). It can then be shown that the dual cone of $\mathcal{K}$, denoted as $\mathcal{K}^*$, is a closed-convex set of probabilities [1, 4].

DPP is a language that allows Alice to express her beliefs about an uncertain experiment by specifying a finite set of gambles $G = \{g_1, \dots, g_n\}$ she is willing to accept. Moreover, it provides three operations:

Compiling a compiler checks this program for type errors, closes Alice's assessments according to Axioms A.1, A.3–A.5 to obtain $\mathcal{K}$ and checks that $\mathcal{K}$ avoids sure loss (A.2 is also satisfied).
Inference this operation (called Natural Extension) allows Alice to determine the implication of desirability, i.e., which gambles are also desirable for her given she accepted the gambles in G.
Updating this allows Alice to revise her set of desirable gambles based on new information (observations).

The verification that $\mathcal{K}$ avoids sure loss means that there does not exist a gamble in $\mathcal{K}$ that can make Alice to loose money deterministically. Avoiding sure loss is equivalent to say that $\mathcal{K}^*$ is not empty, i.e., there exists at least one probabilistic model that is compatible with Alice's assessments of desirability. A set $\mathcal{K}$ that does not satisfy A.2 is not rational and cannot be used for inference. The central operation in DPP is natural extension (inference). When Ω is infinite (e.g., real vector space), the inference procure can be written as an intractable semi-infinite linear programming problem.

[1] Abstract units of utility, indicating the satisfaction derived from an economic transaction.
[2] See coin example in [1].

PROBPROG'18, Oct 4–6 2018, Boston, MA, USA
2018.

ask	53.8	49.5	45.2	41.1	29.3	25.7	22.3	19.1	16.2	13.6	11.3	9.2	7.5	6.1	4.9	3.9	3.2	2.15	1.55	1.15	0.90	0.3
bid	53.3	49	44.8	40.6	28.9	25.3	21.9	18.7	15.9	13.3	11	8.9	7.2	5.8	4.6	3.7	3	2	1.40	1	0.75	0.2
strike	2490	2495	2500	2505	2520	2525	2530	2535	2540	2545	2550	2555	2560	2565	2570	2575	2580	2590	2600	2610	2620	2675

Table 1. Ask and bid price for a call option on the S&P500 index: maturity 30days, quote day 2017-10-03.

However, when the gambles g_j are polynomial or piecewise polynomial functions, an approximate solution of this problem can be obtained by means of sum-of-squares hierarchy (SOS hierarchy) [3]. SOS hierarchy provides a hierarchy of convex relaxations of this optimisation problem of increasing power but increasing computational cost, which has a convergence guarantee (under mild conditions). You can think this approach as the dual of the Markov chain Monte Carlo methods that are used in PP for calculating numerical approximations of multi-dimensional integrals, which also have increasing power at the increasing of the number of samples but increasing computational cost. The advantage of the dual approach is that the approximation is always conservative and is theoretical sound per se [1].

Why should we use DPP instead of PP?

Problem modelling in PP requires to make several assumptions (likelihood model, prior and hyper-prior models etc.) and if these assumptions are not met, then the conclusions from our analysis will not be valid. Bayesian nonparametric models partially overcome this issue by providing a way to automatically learn structure in the data. However, they also require the practitioner to make some choices (e.g., the kernel in a Gaussian Process).

DPP can model knowledge and use data without the need of any additional assumption.

Let us consider a problem from finance: an European call option on an underlying security with strike k and maturity T. It gives the holder the option of buying the underlying security at price k at time T. If the price S_T is more than k, then the holder will exercise the option and make a profit of $S_T - k$. Conversely, if it is less than k, the holder will not exercise and does not make a profit. Thus, the payoff of this option is $\max(S_T - k, 0)$. Since options are traded, a key problem in financial economics is to determine the belief of the market about the future value of S_T from the ask and bid[3] prices of these options. Table 1 shows the ask and bid price for 22 call options on the S&P500 index. What does the first column of the table mean? It means that "the market" believes that the gambles $\max(S_T - 2490, 0) - 53.3$ and $53.8 - \max(S_T - 2490, 0)$ are desirable, since there exists someone that is willing to sell the option $\max(S_T - 2490, 0)$ at price 53.8 and to buy it at price 53.3. As inference, we aim to compute the market's selling and buying price for the gamble $h = I_{\{[c,\infty)\}}(S_T)$.[4]

How do we do that in PP? The state-of-art approach [2] is as follows:

1. compute the mid price for each call option mid=(ask+bid)/2;
2. use Black-Scholes (BS) model to invert option pricing to get a so called implied volatility (IV);
3. interpolate these IVs using a local polynomial smoothing method;
4. compute the second derivative of this curve w.r.t. the strikes to obtain the PDF $p(S_T)$;
5. compute $P(S_T > c) = \int I_{\{[c,\infty)\}}(S_T) p(S_T) dS_T$.

The result is in Figure 1 (blue line). Observe we had to make several assumptions: (i) the true price is assumed to be the mean of ask and bid; (ii) we have to specify the type of local polynomial smoothing; (iii) we use a mathematical model for the dynamics of a financial market (BS).

In DPP, we take the data as they are without extra assumptions. In this case, the set of desirable gambles includes 44 gambles:

$$G = \{\max(S_T - 2490, 0) - 53.3, 53.8 - \max(S_T - 2490, 0), \dots, \max(S_T - 2675, 0) - 0.2, 0.3 - \max(S_T - 2675, 0)\}.$$

This set avoids sure loss (i.e., a DPP program compiles). The inference procedure can be formulated as follows:

$$\sup_{\lambda_0 \in \mathbb{R}, \lambda_j \geq 0} \lambda_0, \; s.t. \; h - \lambda_0 - \sum_{j=1}^{|G|} \lambda_j g_j(S_T) \geq 0, \; \forall \, S_T. \qquad (1)$$

with $h = I_{\{[c,\infty)\}}(S_T)$. This is a semi-infinite linear programming problem. Its (approximate) solution gives the lower probability that $S_T > c$. We can also compute the upper probability by selecting $h = 1 - I_{\{(-\infty,c]\}}(S_T)$. These two curves are shown in Figure 1 and they include the kernel method. Since the lower and upper probabilities are quite far a part, it means that data alone are not enough to produce the blue line and so the model assumptions in [2] have a non negligible effect on the estimate!

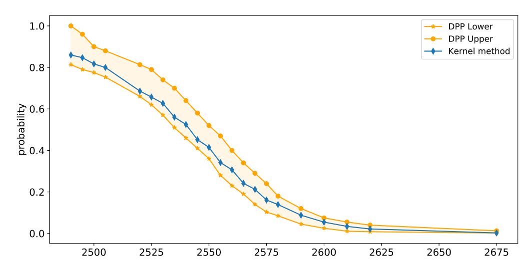

Figure 1. Probability that $S_T > c$ for the kernel method (blue) and DPP (lower and upper probabilities).

This is just an example. DPP can be applied to other problems in finance, engineering, social sciences etc.. We are at the moment developing a Python library for DPP.

[3] The bid price is the max price that a buyer is willing to pay for a security. The ask price is the min price that a seller is willing to receive.

[4] $I_A(a)$ is the indicator function of A, it si one if $a \in A$ and 0 otherwise.

References

[1] Alessio Benavoli, Alessandro Facchini, Dario Piga, and Marco Zaffalon. 2017. SOS for bounded rationality. In *Proc. ISIPTA'17 Int. Symposium on Imprecise Probability: Theories and Applications,* (2017-05-07). PJMLR, 1–12. https://arxiv.org/abs/1705.02663

[2] M. Benko, M. Fengler, W. Härdle, and M. Kopa. 2007. On extracting information implied in options. *Computational Statistics* 22, 4 (01 Dec 2007), 543–553.

[3] Jean Bernard Lasserre. 2009. *Moments, positive polynomials and their applications*. Vol. 1. World Scientific.

[4] P. Walley. 1991. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London.

[5] P. M. Williams. 1975. *Notes on conditional previsions*. Technical Report. School of Mathematical and Physical Science, University of Sussex, UK. Reprinted in [6].

[6] P. M. Williams. 2007. Notes on conditional previsions. *International Journal of Approximate Reasoning* 44 (2007), 366–383. Revised journal version of [5].