

MPhil ACS / Part III

Probabilistic Machine Learning (L173)

DAMON WISCHIK

~~Machine learning is building systems that make good predictions~~

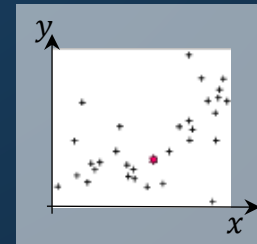
Machine learning is specifying and fitting ^{probabilistic} models

The probabilistic modelling view unifies simple questions and lets us ask deeper questions

- What justifies a model's confidence?
- What can we still predict, when the situation changes?

PREREQUISITES

- You can fit a simple predictive model $Y_i \sim \alpha + \beta x_i + N(0, \sigma^2)$
- You can write down the posterior of α from a Bayesian fit
- You can implement a neural network for simple prediction



EXAMPLE

There are three chess players in a club, $\{A, B, C\}$. They have played $n = 20$ games against each other:

pairing	games	wins
A vs B	10	8 for A
B vs C	10	6 for B
A vs C	0	

There is to be a ten-game match between A and C . What do we expect the outcome to be?

game	u	v	y
1	A	B	1
2	A	B	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$
20	B	C	1

Data: for each game $i \in \{1, \dots, 20\}$ we have a record (u_i, v_i, y_i) where the players are u_i and v_i and the winner is $y_i = 1_{u_i \text{ won}}$

Proposed model: $Y_i \sim \text{Bernoulli}(\sigma(s_{u_i} - s_{v_i}))$ where $\sigma(z) = \frac{1}{1+e^{-z}}$ with parameters $s_A, s_B, s_C \in \mathbb{R}$

Likelihood: The log likelihood of the full dataset, as a function of s , is

$$\log \Pr(\text{data}; s) = \sum_{i=1}^n \log \Pr(Y_i = y_i; s)$$

Fitting: Tune the parameters s_A, s_B, s_C to make the likelihood large

EXAMPLE

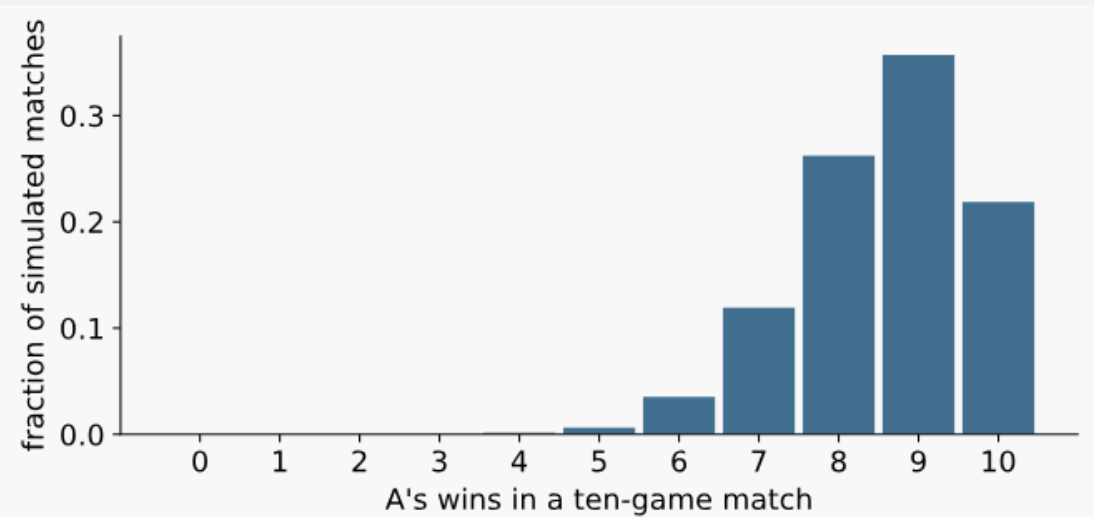
There are three chess players in a club, $\{A, B, C\}$. They have played $n = 20$ games against each other:

pairing	games	wins
A vs B	10	8 for A
B vs C	10	6 for B
A vs C	0	

There is to be a ten-game match between A and C . What do we expect the outcome to be?

After fitting the model, how do we answer the question?

Take the **fitted model**, and simulate from it the outcome we're interested in.



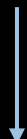
This course is about
formulating
probability models

Data



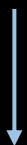
Proposed model

It's a proposal, and no one can tell us it's wrong.
"Modeller's Choice."



Likelihood

The likelihood is the only bridge
between the model and the fitting



Fitting

We'll treat the optimizer as a magic black box.

Fitted model

Our models are generally overparameterized.
And the questions we want to answer have no closed form.
So don't look at the actual fitted parameters, just simulate outcomes.

the maximization we just did

why?

what is this and how do we compute it?

what are these?

Maximizing $\text{Pr}^{\text{model}}(\text{data})$ is the same as minimizing $\text{KL}(\text{Pr}^{\text{true}} \parallel \text{Pr}^{\text{model}})$

KL is a different language for the fit we just did

Why bother?

- Some ideas become easier to think about *e.g. what happens with more data*
- Some choices become more visible, *e.g. why this measure of fit?* and so do their alternatives
- Everyone in machine learning speaks it

what is this?
↓
Maximizing $\Pr^{\text{model}}(\text{data})$ is the same as minimizing $\text{KL}(\Pr^{\text{true}} \parallel \Pr^{\text{model}})$

$\Pr^{\text{true}}$ is the distribution that we suppose the datapoints came from

Our datapoints (u_i, v_i, y_i) represent games.

We'll suppose they are samples of (U, V, Y) drawn from some "truth distribution".

not just the outcome Y
but the full datapoint

Nobody has seen it. All we have is samples.

what is
this?

Maximizing $\Pr^{\text{model}}(\text{data})$ is the same as minimizing $\text{KL}(\Pr^{\text{true}} \parallel \Pr^{\text{model}})$

$\Pr^{\text{model}}$ is a proposed distribution over the same sample space as $\Pr^{\text{true}}$

We've already proposed $Y \sim \text{Bernoulli}(\sigma(s_U - s_V))$

We need to model (U, V) too
e.g. $(U, V) \sim \text{Cat}(r)$

What do you propose?

$$\Pr^{\text{model}}(u, v, y) = r(u, v) \Pr_Y(y|u, v; s)$$

$$\Pr^{\text{model}}(\text{data}) = \prod_{i=1}^n r(u_i, v_i) \Pr_Y(y_i|u_i, v_i; s)$$

notation:

capital letters for random variables, lower case for values
 $y|u, v$ is conditioning on random variables
 $y; s$ lists parameters

what is
this?

Maximizing $\Pr^{\text{model}}(\text{data})$ is the same as minimizing $\text{KL}(\Pr^{\text{true}} \parallel \Pr^{\text{model}})$

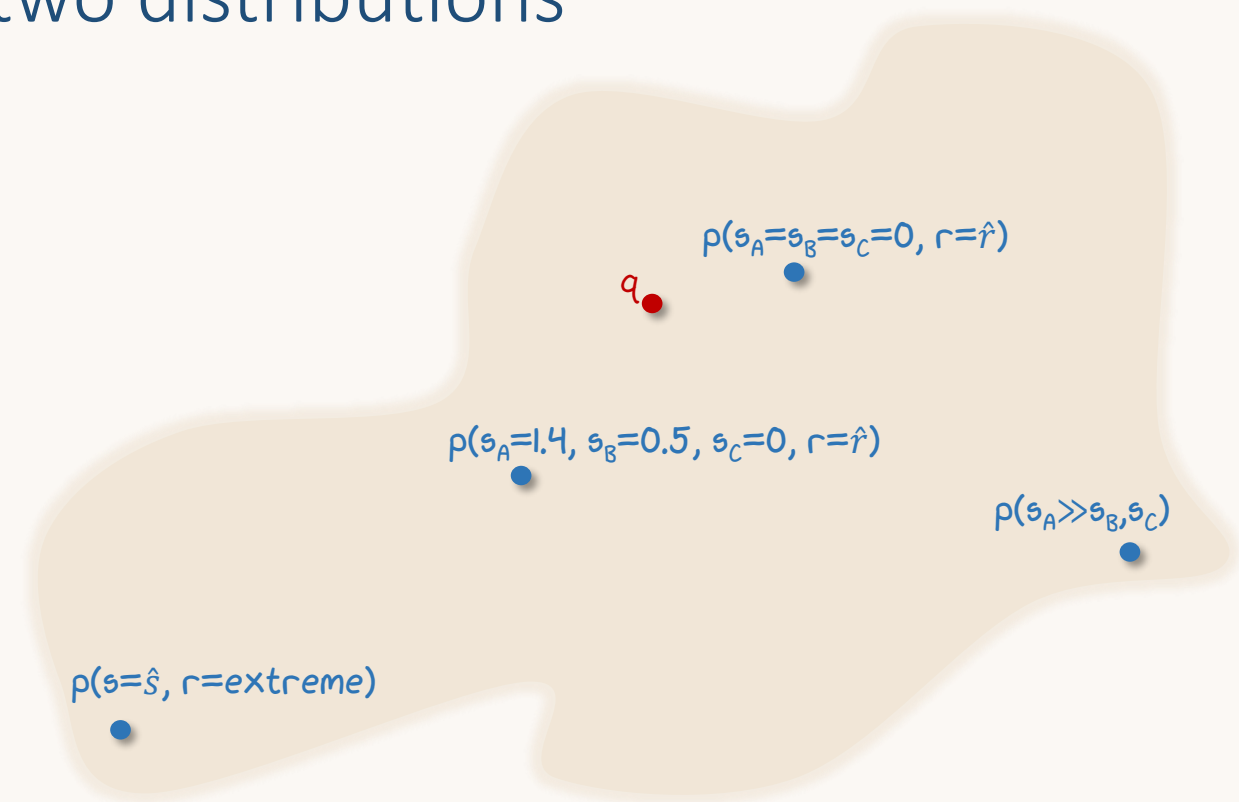
Think of KL as the distance between two distributions

Definition. For two distributions p and q on the same sample space,

$$\text{KL}(q \parallel p) = \mathbb{E}_{X \sim q} \log \frac{q(X)}{p(X)}$$

Lemma. $\text{KL}(q \parallel p) \geq 0$,
and $\text{KL} = 0$ only when $q = p$.

Shorthand. I'll write q for $\Pr^{\text{true}}$ and
 p_θ for $\Pr^{\text{model}}$ with parameters θ



how do we
do this?

Maximizing $\Pr^{\text{model}}(\text{data})$ is the same as minimizing $\text{KL}(\Pr^{\text{true}} \parallel \Pr^{\text{model}})$

We can't compute KL, but we can minimize it

Suppose our model has unknown parameters θ . Fitting means “minimize $\text{KL}(q \parallel p_\theta)$ over θ ”.

$$\text{KL}(q \parallel p_\theta) = \mathbb{E}_{X \sim q} \log \frac{q(X)}{p_\theta(X)} = \underbrace{\mathbb{E}_{X \sim q} \log q(X)}_{\text{we can never know this, but it's a constant so we don't need to}} - \mathbb{E}_{X \sim q} \log p_\theta(X)$$

we don't know q , so how
can we take an
expectation over it?

By Monte Carlo approximation,

$$\mathbb{E}_{X \sim q} \log p_\theta(X) \approx \frac{1}{n} \sum_{i=1}^n \log p_\theta(x_i) \quad \text{where } x_1, \dots, x_n \text{ are real data, i.e. sampled from the true distribution } q$$

Putting this together, $\text{KL}(q \parallel p_\theta) \approx \text{const} - n^{-1} \sum_i \log p_\theta(x_i)$, which we *can* minimize

Another way to write it: $\text{KL}(q \parallel p_\theta) \approx \text{const}' + \text{KL}(\hat{q} \parallel p_\theta)$

why?

Maximizing $\Pr^{\text{model}}(\text{data})$ is the same as minimizing $\text{KL}(\Pr^{\text{true}} \parallel \Pr^{\text{model}})$

EXERCISE

There are three chess players in a club, $\{A, B, C\}$.
They have played $n = 20$ games against each other:

pairing	games	wins
A vs B	10	8 for A
B vs C	10	6 for B

We proposed a model with parameters r and s ,

$$(U, V) \sim \text{Cat}(r)$$

$$(Y|u, v) \sim \text{Bernoulli}(\sigma(s_u - s_v))$$

Explain how to fit this model.

First step. Write out our proposed model's likelihood,

$$\Pr^{\text{model}}(u, v, y) = r(u, v) \Pr_Y(y|u, v; s)$$

Which part answers “who plays next?”?
Which part answers “who will win a match between those players?”?

Supervised learning is just a special case of unsupervised, in which the predictor distribution is left free then optimized out.

Suppose we want to fit a supervised learner, to predict $(Y|x)$

The true joint distribution can be factorized as $q(x, y) = q_X(x) q_Y(y|x)$

Let our proposed model be $\Pr^{\text{model}}(x, y) = \underbrace{r(x)}_{\substack{\text{the dist. of} \\ \text{predictors,} \\ \text{unconstrained}}} \underbrace{p_\theta(y|x)}_{\substack{\text{our parametric} \\ \text{supervised} \\ \text{learning model}}}$

$$\begin{aligned}\text{KL}(q \parallel r p_\theta) &= \mathbb{E}_{X,Y \sim q} \log \frac{q_X(X) q_Y(Y|X)}{r(X) p_\theta(Y|X)} \\ &= \mathbb{E}_{X,Y \sim q} \log \frac{q_X(X)}{r(X)} + \mathbb{E}_{X,Y \sim q} \log \frac{q_Y(Y|X)}{p_\theta(Y|X)} \\ &= \mathbb{E}_{X \sim q_X} \log \frac{q_X(X)}{r(X)} + \mathbb{E}_{X \sim q_X} \left[\mathbb{E}_{Y \sim q(\cdot|X)} \log \frac{q_Y(Y|X)}{p_\theta(Y|X)} \right] \\ &= \text{KL}(q_X \parallel r) + \mathbb{E}_{X \sim q_X} \text{KL}(q_Y(\cdot|X) \parallel p_\theta(\cdot|X))\end{aligned}$$

Explain what would be needed for our full model $\Pr^{\text{model}}$ to fit q perfectly.

$$\begin{aligned}\min_{r, \theta} \text{KL}(q \parallel r p_\theta) &= \min_r \text{KL}(q_X \parallel r) + \min_\theta \mathbb{E}_{X \sim q_X} \text{KL}(q_Y(\cdot|X) \parallel p_\theta(\cdot|X)) \\ &= \min_\theta \mathbb{E}_{X \sim q_X} \text{KL}(\dots) \quad \text{since } r \text{ is unconstrained, we can} \\ &\quad \text{achieve KL=0 for the first term}\end{aligned}$$

we're left with the usual
"fit a supervised learner"

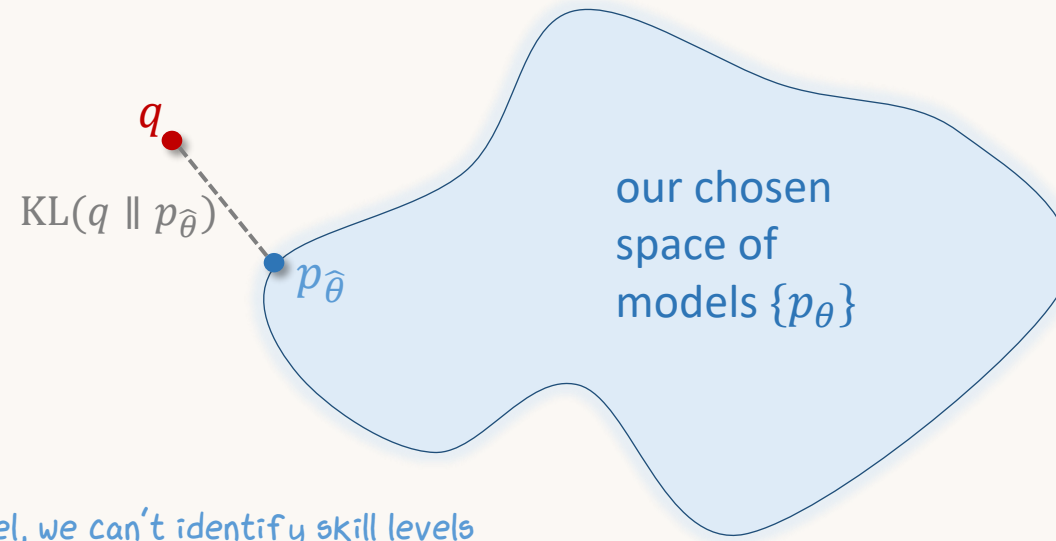
Propose a probability model P_r^{model} (data) for the entire dataset

- For the bits you care about, use a parametric term
- For the bits you don't care about, leave them unconstrained

Fit it by minimizing $KL(P_r^{\text{true}} \parallel P_r^{\text{model}})$
over all parts of your model, parametric and unconstrained

It's helpful to think of KL geometrically, as a distance

- The first step is choosing a class of models
- We'd like to find the model $p_{\hat{\theta}}$ that's closest to the truth q .
- Typically the optimum is non-unique.
- If our class is big enough to include q , we can achieve $KL=0$



*In our chess model, we can't identify skill levels
(since adding a constant doesn't change anything).
But we can identify skill differences.*

EXAMPLE

There are three chess players in a club, $\{A, B, C\}$. The true pairwise win rates are in fact

<u>pairing</u>	<u>wins</u>
A vs B	80% for A
B vs C	60% for B
A vs C	20% for A

We proposed a model with parameters r and s ,

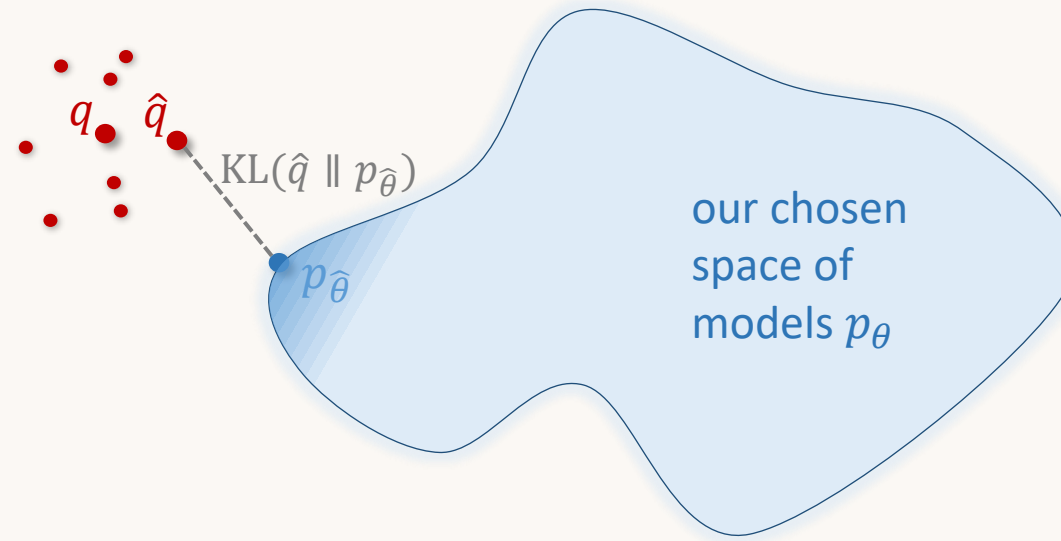
$$(U, V) \sim \text{Cat}(r)$$

$$(Y|u, v) \sim \text{Bernoulli}(\sigma(s_u - s_v))$$

1. Can this model fit the truth perfectly?
2. What is a model that could?
3. Does this model allow us to answer the original question (predict A vs C from A vs B and B vs C data)?

Model confidence also has a geometrical interpretation

- Our only knowledge of q is from a sample, whose empirical distribution $\hat{q}$ is close to q
- If we gathered new data we'd get a different $\hat{q}$ nearby
- There's a spread of likely $\hat{q}$ and thus of $p_{\hat{\theta}(\hat{q})}$ and this tells us how confident we should be
- accuracy = closeness of model-space to q
precision = sharpness of model-space at $p_{\hat{\theta}}$



After the break: We've gotten $\hat{\theta}$, but is $KL(\hat{q} \parallel p_{\hat{\theta}})$ small enough?