Last Time

- control-theoretic safety
- monitoring & fallbacks via VLMs

Lecture 12

EAIS S'26

Andrea Bajcsy

This Time: → ML Safety!

- uncertainty quantification!

CREDIT: Notes inspired by Prof. Eric Nalisnick's lecture @ m²L

Uncertainty Quantification for Predictive Models

So far, we have talked @ length about

safe decision-making / control

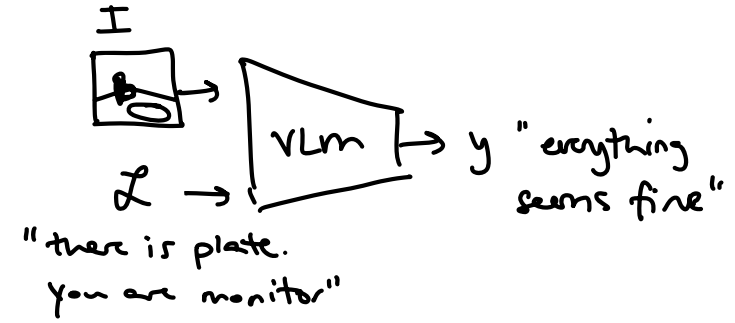
where safety ≡ constraint satisfaction

decision-making ≡ computing a policy/plan that makes sure current actions don't cause future safety violations.

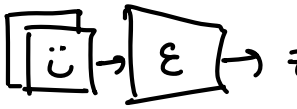
But we have also started to see many more "components" or "models" that our decision-making depends on being driven by data:

These are all **predictive models**

ex. LLM / VLM which: predict next tokens/sentences

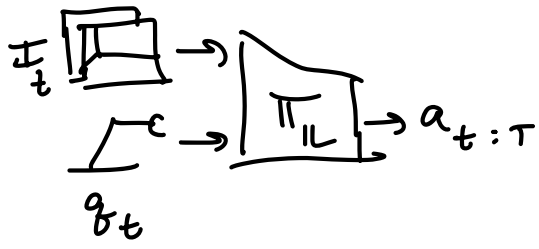


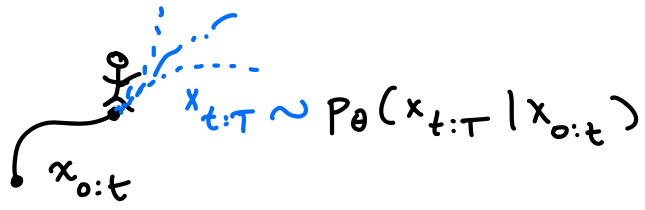
ex. world models which predict next latent state



$$\Rightarrow z_{t+1} \sim p_\phi(z_{t+1} \mid z_t, a_t)$$

ex. imitation robot policies predict next action traj. or human trajectory forecasters predict next state traj





Let's abstract these model architectures a little bit so we can unify our discussion. In general, prediction problems look something like this

"predictor" → $f(x) = \hat{y}$ ← "prediction"

"feature vector" → x

ex 1 diagnosis $\hat{y} \in \{\text{sick}, \text{healthy}\}$

ex 2 future (x,y) positions of human $\hat{y} \in \mathbb{R}^{2 \times T}$ ← pred horiz.

ex 1 health measurements like temp, blood pressure...

ex 2 history of human (x,y) positions over time $x \in \mathbb{R}^{2 \times N}$ ← hist length

❗ note:
x is not state here! it's often measurements / obs. But ML ppl use x often so we use this here too!

But wait, how certain is the model $f(\cdot)$ about its prediction? These predictive models will interact with "downstream" decision-making modules (e.g., a doctor who looks @ medical diagnosis, a robot planner that looks @ the predictions of where the human will move).

What we may want is something like a confidence statement

$$f(x) = \hat{y} \quad \text{AND} \quad \mathbb{P}(y = \hat{y} \mid x)$$

ex 1 85% confident that person is sick.

> Our goal is to know what our predictive models do not know.

Two Types of Uncertainty

We first need to define what is uncertainty.

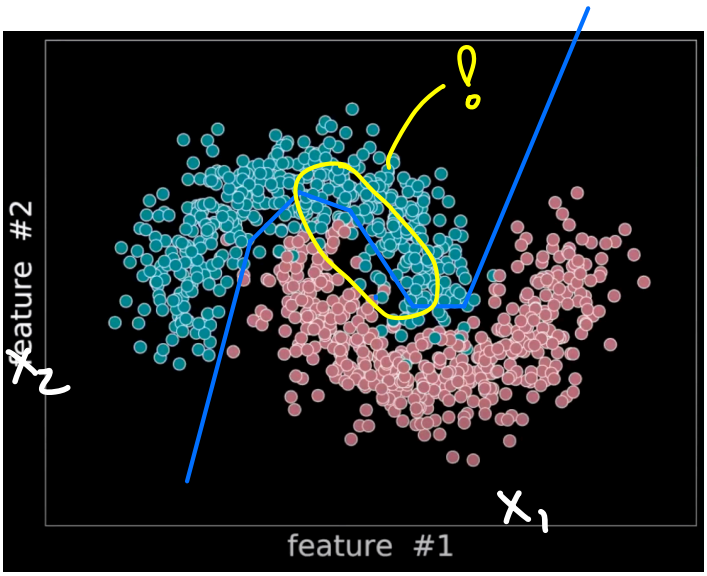
1. **ALEOTORIC** — Bayes E of prob. data dist. is the lowest possible error rate for any classifier: an instance is misclassified by a classifier that knows the true class probabilities given the predictors

- fundamental, related to **Bayes error rate** ⇓ this is ≠ 0 if class labels are not deterministic!
- uncertainty that is "irriducible" even if you collect more data

In regression Bayes Error is = the noise variance.

→ only way possibly around it is to collect more features

ex 1 Train NN to learn decision boundary (classifier: $f(x) = \hat{y}$)



← img. from Prof. Eric Nalisnick's lecture @ m²L

high aleotoric uncertainty in region b/c there is fundamental overlap between the distributions (green pts in red; red pts in green).

ex 2 Suppose true data is linear w/ Gaussian Noise:

$$y \sim \mathcal{N}(a + bx, \sigma^2).$$

The optimal estimator is linear regressor $\hat{y} = \hat{a} + \hat{b}x$. As we add more data, $\hat{a}$ and $\hat{b} \to a, b$. So, the best error we could hope to achieve is σ^2, the irreducible error/noise in the data itself.

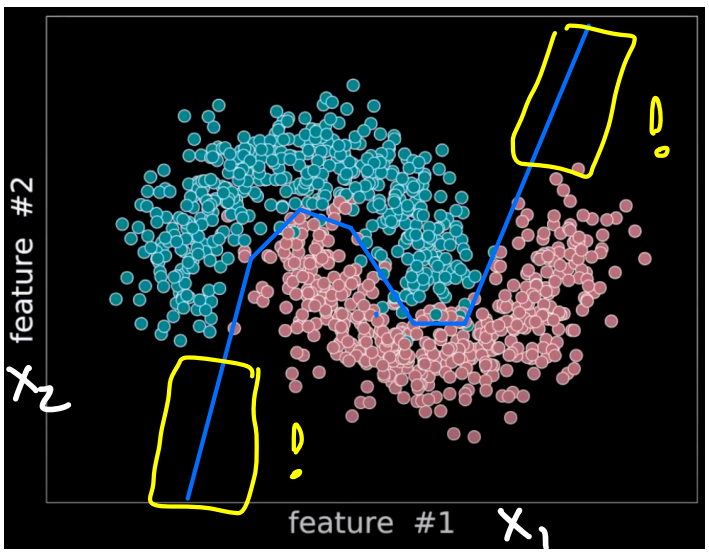
train with squared loss

2. EPISTEMIC

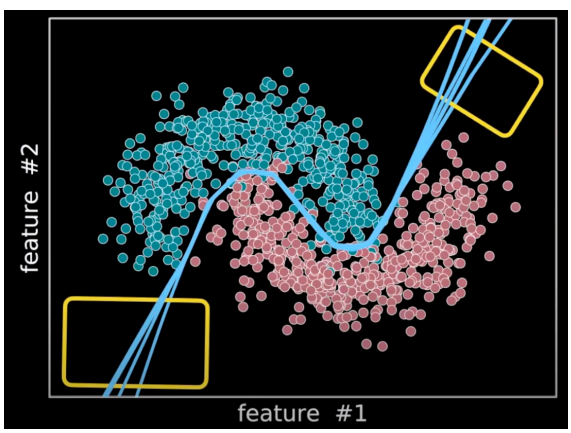
- "easier" to deal with because it relates to a lack of data / experience / observations
- always reduced by collecting more data

ex.

region of high epistemic uncertainty is in region where our model made predictions but has no data to be guided with!



If you were to train this model w/ different seeds, you get slightly different solutions! High epistemic uncertainty b/c no data to guide our model's predictions.



It's important to understand these differences in uncertainty but in practice its very hard to know the difference!

⇒ for most of these lectures, we will brush this distinction under the rug a bit and say uncertainty is high when either type of uncertainty is high.

Notation & Assumptions

For these lectures, we will assume that there is a fixed, unknown distribution that generates data:

$$y \sim \mathbb{P}(y \mid x)$$

(x: features; y: true "labels")

we get to see (finite) samples from this dataset, our training data:

$$\mathcal{D} := \{(x_i, y_i)\}_{i=1}^{N}$$

We fit a model to recover the ground-truth distribution:

$$f(x) := p(y|x) \approx \mathbb{P}(y|x)$$

Modeling Paradigms

Broadly there are two modeling schools of thought: frequentism and Bayesianism. There are many philosophical debates we could have about these but an intuitive separation comes from where we model "randomness" as coming from:

1) FREQUENTISM: randomness comes from data sampling dist.

What this translates to in terms of learning is MLE

MAXIMUM LIKELIHOOD ESTIMATION (MLE): $f_\theta(x) \equiv p(y|x;\theta)$

maximize the log-likelihood of model parameters θ:

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \sum_{i=1}^{N} \log p(y_i | x_i; \theta)$$

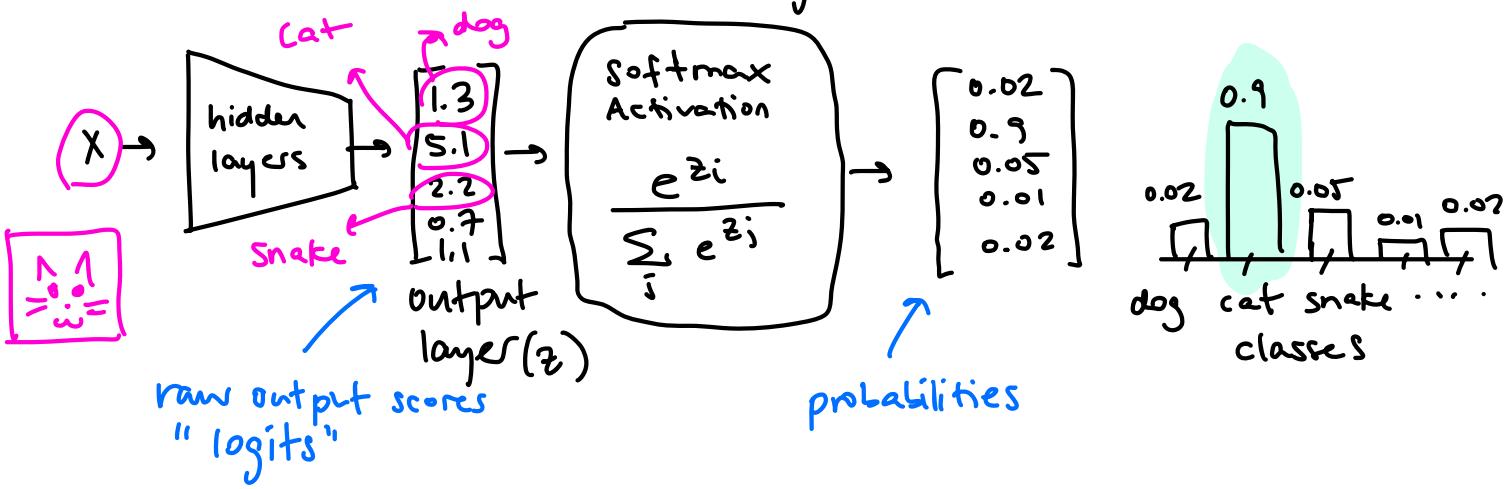
FREQUENTIST IDEAL UQ

IDEALLY, under this paradigm, if I have a really big model and really big dataset, etc. we can quantify uncertainty by simply looking at the model probabilities.

i.e. assume learning worked really well, so

$$p(\hat{y} | x; \hat{\theta}) \approx \mathbb{P}(y = \hat{y} | x)$$

here, uncertainty quantification appears trivial! e.g. if you have a classifier $p(\hat{y} | x; \hat{\theta})$, then you can look at the softmax output layer and read out:



So, are we "done"??

Frequentist learning : Limitations

In practice, as you may have experienced yourself, we can't usually rely on these probabilities directly...

ex. from Guo et al. ICML 2017. "On Calib. of Modern NNs"

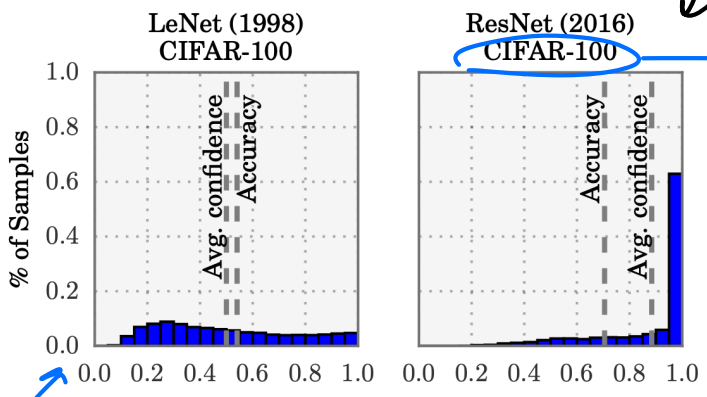


image classification dataset

LeNet is old, smaller NN (5 layer)
softmax probabilities associated w/ predicted label

⇒ old model is "evenly distributed" confidence.
⇒ avg. confidence matches accuracy (~50%)

ResNet bigger more power. (110-layer)

⇒ new model has higher accuracy (70%) but average confidence is MUCH HIGHER than accuracy (~90%)

⚠ you can't just read off softmax outputs b/c you'd get an overconfident estimate of how good you'd actually be.

2) BAYESIANISM: randomness influenced by prior distribution over model parameters.

Bayesian learning comes from first defining a prior distribution over model parameters $p(\theta)$ which "jump starts" your learning — it can constrain your solutions to certain "plausible" sol^{n}s. You multiply your prior by your likelihood (this is where your model comes in!) $p(y_i|x_i;\theta)$ and then you normalize to get a posterior distribution that has updated now you got some data:

$$\underbrace{p(\theta|\mathcal{D})}_{\text{posterior}} = \frac{\overbrace{p(\theta)}^{\text{prior}} \prod_{i=1}^{N} \underbrace{p(y_i|x_i;\theta)}_{\text{likelihood}}}{p(\mathcal{D})}$$

note: here we assume (x_i, y_i) are iid sampled!

Normalizing constant is the hard part about Bayesian learning! it requires you to integrate over all possible alternative parans!

$$= \frac{p(\theta) \prod_{i=1}^{N} p(y_i|x_i;\theta)}{\int_{\bar{\theta}} p(\bar{\theta}) \prod_{i=1}^{N} p(y_i|x_i;\bar{\theta})\, d\bar{\theta}}$$

"marginal likelihood" $= p(\{x_i, y_i\}) = \int_{\theta} p(\{x_i, y_i\}, \theta)$

⚠ When θ is parameters of NN, this is a high-D integral!

Recall: Bayes' Rule $\Rightarrow$ $P(A|B) = \frac{P(A,B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$

Bayesian IDEAL UQ

Assuming you could solve the normalizer, then given a new data point $\tilde{x}$, you can compute the posterior predictive distribution:

$P(A,B) = P(A|B)P(B)$

$$= \int_\theta p(\tilde{y}|\tilde{x},\theta,\mathcal{D})$$

assume: y doesn't depend on $\mathcal{D}$, but θ depends on $\mathcal{D}$

$$p(\tilde{y}|\tilde{x},\mathcal{D}) = \int_\theta p(\tilde{y}|\tilde{x};\theta)\, p(\theta|\mathcal{D})\, d\theta$$

your predictive model

posterior distribution

you would use this to make preds b/c all the uncertainty your model has over different models in the posterior are accounted for.

Under (near) perfect learning, use post. pred. dist as your "ground-truth" probabilities

$$p(\tilde{y}|\tilde{x},\mathcal{D}) \approx \mathbb{P}(y=\tilde{y}|x)$$

You can report confidence just like before...

Bayesian learning: Limitations

Mostly computational, integrating over params hard for NNs, and so computing normalizer or posterior pred. dist is hard.