Artificial Intelligence II (CS4442 & CS9542)

Classification: Generative Models

Boyu Wang
Department of Computer Science
University of Western Ontario

Discriminative model vs. generative model

- Recall: in logistic regression, we model directly $p(y|x)$:

$$p(y = 1|x; w) \triangleq \sigma(h_w(x)) = \frac{1}{1 + e^{-w^\top x}},$$

– This is called discriminative model, because we only care about discriminating examples of the two classes.

Discriminative model vs. generative model

- Recall: in logistic regression, we model directly $p(y|x)$:

$$p(y=1|x;w) \triangleq \sigma(h_w(x)) = \frac{1}{1+e^{-w^\top x}},$$

– This is called discriminative model, because we only care about discriminating examples of the two classes.

- Another way to model $p(y)$ and $p(x|y)$ and then use the Bayes Rule:

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

– This is called generative model, because we can actually use the model to generate data.

Bayes classifier for continuous features

- Idea: Use the training data to estimate $p(y)$ and $p(x|y)$
- $p(y)$ can be estimated by counting the number of data points of each class.
- How to estimate $p(x|y)$?
  - Need additional assumptions (for continuous inputs) – multivariate Gaussian with mean $\mu \in \mathbb{R}^n$, and covariance $\Sigma \in \mathbb{R}^{n \times n}$
  - Each class has mean μ_c and covariance Σ_c, $c \in \{0, 1\}$

Examples of multivariate Gaussian distribution

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

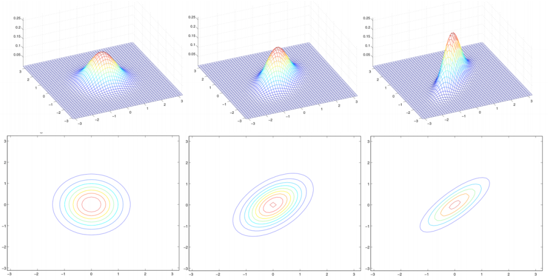


Figure: 2D Gaussian distributions with different Σ

Figure credit: Doina Precup

Gaussian discriminant analysis

- For 2 classes:

$$p(y=1)=\theta; \quad p(y=0)=1-\theta$$

$$p(x|y=1)=\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_1|^{\frac{1}{2}}}e^{-\frac{1}{2}(x-\mu_1)^\top\Sigma_1^{-1}(x-\mu_1)}$$

$$p(x|y=0)=\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_0|^{\frac{1}{2}}}e^{-\frac{1}{2}(x-\mu_0)^\top\Sigma_0^{-1}(x-\mu_0)}$$

- The parameters to estimate are: $\theta, \mu_1, \Sigma_1, \mu_0, \Sigma_0$

- For C classes:

$$p(y=c)=\theta_c; \quad \text{s.t.} \sum_{c=1}^{C}\theta_c=1$$

$$p(x|y=c)=\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_c|^{\frac{1}{2}}}e^{-\frac{1}{2}(x-\mu_c)^\top\Sigma_c^{-1}(x-\mu_c)}$$

- The parameters to estimate are: $\{\theta_c, \mu_c, \Sigma_c\}_{c=1}^{C}$

Estimate the parameters

- We can write down the likelihood function, like linear regression and logistic regression
- Compute the gradient with respect to the parameters and set them to 0.
  - The parameter θ_c is given by $\theta_c = \frac{n_c}{n}$, where n_c is the number of instances of class c.
  - The mean μ_c is given by

$$\mu_c = \frac{1}{n_c} \sum_{x_i : y_i = c} x_i$$

  - The covariance matrix Σ_c is given by

$$\Sigma_c = \frac{1}{n_c} \sum_{x_i : y_i = c} (x_i - \mu_c)(x_i - \mu_c)^\top$$

Other variants to simplify the model

- If we assume the same covariance matrix Σ for all the classes, the maximum likelihood estimation of Σ is

$$\Sigma = \frac{n_c}{n} \sum_{c=1}^{C} \Sigma_c$$

- Covariance matrix can be restricted to diagonal, or mostly diagonal with few off-diagonal elements, based on prior knowledge.
- Covariance matrix can even be identity matrix.
- The shape of the covariance is influenced both by assumptions about the domain and by the amount of data available.
- If the covariance matrices are different for the class, the model is called quadratic discriminant analysis (QDA); if the covariance matrices are are the same, the model is called linear discriminant analysis (LDA); if the covariance matrices are diagonal, the model is called naive Bayes classifier (NBC).

Classification using quadratic discriminant analysis

Recall:

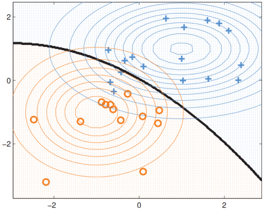
$$p(y = c) = \theta_c; \quad p(x|y = c) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_c|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_c)^\top \Sigma_c^{-1}(x-\mu_c)}$$

Using the Bayes rule, we have

$$p(y = c|x) \propto \theta_c |2\pi\Sigma_c|^{-\frac{1}{2}} e^{-\frac{1}{2}(x-\mu_c)^\top \Sigma_c^{-1}(x-\mu_c)}$$

Predict class label as the most probable label:

$$y = \arg\max_c p(y = c|x)$$



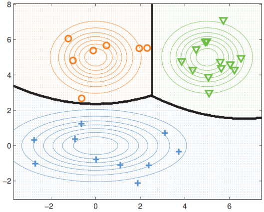


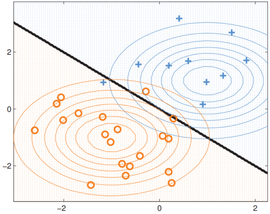
Figure credit: Kevin Murphy

Classification using linear discriminant analysis

If we assume the covariance matrices are the same for all the classes:

$$
\begin{aligned}
p(y = c|x) &\propto \theta_c e^{-\frac{1}{2}(x-\mu_c)^\top \Sigma^{-1}(x-\mu_c)} \\
&= e^{\mu_c^\top \Sigma^{-1} x - \frac{1}{2}\mu_c^\top \Sigma \mu_c + \log \theta_c} \cdot e^{-\frac{1}{2}x^\top \Sigma^{-1} x} \\
&\propto e^{\mu_c^\top \Sigma^{-1} x - \frac{1}{2}\mu_c^\top \Sigma \mu_c + \log \theta_c}
\end{aligned}
$$

Let $w_c = \Sigma^{-1}\mu_c$, and $b_c = -\frac{1}{2}\mu_c^\top \Sigma \mu_c + \log \theta_c \Rightarrow$ we get a linear model!



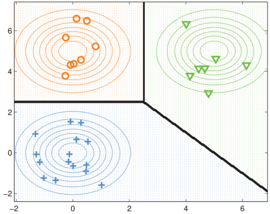


Figure credit: Kevin Murphy

Bayes classifier for discrete features

- Idea: Use the training data to estimate $p(y)$ and $p(x|y)$
- $p(y)$ can be estimated in the same way as for continuous features
- How to estimate $p(x|y)$ for discrete values?
  - Assume $x = [x_1, \dots, x_n]^\top \in \mathbb{R}^n$ has n features. Then using the chain rule, we have

    $$p(x|y) = p(x_1|y)p(x_2|y, x_1) \cdots p(x_n|y, x_1, \dots, x_{n-1})$$

    – even for binary features, it requires $\mathcal{O}(2^n)$ numbers to describe the model!
  - If we assume that the features x_j's are conditionally independent given y: $p(x_j|y) = p(x_j|y, x_k), \forall i, j$, then we have

    $$\begin{aligned} p(x|y) &= p(x_1|y)p(x_2|y, x_1) \cdots p(x_n|y, x_1, \dots, x_{n-1}) \\ &= p(x_1|y)p(x_2|y) \cdots p(x_n|y) \end{aligned}$$

    – only requires $\mathcal{O}(n)$ numbers to describe the model!

Conditional independence: an example

- A box contains two coins: a regular coin (R) and one fake two-headed coin (F). I choose a coin at random and toss it twice. Define the following two events:
  - A = First coin toss results in a head
  - B = Second coin toss results in a head

  Are A and B independent?

Conditional independence: an example

- A box contains two coins: a regular coin (R) and one fake two-headed coin (F). I choose a coin at random and toss it twice. Define the following two events:
  - A = First coin toss results in a head
  - B = Second coin toss results in a head

  Are A and B independent?
- $p(A) = p(B) = p(head) = p(head|R) \times p(R) + p(head|F) \times p(F) = \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} = \frac{3}{4}$
  $p(A, B) = p(head, head) =$
  $p(head, head|R) \times p(R) + p(head, head|F) \times p(F) = \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} = \frac{5}{8}$
  $p(A)p(B) \neq p(A, B) \Rightarrow$ A and B are dependent!

Conditional independence: an example

- A box contains two coins: a regular coin (R) and one fake two-headed coin (F). I choose a coin at random and toss it twice. Define the following two events:
  - A = First coin toss results in a head
  - B = Second coin toss results in a head

  Are A and B independent?

- $p(A) = p(B) = p(head) = p(head|R) \times p(R) + p(head|F) \times p(F) = \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} = \frac{3}{4}$
  $p(A, B) = p(head, head) =$
  $p(head, head|R) \times p(R) + p(head, head|F) \times p(F) = \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} = \frac{5}{8}$
  $p(A)p(B) \neq p(A, B) \Rightarrow$ A and B are dependent!

- Consider an additional event:
  - C = Coin R (regular) has been selected.

  Then it is easy to show that $p(A|R)p(B|R) = p(A, B|R) \Rightarrow$ A and B are conditionally independent given C!

Naive Bayes classifier for binary features

- The model parameters are $\{\theta_c = p(y = c)\}_{c=1}^{C}$ and $\{\beta_{j,c} = p(x_j = 1 | y = c)\}_{j,c=1}^{n,C}$
- Predict class label as the most probable label:

$$y = \arg\max_c \left[p(y = c) \prod_{j=1}^{n} p(x_j | y = c) \right]$$

- In practice, using the log trick to avoid the numerical issue:

$$\begin{aligned} y &= \arg\max_c \log \left[p(y = c) \prod_{j=1}^{n} p(x_j | y = c) \right] \\ &= \arg\max_c \log p(y = c) + \sum_{j=1}^{n} \log p(x_j | y = c) \end{aligned}$$

Maximum likelihood estimation for Naive Bayes

- The log-likelihood function is

$$\log L\left(\{\theta_c\}_{c=1}^{C}, \{\beta_{j,c}\}_{i,c=1}^{n,C}\right) = \sum_{i=1}^{m}\left(\log p(y_i) + \sum_{j=1}^{n}\log p(x_{i,j}|y_i)\right)$$

- Computing the gradient with respect to θ_c and setting it to 0 gives us:

$$\theta_c = \frac{n_c}{n}$$

- Computing the gradient with respect to $\beta_{j,c}$ and setting it to 0 gives us:

$$\begin{aligned}\beta_{j,c} &= p(x_j = 1|y = c) \\ &= \frac{\text{number of the instances for which } x_{i,j} = 1 \text{ and } y_i = c}{\text{number of the instances for which } y_i = c}\end{aligned}$$

Training Naive Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P(play)} = ?$

$\mathrm{P(Sky = sunny \mid play)} = ?$

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = ?$

$\mathrm{P(Sky = sunny \mid \neg play)} = ?$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Silde credit: Eric Eaton

Training Naive Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P(play)} = ?$

$\mathrm{P(Sky = sunny \mid play)} = ?$

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = ?$

$\mathrm{P(Sky = sunny \mid \neg play)} = ?$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Silde credit: Eric Eaton

Training Naive Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\text{play}) = 3/4$

$\mathrm{P}(\text{Sky} = \text{sunny} \mid \text{play}) = ?$

$\mathrm{P}(\text{Humid} = \text{high} \mid \text{play}) = ?$

$\ldots$

$\mathrm{P}(\neg\text{play}) = 1/4$

$\mathrm{P}(\text{Sky} = \text{sunny} \mid \neg\text{play}) = ?$

$\mathrm{P}(\text{Humid} = \text{high} \mid \neg\text{play}) = ?$

$\ldots$

Silde credit: Eric Eaton

Training Naive Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny						*yes*
sunny						*yes*
rainy	cold	high	strong	warm	change	*no*
sunny						*yes*

$\mathrm{P}(\mathrm{play}) = 3/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = ?$

$\ldots$

$\mathrm{P}(\neg\mathrm{play}) = 1/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = ?$

$\ldots$

Silde credit: Eric Eaton

Training Naive Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny						*yes*
sunny						*yes*
rainy	cold	high	strong	warm	change	*no*
sunny						*yes*

$\mathrm{P}(\mathrm{play}) = 3/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = 1$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = ?$

$\ldots$

$\mathrm{P}(\neg\mathrm{play}) = 1/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = ?$

$\ldots$

Silde credit: Eric Eaton

Training Naive Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	**Temp**	**Humid**	**Wind**	**Water**	**Forecast**	**Play?**
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy						*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\text{play}) = 3/4$

$\mathrm{P}(\text{Sky} = \text{sunny} \mid \text{play}) = 1$

$\mathrm{P}(\text{Humid} = \text{high} \mid \text{play}) = ?$

$\ldots$

$\mathrm{P}(\neg\text{play}) = 1/4$

$\mathrm{P}(\text{Sky} = \text{sunny} \mid \neg\text{play}) = ?$

$\mathrm{P}(\text{Humid} = \text{high} \mid \neg\text{play}) = ?$

$\ldots$

Silde credit: Eric Eaton

Training Naive Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy						*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 1$

$\mathrm{P(Humid = high \mid play)} = ?$

...

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = 0$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

...

Silde credit: Eric Eaton

Training Naive Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
		normal				*yes*
		high				*yes*
rainy	cold	high	strong	warm	change	*no*
		high				*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 1$

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = 0$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Silde credit: Eric Eaton

Training Naive Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
		normal				*yes*
		high				*yes*
rainy	cold	high	strong	warm	change	*no*
		high				*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 1$

$\mathrm{P(Humid = high \mid play)} = 2/3$

...

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = 0$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

...

Silde credit: Eric Eaton

Training Naive Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
		high				*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 1$

$\mathrm{P(Humid = high \mid play)} = 2/3$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = 0$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Silde credit: Eric Eaton

Training Naive Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
		high				*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 1$

$\mathrm{P(Humid = high \mid play)} = 2/3$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = 0$

$\mathrm{P(Humid = high \mid \neg play)} = 1$

$\ldots$

Silde credit: Eric Eaton

Laplace smoothing

- Notice that some probabilities estimated by counting might be zero!
- Instead of the maximum likelihood estimate:

$$\beta_{j,c} = \frac{\text{number of the instances for which } x_{i,j} = 1 \text{ and } y_i = c}{\text{number of the instances for which } y_i = c}$$

use:

$$\beta_{j,c} = \frac{(\text{number of the instances for which } x_{i,j} = 1 \text{ and } y_i = c) + 1}{(\text{number of the instances for which } y_i = c) + C}$$

– add 1 to each count

- If a feature appears a lot of times, this estimate is only slightly different from maximum likelihood.

Training Naive Bayes with Laplace smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny						*yes*
sunny						*yes*
rainy	cold	high	strong	warm	change	*no*
sunny						*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 4/5$

$\mathrm{P(Humid = high \mid play)} = ?$

...

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = ?$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

...

Silde credit: Eric Eaton

Training Naive Bayes with Laplace smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy						*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 4/5$

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = 1/3$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Silde credit: Eric Eaton

Training Naive Bayes with Laplace smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
		normal				*yes*
		high				*yes*
rainy	cold	high	strong	warm	change	*no*
		high				*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 4/5$

$\mathrm{P(Humid = high \mid play)} = 3/5$

...

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = 1/3$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

...

Silde credit: Eric Eaton

Training Naive Bayes with Laplace smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
		high				*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\mathrm{play}) = 3/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = 4/5$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = 3/5$

...

$\mathrm{P}(\neg\mathrm{play}) = 1/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = 1/3$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = 2/3$

...

Silde credit: Eric Eaton

Generative model summary

- Advantages:
  - Easy to train
  - Can handle streaming data well
  - Can handle both real and discrete data
- Disadvantages:
  - Requires additional assumptions (e.g., Gaussian distribution, conditional independence of features)
  - Cannot handle high-dimensional data very well