Probabilistic Modelling and Reasoning: Assignment

*Modelling the skills of Go players**

Deadline: Tuesday March 20 2018 at 16:00

Marks: This assignment is out of 100 marks and forms 20% of your final grade for the course.

Group work: You may work alone or in pairs. If the work is done in pairs, both students have to work together on all questions. **Please submit the name(s) and student number(s) of your team using this form[1] as soon as possible** – by March 2nd latest.

Submission instructions: You should submit this assignment manually to the ITO office by the deadline. Handwritten submissions are acceptable if the handwriting is neat and legible. If the work is done in pairs, only a single report needs to be submitted. Note both student names and IDs on the copy and *add a statement confirming that both students contributed equally to answering all questions*.

Data files: Associated data files for the assignment are available as the compressed TAR archive file `go_player_skill_model.tar.gz`.

Academic conduct: Assessed work is subject to University regulations on academic conduct:
http://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct
Do not show your code, answers or write-up to anyone else.
Never copy-and-paste material that is not your own into your assignment and edit it.

Late submissions: We follow the policy of the School of Informatics: http://web.inf.ed.ac.uk/infweb/student-services/ito/admin/coursework-projects/late-coursework-extension-requests

Introduction

A central theme of this course is constructing probabilistic models to express the relationships we believe exist between variables in real world problems. Given a model and observed data we can then make principled predictions of the answers to questions we have and importantly also get some indication of our degree of certainty in those answers. In this coursework assignment we will consider the problem of inferring information about the skills of players from game outcome data, and using the resulting model to predict the results of future games.

In particular you will be constructing a model to infer the skill of professional Go players. As many of you will know, a computer-based Go agent named AlphaGo, developed by Google DeepMind, beat the reigning 3-time European Go champion Fan Hui in October 2015. This was the first time a computer program had beaten a professional level Go player in an even (non-handicapped) game. AlphaGo then went on to compete against Lee Sedol, a South Korean Go player who is regarded as one of the best players in the world in the last decade, in a five game match in March 2016. Of the five games in the match, AlphaGo won four and Lee Sedol won one. AlphaGo has played more games since then and beaten the world number one Ke Jie.

Part of your task in this assignment will be to use a probabilistic model fitted to a dataset of over 50 000 professional Go game results, kindly provided by http://Go4Go.net, as well as some recent AlphaGo match results to find out what the available data suggests the relative skill of AlphaGo is compared to the best human players. Note, no specialist knowledge of the game Go beyond the information given in the assignment is required.

*Based on material courtesy of Dr. Graham and Dr. Storkey.
[1] https://goo.gl/forms/iRyRMXW9UQNLviCk2

Notation

$\mathbb{P}[\cdot]$ indicates a probability mass function on discrete random variables and $\mathbb{p}[\cdot]$ a probability density function on real-valued random variables. $\mathbb{1}[\texttt{condition}]$ is an indicator function which is equal to 1 if `condition` is true and 0 otherwise. $\mathcal{N}\left(x; \mu, \sigma^2\right)$ is a Gaussian probability density function on x with mean μ and variance σ^2 and $\Phi(x)$ is the standard Gaussian cumulative density function (cdf), i.e.

$$\mathcal{N}\left(x; \mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \qquad \Phi(x) = \int_{-\infty}^{x} \mathcal{N}(u; 0, 1)\, \mathrm{d}u.$$

$\left\{x_i\right\}_{i=1}^{N}$ indicates a collection of N variables i.e. $\left\{x_i\right\}_{i=1}^{N} = \left\{x_1, x_2, \ldots x_N\right\}$. $\mathbb{R}$ is the set of real numbers and $\mathbb{N}$ is the set of natural numbers $\{1, 2, 3, \ldots\}$.

Question 1: Player skill graphical models (20 marks)

As a first step we will consider a basic probabilistic graphical model relating the skills of two Go players and the result of the game between them.

In Go the two players lay either black or white stones on the playing board, hence we will refer to the two players as the *black player* and the *white player*. The black player always lays the first stone; to adjust for this there is usually a point offset called *komi* given to the white player in compensation. The *komi* is usually set to have a fractional component (e.g. 6.5 is common).

As the main scoring methods only allow integer scores this generally means games can only result in a win or loss with draws very infrequent. Therefore we will model the result as being a binary outcome of either the black player winning or the white player winning.

In Figure 1 a simple directed graph is given to describe the relationship between the skills of two players and the result of a game between them. The game is indexed by $k \in \mathbb{N}$, and every player is also assigned a unique ID $\in \mathbb{N}$, with the ID of the black player in the k^{th} game being given by b_k and the ID of the white player by w_k. In general there may be more than one (or zero) games between each pair of players. The graph in Figure 1 models the outcome of one game only. The skills of the black and white players in the k^{th} game are represented by s_{b_k} and s_{w_k} respectively. The result of the game is denoted $r^{(k)}$ and is either equal to 1 if the black player wins or 0 if the white player wins.

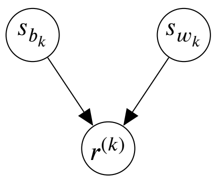


Figure 1: Directed graph for a simple player skill model when observing one game between two players. Games are indexed by $k \in \mathbb{N}$, $b_k \in \mathbb{N}$ is the ID of the black player and $w_k \in \mathbb{N}$ the ID of the white player in game k. The skill of the black player is s_{b_k}, the skill of the white player is s_{w_k} and the result of the game is $r^{(k)} \in \{0, 1\}$, with 0 indicating a white win and 1 a black win.

(a) Consider the case where we have three players and games between each pair of players. [5 MARKS]

Game index k	Black player ID b_k	White player ID w_k
1	1	2
2	2	3
3	3	1

Draw a directed graph to represent the joint distribution over the skills of the three players s_1, s_2 and s_3 and the three match outcomes $r^{(1)}$, $r^{(2)}$ and $r^{(3)}$.

[Expected response: Labelled directed graph.]

(b) Using the directed graph from part (a) and the rules for assessing conditional independence in directed graphical models, state and explain whether s_1 and s_2 are conditionally independent if we know the value of $r^{(2)}$. Similarly state and explain if s_1 and s_2 are conditionally independent if we know $r^{(2)}$ and $r^{(3)}$. Interpret the results in terms of the Go game. [5 MARKS]

[Expected response: Two conditional independency statements, justifications, and interpretation]

(c) Draw an undirected minimal I-map for the directed graphical model specified by the DAG in part (a). [2 MARKS]

[Expected response: Labelled undirected graph.]

(d) Fill in a table like that shown in Table 1. Use a tick or cross to indicate whether the independence relation shown in the table row generally holds under the given model (**UGM** is the Undirected Graphical Model, **DGM** is the Directed Graphical Model). Provide a short justification for your answer, e.g. UGM: path a-b-c is not blocked, DGM: blocked by conditioning on d. [8 MARKS]

[Expected response: A filled out table like that provided]

Independence	**UGM**	**DGM**	**Justification**
$r^{(1)} \perp\!\!\!\perp \{r^{(2)}, r^{(3)}\} \mid \{s_1, s_2, s_3\}$	✓	✓	UGM: cond. on s_1 and s_2 blocks all paths from $r^{(1)}$. DGM: ditto because s_1 and s_2 in diverging configuration.
$s_1 \perp\!\!\!\perp r^{(2)} \mid \{s_2, s_3\}$			
$s_1 \perp\!\!\!\perp r^{(2)} \mid \{r^{(1)}, s_3\}$			
$s_1 \perp\!\!\!\perp s_2$			
$s_1 \perp\!\!\!\perp s_2 \mid r^{(1)}$			
$r^{(1)} \perp\!\!\!\perp r^{(2)}$			
$r^{(1)} \perp\!\!\!\perp r^{(2)} \mid s_2$			
$r^{(1)} \perp\!\!\!\perp r^{(2)} \mid \{s_1, s_2\}$			
$r^{(1)} \perp\!\!\!\perp r^{(2)} \mid \{s_2, r^{(3)}\}$			

Table 1: The table to fill out for Question (d)

Question 2: Inference in a discrete-valued skill model (25 marks)

For this question, we assume that the skill of the players is represented as a discrete random variable taking values between $1, 2, \dots, 5$, where 1 is the lowest skill level and 5 the highest.

(a) How does the joint probability mass function (pmf) $\mathbb{P}\left[s_1, s_2, s_3, r^{(1)}, r^{(2)}, r^{(3)}\right]$ over the skills s_1, s_2, s_3 and the three match outcomes in the table of Question 1(a) factorise? [2 MARKS]

[Expected response: Factorisation of the joint pmf with explanation of the terms involved.]

(b) Observe the undirected factor graph given in Figure 2 and understand what it represents. Write down the equation for each factor. [3 MARKS]

[Expected response: Six equations, one for each factor.]

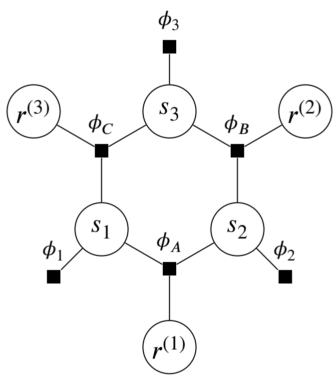


Figure 2: Undirected factor graph representing the joint pmf $\mathbb{P}\left[s_1, s_2, s_3, r^{(1)}, r^{(2)}, r^{(3)}\right]$, i.e. three games between three players.

(c) Assume you observe the following outcomes for the three games [4 MARKS]

$$r^{(1)} = 1, \qquad r^{(2)} = 0, \qquad r^{(3)} = 1. \tag{1}$$

Draw the factor graph for $\mathbb{P}\left[s_1, s_2, s_3 | r^{(1)} = 1, r^{(2)} = 0, r^{(3)} = 1\right]$ and define or provide equations for all factors in the graph.

[Expected response: Labelled undirected factor graph with definitions or equations for the factors.]

We now consider the situation where we have reasonable knowledge about the skills of the first two players, but know nothing about the skill of the third player. Our knowledge about the skills is represented by the probability mass functions $\mathbb{P}\left[s_i\right]$ whose values are shown below as vectors such that the k-th element of the vector equals $\mathbb{P}\left[s_i = k\right]$,

$$\mathbb{P}\left[s_1\right] = \begin{pmatrix} 0.01 \\ 0.01 \\ 0.08 \\ 0.2 \\ 0.7 \end{pmatrix}, \qquad \mathbb{P}\left[s_2\right] = \begin{pmatrix} 0.02 \\ 0.02 \\ 0.06 \\ 0.3 \\ 0.6 \end{pmatrix}, \qquad \mathbb{P}\left[s_3\right] = \begin{pmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \end{pmatrix}. \tag{2}$$

For example, $\mathbb{P}\left[s_1 = 3\right] = 0.08$. The goal of the following questions is to use probabilistic inference to update the belief about the third player's skill given the outcomes of the three games. In other words, we will compute the posterior pmf

$$\mathbb{P}\left[s_3 | r^{(1)} = 1, r^{(2)} = 0, r^{(3)} = 1\right].$$

We assume that the conditional probability for black winning in any game is given by

$$\mathbb{P}\left[r^{(k)} = 1 | s_{b_k}, s_{w_k}\right] = 0.0052(s_{b_k} - s_{w_k})^3 + 0.0292(s_{b_k} - s_{w_k}) + 0.5. \tag{3}$$

The conditional probability is plotted in Figure 3 as a function of the difference between the skill of the black and white player.

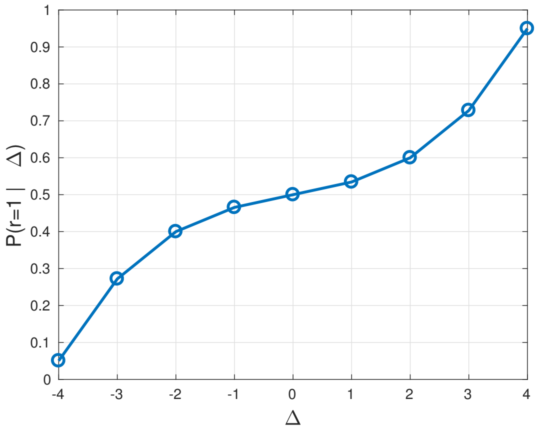


Figure 3: The conditional probability for black winning depends on the difference $\Delta = s_b - s_w$ between the skill of the black and white player $\mathbb{P}\left[r = 1 \mid s_b, s_w\right] = \mathbb{P}\left[r = 1 \mid \Delta\right]$. The conditional probability is defined in (3) and plotted here for reference as a function of Δ.

(d) Can (standard) message passing be used to compute $\mathbb{P}\left[s_3 | r^{(1)} = 1, r^{(2)} = 0, r^{(3)} = 1\right]$? [1 MARKS]

[Expected response: Yes/no answer with justification.]

(e) We can marginalise over s_1 to obtain [7 MARKS]

$$\mathbb{P}\left[s_2, s_3 | r^{(1)} = 1, r^{(2)} = 0, r^{(3)} = 1\right]$$

from

$$\mathbb{P}\left[s_1, s_2, s_3 | r^{(1)} = 1, r^{(2)} = 0, r^{(3)} = 1\right].$$

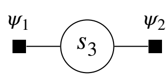
Show that summing out s_1 eliminates some nodes in the factor graph and creates a new factor $\tilde{\phi}_1(s_2, s_3)$. Provide the equation defining $\tilde{\phi}_1(s_2, s_3)$.

Draw the factor graph representing $\mathbb{P}\left[s_2, s_3 | r^{(1)} = 1, r^{(2)} = 0, r^{(3)} = 1\right]$ and provide the numerical values for $\tilde{\phi}_1(s_2, s_3)$ by filling in the table below.

[Expected response: Labelled undirected factor graph, filled-in table, derivation, snippet of code for numerical evaluation]

$s_2 \setminus s_3$	1	2	3	4	5
1					
2					
3					
4					
5					

(f) The desired posterior $\mathbb{P}\left[s_3 | r^{(1)} = 1, r^{(2)} = 0, r^{(3)} = 1\right]$ can be represented with the following factor graph. [4 MARKS]



Derive expressions for $\psi_1(s_3)$ and $\psi_2(s_3)$, and evaluate them for $s_3 \in \{1, 2, \ldots, 5\}$.

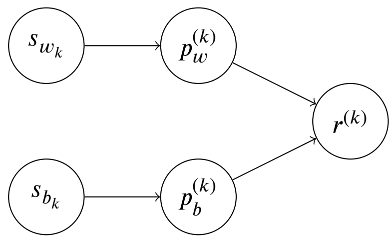
[Expected response: Two vectors with the values of $\psi_1(s_3)$ and $\psi_2(s_3)$. Rescale both vectors such that the maximal value of the elements is one. Derivation of the result and code snippet showing numerical evaluation.]

(g) Indicate the numerical values of the posterior $\mathbb{P}\left[s_3 | r^{(1)} = 1, r^{(2)} = 0, r^{(3)} = 1\right]$ for $s_3 \in \{1, 2, \ldots, 5\}$. Make a plot that compares $\mathbb{P}\left[s_3 | r^{(1)} = 1, r^{(2)} = 0, r^{(3)} = 1\right]$ with $\mathbb{P}\left[s_3\right]$, explain the differences between the prior and posterior. [4 MARKS]

[Expected response: Vector with values of the posterior. Explanation and figure with the plot.]

Question 3: Gaussian player skill model (25 marks)

We will now consider a more concrete model relating the skills of M Go players to the results of N games between them. In particular we will consider the model described by the graph in Figure 4 and work with real-valued skill variables. The skill variables are modelled as having zero-mean Gaussian prior distributions with fixed variance σ^2. As well as the underlying per-player skill variables, we will also model the players in a particular game as having per-game *performances* $p_b^{(k)}$ and $p_w^{(k)}$ for the black and white players in game k respectively. The performances are Gaussian distributed with a constant variance β^2 and means equal to the player skills.



$$
\begin{aligned}
s_{b_k} &\sim \mathcal{N}\left(s_{b_k};\, 0,\, \sigma^2\right)\\
s_{w_k} &\sim \mathcal{N}\left(s_{w_k};\, 0,\, \sigma^2\right)\\
p_b^{(k)} &\sim \mathcal{N}\left(p_b^{(k)};\, s_{b_k},\, \beta^2\right)\\
p_w^{(k)} &\sim \mathcal{N}\left(p_w^{(k)};\, s_{w_k},\, \beta^2\right)\\
r^{(k)} &= \begin{cases} 1 & \text{if } p_b^{(k)} > p_w^{(k)}\\ 0 & \text{if } p_b^{(k)} \leq p_w^{(k)}\end{cases}
\end{aligned}
$$

Figure 4: Gaussian player skill model for a single game. The variables $p_b^{(k)}$ and $p_w^{(k)}$ are the performances of the black and white player respectively in game k.

Our simple model is that the player whose performance is higher in the game is the winner. Therefore if $p_b^{(k)} > p_w^{(k)}$ then the black player is the winner and $r^{(k)} = 1$. If $p_b^{(k)} \leq p_w^{(k)}$ then the white player is the winner and $r^{(k)} = 0$. Note the slight asymmetry in defining the case when $p_b^{(k)} = p_w^{(k)}$ as a win for white will not matter in practice as the probability of this occurring exactly is zero.

We will assume that the result of each game is independently and identically distributed given the player skills and performances. To begin with in parts (a)–(c) we will consider a single game and so for the sake of clarity and brevity will drop the game index k from notation.

(a) Using Figure 4, write down an expression for $\mathbb{p}\left[p_b, p_w \mid s_b, s_w\right]$, the probability density across the player performance levels given known player skill levels. [2 MARKS]

[Expected response: Factorised expression for $\mathbb{p}\left[p_b, p_w \mid s_b, s_w\right]$.]

(b) We will now define a change of variables using [7 MARKS]

$$
\psi = \frac{(p_w - s_w) - (p_b - s_b)}{\sqrt{2}\beta} \qquad \theta = \frac{(p_w - s_w) + (p_b - s_b)}{\sqrt{2}\beta}. \tag{4}
$$

Calculate the conditional expectations $\mathbb{E}\left[\theta \mid s_b, s_w\right]$, $\mathbb{E}\left[\psi \mid s_b, s_w\right]$, $\mathbb{E}\left[\theta^2 \mid s_b, s_w\right]$, $\mathbb{E}\left[\psi^2 \mid s_b, s_w\right]$ and $\mathbb{E}\left[\theta\psi \mid s_b, s_w\right]$ (the expectations are taken with respect to $\mathbb{p}\left[p_b, p_w \mid s_b, s_w\right]$ in all cases). Hence write down the conditional density $\mathbb{p}\left[\theta, \psi \mid s_b, s_w\right]$.

[Expected response: Values for each of the five conditional expectations including working to show how you got results. An expression for $\mathbb{p}\left[\theta, \psi \mid s_b, s_w\right]$ and explanation of how you arrived at it.]

(c) Show that [5 MARKS]

$$
\mathbb{P}\left[r = 1 \mid s_b, s_w\right] = \Phi\left[\frac{s_b - s_w}{\sqrt{2}\beta}\right]. \tag{5}
$$

[Expected response: Derivation of result shown with explanation of steps taken.]

(d) Now considering all N games and M players, write down an expression for $\mathbb{P}\left[\left\{r^{(k)}\right\}_{k=1}^{N} \mid \left\{s_i\right\}_{i=1}^{M}\right]$, the probability of observing a set of game results $\left\{r^{(k)}\right\}_{k=1}^{N}$ between M players with known skills $\left\{s_i\right\}_{i=1}^{M}$. [4 MARKS]

[Expected response: Expression for $\mathbb{P}\left[\left\{r^{(k)}\right\}_{k=1}^{N} \mid \left\{s_i\right\}_{i=1}^{M}\right]$ with any assumptions used to derive this result explicitly stated and explained.]

(e) Show that the posterior density on player skills given observed game outcomes can be written as [7 MARKS]

$$\mathbb{p}\left[\left\{s_i\right\}_{i=1}^{M} \mid \left\{r^{(k)}\right\}_{k=1}^{N}\right] \propto \prod_{k=1}^{N} \left\{\Phi\left[\boldsymbol{s}^{\mathrm{T}}\boldsymbol{x}^{(k)}\right]\right\} \mathcal{N}\left(\boldsymbol{s}; \boldsymbol{m}_0, \mathbf{V}_0\right) \tag{6}$$

where $\boldsymbol{s} = \left[s_1 \ s_2 \ \dots \ s_M\right]^{\mathrm{T}}$ is a vector with the skills of the M players. Give expressions for $\boldsymbol{x}^{(k)}$, $\boldsymbol{m}_0$ and $\mathbf{V}_0$. You may find the identity $\Phi(-x) = 1 - \Phi(x)$ useful.

[Expected response: Derivation showing how to get from your answer to the previous question to an expression for the posterior density in the form given and identification of how to express the new variables $\boldsymbol{x}^{(k)}$, $\boldsymbol{m}_0$ and $\mathbf{V}_0$ in terms of the variables defined in the graph in Figure 4.]

Question 4: Approximate inference with Go game data (30 marks)

The posterior density on the player skills we derived in the previous question from our model cannot be evaluated exactly. In particular we can only express the posterior density up to an unknown normalisation constant as the integral

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \mathbb{p}\left[\left\{s_i\right\}_{i=1}^{M}, \left\{r^{(k)}\right\}_{k=1}^{N}\right] \mathrm{d}s_1 \ldots \mathrm{d}s_M \tag{7}$$

does not have an analytic solution. If we want to perform Bayesian inference with our model, for example to predict the probability of the result $r^\star$ of a new game between two of the players $b_\star$ and $w_\star$, we need to be able to calculate values for integrals like

$$\mathbb{P}\left[r^\star = 1 \mid \left\{r^{(k)}\right\}_{k=1}^{N}\right] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbb{P}\left[r^\star = 1 \mid s_{b_\star}, s_{w_\star}\right] \mathbb{p}\left[s_{b_\star}, s_{w_\star} \mid \left\{r^{(k)}\right\}_{k=1}^{N}\right] \mathrm{d}s_{b_\star} \mathrm{d}s_{w_\star} \tag{8}$$

which are in terms of a (marginal) normalised posterior density $\mathbb{p}\left[s_{b_\star}, s_{w_\star} \mid \left\{r^{(k)}\right\}_{k=1}^{N}\right]$.

As we cannot perform exact inference in the model, we will therefore use an *approximate inference* method which will allow us to estimate integrals like (8). In particular here we will make use of *expectation propagation* [Minka, 2001]. All of the details you need to know about expectation propagation for the purposes of this assignment are covered in appendix A, however if you wish to find out more about the method, possible textbook references are [Bishop, 2006, pp. 505ff.], [Koller and Friedman, 2009, pp. 442ff.], [Murphy, 2012, pp. 787ff.] and [Barber, 2012, pp. 599ff.].

In the provided archive file `go_player_skill_model.tar.gz` there are `python` `numpy` `.npz` and `MATLAB` `.mat` data files containing the parameters of expectation propagation based Gaussian approximations to the posterior density over the skills of 1049 Go players (1048 of the top professional players and AlphaGo) given the results of 51 413 games between them. These can be loaded using either the `numpy.load` function in `python` or the `load` function in `MATLAB`.

The parameters of two different approximate densities are provided. The parameters in the `full_covar.*` files are for a Gaussian approximate density parametrised with a full $M \times M$ covariance matrix $\mathbf{V}$. The parameters in the `diag_covar.*` files are for Gaussian approximation using a diagonal covariance matrix $\mathbf{V} = \operatorname{diag} \boldsymbol{v} \;\; \boldsymbol{v} \in \mathbb{R}^M$. In both a full mean vector $\boldsymbol{m} \in \mathbb{R}^M$ was used. The full list of variables defined in each of the files and mappings between notation in this document and variable names is given in Table 2. Compared to the model in the previous question, we here work with $\beta = 1$.

Description	Symbolic representation	Variable name	Dimensions
Number of players	M	`n_players`	scalar
Number of games	N	`n_games`	scalar
Overall mean vector	$\boldsymbol{m}$	`approx_mean`	M
Overall covariance matrix	$\boldsymbol{v}$ OR $\mathbf{V}$	`approx_covar`	M OR M × M
Skill prior variance	σ^2	`skill_prior_var`	scalar

Table 2: Relationships between variable symbols used in this document and variable names in `.npz` / `.mat` files.

(a) The true and approximate factors $f_k(\boldsymbol{s}) = \Phi\left[\boldsymbol{s}^{\mathrm{T}}\boldsymbol{x}^{(k)}\right]$ and $\tilde{f}_k(\boldsymbol{s}) = \mathcal{N}\left(\boldsymbol{s}; \boldsymbol{m}_k, \mathbf{V}_k\right)$ are defined over the 1049 dimensional player skill space. However the true factor only varies along one direction in that high-dimensional space — in parallel to $\boldsymbol{x}^{(k)}$. To visualise the fit between the approximate and true factors we can therefore plot the variation in the true and approximate factors along this line. [4 MARKS]

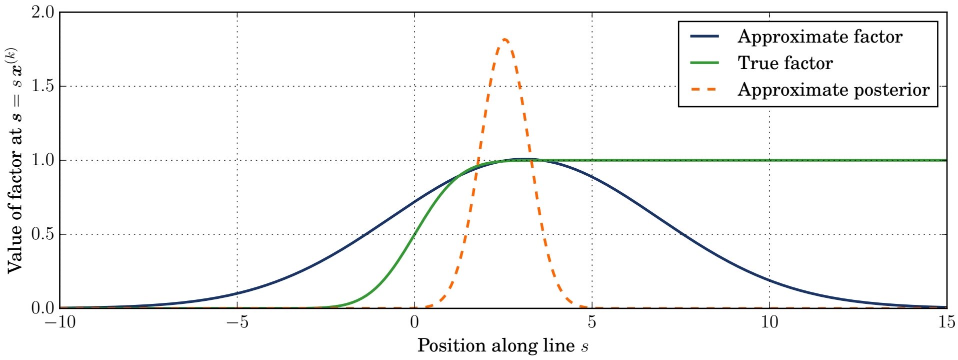


Figure 5: Values of corresponding approximate and true factors and scaled value of approximate posterior density along line $\boldsymbol{s} = s\,\boldsymbol{x}^{(k)}$ for a single factor in the full covariance approximate Gaussian density.

Figure 5 shows one such plot, with three values plotted

- Blue: value of the approximate factor $\tilde{f}_k(\boldsymbol{s})$
- Green: value of the true factor $\Phi(\boldsymbol{s}^{\mathrm{T}}\boldsymbol{x}^{(k)})$
- Orange: value of overall approximation density $\mathcal{N}(\boldsymbol{s};\, \boldsymbol{m},\, \mathbf{V})$ scaled to make it visible

along the line $\boldsymbol{s} = s\,\boldsymbol{x}^{(k)} \,:\, s \in [-10, 15]$ for a single factor in the full-covariance Gaussian approximate posterior skill density corresponding to one of the AlphaGo–Fan Hui games. Interpret what is shown in the plot in the context of how expectation propagation iteratively fits each of the approximate factors.

[Expected response: An explanation of the relationships shown between the curves in the provided plot.]

(b) In the data files containing the fitted model parameters, there are also three integer variables `alpha_go_id`, `lee_sedol_id` and `ke_jie_id` defined. These contain the player IDs / indices corresponding to AlphaGo, Lee Sedol and Ke Jie respectively. Plot the approximate posterior marginal skill densities for each of these three players on the same axis for both the full covariance and diagonal covariance models. Interpret what you see in the plots. [9 MARKS]

[Expected response: Two plot figures - one for each of the full covariance approximation and diagonal covariance approximation, with the three specified approximate posterior marginal distributions plotted on the same axis in each plot. A code snippet showing how you produced the plots. An interpretation of what the distributions shown mean.]

(c) Using (12), calculate the predicted probability of AlphaGo winning when playing as black in a single game against Ke Jie, using both the full and diagonal covariance approximate Gaussian posterior densities. [4 MARKS]

[Expected response: Two predicted win probabilities i.e. values in $[0, 1]$*, one for each of the two approximate densities. A code snippet showing how you calculated these values.]*

(d) State the number of independent parameters for both the full-covariance and diagonal-covariance EP approximation defined in Equations (9) and (10). [4 MARKS]

[Expected response: The number of parameters for each case with justification]

(e) Suggest one advantage and one disadvantage of using expectation propagation with a Gaussian approximating density that is parametrised with a full covariance matrix rather than a diagonal covariance matrix. [2 MARKS]

[Expected response: A positive and negative aspect of using the full covariance Gaussian approximation to the posterior density compared to the diagonal covariance approximation, for example comparing in terms of accuracy or computational cost, including justifications for any assertions made.]

(f) How realistic is the Gaussian skill model? Identify at least two limitations. How would you change the model so that it becomes better at predicting the outcomes of upcoming games? Discuss possible drawbacks of the new model. [7 MARKS]

[Expected response: Concise answers to the questions possibly backed up by suitable graphics]

A Introduction to expectation propagation

For compactness in the description that follows we use the shorthand $p(\boldsymbol{s})$ for the true posterior density we wish to approximate.

Expectation propagation is a form of approximate message passing which generalises loopy belief propagation. The parameters of an approximating density $q(\boldsymbol{s})$, which is chosen from the exponential family, are iteratively updated so as to minimise a measure of the difference between the approximate and true posterior densities.

We can express our true posterior density as a product of factors $p(\boldsymbol{s}) \propto \prod_{k=0}^{N} \{f_k(\boldsymbol{s})\}$ with here $f_0(\boldsymbol{s}) = \mathcal{N}(\boldsymbol{s}, \boldsymbol{m}_0, \mathbf{V}_\mathbf{0})$ and $f_k(\boldsymbol{s}) = \Phi\left[\boldsymbol{s}^\mathrm{T}\boldsymbol{x}^{(k)}\right] \quad \forall k \in \{1 \dots N\}$. In expectation propagation we assume our approximate density factorises equivalently

$$q(\boldsymbol{s}) = \prod_{k=0}^{N} \{\tilde{f}_k(\boldsymbol{s})\} \tag{9}$$

with each of the approximate density factors $\tilde{f}_k(\boldsymbol{s})$ an (unnormalised) exponential family distribution itself, with a property of exponential family meaning that the product of exponential family distributions is always another exponential family distribution.

Here we will choose for each of the approximate terms to be scaled Gaussian densities $\tilde{f}_k(\boldsymbol{s}) \propto \mathcal{N}(\boldsymbol{s}; \boldsymbol{m}_k, \mathbf{V}_k)$, with in this case we being able to exactly incorporate the prior using $\tilde{f}_0(\boldsymbol{s}) = f_0(\boldsymbol{s}) = \mathcal{N}(\boldsymbol{s}; \boldsymbol{m}_0, \mathbf{V}_0)$ and the overall approximate density itself being Gaussian

$$q(\boldsymbol{s}) = \mathcal{N}(\boldsymbol{s}; \boldsymbol{m}, \mathbf{V}) \qquad \text{with } \mathbf{V} = \left[\sum_{k=0}^{N} \left(\mathbf{V}_k^{-1}\right)\right]^{-1} \text{ and } \boldsymbol{m} = \mathbf{V} \sum_{k=0}^{N} \left(\mathbf{V}_k^{-1}\boldsymbol{m}_k\right) \tag{10}$$

One way we might fit our approximate distribution is to individually fit each of our approximate factors $\tilde{f}_k$ to the corresponding true factor f_k. In practice this does not work very well as even if each of the individual factors are well approximated this does not mean the overall approximation will be good.

Expectation propagation instead optimises each approximate factor to fit to the true factor in the context of all the remaining factors. Iteratively for each factor $j \in \{1 \dots N\}$ it sets the approximate factor parameters so as to minimise a 'distance'[2] between the densities proportional to $\tilde{f}_j(\boldsymbol{s}) \prod_{k=0,k\neq j}^{N} \tilde{f}_k(\boldsymbol{s})$ and $f_j(\boldsymbol{s}) \prod_{k=0,k\neq j}^{N} \tilde{f}_k(\boldsymbol{s})$, thus making the approximate factor $\tilde{f}_j$ be a good fit to the true factor f_j in regions of the state space where the remaining approximate density factors currently suggest most of the posterior probability mass is.

Although this is not guaranteed to lead to an optimal overall fit between the true and approximate posterior densities, in practice this method usually leads to a good fit with the algorithm converging to a fixed point after a few sequential passes across all the factors.

Given an approximate posterior density fitted using expectation propagation we can then use the identity

$$\int_{\mathbb{R}^M} \Phi\left[\boldsymbol{s}^\mathrm{T}\boldsymbol{x}\right] \mathcal{N}(\boldsymbol{s}; \boldsymbol{m}, \mathbf{V})\, \mathrm{d}\boldsymbol{s} = \Phi\left[\frac{\boldsymbol{m}^\mathrm{T}\boldsymbol{x}}{\sqrt{\boldsymbol{x}^\mathrm{T}\mathbf{V}\boldsymbol{x} + 1}}\right] \tag{11}$$

to approximately compute predictive probabilities of the form given in (8)

$$\mathbb{P}\left[r^\star = 1 \mid \left\{r^{(k)}\right\}_{k=1}^{N}\right] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathbb{P}\left[r^\star = 1 \mid s_{b_\star}, s_{w_\star}\right] \mathbb{p}\left[s_{b_\star}, s_{w_\star} \mid \left\{r^{(k)}\right\}_{k=1}^{N}\right] \mathrm{d}s_{b_\star}\mathrm{d}s_{w_\star} \tag{12}$$

$$= \int_{\mathbb{R}^M} \mathbb{P}\left[r^\star = 1 \mid \boldsymbol{s}\right] \mathbb{p}\left[\boldsymbol{s} \mid \left\{r^{(k)}\right\}_{k=1}^{N}\right] \mathrm{d}\boldsymbol{s} \tag{13}$$

$$\approx \int_{\mathbb{R}^M} \Phi\left[\boldsymbol{s}^\mathrm{T}\boldsymbol{x}^\star\right] \mathcal{N}(\boldsymbol{s}; \boldsymbol{m}, \mathbf{V})\, \mathrm{d}\boldsymbol{s} \tag{14}$$

$$= \Phi\left[\frac{\boldsymbol{m}^\mathrm{T}\boldsymbol{x}^\star}{\sqrt{\boldsymbol{x}^{\star\mathrm{T}}\mathbf{V}\boldsymbol{x}^\star + 1}}\right] \tag{15}$$

[2]The 'distance' measure is the Kullback–Leibler divergence or relative entropy which is not a true distance measure as it is non-symmetric.

References

D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

R. Coulom. Whole-history rating: A Bayesian rating system for players of time-varying strength. In *Computers and games*, pages 113–124. Springer, 2008.

P. Dangauthier, R. Herbrich, T. Minka, and T. Graepel. Trueskill through time: Revisiting the history of chess. In *Advances in Neural Information Processing Systems*, pages 337–344, 2007.

R. Herbrich, T. Minka, and T. Graepel. TrueSkill: A Bayesian skill rating system. In *Advances in Neural Information Processing Systems*, pages 569–576, 2006.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

T. P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

K. P. Murphy. *Machine Learning: a Probabilistic Perspective*. MIT press, 2012.