Probabilistic DAG Search (Supplementary material)

Julia Grosse[1] **Cheng Zhang**[2] **Philipp Hennig**[1,3]

[1]University of Tübingen, Tübingen, Germany
[2]Microsoft Research, Cambridge, United Kingdom,
[3]Max-Planck-Institute for Intelligent Systems, Tübingen, Germany,

1 A GAUSSIAN APPROXIMATION OF THE MAXIMUM OF GAUSSIAN DISTRIBUTED VARIABLES

Here we give the moments used for the approximations in Section 3 of the paper for the sake of completeness. Their derivations and an extended analysis of the approximation quality of the moment matching can be found in Hennig [2009].

Assume two jointly Gaussian distributed variables x_1 and x_2 with mean $\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$ and covariance $\Sigma = \begin{pmatrix} \sigma_1^2 & \varrho\sigma_1\sigma_2 \\ \varrho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$, where ϱ is their correlation. Given prior information $\mathcal{N}(\mu_0, \sigma_0^2)$ on their maximum m, the moments $\mu_m, \sigma_m^2$of the distribution of m are:

$$\begin{aligned}
\mu_m &= w_1\left[\mu_{c1} + \sigma_{c1}\frac{b_1}{a_1}\frac{\phi(k_1)}{\Phi(k_1)}\right] + w_2\left[\mu_{c2} + \sigma_{c2}\frac{b_2}{a_2}\frac{\phi(k_2)}{\Phi(k_2)}\right] && (1) \\
\sigma_m^2 &= w_1\left\{[2\mu_{c1}\sigma_{c1}] + \left[2\mu_{c1}\sigma_{c1}\frac{b_1}{a_1} - k_1\sigma_{c1}^2\frac{b_1^2}{a_1^2}\right]\frac{\phi(k_1)}{\Phi(k_1)}\right\} \\
&\quad + w_2\left\{[\mu_{c2}^2 + \sigma_{c2}^2] + \left[2\mu_{c2}\sigma_{c2}\frac{b_2}{a_2} - k_2\sigma_{c2}^2\frac{b_2^2}{a_2^2}\frac{\phi(k_2)}{\Phi(k_2)}\right]\frac{\phi(k_2)}{\Phi(k_2)}\right\}, && (2) \\
&\quad - \mu_m^2
\end{aligned}$$

where ϕ is the standard normal probability density function and Φ is the standard normal cumulative distribution function.

Supplement for the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021).

The remaining terms are defined by:

$$w_1 = Z^{-1}\mathcal{N}(\mu_0; \mu_1, \sigma_0^2 + \sigma_1^2)\Phi(k_1) \tag{3}$$

$$w_2 = Z^{-1}\mathcal{N}(\mu_0; \mu_2, \sigma_0^2 + \sigma_2^2)\Phi(k_2) \tag{4}$$

$$a_1 = [\sigma_1^2\sigma_2^2(1 - \varrho^2) + (\sigma_1 - \varrho\sigma_2)^2\sigma_{c1}^2]^{1/2} \tag{5}$$

$$a_2 = [\sigma_1^2\sigma_2^2(1 - \varrho^2) + (\sigma_2 - \varrho\sigma 1)^2\sigma_{c2}^2]^{1/2} \tag{6}$$

$$b_1 = \sigma_{c1}(\sigma_1 - \varrho\sigma_2) \tag{7}$$

$$b_2 = \sigma_{c2}(\sigma_2 - \varrho\sigma_1) \tag{8}$$

$$\sigma_{c1}^2 = \frac{\sigma_1^2\sigma_0^2}{\sigma_{c1}^2 + \sigma_0^2} \tag{9}$$

$$\mu_{c1} = \left(\frac{\mu_1}{\sigma_1^2} + \frac{\mu_0}{\sigma_0^2}\right)\sigma_{c1}^2 \tag{10}$$

$$Z = \mathcal{N}(\mu_0; \mu_1, \sigma_0^2 + \sigma_1^2)\Phi(k_1) + \mathcal{N}(\mu_0; \mu_2, \sigma_0^2 + \sigma_2^2)\Phi(k_2) \tag{11}$$

$$k_1 = \frac{(\sigma_1 - \varrho\sigma_2)\mu_{c1} - \sigma_1\mu_{c1} - \sigma_1\mu_2 + \varrho\sigma_2\mu_1}{[\sigma_1^2\sigma_2^2(1 - \varrho^2) + (\sigma_1 - \varrho\sigma_2)^2\sigma_{c1}^2]^{1/2}} \tag{12}$$

$$k_2 = \frac{(\sigma_2 - \varrho\sigma_1)\mu_{c2} - \sigma_2\mu_1 + \varrho\sigma_1\mu_2}{[\sigma_1^2\sigma_2^2(1 - \varrho^2) + (\sigma_2 - \varrho\sigma_1)^2\sigma_{c2}^2]^{1/2}} \tag{13}$$

2 ADDITIONAL EXPERIMENTAL DETAILS

This section provides additional experimental details for the MNIST pixel selection experiment. The tree/dag search methods can be combined with arbitrary classifiers to evaluate the reward at terminal nodes as long as it can handle partial input. We use a simple, fast Convnet (see Section B.1). The heuristic used for the UCT in MNIST experiment is given in Section B.2.

2.1 CONVNET FOR MNIST EXPERIMENT

The Convnet is built from two convolutional layers of size $32 \times 3 \times 3$ and $64 \times 3 \times 3$ with stride 1 and zero padding. The second convolutional layer is followed by 2×2 max pooling and two fully connected layers of size 9216×128 and 128×10. The inner layers use ReLUs and the softmax function is applied to the output layer. The first convolutional layer is followed by a 0.25 dropout layer and the first fully connected layer by a 0.5 dropout layer. Images are normalized during training and evaluation such that the pixel values are zero mean and unit standard deviation. During training all but randomly selected pixels are turned off, i.e. set to zero. We use stochastic gradient descent with the adaptive learning rate method (ADADELTA) and parameters $\epsilon = 10^{-6}$, $\rho = 0.9$. The learning rate is set to $\lambda = 1$in the beginning and is reduced by a factor of $\gamma = 0.7$ after each epoch. The net is trained for 30 epochs with a batch size of 64.

2.2 RAVE HEURISTIC FOR UCT

UCT in its original form selects new nodes based on the UCB formula. At a MAX node j, the child i that maximizes

$$\mu_j + \beta\sqrt{\frac{ln(n_i)}{n_j}} \tag{14}$$

is chosen, where μ_j is the empirical mean of rewards from roll-outs passing through node j. The variables n_j and n_i count the visits of node i and j and are used together with hyperparameter β to control the exploration. The RAVE-heuristic, orginally proposed by Gelly and Silver [2007] and and adapted to feature selection settings by Gaudel and Sebag [2010], combines the above selection policy with local and global RAVE scores. The global RAVE score of pixel p $g - \text{RAVE}_p$ indicates the global relevance of p and is calculated as the average of the observed rewards at terminal nodes containing p:

$$g - \text{RAVE}_p = \frac{1}{|T_p|}\sum_{i \in T_p} r_i \tag{15}$$

where T_p denotes the set of observed terminal nodes that contain pixel p. The local RAVE-score represents the importance of pixel p at a node i, conditioned on the other pixels from node$i$. Here, the average is taken only from roll-outs that passed through node i.

$$l - \text{RAVE}_{i,p} = \frac{1}{|\{j \in T_p | i \rightsquigarrow j\}|} \sum_{j \in T_p, i \rightsquigarrow j} r_j \tag{16}$$

The modified score trades-off the original UCB term, the global RAVE score and the local one. It further includes a restriction of the exploration term in order to handle problems with high branching factor. The final score of a child j reached from parent i by adding pixel p is defined as:

$$(1-\alpha) \cdot \mu_j + \alpha \cdot ((1-\beta) \cdot l - \text{RAVE}_{i,p} + \beta \cdot g - \text{RAVE}_p) + \sqrt{\frac{c_1 \ln(n_i)}{n_j} \min\left(\frac{1}{4}, \sigma_j^2 + \sqrt{\frac{2\ln(n_i)}{n_j}}\right)} \tag{17}$$

σ_j^2 denotes the empirical variance of rewards observed from roll-outs passing through node j. The trade-off is controlled by parameters α and β:

$$\alpha = \frac{c_2}{c_2 + n_j} \tag{18}$$

$$\beta = \frac{c_3}{c_3 + n_l} \tag{19}$$

n_l denotes the number of observations that are taken into account for the calculation of the l-RAVE score. In the beginning of the search, the RAVE scores that are biased but calculated from more trials count more. In the course of time, their impact decreases and the mean estimate μ_j is valued more as it is unbiased, but requires more trials to be reliable. The heuristic involves three hyperparameters c_1, c_2, c_3 that were set to $c_1 = 10^{-4}, c_2 = 10^4$ and $c_3 = 10^4$, respectively.