Practical Bayesian optimization in the presence of outliers

Thomas Gaviard
Code available online at `https://github.com/tms-gvd/project_bl`
`thomas.gaviard@centrale.centralelille.fr`

Abstract

Bayesian optimization is a powerful tool for optimizing black-box functions. However, it is known to be sensitive to outliers. In this report, I present my study of the paper *Practical Bayesian optimization in the presence of outliers* by Martinez-Cantin et al. (2017) that proposes a new algorithm to filter outliers during the optimization.

1 Settings

The authors study **sample efficient optimization** that is a common task in many fields. It consists of minimizing a function $f : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ that is expensive to evaluate. For instance, f can be the error rate of a machine learning classifier model trained with a given set of hyperparameters. Then, the goal would be to find the hyperparameters that minimize a metric. Yet, evaluating f, i.e. training the model, is expensive and time-consuming that is why we want to minimize the number of evaluations of f.

Bayesian optimization (BO) is a powerful method for sample efficient optimization as it provides a black-box solution. To this end, it uses a probabilistic surrogate model of the function f that is cheap to evaluate and leverages the data observed so far during the optimization. This "memory" helps to sample efficiently and yields great performance.

The optimization is performed as follows:

- **Initialization:** Draw p samples at random from $\mathcal{X}$ and evaluate f at these points.
- **For** $t = 1, \dots, T$**:**
  - Use the previously observed data $\mathbf{y} = \mathbf{y}_{1:t}$ at points $\mathbf{X} = \mathbf{X}_{1:t}$ to fit a surrogate model s_t of f.
  - The next point to evaluate $\mathbf{x}_{t+1}$ is selected by optimizing an *acquisition function* over $\mathcal{X}$.
  - Finally, evaluate f at $\mathbf{x}_{t+1}$ and add the new observation y_t to $\mathcal{D}_t$.

The paper restricts their study to the expected improvement (EI) for the acquisition function, which is defined as follows:

$$EI(\mathbf{x}) = \mathbb{E}_{p(y|s_t(\mathbf{x}))}\left[\max(0, y^* - y)\right], \tag{1}$$

where $y^* = \max(y_1, \dots, y_t)$.

The authors assume that the data is observed under homoscedastic noise, i.e. $y_i = f(\mathbf{x}_i) + \epsilon_i$ where $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$. Then, they choose to use a **Gaussian Process** (GP) for the surrogate model s_t as it is a common choice. They consider a zero-mean GP with covariance $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Therefore, the likelihood can be written as $y|f \sim \mathcal{N}(f, \sigma_n^2)$ where $f \equiv f(\mathbf{x})$. A common choice will be to use a **Gaussian likelihood** which leads to a closed-form expression for the posterior distribution.

Yet, the authors also assume that the data can contain **outliers**. For the previous example of the machine learning model, an outlier could be a model that is not trained properly due to random bugs, I/O errors, convergence issues for some hyperparameters sets, etc. Nevertheless, BO is known to be sensitive to these outliers because of its "memory" component.

Particularly, the Gaussian likelihood is not robust to outliers. Indeed, the Gaussian distribution has light tails, i.e. it assigns low probability to events that are far from the mean. Therefore, the posterior distribution will be strongly

influenced by outliers. To overcome this issue, the authors propose to use a **Student-t likelihood** that has heavier tails. This means that it assigns more probability to events that are further from the mean. Thus, the posterior distribution will be less influenced by outliers.

2 Contributions

Algorithm 1 BO with outliers

Input: Total budget T, rejection threshold α

Initial design of p points (e.g.: LHS)
$\mathbf{X} \leftarrow \mathbf{x}_{1:p}$ $\mathbf{y} \leftarrow y_{1:p}$

for $t = p+1 \dots T$ **do**
  if schedule(t) **then**
    $\mathbf{\Theta_t} \leftarrow$ fitGPwithTlik$(\mathbf{X}, \mathbf{y})$
    $\mathbf{X}_{in}, \mathbf{y}_{in} \leftarrow$ filterOutliers$(\mathbf{X}, \mathbf{y}, \mathbf{\Theta_t}, \alpha)$
  if $length(\mathbf{y}_{in}) < \lfloor length(\mathbf{y})/2 \rfloor$ **or**
  not schedule(t) **then**
    $\mathbf{X}_{in} \leftarrow \mathbf{X}$ $\mathbf{y}_{in} \leftarrow \mathbf{y}$
  $\mathbf{\Theta_g} \leftarrow$ fitGPwithGlik$(\mathbf{X}_{in}, \mathbf{y}_{in})$
  $\mathbf{x}_t = \arg\max_{\mathbf{x}} EI(\mathbf{x}|\mathbf{X}_{in}, \mathbf{y}_{in}, \mathbf{\Theta_g})$
  $y_t \leftarrow f(\mathbf{x}_t)$ $\mathbf{X} \leftarrow add(\mathbf{x}_t)$ $\mathbf{y} \leftarrow add(y_t)$

I will present two aspects of the paper. First, I will re-derive the form of the posterior distribution with a Student-t likelihood and the Laplace approximation. Then, I will test the proposed strategy of the paper on simple examples.

3 Derivation of Laplace approximation for the Student-t Process

Reminder that for a GP with a Gaussian likelihood, the posterior distribution is in closed-form and the predictions at a query point $\mathbf{x}_q$ follows a Gaussian distribution with parameters:

$$\begin{aligned} \mu(\mathbf{x}_q) &= \mathbf{k}(\mathbf{x}_q, \mathbf{X})^T \mathbf{K}^{-1} \mathbf{y}, \\ \sigma^2(\mathbf{x}_q) &= k(\mathbf{x}_q, \mathbf{x}_q) - \mathbf{k}(\mathbf{x}_q, \mathbf{X})^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}_q, \mathbf{X}), \end{aligned} \tag{2}$$

where

$$\begin{aligned} \mathbf{k}(\mathbf{x}_q, \mathbf{X}) &= (k(\mathbf{x}_q, \mathbf{x}_1) \quad \dots \quad k(\mathbf{x}_q, \mathbf{x}_t))^T, \\ \mathbf{K} &= (\mathbf{k}(\mathbf{x}_1, \mathbf{X}) \quad \dots \quad \mathbf{k}(\mathbf{x}_t, \mathbf{X})) + \mathbf{I}\sigma_n^2. \end{aligned} \tag{3}$$

In the paper, the authors use a Matérn kernel with $\nu = 5/2$.

The authors use the following definition of the Student-t likelihood, with $f \equiv f(\mathbf{x})$.

$$t(y; f, \sigma_0, \nu) = \frac{\Gamma\left(\nu + \frac{1}{2}\right)}{\Gamma(\frac{\nu}{2})\sqrt{(\nu\pi)\sigma_0}} \left[1 + \frac{(y-f)^2}{\nu\sigma_0^2}\right]^{-(\nu+1)/2}, \tag{4}$$

The Laplace approximation for the conditional posterior of the latent function, which we write as $q(\mathbf{f}|\mathbf{y}, \mathbf{X}) \approx p(\mathbf{f}|\mathbf{y}, \mathbf{X})$ with $\mathbf{f} = (f(\mathbf{x}_i))_{1 \le i \le t}$, is constructed from the second order Taylor expansion of log posterior around the mode $\hat{\mathbf{f}} = \arg\max_{\mathbf{f}} p(\mathbf{f}|\mathbf{y}, \mathbf{X})$:

$$\log p(\mathbf{f}|\mathbf{y}, \mathbf{X}) = \log p(\hat{\mathbf{f}}|\mathbf{y}, \mathbf{X}) + \frac{1}{2}(\mathbf{f} - \hat{\mathbf{f}})^T \nabla\nabla_{\mathbf{f}} \log p(\mathbf{f}|\mathbf{y}, \mathbf{X})|_{\mathbf{f}=\hat{\mathbf{f}}} (\mathbf{f} - \hat{\mathbf{f}}) + o(||\mathbf{f} - \hat{\mathbf{f}}||^2)$$

as $\nabla_{\mathbf{f}} \log p(\mathbf{f}|\mathbf{y}, \mathbf{X})|_{\mathbf{f}=\hat{\mathbf{f}}} = 0$.

This can be simplified as follows keeping only the terms that depends on $\mathbf{f}$:

$$\log q(\mathbf{f}|\mathbf{y}, \mathbf{X}) = -\frac{1}{2} (\mathbf{f} - \hat{\mathbf{f}})^T \mathbf{\Sigma}_{\text{LA}}^{-1} (\mathbf{f} - \hat{\mathbf{f}}) + \text{const}$$

with $\mathbf{\Sigma}_{\text{LA}}^{-1} = -\nabla\nabla_{\mathbf{f}} \log p(\mathbf{f}|\mathbf{y}, \mathbf{X})|_{\mathbf{f}=\hat{\mathbf{f}}}$.

We recognize that:

$$q(\mathbf{f}|\mathbf{y}, \mathbf{X}) \propto \exp\left(-\frac{1}{2}\left(\mathbf{f} - \hat{\mathbf{f}}\right)^T \mathbf{\Sigma}_{\text{LA}}^{-1}\left(\mathbf{f} - \hat{\mathbf{f}}\right)\right)$$

meaning that the Laplace approximation of $\mathbf{f}|\mathbf{y}, \mathbf{X} \sim \mathcal{N}(\hat{\mathbf{f}}, \mathbf{\Sigma}_{\text{LA}})$.

Using $p(\mathbf{f}|\mathbf{y}, \mathbf{X}) \propto p(\mathbf{y}|\mathbf{f}, \mathbf{X})\, p(f|\mathbf{X})$, we simplify $\mathbf{\Sigma}_{\text{LA}}$ as follows keeping only the terms that depends on $\mathbf{f}$:

$$\begin{aligned}\log p(\mathbf{f}|\mathbf{y}, \mathbf{X}) &= \log p(\mathbf{y}|\mathbf{f}, \mathbf{X}) + \log p(\mathbf{f}|\mathbf{X}) \\ &= \log p(\mathbf{y}|\mathbf{f}, \mathbf{X}) + \left(-\frac{1}{2}\,\mathbf{f}^T\,\mathbf{K}^{-1}\,\mathbf{f}\right) \text{ as } \mathbf{f}|\mathbf{X} \sim \mathcal{N}(0, \mathbf{K}) \text{ by definition of the GP}\end{aligned}$$

Using $-\nabla\nabla_{\mathbf{f}}\left(-\frac{1}{2}\,\mathbf{f}^T\,\mathbf{K}^{-1}\,\mathbf{f}\right) = K^{-1}$, we can rewrite $\mathbf{\Sigma}_{\text{LA}}^{-1} = \mathbf{W} + \mathbf{K}^{-1}$ with $\mathbf{W} = -\nabla\nabla_{\mathbf{f}} \log p(\mathbf{y}|\mathbf{f}, \mathbf{X})|_{\mathbf{f}=\hat{\mathbf{f}}}$. $\mathbf{W}$ is diagonal since the likelihood factorizes over the data points, i.e. the distribution for y_i depends only on f_i.

Eventually, the likelihood follows a Student-t distribution then keeping only the terms that depends on $\mathbf{f}$:

$$\begin{aligned}-\log p(y_i|f_i, \mathbf{x}_i) &= \frac{\nu+1}{2}\,\log\left(1 + \frac{(y_i - f_i)^2}{\nu\sigma_0^2}\right) \\ -\nabla\nabla_{\mathbf{f}} \log p(y_i|f_i, \mathbf{x}_i) = \mathbf{W}_{i,i} &= -(\nu+1)\,\frac{r_i^2 - \nu\sigma_0^2}{(r_i^2 + \nu\sigma_0^2)^2}\end{aligned}$$

with $r_i = y_i - f_i$.

To make **predictions**, we need to compute the posterior distribution $p(f_q|\mathbf{y}, \mathbf{X}, \mathbf{x}_q)$. With the Laplace approximation, we have:

$$\begin{aligned}\mathbb{E}_q[f_q|\mathbf{X}, \mathbf{y}, \mathbf{x}_q] &= \int \mathbb{E}_q[\mathbf{f}|\mathbf{X}, \mathbf{y}, \mathbf{x}_q] p(\mathbf{f}|\mathbf{X}, \mathbf{y}) d\mathbf{f} \\ &= \int \mathbf{k}(\mathbf{x}_q, \mathbf{X})^T \mathbf{K}^{-1}\mathbf{f}\; p(\mathbf{f}|\mathbf{X}, \mathbf{y}) d\mathbf{f} \\ &= \mathbf{k}(\mathbf{x}_q, \mathbf{X})^T \mathbf{K}^{-1} \mathbb{E}[\mathbf{f}|\mathbf{X}, \mathbf{y}] \\ &= \mathbf{k}(\mathbf{x}_q, \mathbf{X})^T \mathbf{K}^{-1} \hat{\mathbf{f}} \end{aligned} \tag{5}$$

Similarly, we get a closed-form expression for the variance:

$$\mathbb{V}_q[f_q|\mathbf{X}, \mathbf{y}, \mathbf{x}_q] = k(\mathbf{x}_q, \mathbf{x}_q) - \mathbf{k}(\mathbf{x}_q, \mathbf{X})^T \left(K + W^{-1}\right)^{-1} \mathbf{k}(\mathbf{x}_q, \mathbf{X}) \tag{6}$$

For a given $\mathbf{X}, \mathbf{y}$, the mode $\hat{\mathbf{f}}$ can be computed using the expectation maximization (EM) algorithm as detailed in Vanhatalo et al. (2009).

4 Experiments

Firstly, I implemented the Gaussian Process with Student-t likelihood using Laplace approximation and check that it works as expected. Secondly, I implemented the proposed Algorithm 1 and tested it on simple examples.

4.1 Gaussian Process with Student-t likelihood

To check that the implementation of GP with Student-t likelihood works as expected, I generated two simple datasets with outliers. The first one is the 1D Forrester function defined as follows:

$$f(x) = (6x - 2)^2 \sin(12x - 4) \tag{7}$$

on the support $x \in [0, 1]$.

The second one is the 2D Branin function defined as follows:

$$f(x_1, x_2, l) = \left(x_2 - \left(\frac{5.1}{4\pi^2} - 0.1(1 - l)\right) x_1^2 + \frac{5}{\pi} x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10 \tag{8}$$

on the support $x_1 \in [-5, 10], x_2 \in [0, 15]$ with $l \in \{0, 1\}$ the degree of fidelity.

I compared the implementation with 3 other GPs:

- GP with Gaussian likelihood, without outliers
- GP with Gaussian likelihood, with outliers
- GP with Student-t likelihood, computed using ApproximateGP with Variational Inference from GPyTorch.

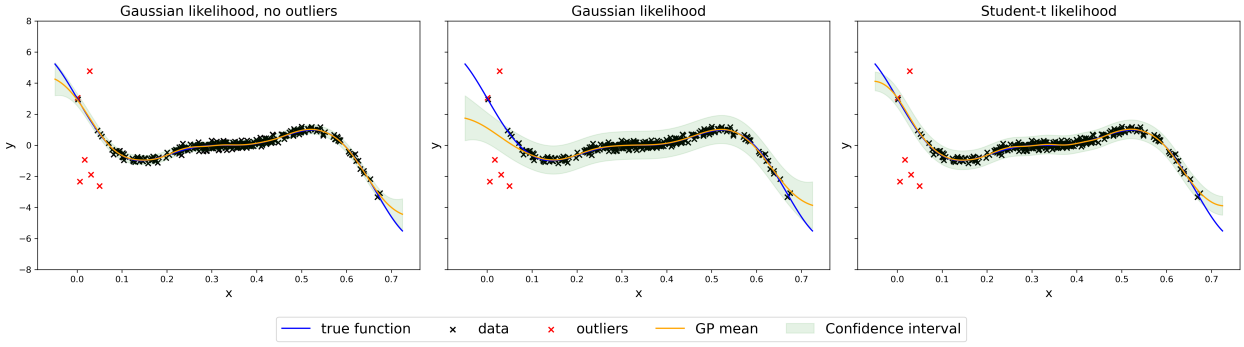

Figure 1: GP for different likelihoods in the presence or not of outliers, for the Forrester function. Points that were generated as outliers are represented in red.

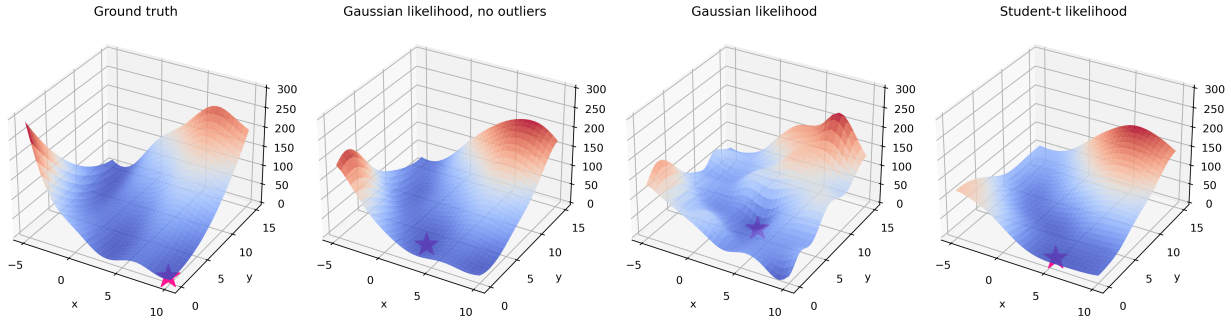

Figure 2: GP for different likelihoods in the presence or not of outliers, for the Branin function with $l = 0.8$. The pink star shows the minimum of the function.

1 and 2 show that the GP with Student-t likelihood is more robust to outliers than the GP with Gaussian likelihood. For the 1D example, the GP with Student-t likelihood is less influenced by the outliers in [0, 0.1] and the confidence interval is tighter on the whole domain than with a Gaussian likelihood. This even more noticeable for the 2D case.

To compute the GP with Student-t likelihood, I implemented the EM algorithm to get $\hat{\mathbf{f}}$ and $\mathbf{W}$ as detailed in Vanhatalo et al. (2009). I also implemented the Laplace approximation to compute the posterior distribution $p(f_q|\mathbf{y}, \mathbf{X}, \mathbf{x}_q)$. Yet, I did not finish the last part that was to optimize the hyperparameters. According to the previous paper, $p(\theta|\mathbf{y}, \mathbf{X})$ is differentiable with respect to θ and can be optimized using gradient descent. I believe that it can be done by writing a custom likelihood in GPyTorch using $\hat{\mathbf{f}}$ and $\mathbf{W}$.

4.2 Bayesian optimization with Student-t likelihood to filter outliers

Using the previous implementation of GP with Student-t likelihood, I implemented the proposed Algorithm 1 and tested it on the Forrester function and the Branin function with $l = 0.8$. I used the same acquisition function as in the paper, i.e. the expected improvement (EI) defined in (1). I did not tested it in higher dimension as the time to compute $\arg\max_{\mathbf{x}} \; EI(\mathbf{x}|\mathbf{X}_{in}, \mathbf{y}_{in}, \mathbf{\Theta}_{\mathbf{g}})$ grows exponentially with the dimension if we compute it naively by sampling points and evaluating the acquisition function at each point. Yet, as EI is differentiable with respect to $\mathbf{x}$, we can use gradient descent to optimize it. Yet, the function is not stricly convex and on the contrary has many local minima. Therefore, we need to use a good initialization to avoid getting stuck in a local minimum. Even though I did not have time to implement it, it is feasible thanks to BoTorch as they explain how to do it in this tutorial. The main idea is to choose a good initialisation to converge to the global minimum by sampling promising points and leveraging parallel computing.

I created animated gifs to show the optimization process for the Forrester function and the Branin function with $l = 0.8$. They are available online at 1D BO and 2D BO. We can observe that the filter is efficient to remove outliers like the outlier at (-0.7, 10) at the ninth iteration for the Forrester function. The oscillations between two states for the Branin function comes to the fact that we fit a Student-t GP every two steps. Then, it filters a lot of outliers and keep the most promising points that emphasize the area that we look at.

In addition, I studied the impact of two parameters for the Forrester function. Firstly, the impact of the rejection threshold α that is the threshold to consider a point as an inlier. It is computed as folows: all points that fall in the α-percentile are considered as outliers. The higher α is, the more points are considered as outliers. Figure 3 presents the evolution of $f(x_{best}) - f(x_t^*)$ during the optimization. We observe that extreme values for α are not efficient. Indeed, if α is too low, we do not filter enough outliers and the optimization is not robust. On the contrary, if α is too high, we filter too many points and the optimization is not efficient either, mostly because of the second if condition in the algortithm that says that there are at least half points that are inliers and if it is not the case during the BO, then it shoudl be a mistake and we take all the points as inliers for the current step.

Secondly, I studied the impact of the probability of outliers generated. Indeed, the outliers are generated randomly with a probability p_{out}. Figure 4 presents the evolution of $f(x_{best}) - f(x_t^*)$ during the optimization. We observe that the optimization is not robust to outliers if p_{out} is too high. It is relevant as in this case we generate too many outliers and the optimization is not robust. In addition, too many outliers can mislead the GP, even with Student-t likelihood.

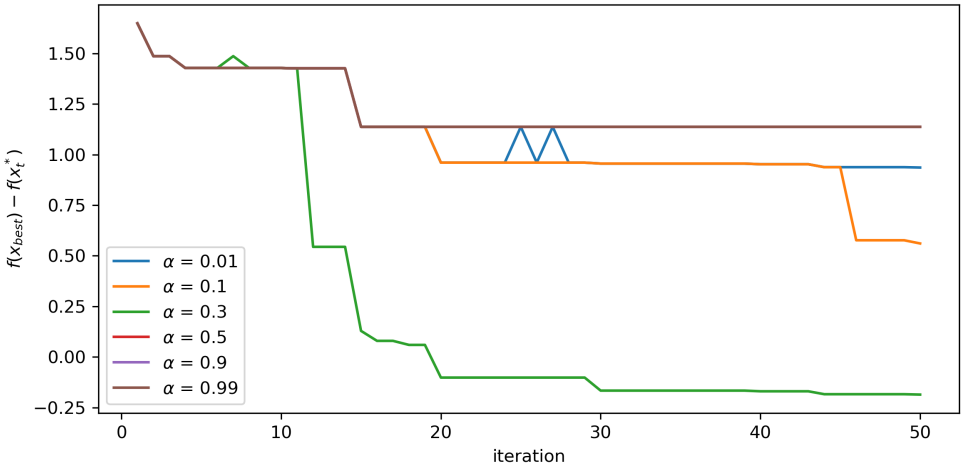

Figure 3: Impact of the rejection threshold α on the BO for the Forrester function.

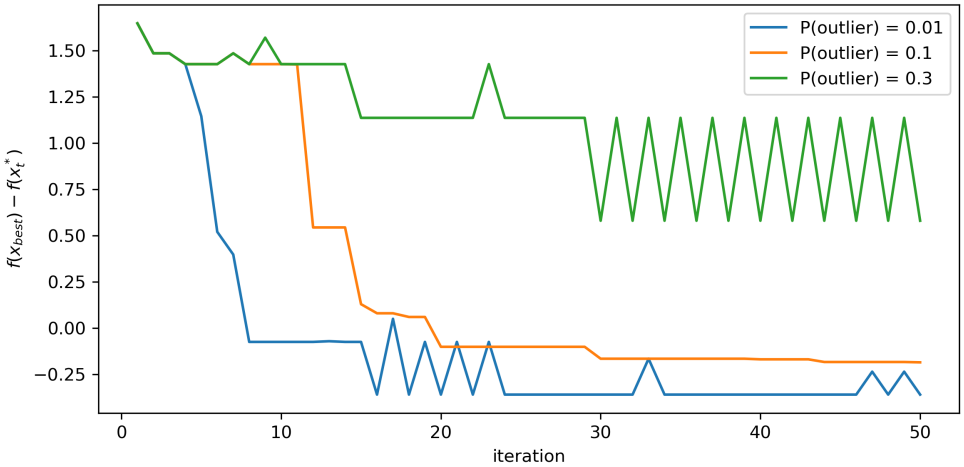

Figure 4: Impact of the probability of outliers generated on the BO for the Forrester function.

References

Martinez-Cantin, R., Tee, K., and McCourt, M. (2017). Practical Bayesian optimization in the presence of outliers.

Vanhatalo, J., Jylänki, P., and Vehtari, A. (2009). Gaussian process regression with Student-t likelihood. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A., editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.