IKER GAZTE
NAZIOARTEKO IKERKETA EUSKARAZ

I. IKERGAZTE
NAZIOARTEKO IKERKETA EUSKARAZ

2015eko maiatzaren 13, 14 eta 15
Durango, Euskal Herria

ANTOLATZAILEA:
Udako Euskal Unibertsitatea (UEU)

INGENIARITZA ETA ARKITEKTURA

Kernel hautapen dinamikoa Optimizazio Bayesiarrean

Ibai Roman, Roberto Santana, Alexander Mendiburu
eta *Jose A. Lozano*

690-695 or.
https://dx.doi.org/10.26876/ikergazte.i.95

ANTOLATZAILEA: BABESLEAK: LAGUNTZAILEAK:
ueu udako euskal unibertsitatea
EUSKO JAURLARITZA GOBIERNO VASCO
BFA DFB Bizkaiko Foru Aldundia Diputación Foral de Bizkaia
UPV EHU
Deusto Universidad de Deusto Deustuko Unibertsitatea
MONDRAGON UNIBERTSITATEA
UDALBILTZA
upna

Kernel hautapen dinamikoa Optimizazio Bayesiarrean

Ibai Roman, Roberto Santana, Alexander Mendiburu eta **Jose A. Lozano**

Intelligent Systems Group
Euskal Herriko Unibertsitatea (UPV/EHU)
Manuel de Lardizabal 1. 20018. Donostia. Gipuzkoa.

Laburpena

Optimizazio Bayesiarra Prozesu Gaussiarren bitartez egiten denean, kernel batzuk beste batzuk baino hobeto egokitzen dira helburu-funtziora. Lan honetan, kernel hauek dinamikoki aldatzeko aukera aztertu dugu, hobekuntza-probabilitatean oinarriturik. Kernelen hautaketa aurrera eramateko bost irizpide aurkeztu eta helburu-funtzio ezagunen bidez ebaluatu ditugu. Lortutako emaitzen arabera, irizpide hauek algoritmoaren errendimendua hobetzen dute kernel egokiena aurretiaz ezezaguna denean.

Hitz gakoak: Optimizazio Bayesiarra, Prozesu Gaussiarrak, Optimizazio Orokorra

Abstract

In Bayesian Optimization, when using a Gaussian Process prior, some kernels adapt better than others to the objective function. This research evaluates the possibility of dynamically changing the kernel function based on the probability of improvement. Five kernel selection strategies are proposed and tested in well known synthetic functions. According to our preliminary experiments, these methods can improve the efficiency of the search when the best kernel for the problem is unknown.

Keywords: Bayesian Optimization, Gaussian Process, Global Optimization

1 Sarrera eta motibazioa

Optimizazio orokorraren xedea helburu-funtzio baten maximo edo minimo globala bilatzea da. Jorratu ditugun optimizazio problemetan helburu-funtzioaren adierazpen analitikoa ezezaguna da. Problema hauei kaxa-beltzeko funtzio deritze (Jones *et al.*, 1998). Demagun, adierazpen analitikoa ezezaguna izateaz gain, helburu-funtzioa ebaluatzearen kostua handia dela. Gure problemetan helburu-funtzioaren lagin bat hartzeko, denbora edo bestelako baliabide ugari kontsumitu beharko ditugu. Bestalde, funtzioaren aldagaiak jarraiak izango dira, eta bilaketa aldagai hauen mugen barruan gauzatuko da.

Gehienetan, ebaluatzeko garestiak diren kaxa-beltzeko funtzio hauek optimizatzeko, bilaketa prozesu iteratibo bat jarraitzen da. Ondorioz, bilaketan zehar eman beharreko pausuak kontu handiz hautatu behar dira. Optimizazio problema honi aurre egiteko bi estrategia posible proposatu dira literaturan:

Esperimentuen diseinua Helburu-funtzioaren ebaluazio kopurua gutxitzea, pausu bakoitzean beronen ezagutza-irabazia maximizatuz.

"Multi-armed bandit" Metatutako irabazia handitzea, esplorazio eta esplotazioaren arteko oreka bilatuz.

Bata edo bestea gauzatzeko Optimizazio Bayesiarra-k (OB) (Mockus, 1989) emaitza itxaropentsuak eman ditu dagoeneko (Jalali *et al.*, 2013; Garnett *et al.*, 2010). Kontrakoa dirudien arren, 2010-ean Srinivas *et al.*-ek bi estrategia hauek erlazionatuta daudela frogatu zuten. OB-ean helburu-funtzioa ausazko funtzio bezala hartzen da, "a priori" banaketa bat esleituz. Ondoren, funtzioa ebaluatzearekin batera, banaketa hau informazio berriarekin eguneratzen da, "a posteriori" banaketa bat kalkulatuz.

Helburu-funtzioaren ezaugarriren bat ezaguna baldin bada, leuntasuna adibidez, “a priori” gisa Prozesu Gaussiarrak (PG) erabili daitezke ezagutza honi etekina ateratzeko (Mahendran *et al.*, 2012; Wilson *et al.*, 2014). PG-ak dimentsio anitzeko banaketa gaussiar edo normalak dira, non bilaketa esparruko puntu bakoitzerako banaketa normal bat emango baituten.

Kernelaren arabera, PG-ak helburu-funtzioaren ezaugarri batzuk ala besteak aintzat hartzen ditu eta baita optimoa bilatzeko eman beharreko pausu kopurua nabarmenki jaitsi ere. Horregatik, teknika hauek erabiltzean kernelaren hautapena erabaki garrantzitsua da.

2 Arloko egoera eta ikerketaren helburuak

PG-en arloan kernelaren hautapena sakonki aztertu den arren (Rasmussen eta Williams, 2006), beti ikuspuntu estatiko batetik planteatu da. Funtzioaren ezaugarriak aztertu, kernela hautatu, eta ondoren hasten da optimizazio prozesua.

Kaxa-beltz funtzio garestien kasuan, optimizazioaren kostua arbuiagarria bihurtzen da ebaluazioaren kostuaren aurrean. Posible da hortaz, kernela hasieratik hautatu beharrean, prozesuan zehar PG anitz kalkulatzea, bakoitza kernel ezberdin batekin, kostu globala nabarmen aldatu gabe. Pausu bakoitzean kernel baten edo bestearen arteko hautua egingo litzateke, ordura arte irabazitako ezagutzaz baliatuz.

Lan honen helburua posibilitate hau ikertzea da. Horrez gain, kernela dinamikoki hautatzeko irizpideak proposatu eta aztertzea.

3 Ikerketaren muina

Atal honen xedea esperimentuak aurrera eramateko hartu ditugun erabakiak azaltzea da. Lehenik eta behin, kernel funtzioak deskribatu ditugu. Ondoren, eskuratze-funtzioa aurkezten da, hautapen irizpideekin batera. Azkenik, esperimentuen emaitzak argitaratu ditugu.

3.1 Kernel Funtzioak

Kernel funtzioak PG-aren funtsezko atala dira. Funtzio hauek PG-aren kobariantza matrizea sortu, eta prozesuaren erabilgarritasuna erabat baldintzatzen dute. Lan honetan, kernel finkoak bakarrik izan ditugu kontutan. Kernel hauen emaitza puntuen arteko distantziaren araberakoa da, beraien posizio absolutuak arbuiatuz ($r = ||x' - x||$).

1 Taulak erakusten duen moduan, sei kernel finko hautatu dira. l eskala hyper-aldagaia da eta σ_n-k zarata adierazten du.

Kernel Funtzioak	
$k_{SE}(r) = exp\left(-r^2/2l^2\right) + \sigma_n$	$k_{M32}(r) = \left(1 + \sqrt{3}r/l\right) exp\left(-\sqrt{3}r/l\right) + \sigma_n$
$k_{M52}(r) = \left(1 + \sqrt{5}r/l + 5r^2/3l^2\right) exp\left(-\sqrt{5}r/l\right) + \sigma_n$	$k_E(r) = exp\left(-r/l\right) + \sigma_n$
$k_{E15}(r) = exp\left(-(r/l)^{1.5}\right) + \sigma_n$	$k_{RQ2}(r) = \left(1 + r^2/4l^2\right)^{-2} + \sigma_n$

1 Taula: Kernel Funtzioak. k_{SE}: *Squared Exponential.* k_{M32}: *Matern* 32. k_{M52}: *Matern* 52. k_E: *Exponential.* k_{E15}: *γ-exponential.* k_{RQ2}: *Rational Quadratic.*

3.2 Hobekuntza Probabilitatea eskuratze-funtzio bezala

Ebaluatuko den hurrengo puntua hautatzeko, OB-ak eskuratze-funtzio bat erabiltzen du ($u(\cdot)$). Funtzio honek, bilaketa espazioko puntu bakoitzak duen baliagarritasuna [1] adierazten du. Horretarako, orain arte ebaluatutako puntuek sortutako PG-ean oinarritzen da. Gainera, funtzio honek esplorazio eta esplotazioaren arteko oreka bilatzeko ere balio du.

Era berean, bilaketa gidatzeko eskuratze-funtzioa erabiliko da esperimentu honetan. Haatik, PG sorta bat izatean, bakoitzak eskuratze-funtzio ezberdin bat sortzen du. Ondorioz, ebaluatuko den hurrengo puntua hautatzeko, funtzio hauek bata bestearekin alderagarriak izatea beharrezkoa da. Horregatik,

[1] ingelesezko literaturan *utilty* deritzo

Hobekuntza Probabilitatea(Kushner eta Yin, 1997) deritzon eskuratze-funtzioa erabili dugu. Funtzio honek orain arteko emaitzarik onena hobetzeko probabilitatea neurtzen du.

Jarraian, eskuratze-funtzioaren adierazpen matematikoa deskribatzen dugu:

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

μ eta σ PG-ak ematen dituen media eta bariantza dira, x^+ orain arteko punturik onena, Φ banaketa normala jarraitzen duen zorizko aldagai baten banaketa-funtzioa, eta ξ esplorazioa eta esplotazioaren arteko oreka bermatzen duen aldagaia. Aurreko ikerketek (Lizotte, 2008) horrela iradokita $\xi = 0.01$-n finkatu dugu.

3.3 Kernela dinamikoki hautatzeko irizpideak

Gure proposamenak, ohizko OB algoritmo bat darrai (Brochu *et al.*, 2009), aldaketa txiki batzuekin. Lehenik, PG-en berritze prozesuan, PG bakoitzak ebaluazioen datu sorta jasotzen du ($D_{1:t}$). Bigarren, puntu berria hautatzen denean, eskuratze-funtzio anitz optimizatzen dira. Azkenik, ondoren azalduko ditugun kernel hautapen irizpideak jarraiki, PG bakoitzak proposatutako puntu onenen artean onena hautatzen da.

1 Algoritmoa: **Proposaturiko optimizazio algoritmoa**

Puntu bat zoriz hautatu eta ebaluatu: $y_1 = f(x_1)$
Datu-sorta hasieratu $D_1 = \{(x_1, y_1)\}$
PG-ak hasieratu $\leftarrow D_1$
$t = 2$
do {
  PG bakoitzak (k) eskuratze-funtzioa optimizatuz puntu bat proposatzen du :
  $x_{t,k} = argmax_x u_k(x|D_{1:t-1})$
  Puntu onena hautatu $x_t \leftarrow$ *choose* $(x_{t,k})$
  Helburu-funtzioa ebaluatu: $y_t = f(x_t)$
  Datu-sorta eguneratu $D_{1:t} = \{D_{1:t-1}, (x_t, y_t)\}$
  PG-ak eguneratu $\leftarrow D_{1:t}$
  $t+ = 1$
} **until** geldiera baldintzak bete

1 Algoritmoan oinarriturik, kernela dinamikoki hautatzeko ondorengo irizpideak jarraitu ditugu:

DynamicRandom PG-ek proposatutako puntuak ausaz hautatzen dira optimizazio prozesuan zehar.

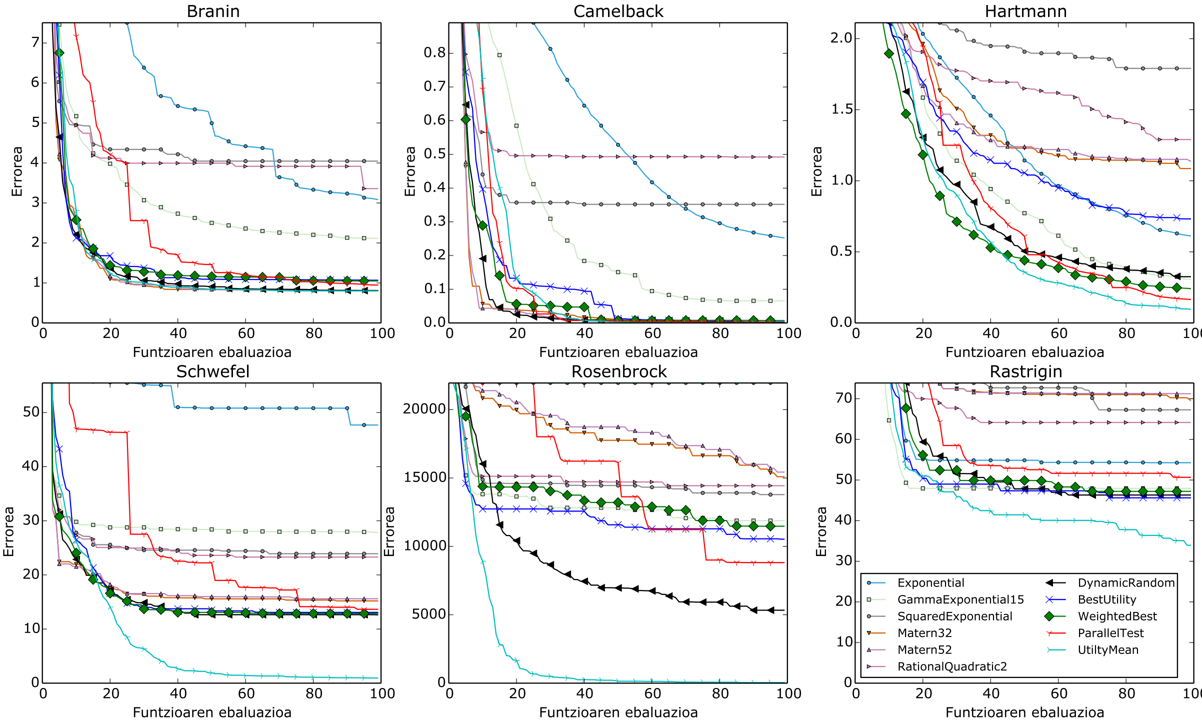
BestUtility PG-ek proposatutako puntuen artean, eskuratze-funtzioan duten baliagarritasunaren arabera onena hautatzen da hurrengo pausurako.

WeightedBest eskuratze-funtzio bakoitzari pisu bat esleitzen zaio ($u_k^*(x) = u_k(x)w_{k,t}$). Hasiera baten pisu guztiek 0.5 balio dute. Ebaluazioaren ostean, hobekuntza neurtzen da, eta horren arabera hautatua izan den eskuratze-funtzioaren pisua eguneratzen da ($w_{k,t+1} = w_{k,t}(\Phi(i_{t-1}) + 0.5)$), besteen pisuak bere horretan geratzen direlarik ($w_{k,t+1} = w_{k,t}$).

ParallelTest Hasierako pausuan, PG guztiek puntu bat proposatzen dute, eta puntu guzti hauek ebaluatzen dira. Eskuratze-funtzioaren arabera baliagarritasun handiena duen puntua eman duen PG-a hautatzen da. Hurrengo N pausuetan PG hau bakarrik hautatzen da, eta ondoren prozesua errepikatzen da. Gure esperimentazioan $N = 20$ erabili dugu.

	Exponential	GammaExponential15	SquaredExponential	Matern32	Matern52	RationalQuadratic2
Branin	653.83	382.67	508.29	207.35	209.06	492.32
Camelback	71.02	38.27	44.03	10.88	10.29	57.31
Hartmann	141.06	104.81	205.18	149.69	146.79	174.52
Schwefel	5934.99	3610.52	3267.92	2518.78	2560.76	3220.60
Rosenbrock	3266480.68	2385382.31	2559072.46	2892357.10	2941860.40	2582973.94
Rastrigin	6120.38	5443.13	7617.92	7830.09	7844.48	7047.10

	DynamicRandom	BestUtility	WeightedBest	ParallelTest	UtiltyMean
Branin	207.26	234.54	240.55	362.29	224.62
Camelback	12.52	17.11	14.09	23.14	20.94
Hartmann	90.22	125.37	77.02	99.44	78.82
Schwefel	2325.86	2427.06	2276.88	3865.39	1509.75
Rosenbrock	1949197.30	2295975.53	2430911.09	2572203.21	1304177.74
Rastrigin	5803.88	5499.70	5772.56	6419.66	4940.36



1 Irudia: Bataz-besteko errorea funtzio ebaluazio bakoitzean.

UtilityMean eskuratze-funtzio bakoitza optimizatu beharrean, eskuratze-funtzio orokor bat sortzen da beste hauen media eginez ($u(x) = \sum_{k=1}^{n} \frac{u_k(x)}{n}$). eskuratze-funtzio orokor honen optimoa izango da ebaluatuko den hurrengo puntua.

3.4 Esperimentuen emaitzak

Esperimentuen helburua kernela dinamikoki hautatzeko irizpideen errendimendua ebaluatzea da eta kernela estatiko mantentzen duten estrategiekin alderatzea. Horretarako, 3.3 Atalean azaldu ditugun irizpide dinamikoak eta 3.1 Ataleko kernelak estatikoki erabiliz, sei test funtzio ezagun optimizatu ditugu: Branin-Hoo, Camelback, Hartmann 6d, Schwefels, Rosenbrock and Rastrigin (Eggensperger *et al.*, 2013).

Branin eta Camelback funtzio sinpleak dira eta bi aldagai dituzte soilik. Schwefel, Rosenbrock eta Rastrigin funtzioak berriz, 4 aldagairekin ebaluatu ditugu. Azkenik, Hartmann funtzioa da optimizatuko dugun konplexuena, 6 aldagai baititu.

Estrategia dinamiko eta estatiko hauekin, sei funtzioetan, 100 ebaluazio egin ditugu. Ebaluazio bakoitzean, helburu-funtzioaren maximoaren ($f(x^*)$) eta orain arteko puntu onenaren ($f(x_t^+)$) arteko tartea kalkulatu dugu. Neurri honi, errorea(e_t) deituko diogu eta honela definituko dugu: $e_t = f(x^*) - f(x_t^+)$. Metatutako errorea (E_t) erabiliko dugu funtzio bakoitzean estrategia onena erabakitzeko: $E_t = \sum_{t=1}^{T} e_t$. Prozesu estokastikoa denez, froga bakoitza 25 aldiz exekutatu dugu, errorearen media aritmetikoa kalkulatuz.

Esperimentuak gauzatzeko software berri bat garatu dugu, 1 Algoritmoa eta *choose* funtzioak inplementatuz. Kernel bakoitzaren hyper-aldagaiak doitzeko Rasmussen eta Williams-ek aurkeztutako teknika erabili dugu, egiantz handiena duen aldagai sorta bilatuz. Bestalde, eskuratze-funtzioa optimizatzeko, DIRECT algoritmoa erabili dugu (Jones *et al.*, 1993), Brochu *et al.*-k 2009-n erabili bezala.

1 Irudiak esperimentuaren emaitzak erakusten ditu. Bertan, kernel estatikoarekin eta hautapen irizpide dinamikoekin egindako frogak alderatzen dira. Lerroetako puntu bakoitzak 25 ebaluazioetan izandako bataz-besteko errorea azaltzen du. 2 Taulak berriz, irizpide bakoitzak prozesuan zehar metatutako errorea erakusten du.

Branin-Hoo, Camelback funtzioetan, $Matern$ kernelek dute errendimendurik onena estrategia estatikoen artean. Estrategia dinamikoek ez dute lortzen emaitzak hauek hobetzerik. Alabaina, beste estrategia estatikoen alderatuz errore txikiagoa metatzen dute. 4 aldagai dituzten Schwefel, Rosenbrock eta Rastrigin funtzioetan berriz, pareko emaitzak lortzen dituzte estrategia dinamiko eta estatikoek. Bertan, nabarmentzekoa da $UtilityMean$ irizpidearen errendimendua, hiru kasuetan emaitza onena lortzen duelarik. Azkenik, Hartmann funtzioan, estrategia dinamikoek estrategia estatikoen emaitzak hobetzen dituzte ia kasu guztietan. $GammaExponential15$ kernela estatikoki darabilen estrategia soilik da gai $BestUtility$-ren emaitzak gainditzeko.

4 Ondorioak

Prozesu Gaussiarren kernela dinamikoki aldatuz, Optimizazio Bayesiarraren emaitzak hobetzeko aukera aztertu dugu. Horretarako, Optimizazio Bayesiarraren algoritmoa egokitu eta bost irizpide dinamiko aurkeztu ditugu. Irizpide hauek eskuratze-funtzioak adierazten duen baliagarritasunean oinarritzen dira. Esperimentuetan, estrategia dinamiko hauen errendimendua, orain arteko estrategia estatikoekin alderatu dugu. Horrela, software berri bat garatu dugu, estrategia dinamiko eta estatiko hauek 6 helburu-funtzio ezagunetan frogatzeko. Emaitzen arabera, OB-ean kernel ezberdinak erabiltzeak prozesuaren errendimendua asko hobetu dezake. Hala ere, emaitza hauek helburu funtzioaren araberakoak dira eta aldagai kopuruaren menpekotasuna dute. Nabarmentzekoa da, $UtilityMean$ estrategiak erakutsi duen errendimendua frogatu dugun helburu-funtzio guztietan.

5 Etorkizunerako planteatzen den norabidea

Lan honen emaitzek ikerketa ildo honetan jarraitzera bultzatzen gaituzte. Alde batetik, kernelen hiperaldagaiak doitzeko teknika berriak frogatu daitezke. Oraingoan, 2014-an egin genuen esperimentazioa (Roman *et al.*, 2014) hobetu dugu, non aldagai hauek aurretiaz adierazten baikenituen. Ildo honetan Snoek *et al.* (2012)-ek, eta Wang eta de Freitas (2014)-ek planteatzen dituzten teknikak ere erabil daitezke emaitzak hobetzeko. Bestalde, kernela dinamikoki hautatzeko irizpide berriak bilatzea interesgarria litzateke, batez ere, eskuratze-funtzioak uztartzearen ideiari jarraiki.

Erreferentziak

Brochu, Eric, Vlad M. Cora, eta Nando de Freitas. 2009. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. Technical Report TR-2009-23, Department of Computer Science, University of British Columbia.

Eggensperger, Katharina, Matthias Feurer, Frank Hutter, James Bergstra, Jasper Snoek, H Hoos, eta Kevin Leyton-Brown. 2013. Towards an empirical foundation for assessing Bayesian optimization of hyperparameters. In *Workshop on Bayesian Optimization in Theory and Practice*.

Garnett, Roman, Michael A Osborne, eta Stephen J Roberts. 2010. Bayesian optimization for sensor set selection. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 209–219. ACM.

Jalali, Ali, Javad Azimi, Xiaoli Fern, eta Ruofei Zhang. 2013. A Lipschitz Exploration-Exploitation Scheme for Bayesian Optimization. In *Machine Learning and Knowledge Discovery in Databases*, 210–224. Springer.

Jones, Donald, Matthias Schonlau, eta William J. Welch. 1998. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization* 13.455–492.

Jones, Donald R., Cary D. Perttunen, eta Bruce E. Stuckman. 1993. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications* 79.157–181.

Kushner, Harold J., eta G. George Yin. 1997. *Stochastic Approximation Algorithms and Applications*. New York, NY: Springer New York.

Lizotte, Daniel James, 2008. *Practical Bayesian Optimization*. Edmonton, Alta., Canada: University of Alberta tesia. AAINR46365.

Mahendran, Nimalan, Ziyu Wang, Firas Hamze, eta Nando D. Freitas. 2012. Adaptive MCMC with Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, 751–760.

Mockus, Jonas. 1989. *Bayesian approach to global optimization*. Springer.

Rasmussen, C. E., eta C. K. Williams. 2006. *Gaussian processes for machine learning*. MIT Press.

Roman, Ibai, Roberto Santana, Alexander Mendiburu, eta Jose A. Lozano. 2014. Dynamic Kernel Selection Criteria for Bayesian Optimization. Montreal, Canada.

Snoek, Jasper, Hugo Larochelle, eta Ryan P Adams. 2012. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems 25*, ed. by F. Pereira, C. J. C. Burges, L. Bottou, eta K. Q. Weinberger, 2951–2959. Curran Associates, Inc.

Srinivas, Niranjan, Andreas Krause, Sham Kakade, eta Matthias Seeger. 2010. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, 1015–1022.

Wang, Ziyu, eta Nando de Freitas. 2014. Theoretical Analysis of Bayesian Optimisation with Unknown Gaussian Process Hyper-Parameters. *arXiv:1406.7758 [cs, stat]* . arXiv: 1406.7758.

Wilson, Aaron, Alan Fern, eta Prasad Tadepalli. 2014. Using Trajectory Data to Improve Bayesian Optimization for Reinforcement Learning. *J. Mach. Learn. Res.* 15.253–282.