Last Time

Lecture 10

EAIS S'25

Andrea Bajcsy

- ☐ uncertainty quantification!
- ☐ epistemic / aleotoric
- ☐ modeling paradigms

This Time:

- ☐ practical methods for UQ

Announcement: midterm report due March 14th (Friday)

CREDIT: Notes inspired by Prof. Eric Nalisnick's lecture @ m^2L

Summary of UQ Methods

	Frequentism	Bayesianism
✓	data-driven, easy comp. MLE $\hat{\theta} = \arg\max_{\theta} p(y \mid x; \theta)$	prior dist "jump starts" learning, posterior models uncertainty over θ params $p(\theta \mid D) = \frac{p(\theta) \prod p(y \mid x; \theta)}{p(D)}$
✗	misled by sampling noise, dataset size, etc.	computation usually too costly for exact solution

frontiers: ⇒ "beef them up" ⇒ approximate this!

Practical Methods for UQ

Frequentism

1) bootstrap aggregation ("bagging") "ensemble"
2) conformal prediction ← next week: Prof. Anushri Dixit will lecture

Bayesianism

1) sample-then-optimize ensembling "ensemble"
2) Gaussian linear regression ⇒ Gaussian Processes (GPs)
3) Laplace Approximation

BOOTSTRAP AGGREGATION (BAGGING)

Recall how frequentism assumes that the randomness comes from the data sampling process. But, we have fixed dataset!

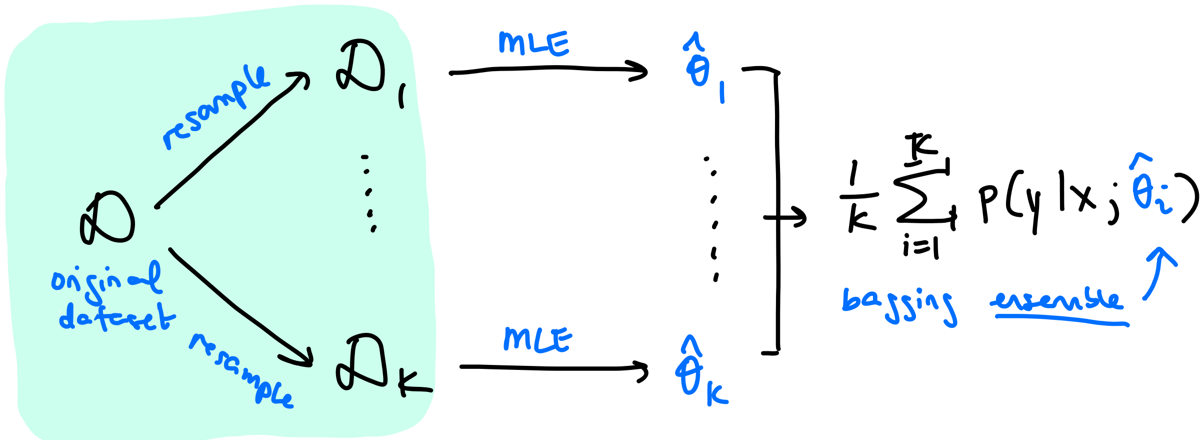
Bootstrapping synthesizes additional datasets by resampling from the fixed training set.

Dirac delta

$$\{\mathcal{D}_k\}_{k=1}^{K} \sim \frac{1}{N}\sum_{i=1}^{N} \delta[(x_i, y_i)]$$

"sample with replacement"

Intuition: w/ equal probability I will sample a data pt. from the existing dataset, draw a new data pt & put it into a new dataset. Do this K times.

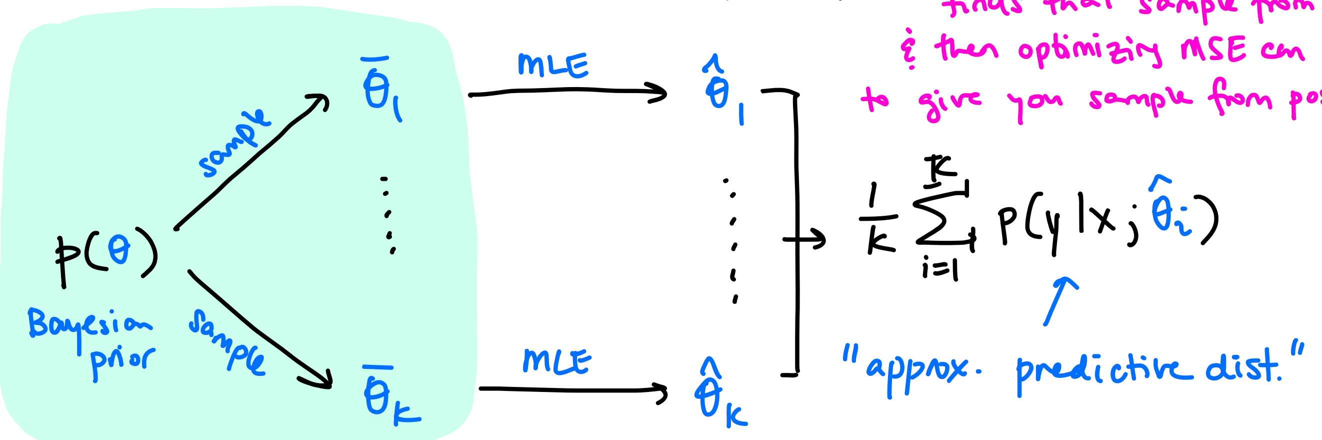


Further Reading: "Intro to the Bootstrap" by Efron & Tibshirani

Sample-then-Optimize Ensemble

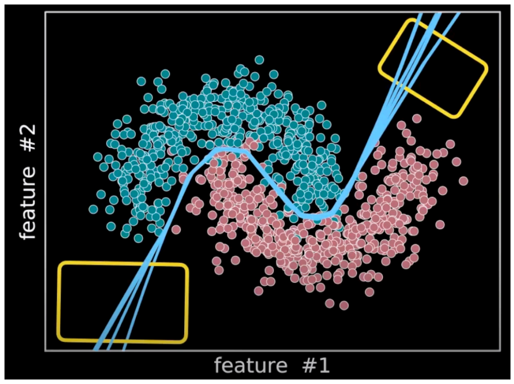
Bayesians model randomness in the prior too. We can perform a bagging-like procedure but using samples from the prior to initialize training! $\{\bar{\theta}_k\}_{k=1}^{K} \sim p(\theta)$

$\boxed{\text{Matthews et. al 2017}}$ finds that sample from prior & then optimizing MSE can be shown to give you sample from posterior.



NOTE: $\boxed{\text{Izmailov et. al. ICML 2021}}$ finds surprisingly comparable results btwn. this approach & high-fidelity Bayesian inference.

ASIDE: this is what generated these K decision boundaries from K NN's. ⟶



← from Prof. Eric Nalisnick's lecture @ M2L.

Re-initialize NN parameters, prior comes from the initialization scheme implemented in scikit-learn

Gaussian (linear) Regression

Is there any way to model the Bayesian uncertainty explicitly/exactly? i.e.

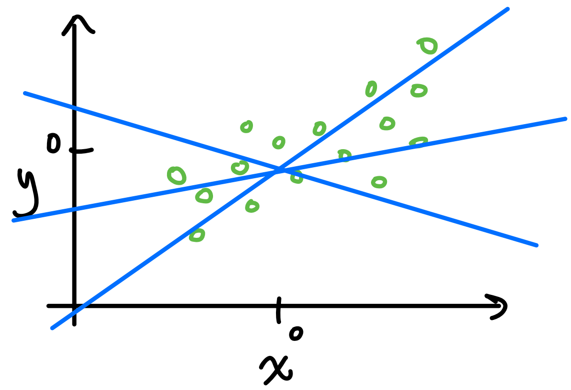
$$p(\tilde{y} \mid \tilde{x}, \mathcal{D}) = \int_\theta p(\tilde{y} \mid \tilde{x}; \theta)\, \underbrace{p(\theta \mid \mathcal{D})}_{\text{posterior}}\, d\theta$$

The core assumption / condition under which you can get this exactly is that all aspects of our model + world are Gaussian. The reason why this is helpful is b/c of the properties of Gaussians:

$$\boxed{\text{ONCE A GAUSSIAN, ALWAYS A GAUSSIAN}}$$

Specifically, we will start w/ regression problems but we will use a running example of linear models.

data: $y \sim \mathcal{N}(\underbrace{\theta^T x}_{\text{mean}}, \underbrace{\Sigma_1}_{\text{covariance}})$ $\quad x \in \mathbb{R}^n$



model: $\hat{y} = \hat{\theta}^T x$

I could just fit a line here, but I also want some uncertainty estimate over alternative lines I could have chosen ($p(\theta | \mathcal{D})$)

PRIOR: $p(\theta) = \mathcal{N}(\mu_0, \Sigma_0)$

LIKELIHOOD (i.e. MODEL): $p(y | x ; \theta) = \mathcal{N}(\theta^T x, \Sigma)$

POSTERIOR: $$\boxed{p(\theta | \mathcal{D})} = \frac{\boxed{p(\theta)} \prod_{i=1}^{N} \boxed{p(y_i | x_i ; \theta)}}{\int_{\bar{\theta}} p(\bar{\theta}) \prod_{i=1}^{N} p(y_i | x_i ; \bar{\theta}) d\bar{\theta}}$$

Gaussian

Gaussian

Closed-form expression of the posterior which looks like Gaussian!

$$p(\theta | \mathcal{D}) = \mathcal{N}(\mu_N, \Sigma_N)$$

a function of Σ_0, Σ, θ

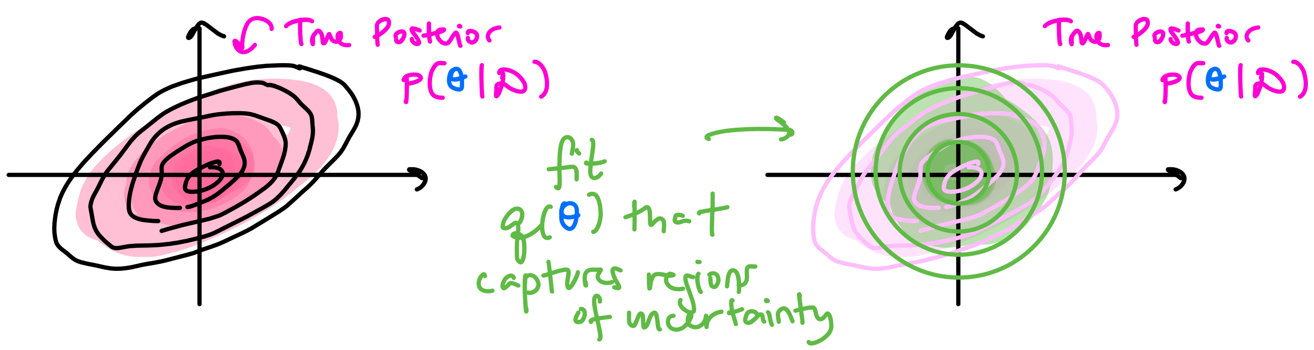
function of $\Sigma_N, \Sigma_0, \mu_0, \Sigma, \theta, y_i$'s

! ULTIMATELY, these operations are just matrix-vector mult & addition

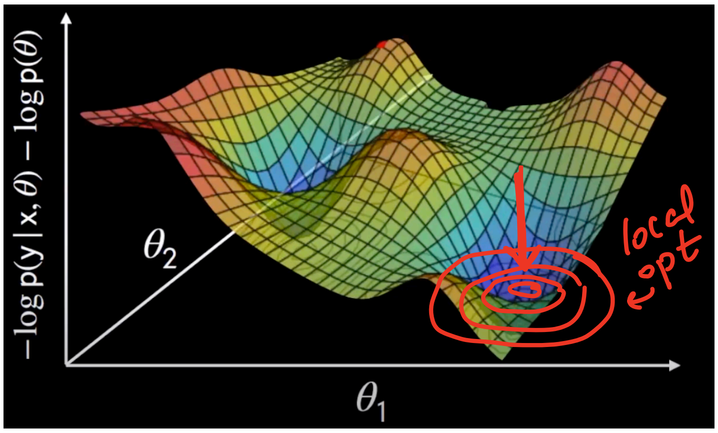
LAPLACE APPROXIMATION (LA)

What if my models aren't Gaussian?

We want to compute a tractable approx to the true Bayesian posterior by taking the intractable posterior dist and fit a simple dist to it!



Laplace approx. is old technique w/ recent resurgence (~2021)



Normally we'd stop here ↓ & return this local opt, but Laplace says lets find a local distribution around this pt. estimate $\hat{\theta}$

LA says that the posterior can be approx. by Gaussian dist where the mean is that $\hat{\theta}$ and its variance is the inverse Hessian matrix.

$$p(\theta \mid \mathcal{D}) \approx \mathcal{N}(\hat{\theta}, H^{-1}(\hat{\theta}))$$

$\boxed{Q}$ where does this come from?

$$p(\theta \mid \mathcal{D}) := \frac{1}{\int \boxed{p(\mathcal{D} \mid \theta) p(\theta)} d\theta} \underbrace{\boxed{p(\mathcal{D} \mid \theta) p(\theta)}}_{h(\theta)} \quad \text{// Bayes Rule}$$

$$= \frac{1}{Z} h(\theta)$$

we want to approx. w/ Gaussian. First, note that:

$$Z = \int \exp[\log h(\theta)] \, d\theta$$

let $\hat{\theta}$ be local min.

FACT: $$\int \exp\left[-\frac{1}{2} x^T A x\right] dx^n = \frac{\sqrt{(2\pi)^n}}{\sqrt{\det A}}$$ ← dim. of x

☺ integral of Gauss. func is closed form!

STEP 1: 2^{nd} order Taylor Series Expansion about $\hat{\theta}$

$$\log h(\theta) \approx \log h(\hat{\theta}) + \underbrace{\boxed{\nabla_\theta \log h(\hat{\theta})^T (\theta - \hat{\theta})}}_{= 0 \text{ @ optimum.}}$$
$$+ \frac{1}{2}(\theta - \hat{\theta})^T \nabla_\theta^2 \log h(\hat{\theta})(\theta - \hat{\theta})$$

$$\log h(\theta) \approx \log h(\hat\theta) - \left(-\tfrac{1}{2}(\theta-\hat\theta)^T \nabla^2_\theta \log h(\hat\theta)(\theta-\hat\theta)\right)$$

$$= \log h(\hat\theta) - \left(\tfrac{1}{2}(\theta-\hat\theta)^T \boxed{\Lambda}(\theta-\hat\theta)\right)$$

$\Lambda := -\nabla^2_\theta \log h(\hat\theta)$

STEP 2: Plug into integral!

$$\int \exp[\log h(\theta)]\, d\theta$$

$$\approx \int \exp\left[\log h(\hat\theta) - \left(\tfrac{1}{2}(\theta-\hat\theta)^T \boxed{\Lambda}(\theta-\hat\theta)\right)\right] d\theta$$

$$= h(\hat\theta) \int \exp\left[-\left(\tfrac{1}{2}\underbrace{(\theta-\hat\theta)^T}_{:=\text{"x"}} \underbrace{\boxed{\Lambda}}_{:=\text{"A"}}(\theta-\hat\theta)\right)\right] d\theta$$

$$= h(\hat\theta)\, \frac{\sqrt{(2\pi)^n}}{\sqrt{\det \Lambda}}$$

hessian matrix! eval @ $\hat\theta$

STEP 3: Plug approx back into $p(\theta \mid \mathcal{D})$ posterior!

$$p(\theta \mid \mathcal{D}) := \frac{1}{\int \boxed{p(\mathcal{D}\mid\theta)p(\theta)} d\theta} \underbrace{\boxed{p(\mathcal{D}\mid\theta)\,p(\theta)}}_{h(\theta)}$$

$$= \frac{\sqrt{\det \Lambda}}{\sqrt{(2\pi)^n}} \cdot \exp\left[-\tfrac{1}{2}(\theta-\hat\theta)^T \Lambda (\theta-\hat\theta)\right]$$

Hessian!

This is just Gaussian density $\theta \sim \mathcal{N}(\underbrace{\hat\theta}_{\text{mean}}, \underbrace{\Sigma}_{\text{covariance}})$

by def^n of multivar Gaussian: $\Sigma := \Lambda^{-1}$ Hessian!

If $\theta \in \mathbb{R}^m$ then the hessian $\Lambda \in \mathbb{R}^m \times \mathbb{R}^m$. But, if θ is weights of NN, this could be HUGE (e.g. GPT!)

$\Rightarrow$ in practice, use only last layer θ's & impose low-rank Λ structure!

"Laplace Redux" by Daxberger et.al. Neurips 2021