CSDS 440: Machine Learning

Soumya Ray (he/him, sray@case.edu)

Olin 516

Office hours T, Th 11:15-11:45 or by appointment

Zoom Link

Recap

- We may not use MLEs in generative models because _____. For NB, an alternative is to use _____. Here we add ____ to the numerator and ___ to the denominator.
- Continuous features can be modeled in Naïve Bayes using G___ distributions.
- Naïve Bayes produces a l____ decision boundary under a l___ transform.
- A discriminative learning algorithm is L____ R____. It models the l___ o____ as a l___ f____.
- How do we classify a new example with LR?
- To estimate parameters we optimize the c___ l___ l____. This has no a____ s____, so we solve with g____ d____ or variants. We can add a c____ p_____, in which case we optimize the n____ l____ l___.
- LR produces a l_____ decision boundary. However it is different from naïve Bayes because _____.
- In a generative-discriminative pair, the _____ approach generally converges faster, however the ____ approach generally has a better asymptote.

Today

- Generative Machine Learning

Pros and Cons of Probabilistic Classification

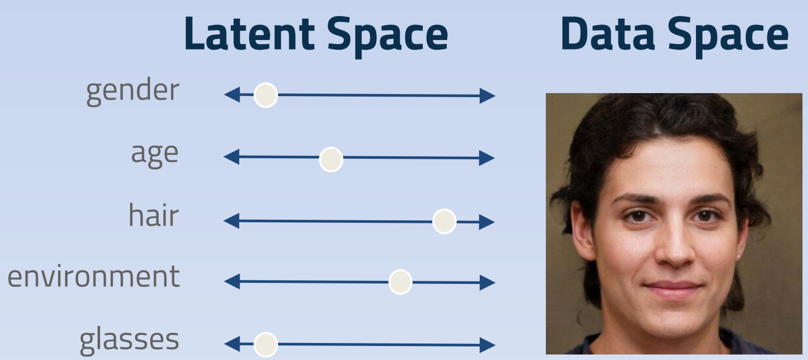
- \+ Optimal approach in decision-theoretic terms
- \+ Can incorporate prior knowledge (possibly later)
- \+ Produces confidence measures
- \+ Very well studied
- \+ Simple models are easy to implement
- \+ Can nicely capture causal influences (CSDS 442)
- – Inference and estimation are in general hard
- – Discriminative approaches can be hard to interpret

High dimensional Generative Models

- Suppose we wish to generate an image of a face
- This is hard!
- These are samples from a VERY high dimensional distribution
- And, the axes are not independent

Latent Variable Models

- To make the problem easier, we introduce "latent variables" z
  - These are hidden features which capture abstractions that constrain the space of images



Maximizing Likelihood

- Observe

$$p(X) = \int p(X \mid z)\, p(z)\, dz$$

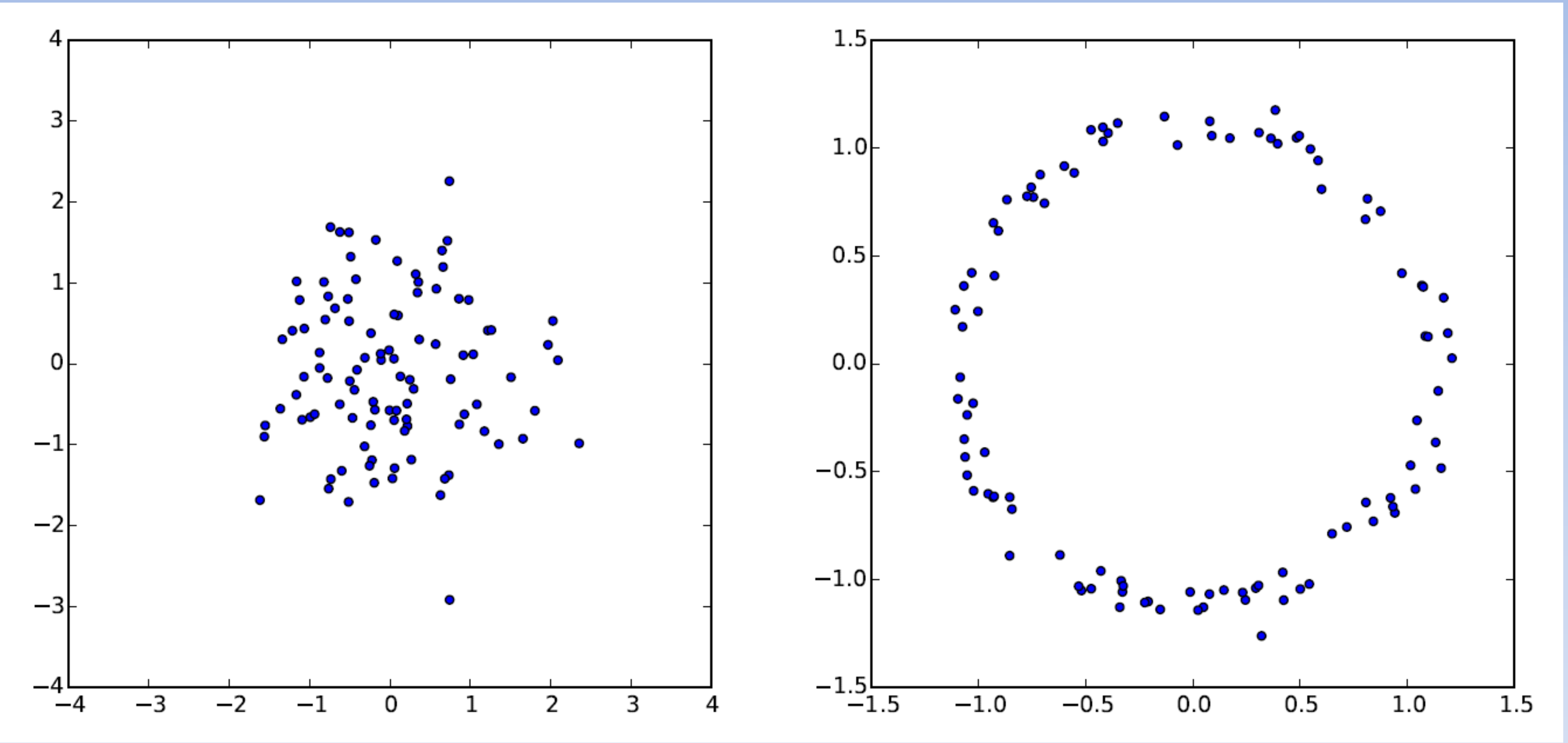
- To train a model, want to maximize LHS as before

1. What should z be?
2. How to compute the $p(X)$ above?

First clever idea

- Suppose we have no idea what z could be
- Let us just sample z from $N(\mathbf{0}, \boldsymbol{I})$ and use a nonlinear function to *transform* this input into the output we need
  - Does such a function exist?
  - In many cases yes! under "compatibility" conditions for $p(X)$ and sufficiently rich nonlinear transformations

You’re Joking, Right?



Left: samples z from 2D $\mathrm{N}(0,\mathrm{I})$. Right: $f(z)=z/10+z/\|z\|$

And so

- We'll choose a trainable family $f_\theta(z)$, typically a neural network

- With this choice, $p(X \mid z) = N(f_\theta(z), \sigma^2 I)$

Evaluating Likelihood

$$p(X) = \int p(X \mid z)\, p(z)\, dz,\ z \sim N(0, I)$$

$$\approx \frac{1}{n} \sum_i p(X \mid z_i)$$

- Unfortunately, in high dimensions, most $p(X \mid z_i)$ will be near zero, so this is going to be VERY inefficient

Second key idea

- What if we had a function $Q(z \mid X)$, that could return a distribution over those z's that are likely to produce X?

- Then maybe we could use $E_{z \sim Q}\, p(X \mid z)$ to get a good approximation to the likelihood?

Aside: Kullback-Liebler divergence

- One way to measure the "dissimilarity" between two distributions

$$D(X(z) \| Y(z)) = E_{z \sim X}\left(\log\left(X(z)\right) - \log\left(Y(z)\right)\right)$$

Relationship between $E_{z\sim Q}\, p(X \mid z)$ and $p(X)$

$$D(Q(z \mid X) \,\|\, p(z \mid X))$$

$$= E_{z\sim Q}\left(\log\left(Q(z \mid X)\right) - \log\left(p(z \mid X)\right)\right)$$

$$= E_{z\sim Q}\left(\log\left(Q(z \mid X)\right) - \log\left(p(X \mid z)\right) - \log\left(p(z)\right)\right)$$

$$+ \log\left(p(X)\right)$$

So

$$\log\left(p(X)\right) - D(Q(z \mid X) \,\|\, p(z \mid X)) =$$

$$E_{z\sim Q}\left(\log\left(p(X \mid z)\right)\right) - D\left(Q(z \mid X) \,\|\, p(z)\right)$$

Observations

- If we can find a good Q, the LHS $\approx p(X)$
- The RHS can be optimized via SGD! (w/suitable choices)
  – Not the LHS due to $p(z|X)$
- The RHS is called an "encoder-decoder" architecture
  – Q is given X and is "encoding" it into z
  – p (through the unknown f introduced before) will take z and "decode" it into X