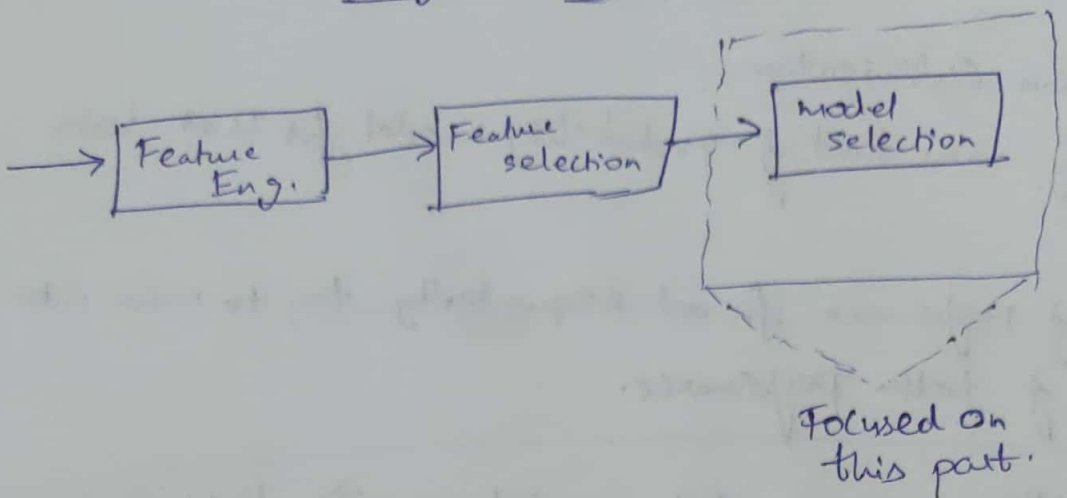
Bayesian Optimization

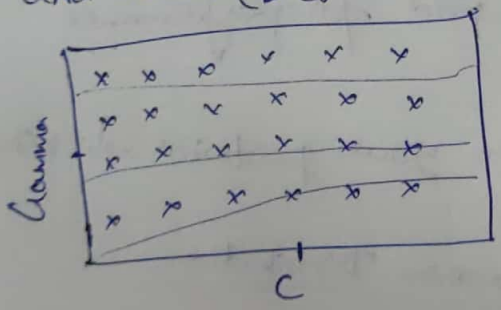


Bayesian optimization in general can be used to optimize any black box function.

f_m, hold out
↓
θ h.ps.

$f_m(\theta|D)$ → Conditional on the data.

manual Grid Search (L(gamma, c) → AVC → works for low-dimensional problems.



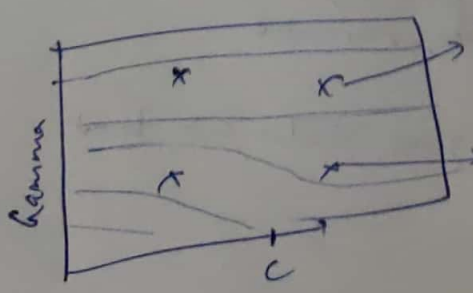
Contour plots of loss function

Does not scale well to higher Dimensions.

X (Big models on Big datasets).

Curse of Dimensionality.
↳ more infeasable with higher dimensions

Random Grid Search



→ low value of AVC so shldn't focus here.

→ high value of AVC, so better to focus here.

* Steer your Algorithm to the region of better performance.

Scanned with CamScanner

Sequential Model based Optimization Algorithms

↳ Bayesian Optimization.

Building some kind of Probability model for that loss function.

Build a Model of performance fn and sequentially. try to move into the Region of better performance.

> Building a probabilistic model over the loss surface

fn(0) → if this takes 1hrs to evaluate on out-of-sample

Better to spend 2mins on updating your beliefs.

In Bayesian optimization, we use something called Gaussian processes to actually model the model of a performance function.

Gaussian processes as a prior for performance functions we have.

Use Bayes Rule to update your posterior distribution

Gaussian Process	Gaussian Distribution
GP's are a generalization of GDs over functions	Random variables.
Specified by Mean function, Covariance fn.	Specified by. mean, variance
GP gives mean, variance of a normal distribution over the possible values of f at x.	N.D gives Scalar no. back.

Scanned with CamScanner

GP → $f_{m_1}(\theta)$

You can build a normal distribution over the potential values of the performance function at that point

Where am i going to sample next?

Utility fns → Called Aquisition fns (in Bayesian context)
(Next best guess)

(Expected improvement Aquisition fn.)

Exploration Vs Exploitation tradeoff. [Exploit a current region which is optimal & Explore a new region]

Tune the Algorithm which Tunes the performance function

i.e G.P's kernals are a H.P.

Use Pre-packaged model | Code it yourself

Bayesian SearchCV | Sklearn.gaussian_process.

→ Use log-uniform domain instead of uniform domain for H.Ps
→ G.P's kernel as H.P.

Production-ready modules for Bayesian optimization

1) Spearmint
2) Hyperopt
3) MOE
4) Hyperband (model free).
5) SMAC (model free)

Scanned with CamScanner