Bayesian Optimization with Shape Constraints

Michael Jauch
Víctor Peña
Department of Statistical Science
Duke University

Internal Statistics Seminar, UPF
Jan 20th, 2016

You are here

Probabilistic Numerics

Bayes Opt

Our paper

Probabilistic Numerics

- Suppose you're using a numerical method.
- When you stop, you don't know what the "right answer" is.
- How can you quantify that uncertainty?
- If you buy Cox's axioms, probability is the way to go.

Probabilistic Numerics

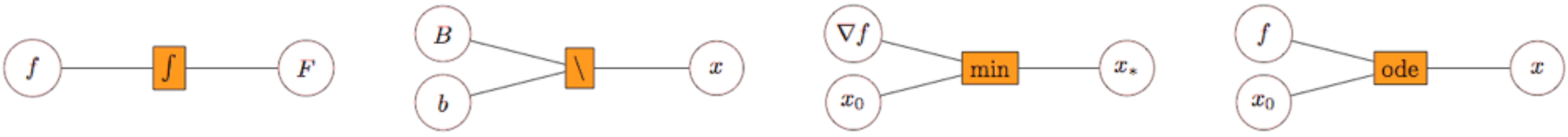
- The first application of this idea is attributed to Poincaré (1896).

- Diaconis (1988) tells us that Poincaré (1896) used Gaussian processes (GPs) for numerical interpolation (think computer models).

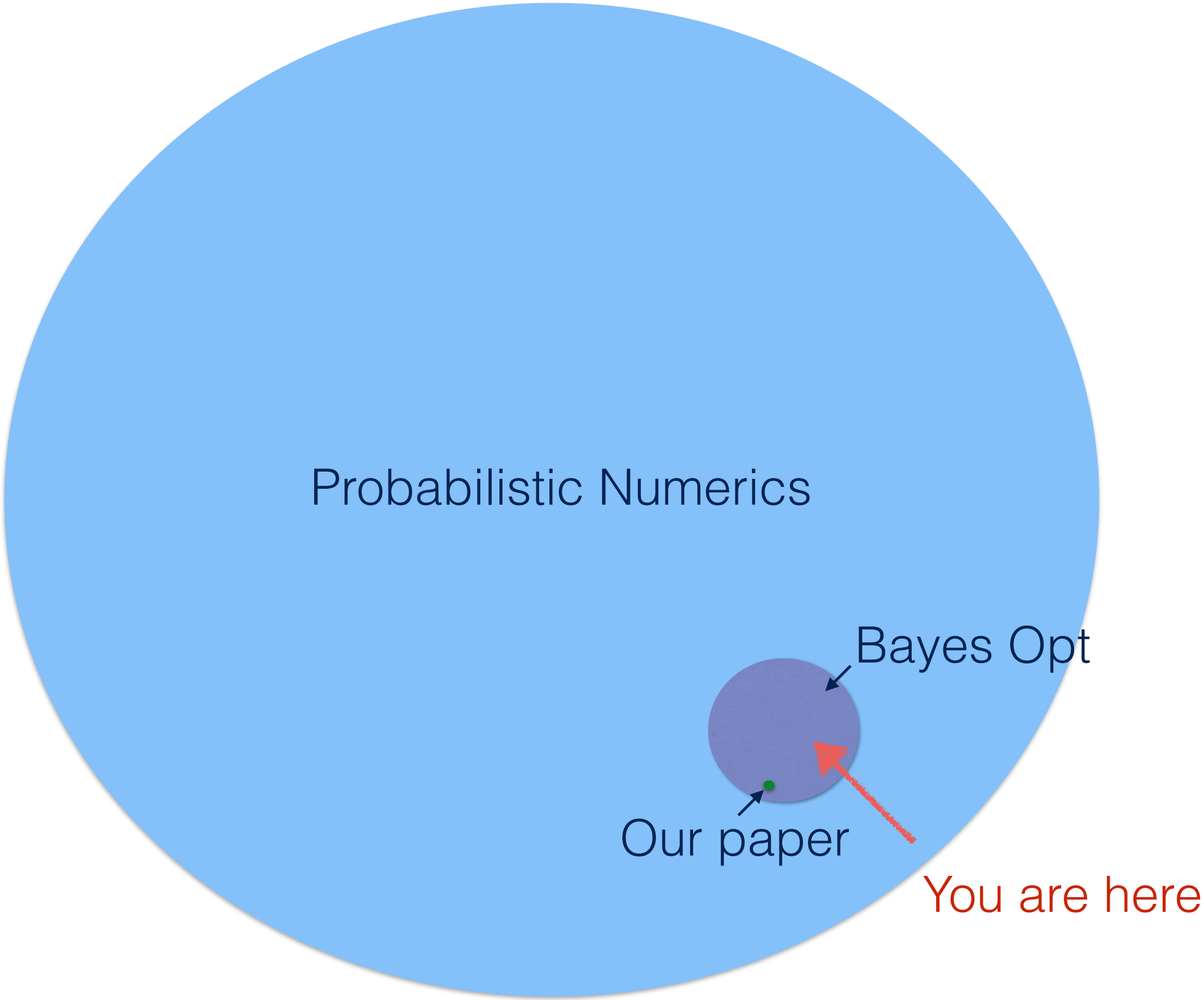
- None of the GP theory was available back then.

Henri Poincaré

Probabilistic Numerics



- I’m new to the field and don’t know much about it.
- Hennig et al. (2015) is a great introduction.
- The community is active and growing.
- Check out probabilistic-numerics.org


Probabilistic Numerics
Bayes Opt
Our paper
You are here

Bayesian Optimization

What is it about?

- Using **Gaussian processes** as **surrogate models for objective functions.**

How does it work?

- Get started with some initial point(s). In the absence of prior information, I'd suggest starting with at least as many points as parameters in the model, which you may sample from some LHS.

- Estimate where the optimum is and quantify your uncertainty. If you're not certain enough, decide where to sample next using some rules that "improve your chances of improvement" or "getting more information about the optimum" (these are called ***acquisition functions***).

Bayesian Optimization

Why would you do that?

- Objective functions can be **noisy**, **not available in closed-form**, and/or **expensive to evaluate**.

- GP theory tells us that we can **estimate consistently functions that are in large classes** (which aren't necessarily "nice," e.g. convex).

- Given a finite set of (possibly noisy) evaluations of the objective function, we can estimate where the optimum is and **quantify our uncertainty**. This is crucial if we can't afford to evaluate our objective function many times (one could say it's a *moral obligation*!).

- If you know something about the problem, you can **use prior information**.

Applications

Hyperparameter tuning

- Bayesian optimization has been very successful in training models such as SVMs, neural nets, etc.

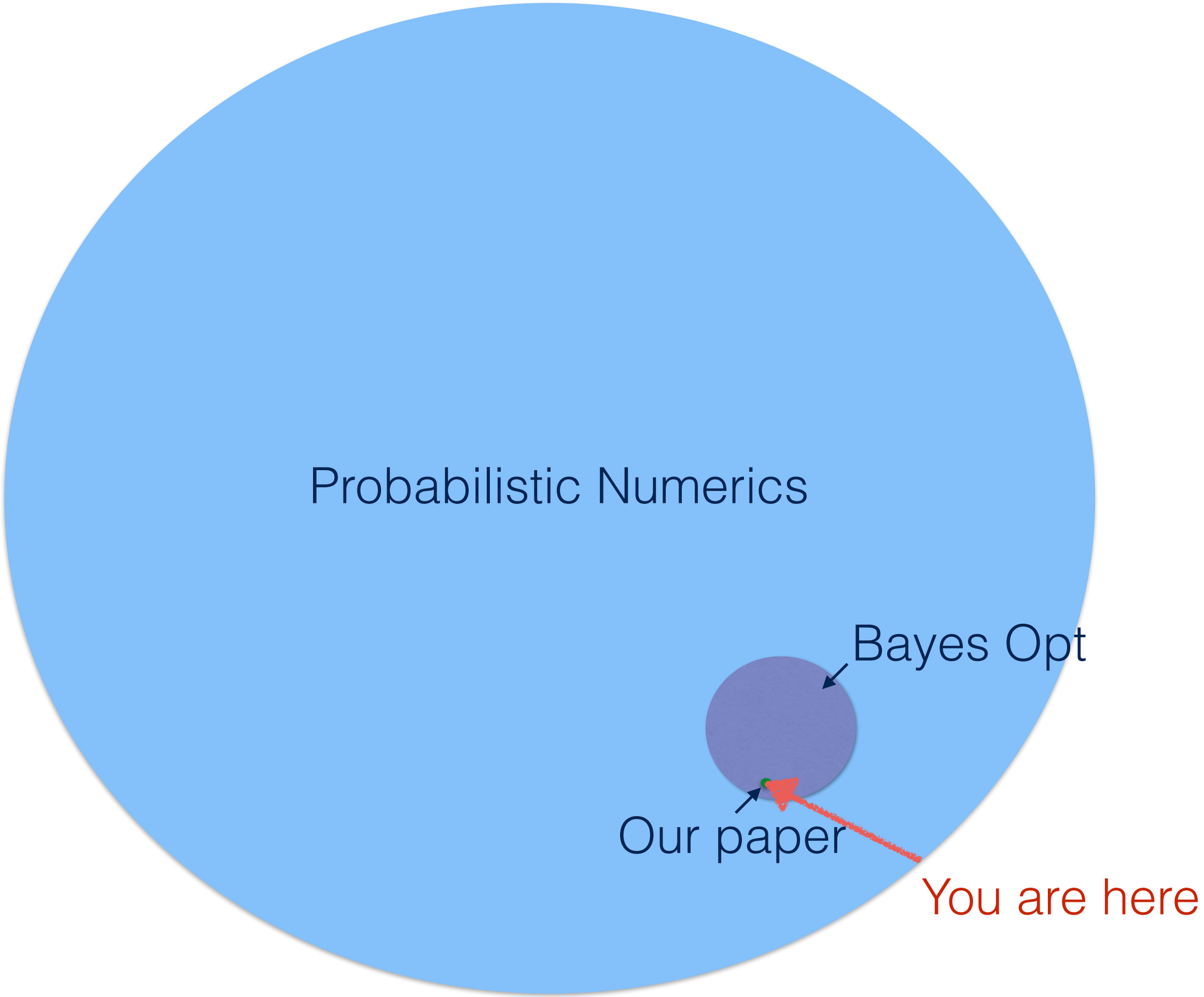
- What is the objective function here? Some measure of "predictive loss" (e.g. cross-validation error) or log-likelihood.

Optimal design based on maximizing posterior utility (see Lindley (1956)):

- In non-trivial models with even moderate sample sizes, one has to make decisions from (noisy) MCMC samples at a limited number of designs.

    - For an application where this the case, see Stephan-Otto Attolini et al. (2015). *[NB: this is a shameless plug; David and I are the "et al."!]*

- Müller & Parmigiani (1995) note this and decide to fit some smoothed curve to posterior utility samples.


Probabilistic Numerics
Bayes Opt
Our paper
You are here

Bayesian Optimization with Shape Constraints

- In some cases, **there is prior information** about the **shape** of the objective function with respect to one or more of its arguments.

- For smooth covariance functions, derivatives of GPs are **jointly Gaussian** with the observations (see e.g. Papoulis & Pillai (2002) or Rasmussen & Williams (2006)).

- Our NIPS workshop paper:

  - Argues in favor of the appropriateness of shape constraints in a couple of explicit applications.

  - Introduces a conceptually simple (albeit approximate and not as efficient as we'd like to) way to implement them.

  - Presents a couple of toy examples with promising results.

Applications

Hyperparameter tuning

- In the case of “regularization” or “capacity” parameters, there’s a **trade-off between over- and under-regularizing**. As a result, one would expect “error curves” or more generally “predictive losses” to be **unimodal** (at least wrt those variables).

Maximization of (posterior) utilities:

- In **multiattribute decision problems**, eliciting preferences for all the possible combinations of the attributes becomes unfeasible. However…
  - Attributes related to monetary value are concave and monotonically increasing for **risk-averse** agents.
  - In other cases, it might make sense to impose a **unimodality** constraint: too much of a good thing can wind up being bad! (think beer).
- In **experimental design**, the utility of some design variables is convex or at least unimodal, reflecting the **trade-off between “statistical accuracy” and sampling costs.**

Componentwise Constraints

- Each of the motivating examples involves prior knowledge of the shape of the objective function wrt an argument, holding the others *fixed*.

- For example, let $f : \mathbb{R}^d \to \mathbb{R}$ and the program be

$$\min f(x),\ x \in \mathcal{A} \subset \mathbb{R}^d$$
$$\mathcal{A} \text{ convex},\ f \in \mathcal{C}^2$$

- We might want to impose that the function is componentwise convex on, say, the first component, that is:

$$\frac{\partial^2}{\partial x_1^2} f(x_1, x_2, \ldots, x_d) > 0,\ \forall x \in \mathcal{A}$$

Derivatives of GPs

- Suppose the covariance function k of the GP is smooth enough.

- Let $x, x' \in \mathbb{R}^d$ and o, o' be the orders of their derivatives.

- Then,

$$\mathrm{Cov}\left(\frac{\partial^o}{\partial x_j^o} f(x), \frac{\partial^{o'}}{\partial x_k'^{o'}} f(x')\right) = \frac{\partial^{o+o'}}{\partial x_j^o x_k'^{o'}} k(x, x')$$

- And you can take a bunch of those and everything is jointly Gaussian.

Examples of Constraints

- We can impose monotonicity and convexity or concavity by imposing the appropriate constraints in their derivatives.

- We also consider ***quasiconvexity***, which includes functions that are unimodal, but not necessarily convex.

- In 1D, quasiconvex functions are either (1) monotone or (2) nonincreasing up to some point, and then nondecreasing from that point onward.

    - Minor observation: This definition gives us a straightforward way to implement an accept-reject sampler in 1D, which can be extended to "slices" in higher dimensions (we do that in 2D).

Some Challenges!

- We'd like to enforce the constraints for all possible values of x, but we can't. We can enforce them on a grid, though.
    - **Solution:** Wang and Berger (2016) propose a (sequential) algorithm to keep on adding points until the probability that the constraint is violated is low enough.
- Covariances get big *fast*:
    - **Solution:** Thankfully, the construction of those can be vectorized (for the most part), and we can resort to approximate methods for inverting matrices (we used incomplete Choleski; there might be better ways).
- The likelihoods/predictives are high-dimensional truncated distributions.
    - **"Solution:"** In our examples, we take a "partial likelihood" approach (i.e. we ignore the constraints) for estimating the MAPs of the hyperparameters, and we use the methods in Botev (2016) to sample from the truncated posterior predictive distributions, given the (wrong) MAPs.

Some Caveats

- Our estimates of the parameters are **off** (because they ignore the constraints), and **we underestimate our uncertainty because we're taking MAPs**. We acknowledge this is not a great solution.

- However, all our decisions, given the wrong estimates, are made via sampling from the ***correct*** truncated distributions.

  - Our acquisition function (which tells us where to sample next) is based on a MC estimate of the posterior expectation of *f* **given the constraints** (truncated normals).

  - At any iteration, our estimation of the optimum and predictive intervals are taken from the ***correct*** distribution, given the ***wrong*** estimates.

Example: Optimal Design

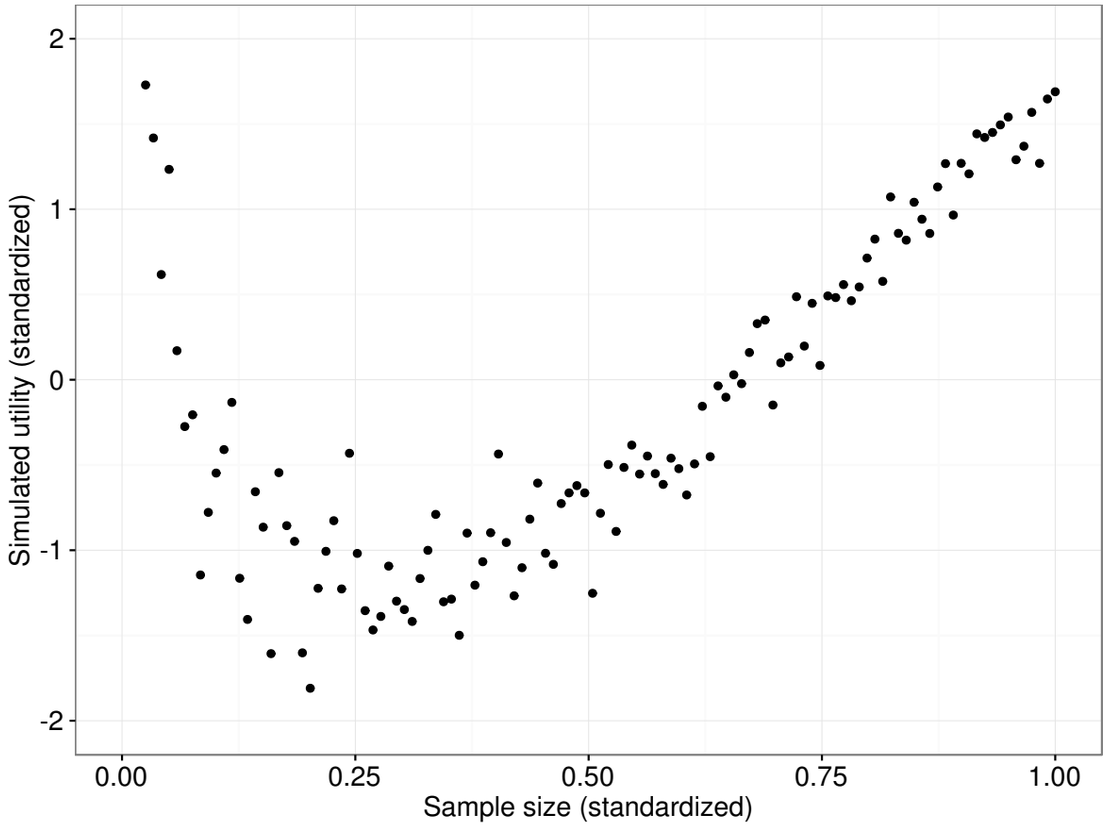
- Optimal sample size of a $\text{Binomial}(n, \theta)$ experiment where the prior on θ is an equal-weighted mixture of a Beta(3,1) and a Beta(3,3).

- The loss of an experiment with sample size *n* given data *y* is

$$L(n, y, \theta) = |\theta - m_y| + 0.0008n$$

- The loss is estimated as the MC average of 100 simulations from the joint distribution of the data and θ.

This is Example 1 in Müller & Parmigiani (1995)



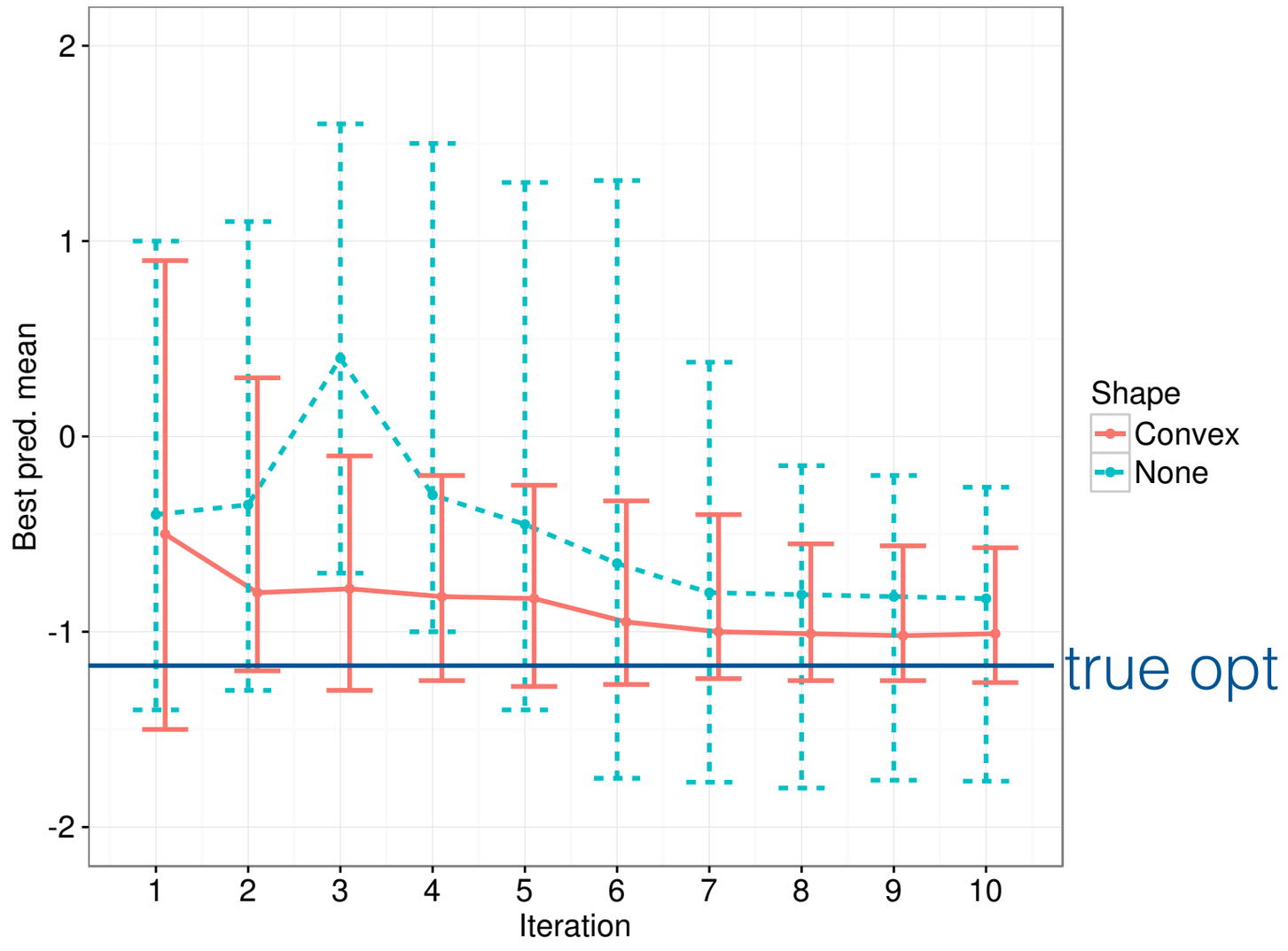
MC averages for sample size ranging from 1 to 120

Example: Optimal Design

- Pick an initial point (sample size) at random and find minimum loss; use acquisition function to select where to go at next iteration. We stop at 10 iterations.

- We do this a 100 times and plot the MC median along with quantiles 2.5% and 97.5% (these are not std. errors).

- The convex GP outperforms the unconstrained GP and appears to be more **robust to the choice of starting points**.



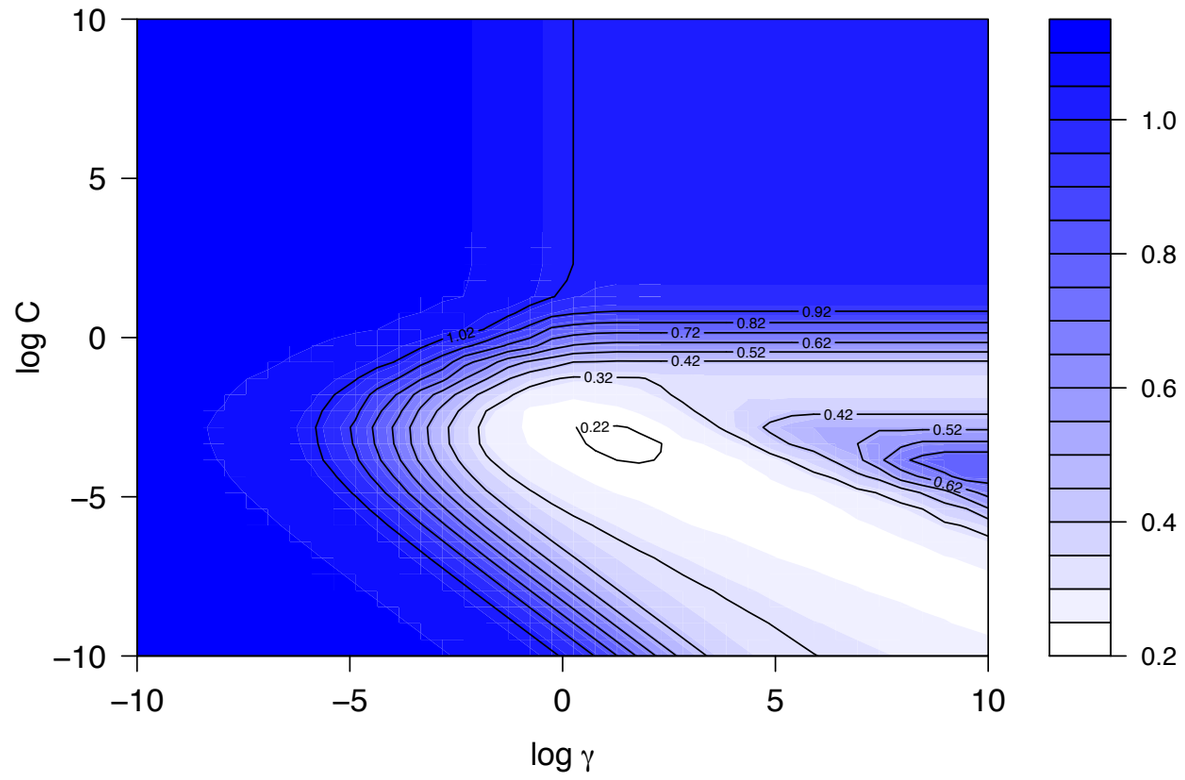
If this were a real application, we'd show optimal sample sizes… (sorry!)

Example: SVM

- Training an SVM with a squared-exponential kernel on the “famous” Ozone dataset.

- The task is predicting ozone readings in LA using 12 other variables.

- The performance metric is 10-fold CV error.

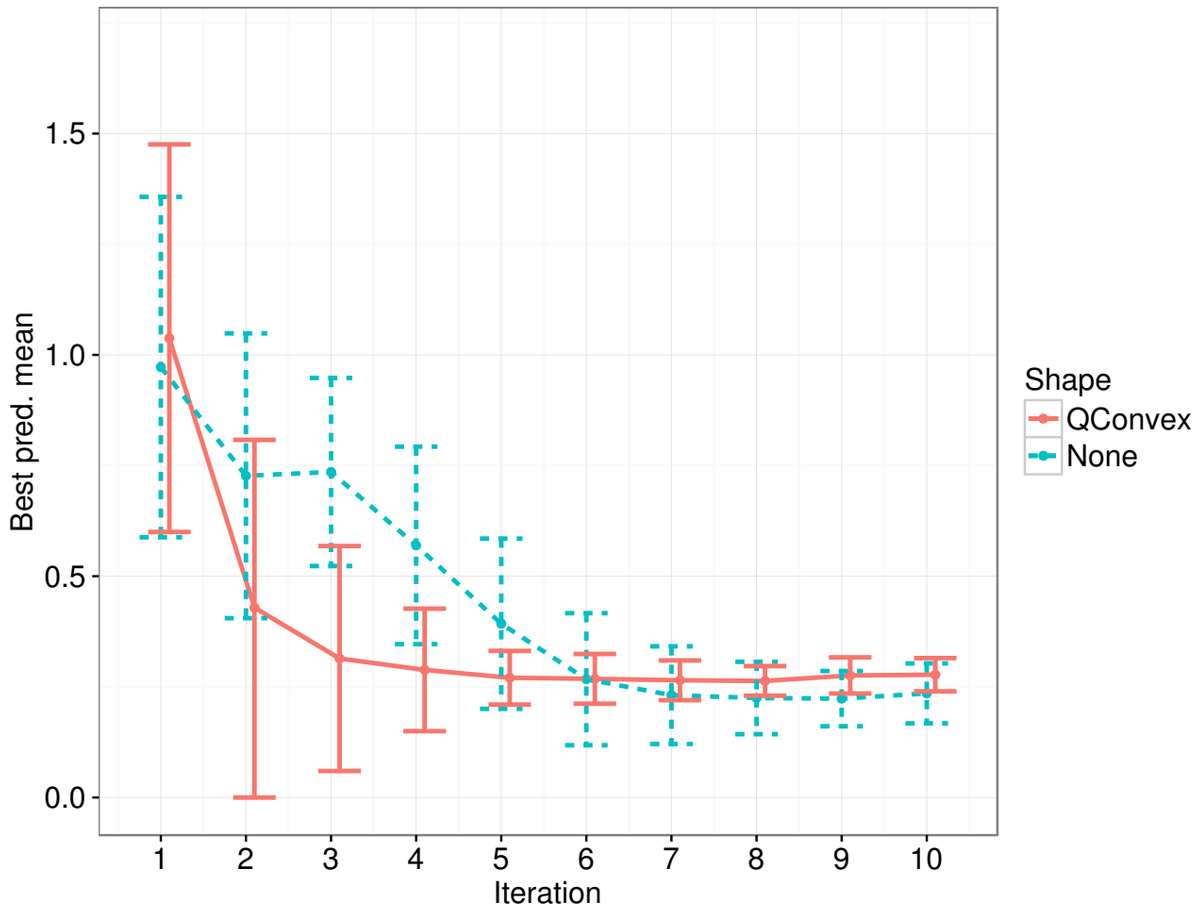


Example: SVM

- We compare GP with quasiconvexity constraints on both (logged) parameters vs unconstrained GP.

- This time, **one starting point** and compare the uncertainty in the minimum expected CV error in the observed samples.

- Quasiconvex GP **stabilizes more quickly** to a solution with **lower uncertainty** than the unconstrained version.



Wrap-up

- When we run a numerical method, especially if we can't afford to run it "long enough," **we don't know what the right answer is**.

- It makes sense to quantify that uncertainty **probabilistically**.

- Bayesian Optimization uses GPs as surrogate models for objective functions.

- Prior information about the shape of the objective function is often available, se we could try to introduce it.

Future Work

- Incorporating shape constraints presents serious computational challenges.

- Our approximations seem to work OK in our toy problems, but we haven't tried to use them in real problems yet.

- The acquisition function we used doesn't use first derivative information. It'd make sense to make that decision using information about the minimum expected first derivatives.

Thanks!

References

Poincaré, H. (1896). *Calcul des probabilités*. Paris: Gauthier-Villars.

Diaconis, P. (1988). Bayesian numerical analysis. *Statistical Decision Theory and Related Topics IV, 1*, 163–175.

Hennig, P., Osborne, M. A., & Girolami, M. (2015). Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 471(2179).*

Lindley, D. V. (1956), On a measure of information provided by an experiment. *Annals of Mathematical Statistics, 27 (4):* 986–1005.

Attolini, C. S. O., Peña, V., & Rossell, D. (2015). Designing alternative splicing RNA-seq studies. Beyond generic guidelines. *Bioinformatics, 31(22), 3631-3637.*

*Müller, P., & Parmigiani, G. (1995). O*ptimal design via curve fitting of Monte Carlo experiments. *Journal of the American Statistical Association, 90(432),* 1322-1330.

Papoulis, A., & Pillai, S. U. (2002). *Probability, random variables, and stochastic processes*. McGraw-Hill Education.

Williams, C. K., & Rasmussen, C. E. (2006). *Gaussian processes for regression.* MIT Press.

Wang, X. & Berger, J.O. (2016). Estimating shape-constrained functions using Gaussian processes. *SIAM/ASA Journal on Uncertainty Quantification, 4(1).* 1-25.

Z. I. Botev. (2016) The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology).*

Check out probabilistic-numerics.org/literature/ and bayesopt.com for more.