Chapter 4
The Acquisition Function

The acquisition function is the mechanism to implement the trade-off between *exploration* and *exploitation* in BO. More precisely, any acquisition function aims to guide the search of the optimum towards points with potential low values of objective function either because the prediction of $f(x)$, based on the probabilistic surrogate model, is low or the uncertainty, also based on the same model, is high (or both). Indeed, *exploiting* means to consider the area providing more chance to improve the current solution (with respect to the current surrogate model), while *exploring* means to move towards less explored regions of the search space where predictions based on the surrogate model are more uncertain, with higher variance.

This chapter is devoted to present some of the most relevant acquisition functions, from the "traditional" towards the most recent ones.

4.1 Traditional Acquisition Functions

This section summarizes the most widely used acquisition functions.

4.1.1 Probability of Improvement

Probability of improvement (PI) was the first acquisition function proposed in the literature Kushner (1964):

$$PI(x) = P\left(f(x) \leq f\left(x^+\right)\right) = \boldsymbol{\Phi}\left(\frac{f\left(x^+\right) - \mu(x)}{\sigma(x)}\right)$$

F. Archetti and A. Candelieri, *Bayesian Optimization and Data Science*,
SpringerBriefs in Optimization, https://doi.org/10.1007/978-3-030-24494-1_4

where $f(x^+)$ is the best value of the objective function observed so far, $\mu(x)$ and $\sigma(x)$ are mean and standard deviation of the probabilistic surrogate model, such as a GP, and $\mathbf{\Phi}(\cdot)$ is the normal cumulative distribution function.

One of the drawbacks of PI is that it is biased towards exploitation. To mitigate this effect, one can introduce the parameter ξ which modulates the balance between exploration and exploitation. The resulting equation is:

$$\text{PI}(x) = P\big(f(x) \leq f(x^+) + \xi\big) = \mathbf{\Phi}\left(\frac{f(x^+) - \mu(x) - \xi}{\sigma(x)}\right)$$

Low values of ξ favour exploitation while large values exploration.

Finally, the next point to evaluate is chosen according to: $x_{n+1} = \text{argmax}_{x \in X} \text{PI}(x)$.

However, a weak point of PI is to assign a value to a new point irrespective of the potential magnitude of the improvement. This is the reason why the next acquisition function was proposed.

4.1.2 Expected Improvement

Expected improvement (EI) was proposed initially in Mockus et al. (1978) and then made popular in Jones et al. (1998) which measures the expectation of the improvement on $f(x)$ with respect to the predictive distribution of the probabilistic surrogate model.

$$\text{EI}(x) = \begin{cases} \big(f(x^+) - \mu(x)\big)\mathbf{\Phi}(Z) + \sigma(x)\phi(Z) \text{ if } \sigma(x) > 0 \\ 0 \text{ if } \sigma(x) = 0 \end{cases}$$

where $\phi(Z)$ and $\mathbf{\Phi}(Z)$ are the probability distribution and the cumulative distribution of the standardized normal, respectively, where

$$Z = \begin{cases} \frac{f(x^+) - \mu(x)}{\sigma(x)} \text{ if } \sigma(x) > 0 \\ 0 \text{ if } \sigma(x) = 0 \end{cases}$$

The EI is made up of two terms: the first is increased by decreasing the predictive mean; the second by increasing the predictive uncertainty. Thus, EI, in a sense, automatically balances, respectively, exploitation and exploration. When we want to actively manage the trade-off between exploration and exploitation, we can introduce the parameter ξ. When exploring, points associated with high uncertainty of the probabilistic surrogate model are more likely to be chosen, while when exploiting, points associated with low value of the mean of the probabilistic surrogate model are selected.

$$\mathrm{EI}(x) = \begin{cases} \left(f\left(x^{+}\right) - \mu(x) - \xi\right)\boldsymbol{\Phi}(Z) + \sigma(x)\phi(Z) \text{ if } \sigma(x) > 0 \\ 0 \text{ if } \sigma(x) = 0 \end{cases}$$

and

$$Z = \begin{cases} \frac{f\left(x^{+}\right) - \mu(x) - \xi}{\sigma(x)} \text{ if } \sigma(x) > 0 \\ 0 \text{ if } \sigma(x) = 0 \end{cases}$$

The following figure shows how the selected points change depending on ξ. Ideally, ξ should be adjusted dynamically to decrease monotonically with the function evaluations. Compared to PI (Fig. 4.1), it is possible to notice that EI (Fig. 4.2) is less biased towards exploitation.

Finally, the next point to evaluate is chosen according to: $x_{n+1} = \mathrm{argmax}_{x \in X}\, \mathrm{EI}(x)$.

EI has been largely used since 1998 and specialized to specific contexts: Astudillo and Frazier (2019) propose EI-CF a version of composite functions which leads to a multi-output GP: the authors also note that constrained optimization can be regarded as a special case of the optimization of composite functions and that EI-CF reduces

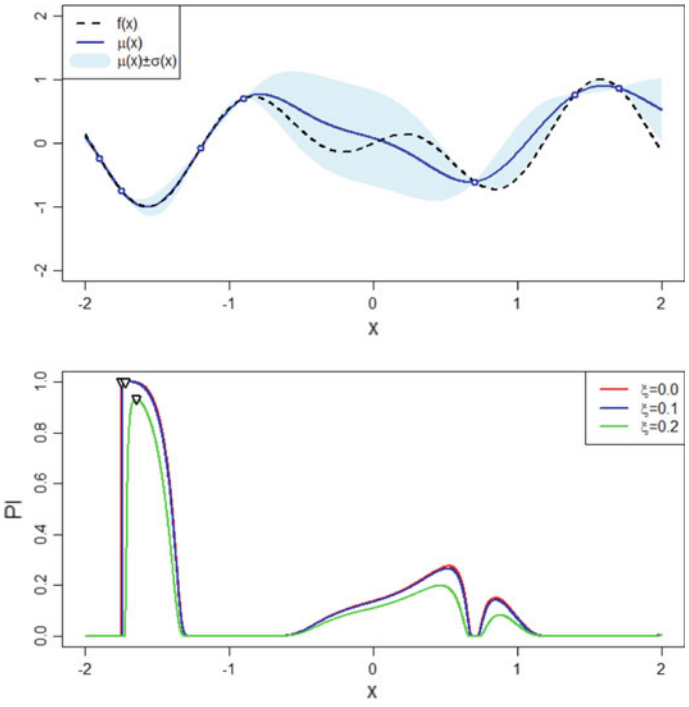

Fig. 4.1 GP trained depending on seven observations (top), PI with respect to different values of ξ and max values corresponding to the next point to evaluate (bottom)

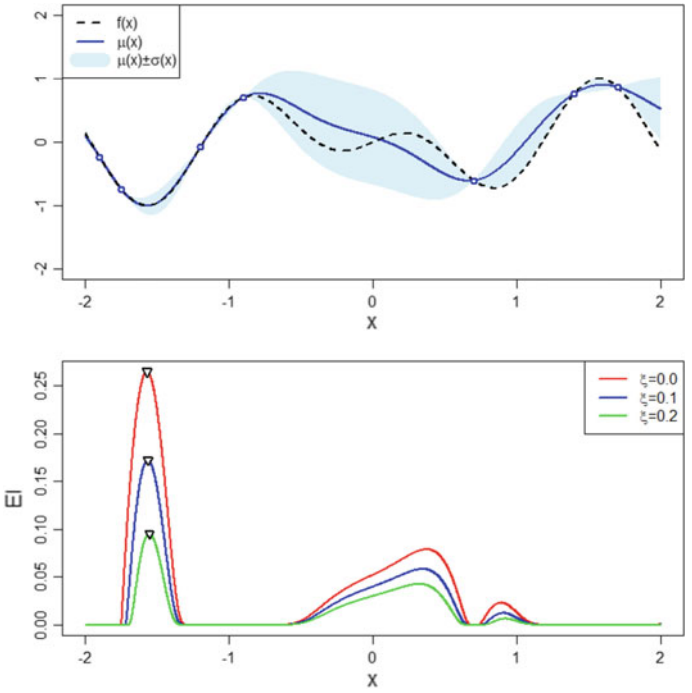

Fig. 4.2 GP trained depending on seven observations (top), EI with respect to different values of ξ and max values corresponding to the next point to evaluate (bottom)

to the expected improvement for constrained optimization. More importantly, both PI and EI are structurally exploitative, increasing ξ from 0 to 0.2 - which is the 10% of the min-max range of $f(x)$ - does not shift substantially x_{n+1}. Higher values of ξ flatten the acquisition function making it close to Random Search. To manage effectively the exploitation-exploration balance a more effective acquisition function is Upper-Lower Confidence Bound, given in the following section.

4.1.3 Upper/Lower Confidence Bound

Confidence bound—where upper and lower are used, respectively, for maximization and minimization problems—is an acquisition function that manages exploration—exploitation by being optimistic in the face of uncertainty, in the sense of considering the best-case scenario for a given probability value (Auer 2002).

For the case of minimization, LCB is given by:

$$\text{LCB}(x) = \mu(x) - \xi\sigma(x)$$

where $\xi \geq 0$ is the parameter to manage the trade-off between exploration and exploitation ($\xi = 0$ is for pure exploitation; on the contrary, higher values of ξ emphasizes exploration by inflating the model uncertainty). For this acquisition function, there are strong theoretical results, originated in the context of multi-armed bandit problems, on achieving the optimal regret derived by Srinivas et al. (2012). For the candidate point x_n, we observe instantaneous regret $r_n = f(x_n) - f(x^*)$. The cumulative regret R_N after N function evaluations is the sum of instantaneous regrets: $R_N = \sum_{n=1}^{N} r_n$. A desirable asymptotic property of an algorithm is to be no-regret: $\lim_{N \to \infty} \frac{R_N}{N} = 0$. Bounds on the average regret $\frac{R_N}{N}$ translate to convergence rates: $f\left(x^+\right) = \min x_{n \leq N} f(x_n)$ in the first N function evaluations is no further from $f(x^*)$ than the average regret. Therefore, $f\left(x^+\right) - f(x^*) \to 0$, with $N \to \infty$.

The following figure shows how the selected points changes depending on ξ. Ideally, ξ should be adjusted dynamically to decrease monotonically with the function evaluations. Compared to PI (Fig. 4.1) and EI (Fig. 4.2), is clear the leaning of LCB towards exploration and how this effect can be managed through the value of ξ (Fig. 4.3).

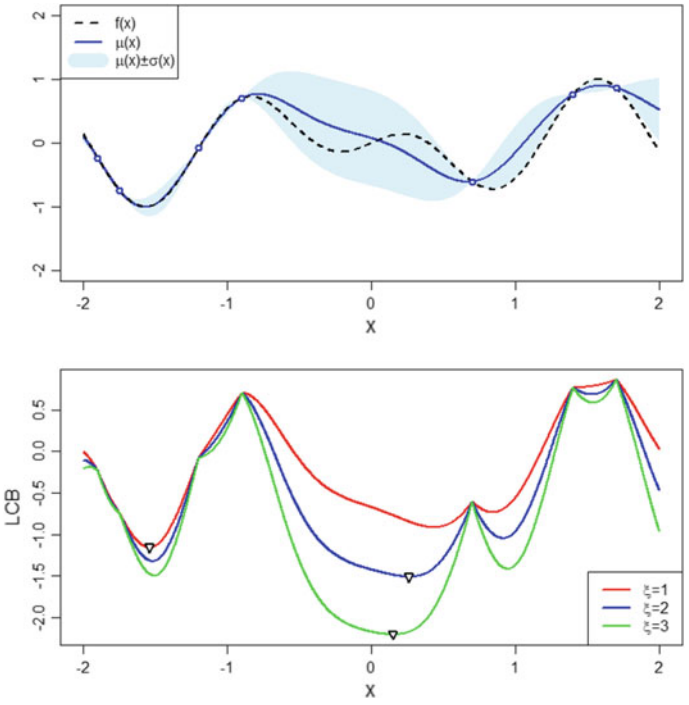

Fig. 4.3 GP trained depending on seven observations (top), LCB with respect to different values of ξ and min values corresponding to the next point to evaluate (bottom). Contrary to the other acquisition functions, LCB is minimized instead of maximized

Finally, the next point to evaluate is chosen according to $x_{n+1} = \text{argmin}_{x \in X} \text{LCB}(x)$, in the case of a minimization problem, or $x_{n+1} = \text{argmax}_{x \in X} \text{UCB}(x)$ in the case of a maximization problem.

4.2 New Acquisition Functions

4.2.1 Scaled Expected Improvement

Recently, a new improvement-based acquisition function has been proposed in Noè & Husmeier (2018). The motivation is that EI does not account for the uncertainty in improvement. The proposed acquisition function chooses the next point to evaluate where the improvement is expected to be high with small variance.

More precisely, the variance of the improvement is computed analytically as:

$$\mathbb{V}[I(x)] = k(x,x)\left[\left(Z^2+1\right)\Phi(Z) + Z\phi(Z)\right] - [EI(x)]^2$$

where Z is defined as in EI. The new acquisition function consists in scaling EI by $\{\mathbb{V}[I(x)]\}^{1/2}$, that is:

$$\text{Scaled EI}(x) = \text{EI}(x)/\{\mathbb{V}[I(x)]\}^{1/2}$$

The main result on a set of benchmarks is that the proposed acquisition function compares favourably with PI, EI, UCB/LCB and the entropy-based acquisition functions to be presented in Sect. 4.2.4.

4.2.2 Portfolio Allocation

Several acquisition functions have been proposed, as reported in this book, so choosing a good acquisition function is not trivial. Indeed, there are no guidelines in the choice of acquisition function. Furthermore, the same acquisition function is not necessarily the best choice on the entire optimization process.

An interesting mixed strategy has been proposed where the acquisition function is chosen, adaptively at each iteration, from a "portfolio" (Brochu et al. 2010). The selection mechanism is formalized as an on online multi-armed bandit problem: different strategies are investigated with the result that a hierarchical hedging approach is the winning one. The basic algorithm, so-called GP-Hedge, is as follows:

1: $D_{1:n} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ is the initial set of evaluated point
2: N is the overall number of available function evaluations
3: Set initial "gains" $g_n = 0$ with $g_n \in \mathbb{R}^M$, where M is the number of acquisition functions in the portfolio
4: for $i = n + 1, \ldots, N$ do
5: compute the next point to evaluate, according to each acquisition function: $x_{i,k} = \underset{x}{\text{argmax}}\, \alpha_k(x \mid D_{1:n}), k = 1, \ldots, M$
6: select one point x_i, among the alternative $x_{i,k}$, with probability: $p_{i,k} = \frac{e^{g_{i[k]}}}{\sum_{l=1}^{M} e^{g_{i[l]}}}$ (i.e., soft − max policy) where $g_{i[k]}$ is the gain of the kth acquisition function at iteration i
7: evaluate objective function, eventually with noise, $y_i = f(x_i) + \varepsilon$
8: update the function evaluations dataset $D_{1:i} = D_{1:n} \cup \{(x_i, y_i)\}$
9: update GP and update gains $g_{i[k]} = g_{i-1[k]} + \mu_i(x_{i,k})$
10: endfor

Empirical evidence shows a significant improvement over EI and LCB, which is anyway offset by the much higher computational cost.

4.2.3 Thompson Sampling

As described in Chap. 3, Thompson Sampling (TS) is a sequential optimization process based on performing, iteratively, the following block of steps: updating a posterior depending on a set of observations, drawing a sample from posterior as an approximation to the function to be optimized, minimizing this sample function to identify the next candidate point and evaluating the objective function at that point.

In a sense TS is in itself a sequential decision-making process, like BO, indeed the acquisition function is replaced by a sample from the current probabilistic surrogate model. However, in the literature, TS is considered just as an acquisition function in BO—this is the reason why we have decided to reserve a small section also in this chapter.

In the following figure, the difference in choosing the next point, depending on TS and LCB, is depicted. The (TS) sample function from the GP is quite different from LCB and, at least in this case, TS is less explorative than LCB. Indeed, TS is in principle exploitative due to the sample minimization step. The exploration is given by its Monte Carlo basis and can be enhanced by adopting an ε-greedy strategy (as already seen in Chap. 3) (Fig. 4.4).

The following figure shows another case where TS is, instead, more explorative than LCB (Fig. 4.5).

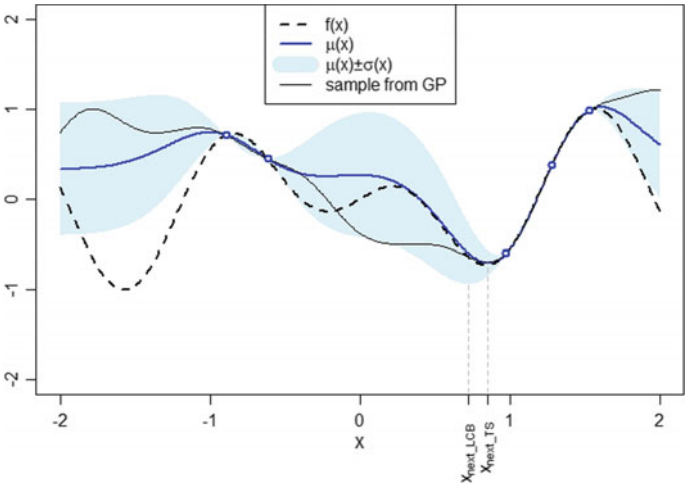

Fig. 4.4 Next point to evaluate according to TS ($\varepsilon = 0.4$) and LCB: the sample from GP posterior implies a more exploitative choice than LCB

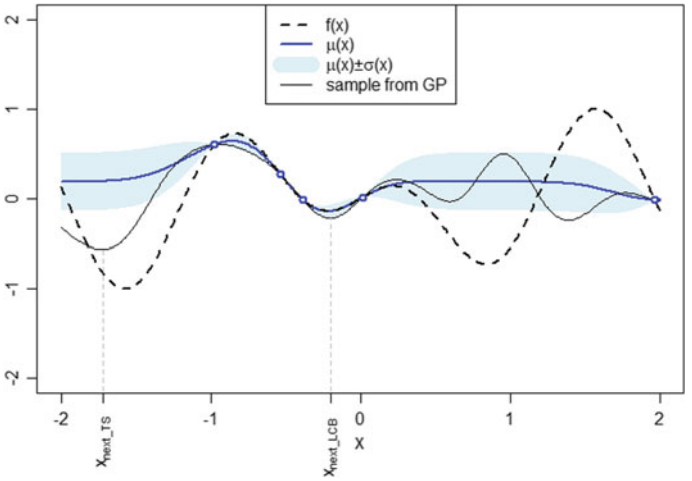

Fig. 4.5 Next point to evaluate according to TS ($\varepsilon = 0.4$) and LCB: the sample from GP posterior implies a more explorative choice than LCB

Theoretical analysis was provided for the classical multi-armed bandit problem and later extended to the analysis of the continuous case in Russo et al. (2018) drawing on an analogy between TS and UCB.

4.2.4 Entropy-Based Acquisition Functions

The traditional acquisition functions, presented in Sect. 4.1, are based on probabilistic measures of improvement in the f domain. Exploitation and exploration are represented, respectively, by the mean value and the "uncertainty bonus" represented by the variance. Recent interest has been focused on querying at points that can help to learn most about the location of the unknown minimum, leading to information-based acquisition functions. This informational approach was originally proposed in Villemonteix et al. (2009), and developer into the Entropy Search (Henning and Schuler (2012)), Predictive Entropy Search (Hernández-Lobato et al. 2014). In both Entropy Search (ES) and Predictive Entropy Search (PES), the basic idea is to maximize the information about the global optimizer.

The current information about the global optimizer can be computed as the negative differential entropy of $p(x^*|D_{1:n})$ and the next point to evaluate is given by the maximization of the expected reduction in this quantity. The ES and PES acquisition functions have the following equations, respectively:

$$\begin{aligned} \text{ES} : \alpha(x) &= H\big(p\big(x^*|D_{1:n}\big)\big) - \mathbb{E}\big[H\big(p\big(x^*|D_{1:n} \cup \{x, y\}\big)\big)\big] \\ \text{PES} : \alpha(x) &= H(p(y|D_{1:n}, x)) - \mathbb{E}\big[H\big(p\big(y|D_{1:n}, x, x^*\big)\big)\big] \end{aligned}$$

where $H[p(\alpha)] = -\int p(\alpha) \log p(\alpha) d\alpha$ represents the differential entropy.

While ES uses the expectation is over $p(x^*|D_{1:n})$, in PES the expectation is over $p(y|D_{1:n}, x)$. Both $p(x^*|D_{1:n})$ and its entropy are analytically intractable and must be approximated through expensive simulation. This approach needs several approximations and is anyway beset by numerical difficulties. To add further complexity to the computation of this distribution, even if the global optimizer is unique, it is not stable meaning that a small perturbation can result in an approximation far away from the global optimum. PES is inspired by Thompson sampling and uses several samples from posterior, at every iteration, to solve probabilistically the optimization of the acquisition function.

The following figure depicts how PES improves over ES, when compared to the ground truth (Fig. 4.6).

In Wang and Jegelka (2017), a computationally more effective solution is provided, based on sampling from the conditional distributions of a global optimum given the observed data. A much cheaper and more robust acquisition function, namely Max-value Entropy Search (MES) is proposed: instead of measuring the information about the optimizer x^*, they use information about the optimal value y^*. MES measures the gain in mutual information between the optimal value y^* and

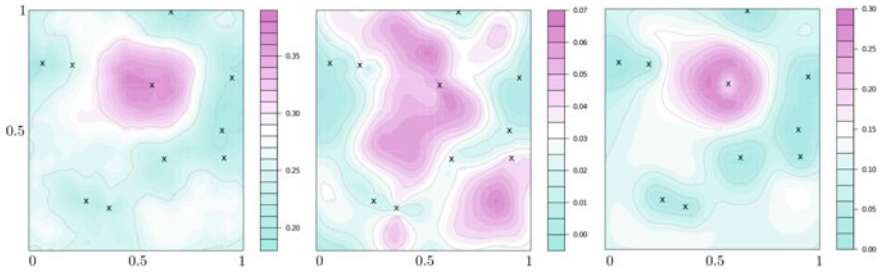

Fig. 4.6 Ground truth (left), approximation produced by the ES method (middle) and approximation produced by the PES method (right). (*Source* Hernandez-Lobato et al. 2014)

the next point to evaluate, which can be approximated analytically by evaluating the entropy of the predictive distribution:

$$\text{MES} : \alpha(x) = H(p(y|D_{1:n}, x)) - \mathbb{E}\big[H\big(p\big(y|D_{1:n}, x, y^*\big)\big)\big]$$

In MES, the expectation is approximated using Monte Carlo estimation. Importantly, while ES and PES rely on the expensive d-dimensional distribution $p(x^*|D_{1:n})$, MES relies on the one-dimensional $p(y^*|D_{1:n})$ which is computationally much easier. The software for PES and MES can be downloaded at https://bitbucket.org/jmh233/codepesnips2014 and https://github.com/zi-w/Max-value-Entropy-Search, respectively. In the case that the value of f^* is known a priori, a more efficient search for x^* is propsed in Nguyen and Osborne (2019).

According to another approach Volpp et al. (2019), the acquisition function is given by a neural network which uses as input the posterior prediction of the GP.

4.2.5 *Knowledge Gradient*

Knowledge gradient (KG) is an acquisition function based on revising the typical assumption made in the BO process that the best evaluated point is returned as final solution. KG revises this assumption by allowing to return any point as a solution, even if it has not been previously evaluated. Another important difference with other acquisition functions, which are usually "risk-seeker", is that KG assumes risk neutrality (Berger (2013), meaning that any random x is evaluated depending on the expected value $f(x)$.

Let us denote this solution with $\bar{x}_n$; the value $f(\bar{x}_n)$ is random under the posterior and has the following expected value conditioned to $D_{1:n}$:

$$\mu_n^* := \mu_n(\bar{x}_n) = \min_{x'} \mu_n\big(x'\big)$$

If a further function evaluation were available, we could sample x one more time and obtain the additional observation (x_{n+1}, y_{n+1}) and, consequently, the updated posterior mean $\mu_{n+1}(\cdot)$. The expected value of the solution after this further function evaluation would be $\mu^*_{n+1} = \min_{x'} \mu_{n+1}(x')$. Thus, the improvement in conditional expected solution value is given by: $\mu^*_n - \mu^*_{n+1}$. As this quantity is unknown, before evaluating $f(x_{n+1})$, we can only compute its expected value, conditioned to the previous observations at $x_1, \ldots, x_n$. This quantity is named knowledge gradient and can be computed at any x:

$$KG_n(x) = \mathbb{E}[\mu^*_n - \mu^*_{n+1} | x_{n+1} = x]$$

The simplest way to compute KG is through simulation: a possible value y_{n+1} is simulated at a certain x_{n+1}, then the optimum of the new posterior mean μ^*_{n+1} is computed, by considering the sampled value y_{n+1} as the actual value obtained through function evaluation at x_{n+1}. The simulation can be performed either by Thompson sampling or directly by using $\mu_n(x_{n+1})$ and $\sigma_n(x_{n+1})$. Finally, this value is subtracted from μ^*_n to obtain the corresponding improvement in solution quality. The entire procedure is repeated many times and the differences $\mu^*_n - \mu^*_{n+1}$ are averaged on the simulated values y_{n+1}. This allows to compute the expected value required for the computation of the estimate of $\mathrm{KG}_n(x)$, that converges to the actual value as the number of samples increases.

Therefore, contrary to other acquisition functions, KG computes the posterior not only at the sampled points but on the entire domain, estimating how a new function evaluation will change the posterior. KG assigns a positive value on measurements that cause the minimum of the posterior mean to improve. Thus, the next point to evaluate is given by the maximization of KG.

According to Frazier et al. (2009), this provides a small performance benefit in the case of BO with noise-free evaluations, but a substantial improvement in problems with noisy, derivative, multi-fidelity observations and other "exotic" problem features. In these cases, the value of sampling comes not through an improvement in the best solution at the sampled point, but through an improvement in the minimum of the posterior mean across possible solutions. For instance, a derivative observation can provide information that the function is decreasing, along with a specific direction, in the neighbourhood of the sampled point. Consequently, the minimum of the posterior mean could be significantly smaller than the previous minimum, even if the function value at the sampled point is worse than the best previously sampled point. In these cases, KG can reportedly outperform EI (Wu et al. 2017; Poloczek et al. 2017; Wu and Frazier 2016; Toscano-Palmerin and Frazier 2018) (Fig. 4.7).

The same variance reduction effect that gradients availability has on GP—as shown in Chap. 3—is also observed in acquisition function. In particular, the following figure shows that both KG and EI, computed on a GP with derivatives, make different sampling decisions (Wu and Frazier 2017) (Fig. 4.8).

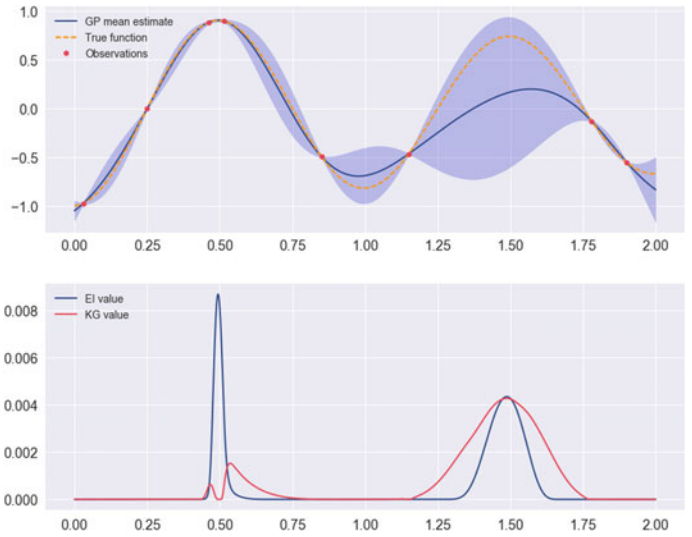

Fig. 4.7 Comparison between KG and EI on a maximization problem. The EI acquisition function prefers to sample in the region around 0.5 and the KG policy prefers to sample in the region around 1.5. Thus, KG gives a chance to exploration while EI is biased towards exploitation. (*Source* https://sigopt.com/blog/expected-improvement-vs-knowledge-gradient/)

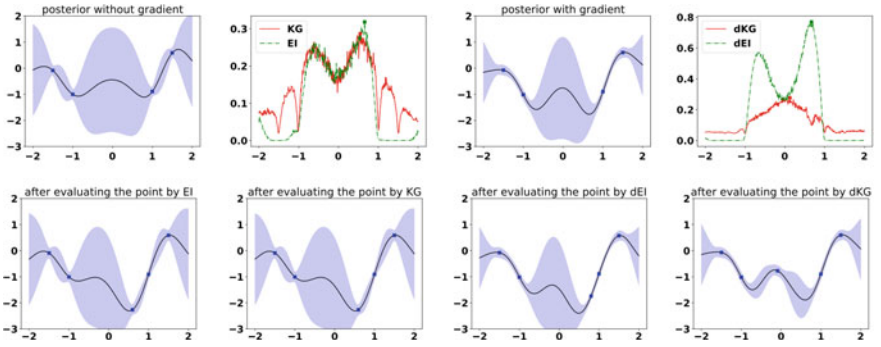

Fig. 4.8 Plots in the top row show the approximations of $f(x)$ modelled through a GP without and with incorporating gradients observations. The posterior variance is smaller if the gradients are incorporated. Both KG and EI are depicted, with and without embedding derivative information. The plots in the bottom row depict the resulting GPs, without and with gradient, after the evaluation at the new point suggested by EI and KG. (*Source* Wu and Frazier 2017)

4.2.6 Look-Ahead

Most of the acquisition functions consider only the impact of the next function evaluation: in this sense, they are also called "myopic". This is a limitation addressed in several papers: in Osborne et al. (2009), a two-step-ahead solution is presented; while

in Marchant et al. (2014), partially observable Markov decision process (POMDP) is used as a model and a Monte Carlo tree search is adopted to solve it. Another look-ahead acquisition function has been proposed in (Lam et al. 2016) where approximate dynamic programming is adopted to solve the problem of selecting the next candidate point to evaluate. More specifically, a rollout heuristic was proposed for BO.

Another approach, GLASSES, has been suggested in Gonzalez et al. (2016), proposing to model the n-step look-ahead problem as a Bayesian network and then suggesting a computationally efficient approximation to solve the otherwise prohibitively computational expensive inference process. At our knowledge, GLASSES is no longer maintained, last update is from November 2015, based on Python 2.7, while other tools presented in Chap. 6 are now based on Python 3.0 and higher.

4.2.7 K-Optimality

A very interesting approach has been recently proposed in Yan et al. (2018), addressing the issue of numerical instability in the kernel matrix K. If the next point to evaluate, selected depending on the acquisition function, is too close to one of the previous observations, it might lead to the computational issue reported in Chap. 3 that is the impossibility to compute the inverse of the matrix K and, consequently, make inference through the GP. The new acquisition function proposed in Yan et al. (2018) consists in the minimization of the condition number of K, leading to a method named Sequentially Bayesian K-optimal (SBKO) design, which tries to avoid sampling close to previous observations. Therefore, SBKO naturally forces the samples to spread sparsely in the search space, providing an alternative exploration mechanism. A possible integration of K-optimality into BO is to consider one of the possible acquisition functions, such as EI, and then select, among possible candidate points with similar values of acquisition the one with smaller condition number. However, this kind of approach leaves some cases undecided. More precisely, given two candidate points, x' and x'', it is not clear which select in the cases: (a) x' is better than x'' in terms of acquisition function but has a higher condition number or (b) x' is worse than x'' in terms of acquisition function but has a lower condition number. To solve these cases, Yan et al. (2018) proposes to consider the equation of parametrized EI (reported in Sect. 4.1.2):

$$\mathrm{EI}(x) = \begin{cases} \left(f\left(x^+\right) - \mu(x) - \xi\right)\boldsymbol{\Phi}(Z) + \sigma(x)\phi(Z) \,\text{if}\, \sigma(x) > 0 \\ 0 \,\text{if}\, \sigma(x) = 0 \end{cases}$$

where ϕ and $\boldsymbol{\Phi}$ are the probability distribution and the cumulative distribution functions, respectively, and

$$Z = \begin{cases} \frac{f\left(x^+\right) - \mu(x) - \xi}{\sigma(x)} \,\text{if}\, \sigma(x) > 0 \\ 0 \,\text{if}\, \sigma(x) = 0 \end{cases}$$

The approach consists in making the parameter ξ dependent on the condition number κ through:

$$\xi(\kappa) = \frac{\log \kappa}{\log \kappa + c \log \kappa_T}$$

where κ_T is a suitable threshold of the condition number and c is a constant that controls the shape of $\xi(\kappa)$. The approach addresses an important issue in BO but numerical evidence is still partial.

4.3 Optimizing the Acquisition Function

The optimization of acquisition functions has been receiving comparably less attention than the development of new ones and in general of the other components of the GP-BO machinery. Indeed, the traditional acquisition functions are available analytically and the computational cost of their optimization is usually much smaller than a single evaluation of the objective function: still the optimization of the acquisition function plays a crucial role as it suggests the next point and it is quite difficult anyway, in high-dimensional spaces, given their non-convexity and specific shape features.

Commonly adopted techniques are random multi-start with local searches or genetic algorithms. More precisely, taking into account the BO software and platforms presented in Chap. 6, genetic algorithm (GA) and evolutionary algorithm (EA) are the most widely adopted techniques to optimize the acquisition function. DIRECT is also another common option; BFGS and L-BFGS implementations can be also adopted in the case of a continuous probabilistic surrogate model, such as GP.

A recent paper Wilson et al. (2018) takes a new approach to the optimization of the acquisition function, looking especially at two different topics:

1. The gradient of acquisition function is estimated via Monte Carlo integration and then used in gradient-based optimization.
2. Greedy maximization of myopic acquisition function is shown, using submodularity, to converge with a guaranteed accuracy.

In the first topic, they use infinitesimal perturbation analysis (Glasserman 1988), also proposed in Kingma and Welling (2013) under the name of *reparametrization trick*. The second topic is concerned with the family of myopic maximization (MM) functions, defined as the expected max of a point-wise utility function: $MM(\mathbf{X}) = \mathbb{E}_{\mathbf{y}}\left[\alpha(\mathbf{y})\right]$.

The family of MM functions is submodular (SM) and greedly maximizing SM functions are guaranteed to produce near-optimal results Krause and Golovin (2014).

References

Astudillo, R., Frazier, P.: Bayesian optimization of composite functions. In: International Conference on Machine Learning, pp. 354–363 (2019, May)

Auer, P.: Using confidence bounds for exploitation-exploration trade-offs. J. Mach. Learn. Res. **3**(3), 397–422 (2002)

Basu, K., Ghosh, S.: Analysis of Thompson sampling for Gaussian process optimization in the bandit setting (2017). arXiv preprint arXiv:1705.06808

Berger, J.O.: Statistical Decision Theory and Bayesian Analysis. Springer Science & Business Media (2013)

Brochu, E., Cora, V.M., de Freitas, N.: A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning (2010). arXiv preprint arXiv:1012.2599

Frazier, P., Powell, W., Dayanik, S.: The knowledge-gradient policy for correlated normal beliefs. Informs J. Comput. **21**(4), 599–613 (2009)

Glasserman, P.: Performance continuity and differentiability in Monte Carlo optimization. In: 1988 Winter Simulation Conference Proceedings, pp. 518–524. IEEE (1988)

González, J., Osborne, M., Lawrence, N.D.: GLASSES: Relieving the Myopia of Bayesian Optimisation (2016)

Hennig, P., Schuler, C.J.: Entropy search for information-efficient global optimization. J. Mach. Learn. Res. **13**, 1809–1837 (2012)

Hernández-Lobato, J.M., Hoffman, M.W., Ghahramani, Z.: Predictive entropy search for efficient global optimization of black-box functions. Adv. Neural. Inf. Process. Syst. **25**, 144–149 (2014). https://doi.org/10.1016/j.molstruc.2009.06.011

Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. J. Glob. Optim. **13**(4), 455–492 (1998)

Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2013). arXiv preprint arXiv:1312.6114

Krause, A., Golovin, D.:. Submodular function maximization. In: Tractability: practical Approaches to Hard Problems, pp. 71–104. Cambridge University Press (2014)

Kushner, H.J.: A new method of locating the maximum point of an arbitrary multi-peak curve in the presence of noise. J. Basic Eng. **86**, 97–106 (1964)

Lam, R., Willcox, K., Wolpert, D.H.: Bayesian optimization with a finite budget: an approximate dynamic programming approach. Adv. Neural. Inf. Process. Syst. **30**, 883–891 (2016). https://doi.org/10.1186/1471-2407-4-76

Marchant, R., Ramos, F., Sanner, S.: Sequential Bayesian optimisation for spatial-temporal monitoring. In: International Conference on Uncertainty in Artificial Intelligence, pp. 553–562 (2014)

Mockus, J., Tiesis, V., and Zilinskas, A.: The application of Bayesian methods for seeking the extremum. In: Dixon, L., Szego, G. (eds.) Towards Global Optimisation, 2, pp. 117–130. Elsevier (1978)

Nguyen, V., Osborne, M.A.: Knowing the what but not the where in Bayesian optimization (2019). arXiv preprint arXiv:1905.02685

Noè, U., Husmeier, D.: On a new improvement-based acquisition function for Bayesian optimization (2018). arXiv preprint arXiv:1808.06918

Osborne, M.A., Garnett, R., Roberts, S.J.: Gaussian processes for global optimization. In: 3rd International Conference on Learning and Intelligent Optimization (LION3), vol. 2009 (2009)

Poloczek, M., Wang, J., Frazier, P.: Multi-information source optimization.In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems, 30, pp. 4291–4301. Curran Associates, Red Hook, NY (2017)

Russo, D., Van Roy, B., Kazerouni, A., Osband, I., Wen, Z.: A tutorial on Thompson sam-pling. Found. Trends Mach. Learn. **11**, 1–96 (2018). https://doi.org/10.1561/2200000070

Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.W.: Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. In: IEEE Transactions on Information Theory, pp. 3250–3265 (2012)

Toscano-Palmerin, S., Frazier, P.I.: Bayesian optimization with expensive integrands (2018). arXiv preprint arXiv:1803.08661

Villemonteix, J., Vazquez, E., Walter, E.: An informational approach to the global optimization of expensive-to-evaluate functions. J. Global Optim. **44**(4), 509 (2009)

Volpp, M., Fröhlich, L., Doerr, A., Hutter, F., Daniel, C.: Meta-learning acquisition functions for Bayesian optimization (2019). arXiv preprint arXiv:1904.02642

Wang, Z., Jegelka, S.: Max-value entropy search for efficient Bayesian optimization. In: Proceedings of the 34th International Conference on Machine Learning. Sydney, Australia (2017)

Wilson, J., Hutter, F., Deisenroth, M.: Maximizing acquisition functions for Bayesian optimization. In: Advances in Neural Information Processing Systems, pp. 9906–9917 (2018)

Wu, J., Frazier, P.: The parallel knowledge gradient method for batch bayesian optimization. In: Advances in Neural Information Processing Systems, pp. 3126–3134 (2016)

Wu, J., Frazier, P.I.: Discretization-free knowledge gradient methods for bayesian optimization (2017). arXiv preprint arXiv:1707.06541

Wu, J., Poloczek, M., Wilson, A.G., Frazier, P.: Bayesian optimization with gradients. In: Advances in Neural Information Processing Systems, pp. 5273–5284, (4.2.5) (2017)

Yan, L., Duan, X., Liu, B., Xu, J.: Bayesian optimization based on K-optimality. Entropy **20**(8), 594 (2018)