Last Time

- ☐ latent-space safety
- ☐ VLMs for monitoring & recovery

Lecture 9

EAIS S'25

Andrea Bajcsy

This Time:

- ☐ uncertainty quantification!

Announcement: midterm report due March 14^{th} (Friday)

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

Uncertainty Quantification for Predictive Models

So far, we have talked @ length about

safe decision-making / control

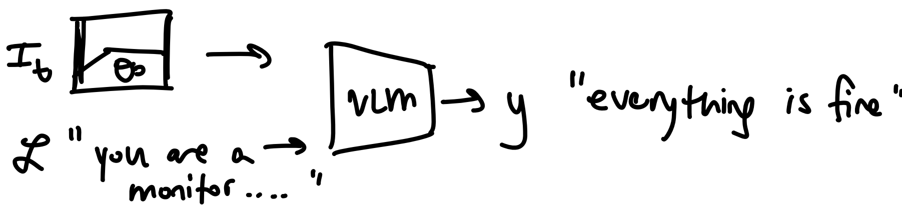
where $\boxed{\text{safe}}$ ≡ constraint satisfaction

$\boxed{\text{decision-making}}$ ≡ computing a policy/plan that makes sure that current action don't lead to future safety violations.

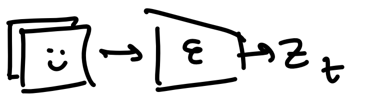
But, as we have pushed into the frontier, we have seen many more "components" or "models" that our decision-making depends on being "driven" by data.

These are all $\boxed{\text{predictive models}}$

ex. LLM / VLM which predicts next token

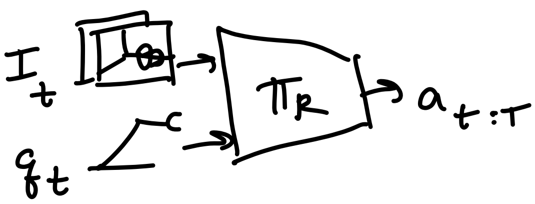


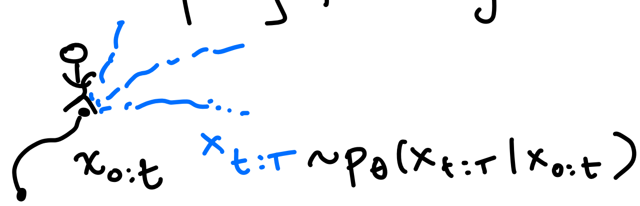
ex. world models which predict next latent state



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

ex. imitation policies OR human trajectory forecasting





Let's abstract these models, their architectures, etc. so we can unify our discussion.

In general, prediction problems look something like this.

$$f(x) = \hat{y}$$

"predictor" → f

"feature vector" → x

"prediction" → $\hat{y}$

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

ex. future (x,y) pos. of human

ex. health measurements (blood pressure, temp...)

ex. history (x,y) states of person

! x is not necessarily the true state. It can be measurements / observations.

Wait, what we want is to know how certain is $f(\cdot)$ about its prediction! These predictive models will interact with "downstream" decision-making modules (e.g. doctor who gives you drugs, robot planner which looks @ predictions to take actions)

What we would like is something like a confidence statement

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

ex. 80% confident the person is sick.

Our goal is to know what our predictive models do AND do not know

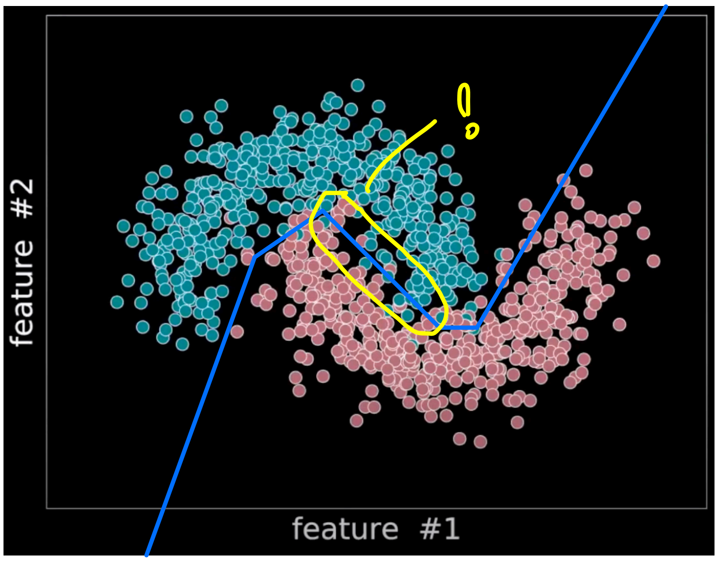
Two Types of Uncertainty

We first need to define what we mean by uncertainty.

1) ALEOTORIC

- fundamental, related to the Bayes error rate — "lowest possible error rate of a classifier"
- "irreducible" uncertainty even if you collect more data

⇒ only way around this is to collect more features ex. train NN to learn classifier $f(x) = \hat{y}$



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

high aleatoric uncertainty in region b/c there is fundamental overlap btwn. the distributions (green in red; red in green)

ex. suppose the true data is linear w/ Gaussian noise

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

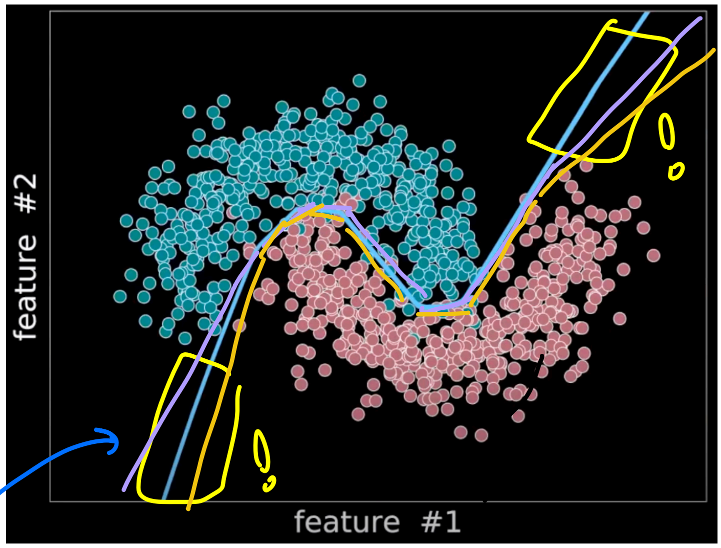
The optimal estimator is linear regressor $\hat{y} = \hat{a} + \hat{b}x$. As we collect more data, $\hat{a}, \hat{b} \to a, b$. So the best error we could get is σ^2, the irreducible error (noise in the data itself).

2. EPISTEMIC

- "easier" to deal with b/c it relates to a lack of data / experience
- always reduced by collecting more data!

! BUT in practice, hard! e.g. safety-critical data is less freq. and hard to get

ex. region of high epistemic uncertainty is in where our model makes predictions but has no data to be grounded with!



multiple training runs of NN with different rand. seeds.

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

⇒ for most of these lectures, we will brush this distinction under the rug a bit; say uncertainty is high if either types are high.

Notation & Assumptions

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

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

features

true "labels"

we get to see a (finite) number of samples to form our training data:

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

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

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

Modeling Paradigms

Broadly speaking, there are two modeling schools of thought: frequentism and Bayesianism. An intuitive separation comes from where we model "randomness" as coming from.

1) FREQUENTISM: randomness comes from the data distribution.

what this translates to in terms of model learning is learning the maximum likelihood estimator (MLE):

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

$f_\theta(x) := p(y \mid x; \theta)$

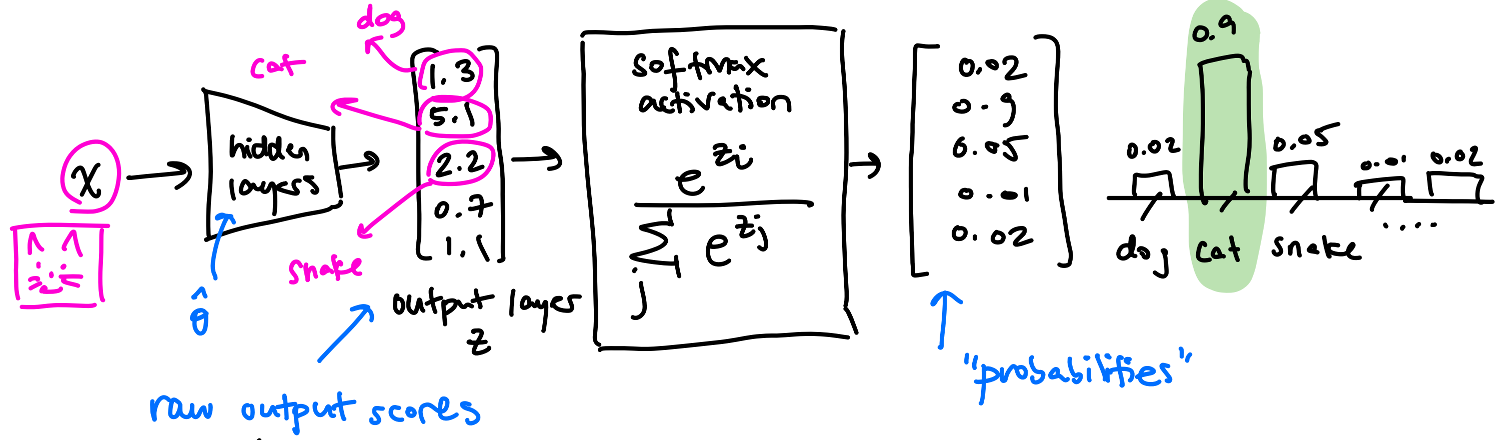
maximize the log likelihood of model parameters θ.

FREQUENTIST IDEAL UQ — uncertainty quantific.

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

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

Here, UQ looks trivial! If I have classifier $p(\hat{y} \mid x; \hat{\theta})$

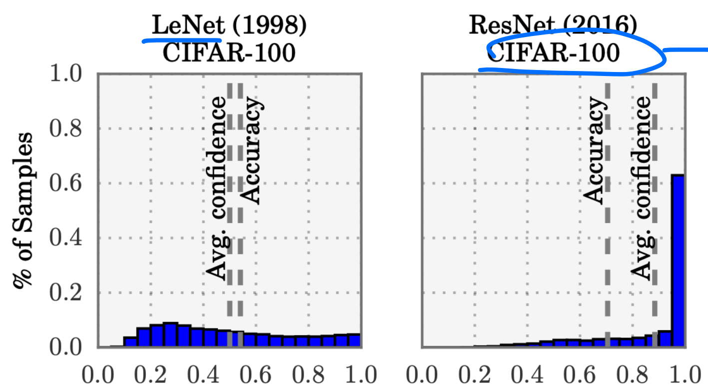


So are we "done"?

Frequentist Learning: Limitations

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

ex. $\boxed{\text{Guo et al. ICML 2017. "On calib. of Modern NNs"}}$



→ img. classific. dataset

$\boxed{\text{LENET}}$ is an old, smaller NN (5 layers). Showing softmax probabilities associated w/ each label

⇒ old model is "evenly distributed" confidence

⇒ avg. confidence matches accuracy (~50%)

$\boxed{\text{RESNET}}$ bigger, more powerful model (110-layers)

⇒ new model has higher accuracy (~70%) BUT the average confidence is much HIGHER (~90%)

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

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

In Bayesian learning, we define a prior distribution $p(\theta)$ over model parameters to "jump start" your learning – it can constrain your solutions to certain "plausible" solutions. You multiply your prior by your likelihood (this is where your model $p(y_i | x_i ; \theta)$ comes in) and then you normalize to get a posterior distribution that has been updated b/c you have seen data:

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

Normalizing constant is the hard part about Bayesian learn!

when this is NN params, hard!

IDEAL BAYESIAN UQ

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

$$p(\tilde{y} | \tilde{x}, \mathcal{D}) = \int_{\theta} \overbrace{p(\tilde{y} | \tilde{x} ; \theta)}^{\text{pred. model}} \overbrace{p(\theta | \mathcal{D})}^{\text{posterior}} \, d\theta$$

What you would use to make preds! B/c it has ALL the uncertainty of your model baked into it -- uncertainty over different models accounted for in posterior

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

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

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.