COMP 432 Machine Learning

Generative vs Discriminative Modeling for Classification

Computer Science & Software Engineering
Concordia University, Fall 2021

UNIVERSITÉ Concordia UNIVERSITY

Generative models *in classification*

- “Approaches [to classification] that model the [joint] distribution of inputs and outputs are known as *generative models*, because by sampling from them it is possible to generate synthetic data points in the input space.” (Bishop §1.5.4)

Generative models (unsupervised)

- Gaussian Mixture Models (GMMs)
- Generative Adversarial Networks (GANs, later)
- Variational Autoencoders (VAEs, later)
- (etc, can sample synthetic data)

Bayes rule applied to classification

- Tells how to calculate feature-conditional class probabilities $p(y \mid \mathbf{x})$ in terms of class-conditional feature probabilities $p(\mathbf{x} \mid y)$:

probability of class y after observing $\mathbf{x}$ is...

...the probability of observing $\mathbf{x}$ under class y...

...weighted by overall probability of class y...

$$\boxed{p(y \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid y)p(y)}{p(\mathbf{x})}}$$

$$= \sum_{k=1}^{K} p(\mathbf{x} \mid y = k)p(y = k)$$

...relative to the overall probability of observing $\mathbf{x}$

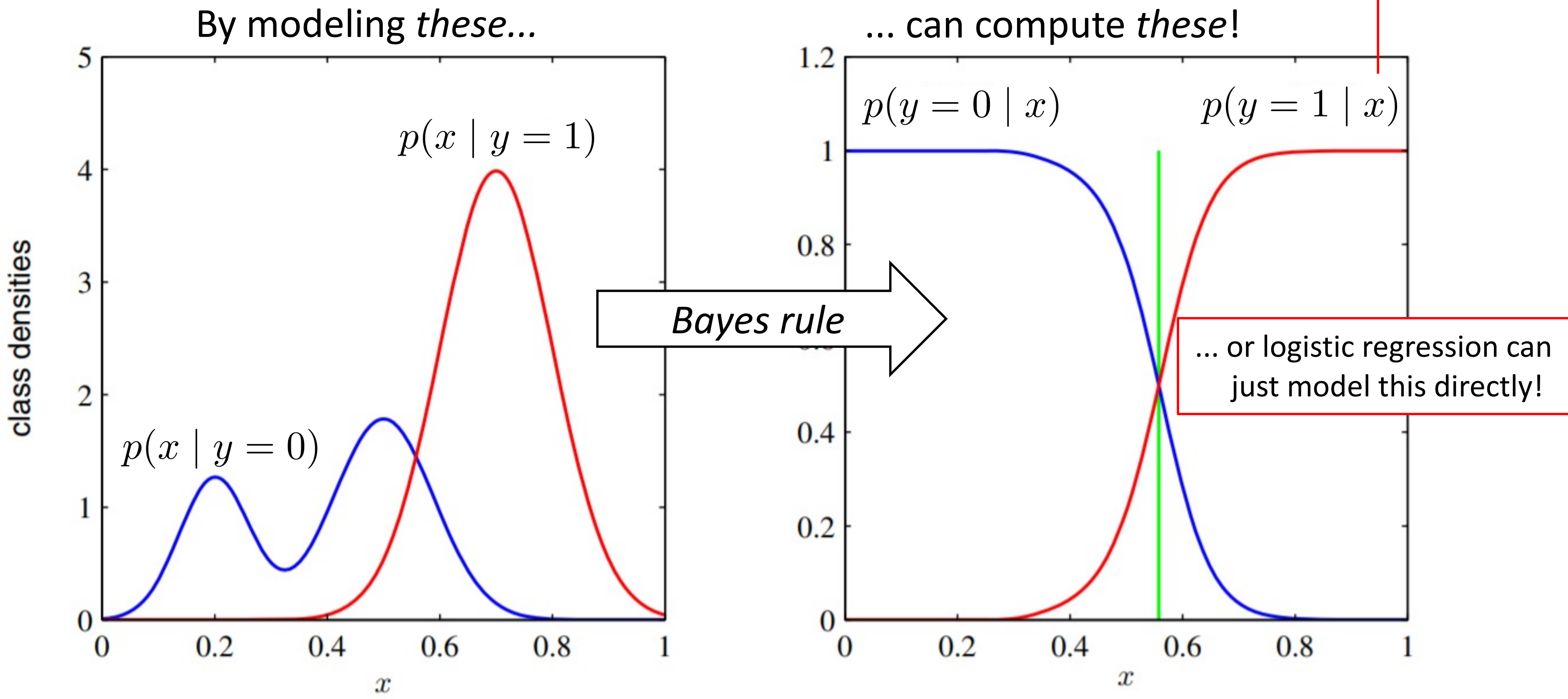
- Follows directly from product rule of probability:

$$p(\mathbf{x}, y) = p(y \mid \mathbf{x})p(\mathbf{x})$$

$$p(\mathbf{x}, y) = p(\mathbf{x} \mid y)p(y)$$

Example

$$\begin{aligned} p(y=1 \mid x) &= \frac{p(x \mid y=1)p(y=1)}{p(x)} \\ &= \frac{p(x \mid y=1)p(y=1)}{p(x \mid y=1)p(y=1) + p(x \mid y=0)p(y=0)} \end{aligned}$$



By modeling *these*...

... can compute *these*!

Bayes rule

... or logistic regression can just model this directly!

Figure 1.27 Example of the class-conditional densities for two classes having a single input variable x (left plot) together with the corresponding posterior probabilities (right plot). Note that the left-hand mode of the class-conditional density $p(x \mid y=0)$ in blue on the left plot, has no effect on the posterior probabilities. The vertical green line in the right plot shows the decision boundary in x that gives the minimum misclassification rate.

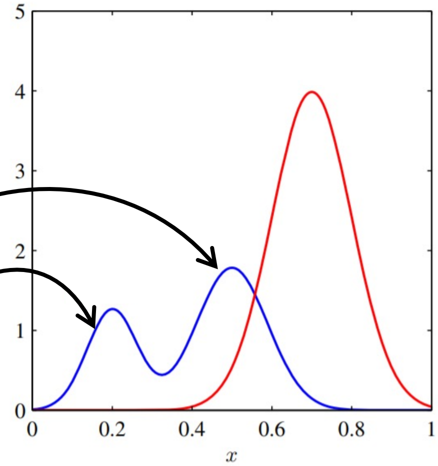
Image credit: Christopher M. Bishop

Generative vs Discriminative

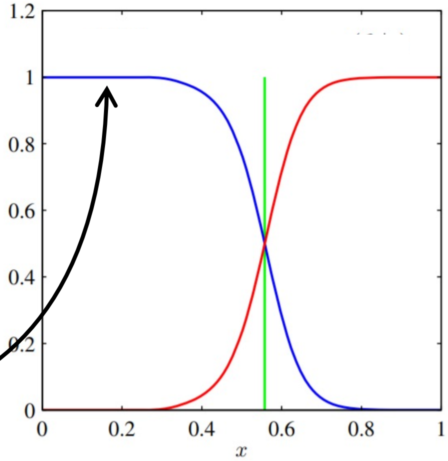
- **Generative modeling:** given data $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ build a model of $p(\mathbf{x})$, the joint distribution over all components $x_1, \ldots, x_D$ of the feature vector
    - Could be for estimating density $p(\mathbf{x})$, e.g. KDE, VAE, GMM
    - Could be for sampling $\mathbf{x} \sim p(\mathbf{x})$, e.g. KDE, GMM, VAE, GAN
    - Could be for classification, first by building a generative model of $p(\mathbf{x}, y)$ from $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ and then using Bayes rule to evaluate $p(y \mid \mathbf{x})$, e.g. Naive Bayes
- **Discriminative modeling:** given data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ build a model of $p(y \mid \mathbf{x})$ *directly*, without attempting to model $p(\mathbf{x} \mid y)$ or $p(\mathbf{x})$, e.g. logistic regression, SVM, decision tree. (But can’t ‘generate’ new data!)

Generative vs Discriminative Classifiers

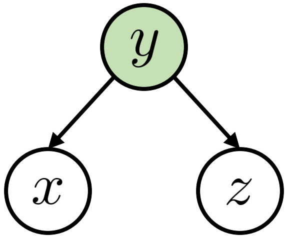
- Generative classifiers need to model the 'structure' of $p(\mathbf{x} \mid y)$, even far from the decision boundary
  - Some structure in $p(\mathbf{x} \mid y)$ may be relevant to classification, whereas some may not.
  - Tends to require more training data, unless we make simplifying assumptions.
  - But when class y is known to *cause* x_j, can exploit prior knowledge of $p(\mathbf{x} \mid y)$!



- Discriminative classifiers need only model the structure of $p(y \mid \mathbf{x})$, which can often be simpler.



Conditional independence



- A common simplifying assumption is that of *conditional independence*.
- Consider variables x, y, z . We say "x is conditionally independent of z given y" if and only if

$$\boxed{p(x \mid y, z) = p(x \mid y)}$$

observing z doesn't tell us anything new about x

- Equivalently

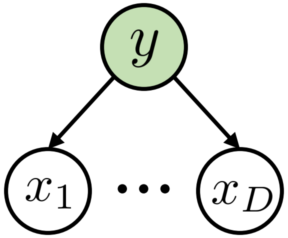
observing y lets us treat x and z as independent

$$\boxed{p(x, z \mid y) = p(x \mid y)p(z \mid y)}$$

i.e. we can factor their marginal distributions conditioned on y

$$\boxed{p(y \mid \mathbf{x}) = \frac{\boxed{p(\mathbf{x} \mid y)}\, p(y)}{p(\mathbf{x})}}$$

Naive Bayes assumption



- In generative modeling, the choice of how to model $p(\mathbf{x} \mid y)$ is the main decision. ($p(y)$ is obvious.)
  - How should we assume that the features are jointly distributed (generated) for each class $y = k$?
- **Main idea:** when modeling $p(\mathbf{x} \mid y)$, make the 'naive' assumption that each feature x_j is *conditionally independent* of the other features

$$\boxed{p(\mathbf{x} \mid y) = \prod_{j=1}^{D} p(x_j \mid y)}$$

feature x_j is generated independently of other features, given class y

Not an assumption about the *form* of $p(x_j \mid y)$ — that's separate assumption!

Naive Bayes assumption

(extra assumption for sake of concrete example; not part of NB assumption)

- Strong assumption, but learns from much less data.
- **Example:** Suppose $y \in \{1, \ldots, K\}$ and $\mathbf{x} \in \{0,1\}^D$ and we make *no assumptions* about $p(\mathbf{x} \mid y)$.
  - Requires $K(2^D - 1)$ parameters to fully specify $p(\mathbf{x} \mid y)$ over all possible classes.
- Suppose we also assume $p(\mathbf{x} \mid y) = \prod_{j=1}^{D} p(x_j \mid y)$.
  - Requires only KD parameters, one Bernoulli parameter $\pi_{jk} = p(x_j = 1 \mid y = k)$ for each combination.
- Conditional independence bought us a reduction in parameters that was exponential in dimension of $\mathbf{x}$!
  - Works surprisingly well despite loss in modeling capacity

Naive Bayes and Logistic Regression

- Assume conditional indepence (Naive Bayes)
- Assume target $y \in \{0, 1\}$ has Bernoulli distribution

$$p(y) = \text{Bernoulli}(q) = \begin{cases} q & \text{if } y = 1 \\ 1 - q & \text{if } y = 0 \end{cases}$$

- Assume features $\mathbf{x} \in \mathbb{R}^D$ have Gaussian distribution when conditioned on class, with per-feature variance

$$p(x_j \mid y = k) = \mathcal{N}(\mu_{jk}, \sigma_j^2) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(x_j - \mu_{jk})^2}{2\sigma_j^2}\right)$$

- A particular ***Gaussian Naive Bayes*** classifier, where here we assume variance is same for all classes

Naive Bayes and Logistic Regression

- Plug this in to generative modeling scheme:

$$p(y=1 \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid y=1)p(y=1)}{p(\mathbf{x} \mid y=1)p(y=1) + p(\mathbf{x} \mid y=0)p(y=0)}$$

$$= \frac{1}{1 + \frac{p(\mathbf{x}|y=0)p(y=0)}{p(\mathbf{x}|y=1)p(y=1)}} \quad = \frac{1}{1 + \exp\left(\ln \frac{p(\mathbf{x}|y=0)p(y=0)}{p(\mathbf{x}|y=1)p(y=1)}\right)}$$

$$= \frac{1}{1 + \exp\left(\sum_j \ln \frac{p(x_j|y=0)}{p(x_j|y=1)} + \ln \frac{q-1}{q}\right)}$$

$$\frac{p(x_j \mid y=0)}{p(x_j \mid y=1)} = \exp\left(-\frac{(x_j - \mu_{j0})^2 - (x_j - \mu_{j1})^2}{2\sigma_j^2}\right)$$

$$= \exp\left(-\frac{(\mu_{j1} - \mu_{j0})}{\sigma_j} x_j - \frac{\mu_{j0}^2 - \mu_{j1}^2}{2\sigma_j^2}\right)$$

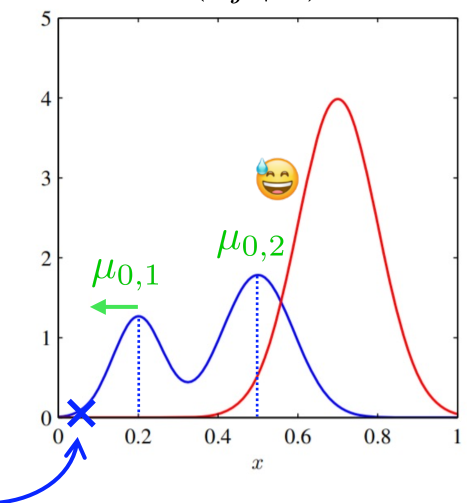
$$= \frac{1}{1 + \exp\left(-\sum_j w_j x_j - w_0\right)}$$

... reduces to form with coefficients $w_0, w_1, \ldots, w_D$, so can represent the exact same decision boundaries as LR! But LR makes no assumptions on $p(\mathbf{x} \mid y)$

Naive Bayes

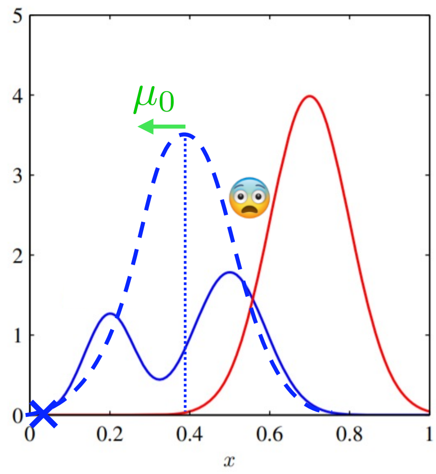
- Remember, in general Naive Bayes says nothing about form of $p(x_j \mid y = k)$, so relation to LR exists only for the special case given on prev slide.
  - Here NB still has $KD + D + 1$ params, LR has only $D + 1$
- Unlike LR, Naive Bayes can be influenced by data far from decision boundary, depending on assumptions

assume $p(x_j \mid y)$ is GMM

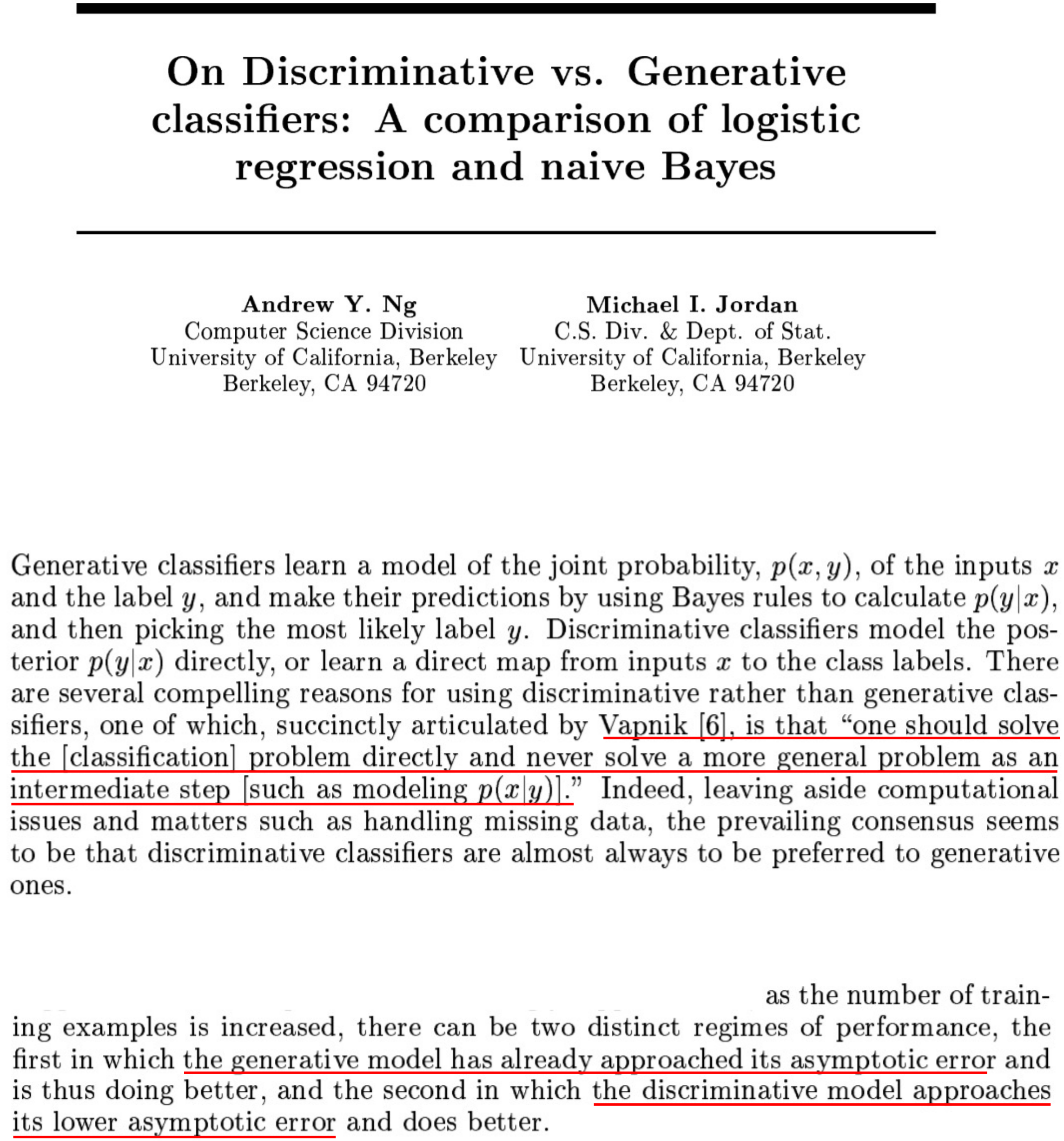


new data point x_i
with $y_i = 0$

assume $p(x_j \mid y)$ is Gaussian



may or may not be good thing!



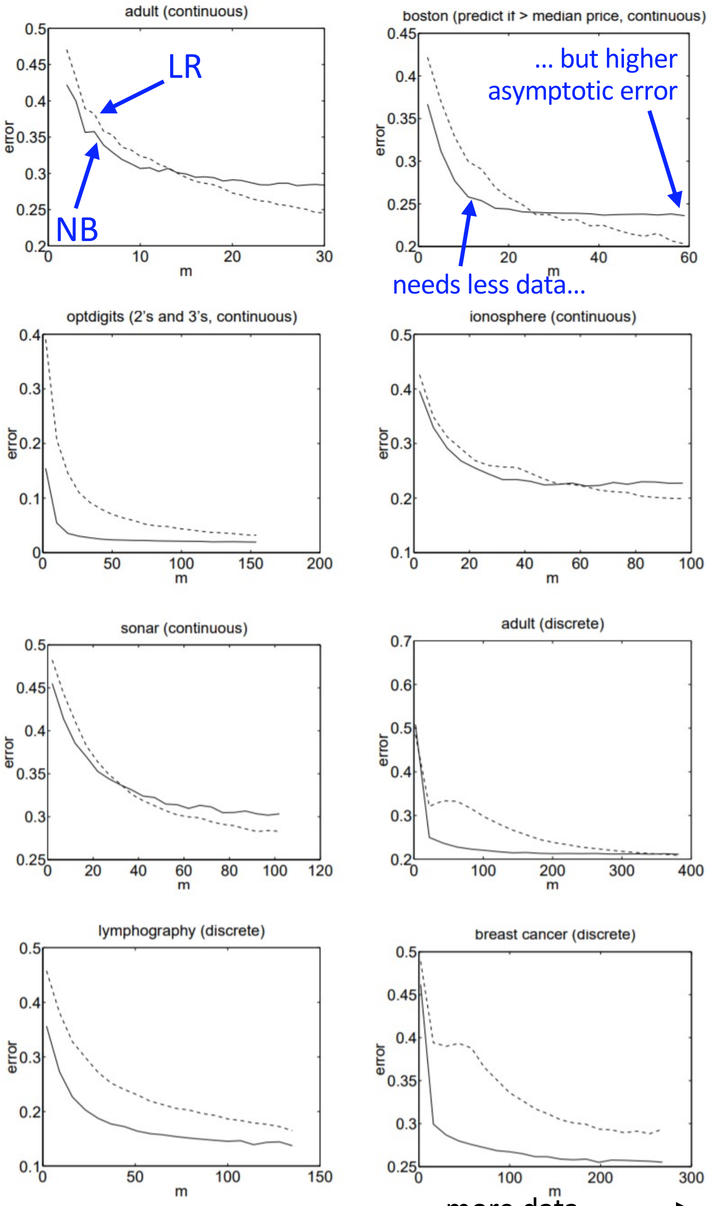
On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes

Andrew Y. Ng
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720

Michael I. Jordan
C.S. Div. & Dept. of Stat.
University of California, Berkeley
Berkeley, CA 94720

Generative classifiers learn a model of the joint probability, $p(x, y)$, of the inputs x and the label y, and make their predictions by using Bayes rules to calculate $p(y|x)$, and then picking the most likely label y. Discriminative classifiers model the posterior $p(y|x)$ directly, or learn a direct map from inputs x to the class labels. There are several compelling reasons for using discriminative rather than generative classifiers, one of which, succinctly articulated by Vapnik [6], is that "one should solve the [classification] problem directly and never solve a more general problem as an intermediate step [such as modeling $p(x|y)$]." Indeed, leaving aside computational issues and matters such as handling missing data, the prevailing consensus seems to be that discriminative classifiers are almost always to be preferred to generative ones.

as the number of training examples is increased, there can be two distinct regimes of performance, the first in which the generative model has already approached its asymptotic error and is thus doing better, and the second in which the discriminative model approaches its lower asymptotic error and does better.



LR

NB

... but higher asymptotic error

needs less data...

more data →

https://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf

***Tip:* if a feature is inherently positive-valued, log-transforming can sometimes make it “more Gaussian”!**

Naive Bayes summary

- Strong conditional independence assumption: for each class, features are generated independently
- Helps when dimensionality of raw features is high
- When assumption approximately holds, better performance with little data, but worse asymptotics
- Additional assumptions about the form of $p(x_j \mid y)$ lead to specializations (Gaussian Naive Bayes, Binomial Naive Bayes, Multinomial Naive Bayes,...)
  - Note: sklearn’s GaussianNB models per-class σ_{jk}, not just σ_j
- Can mix discrete and continuous features
- Probabilities (*predict_proba*) not usually accurate