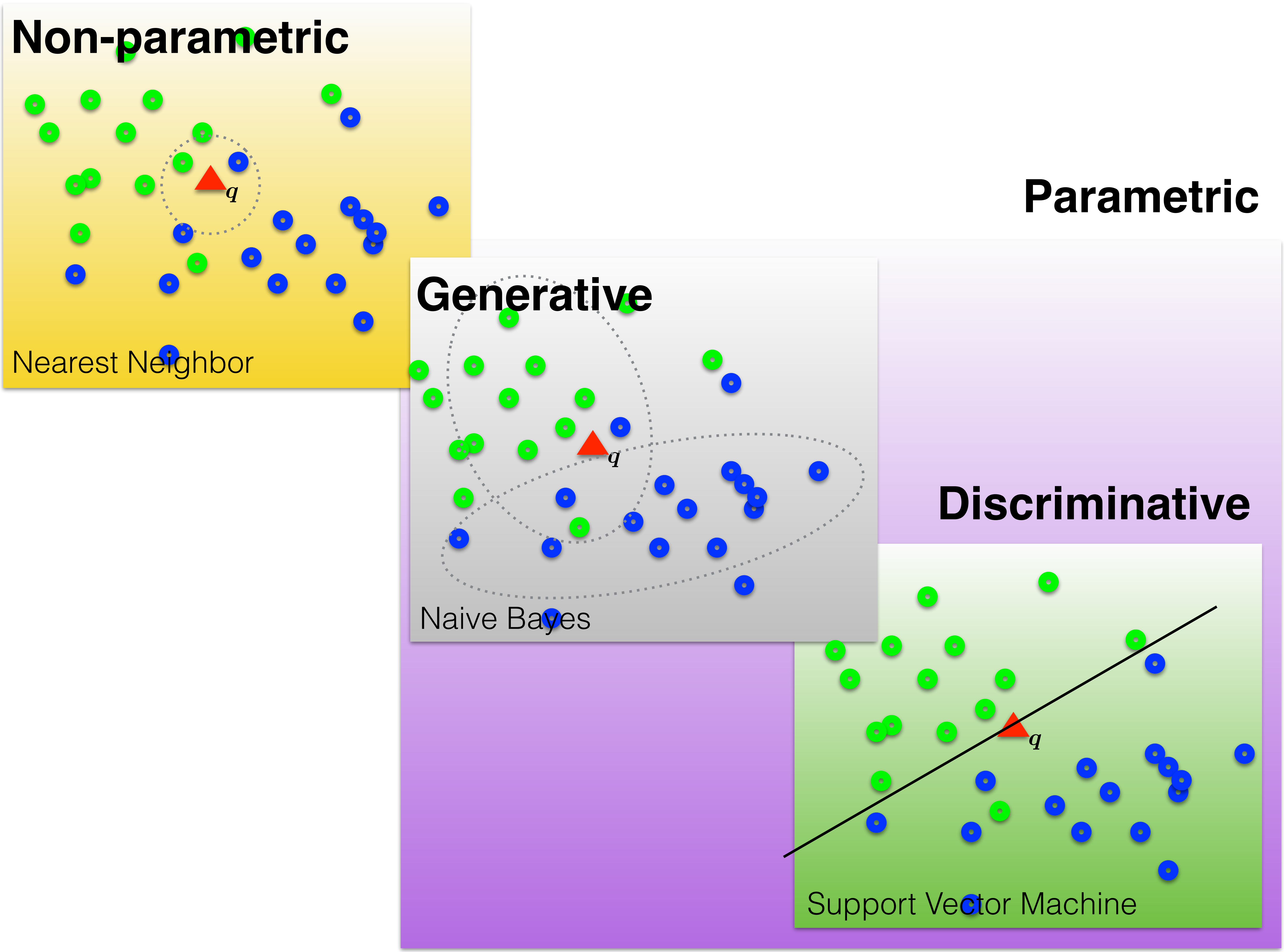
‘Classical’

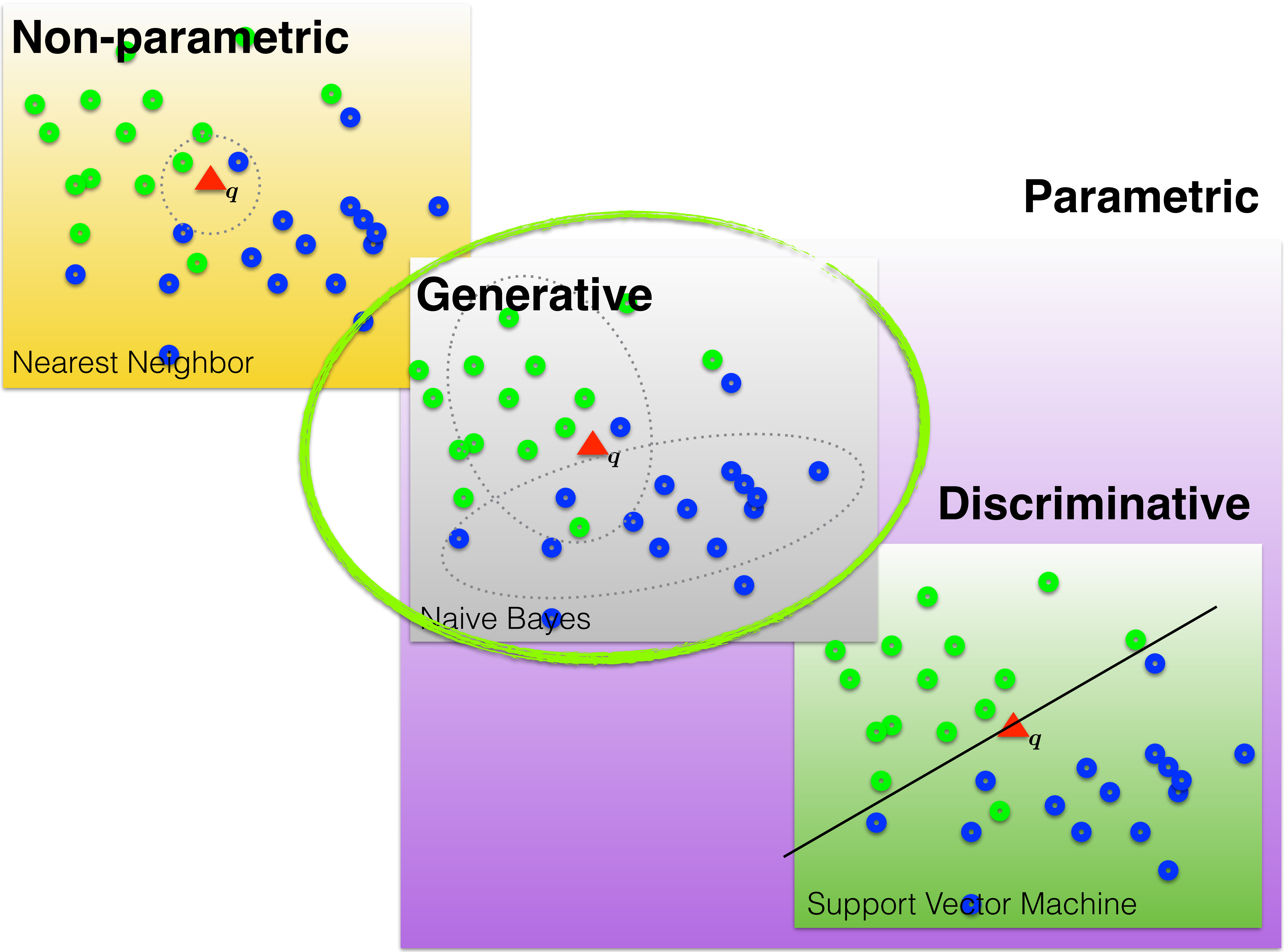
Image Classification Pipeline

Input	➔	encoder	➔	decoder	➔	output
image(s)		Extract features Build descriptor		Classifier		semantic label

Today

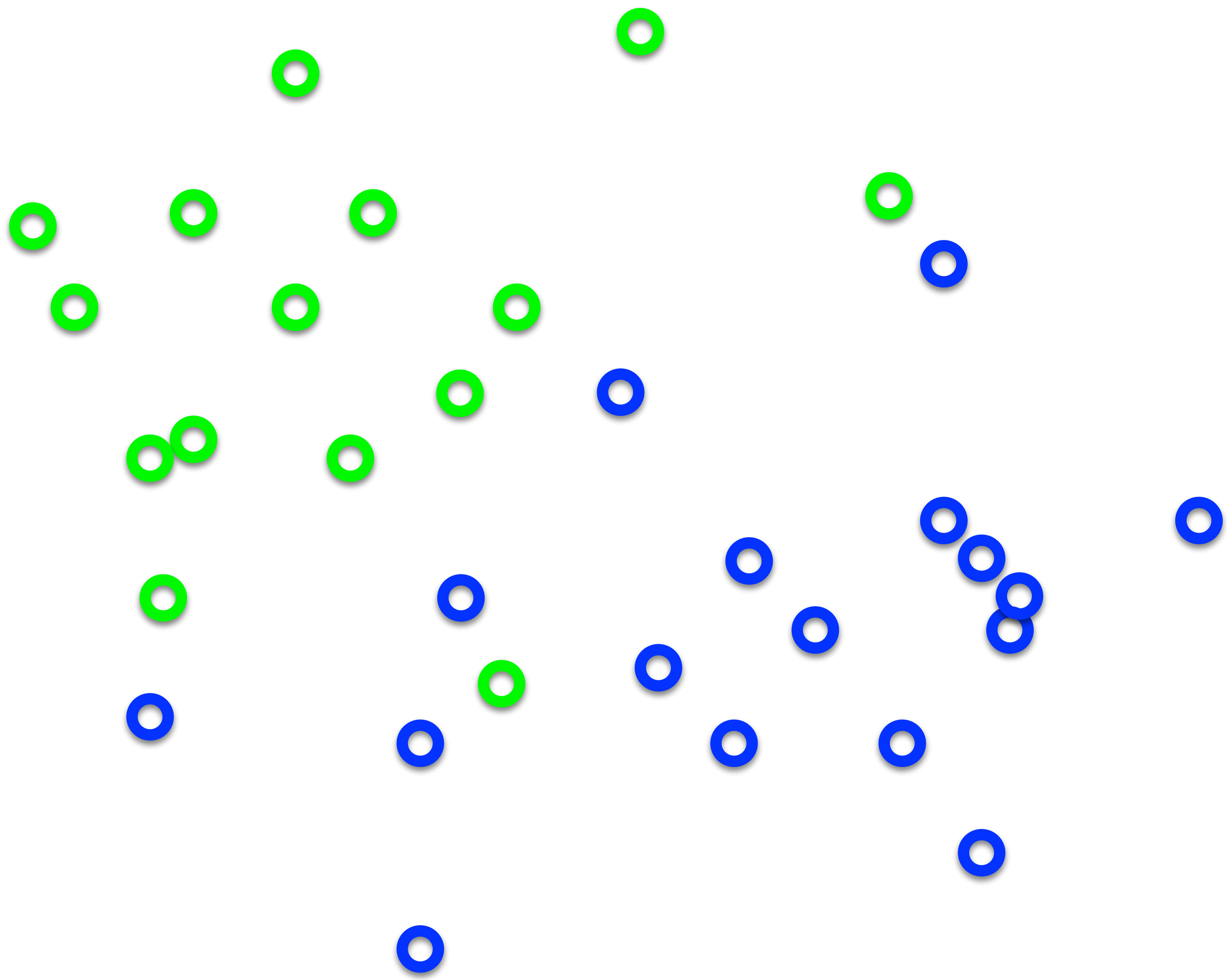


Non-parametric
q
Nearest Neighbor
Parametric
Generative
q
Naive Bayes
Discriminative
q
Support Vector Machine

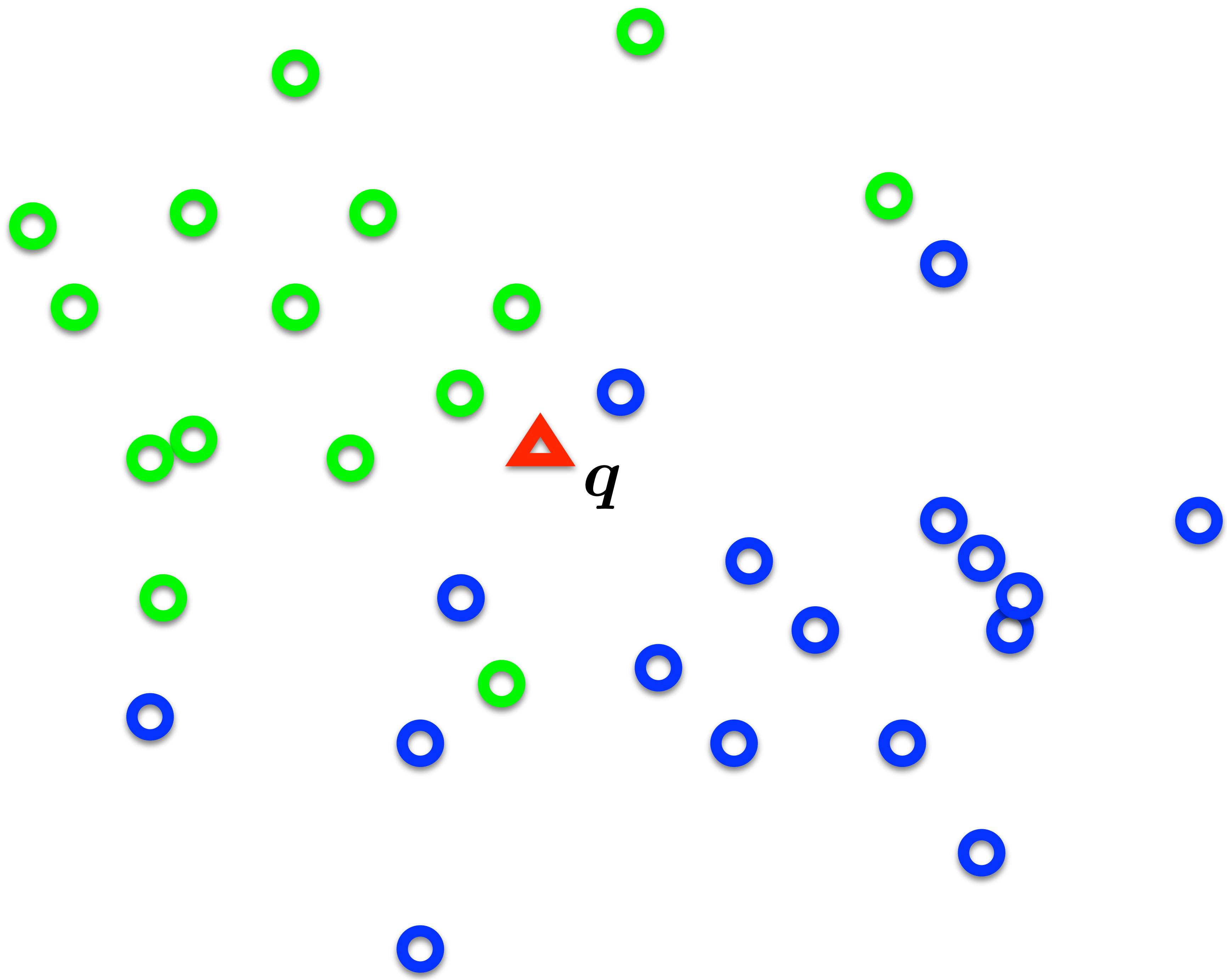


Non-parametric
q
Nearest Neighbor
Parametric
Generative
q
Naive Bayes
Discriminative
q
Support Vector Machine

Distribution of data from two classes

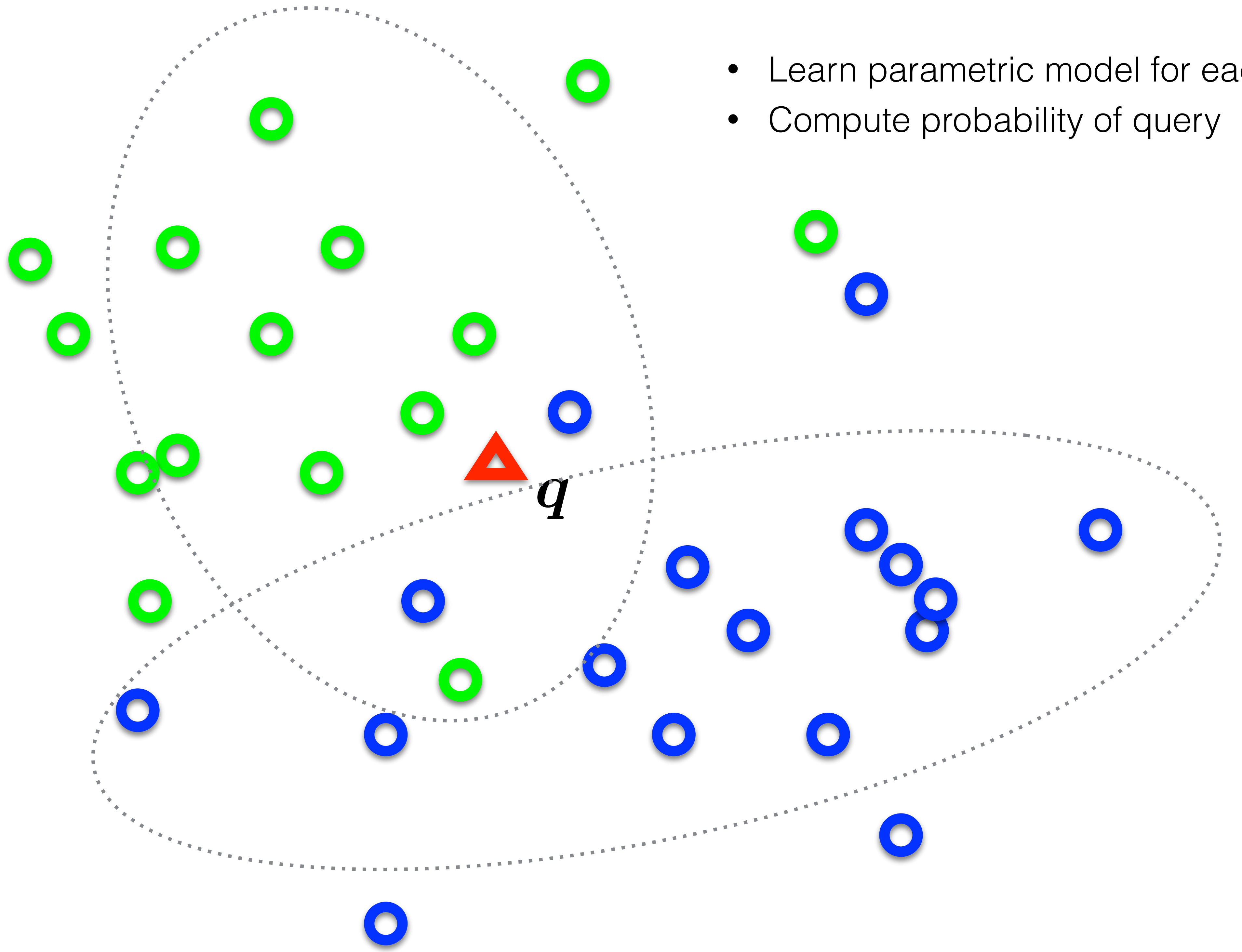


Distribution of data from two classes



Which class does q belong too?

Distribution of data from two classes



- Learn parametric model for each class
- Compute probability of query

Let's consider the probability of q…

This is called the posterior.

the probability of a class z given the observed features X

$$p(\boldsymbol{z}|\boldsymbol{X})$$

Recall: Bayes' Rule

$$\underbrace{p(x|y)}_{\text{posterior}} = \frac{\overbrace{p(y|x)}^{\text{likelihood}}\overbrace{p(x)}^{\text{prior}}}{\underbrace{p(y)}_{\text{evidence}}}$$

This is called the posterior.

the probability of a class z given the observed features X

$$p(\boldsymbol{z}|\boldsymbol{X})$$

For classification, z is a
discrete random variable
(e.g., car, person, building)

This is called the posterior.

the probability of a class z given the observed features X

$$p(\boldsymbol{z}|\boldsymbol{X})$$

For classification, z is a discrete random variable
(e.g., car, person, building)

X is a **set** of observed feature
(e.g., features from a single image)

(it's a function that returns a single probability value)

This is called the posterior.

the probability of a class z given the observed features X

$$p(\boldsymbol{z}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$$

For classification, z is a discrete random variable (e.g., car, person, building)

Each x is an observed feature (e.g., visual words)

(it's a function that returns a single probability value)

The **naive Bayes' classifier** is solving this optimization

$$\hat{z} = \arg\max_{z \in \boldsymbol{z}} p(z|\boldsymbol{X})$$

MAP (maximum a posteriori) estimate

The naive Bayes' classifier is solving this optimization

$$\hat{z} = \arg\max_{z \in \boldsymbol{z}} p(z|\boldsymbol{X})$$

MAP (maximum a posteriori) estimate

$$\hat{z} = \arg\max_{z \in \boldsymbol{z}} \frac{p(\boldsymbol{X}|z)p(z)}{p(\boldsymbol{X})}$$

Bayes' Rule

The naive Bayes' classifier is solving this optimization

$$\hat{z} = \arg\max_{z \in \boldsymbol{z}} p(z|\boldsymbol{X})$$

MAP (maximum a posteriori) estimate

$$\hat{z} = \arg\max_{z \in \boldsymbol{z}} \frac{p(\boldsymbol{X}|z)p(z)}{p(\boldsymbol{X})}$$

Bayes' Rule

$$\hat{z} = \arg\max_{z \in \boldsymbol{z}} p(\boldsymbol{X}|z)p(z)$$

Remove constants

To optimize this…we need to compute this

How should we compute the likelihood? Make a naive assumption!

A naive Bayes' classifier assumes all features are
conditionally independent

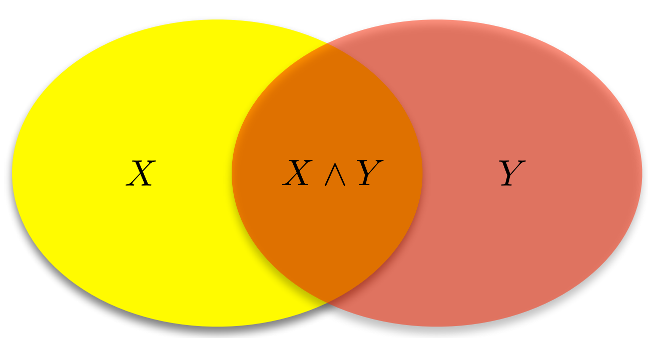
so…

$$\hat{z} = \arg\max_{z \in \boldsymbol{z}} \boxed{p(\boldsymbol{X}|z)}\, p(z)$$

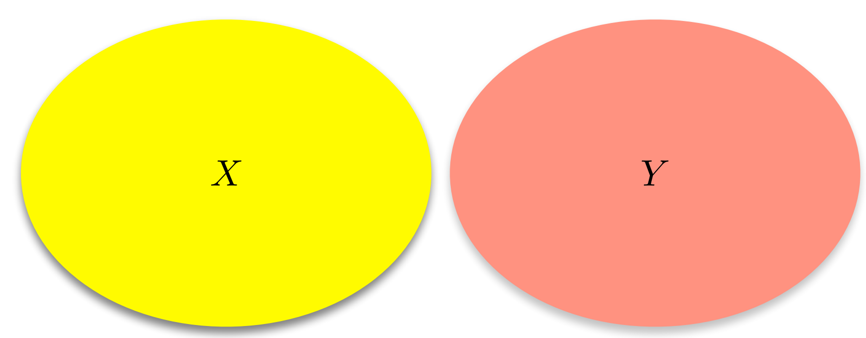
$$\begin{aligned} p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N | \boldsymbol{z}) &= p(\boldsymbol{x}_1|\boldsymbol{z})p(\boldsymbol{x}_2, \ldots, \boldsymbol{x}_N|\boldsymbol{z}) \\ &= p(\boldsymbol{x}_1|\boldsymbol{z})p(\boldsymbol{x}_2|\boldsymbol{z})p(\boldsymbol{x}_3, \ldots, \boldsymbol{x}_N|\boldsymbol{z}) \\ &= p(\boldsymbol{x}_1|\boldsymbol{z})p(\boldsymbol{x}_2|\boldsymbol{z}) \cdots p(\boldsymbol{x}_N|\boldsymbol{z}) \end{aligned}$$

Each features z only depends on class x

Recall:



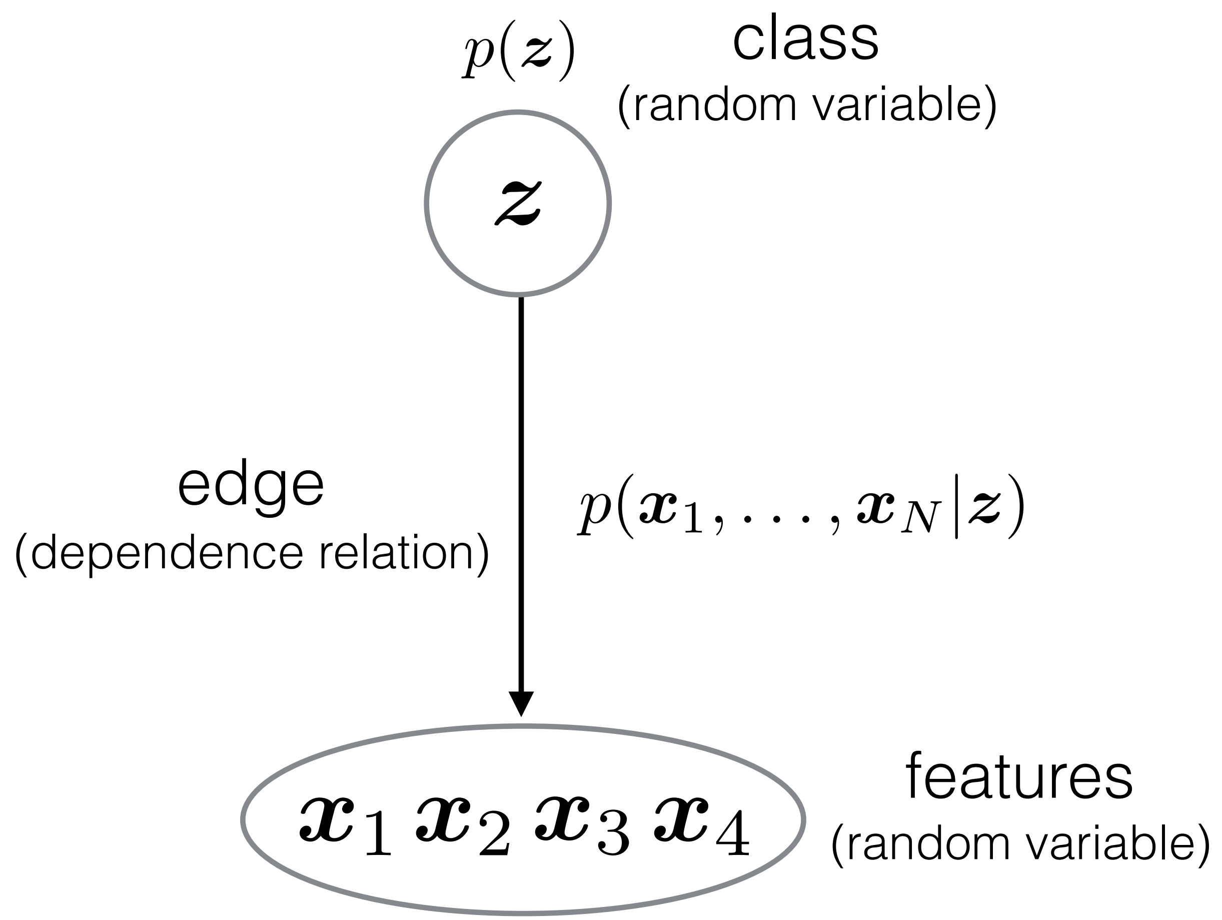
$$p(x, y) = p(x|y)p(y)$$



$$p(x, y) = p(x)p(y)$$

$$
\begin{aligned}
p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N | \boldsymbol{z}) &= p(\boldsymbol{x}_1|\boldsymbol{z}) p(\boldsymbol{x}_2, \ldots, \boldsymbol{x}_N|\boldsymbol{z}) \\
&= p(\boldsymbol{x}_1|\boldsymbol{z}) p(\boldsymbol{x}_2|\boldsymbol{z}) p(\boldsymbol{x}_3, \ldots, \boldsymbol{x}_N|\boldsymbol{z}) \\
&= p(\boldsymbol{x}_1|\boldsymbol{z}) p(\boldsymbol{x}_2|\boldsymbol{z}) \cdots p(\boldsymbol{x}_N|\boldsymbol{z})
\end{aligned}
$$

You can visualize the distribution using a graphical model:

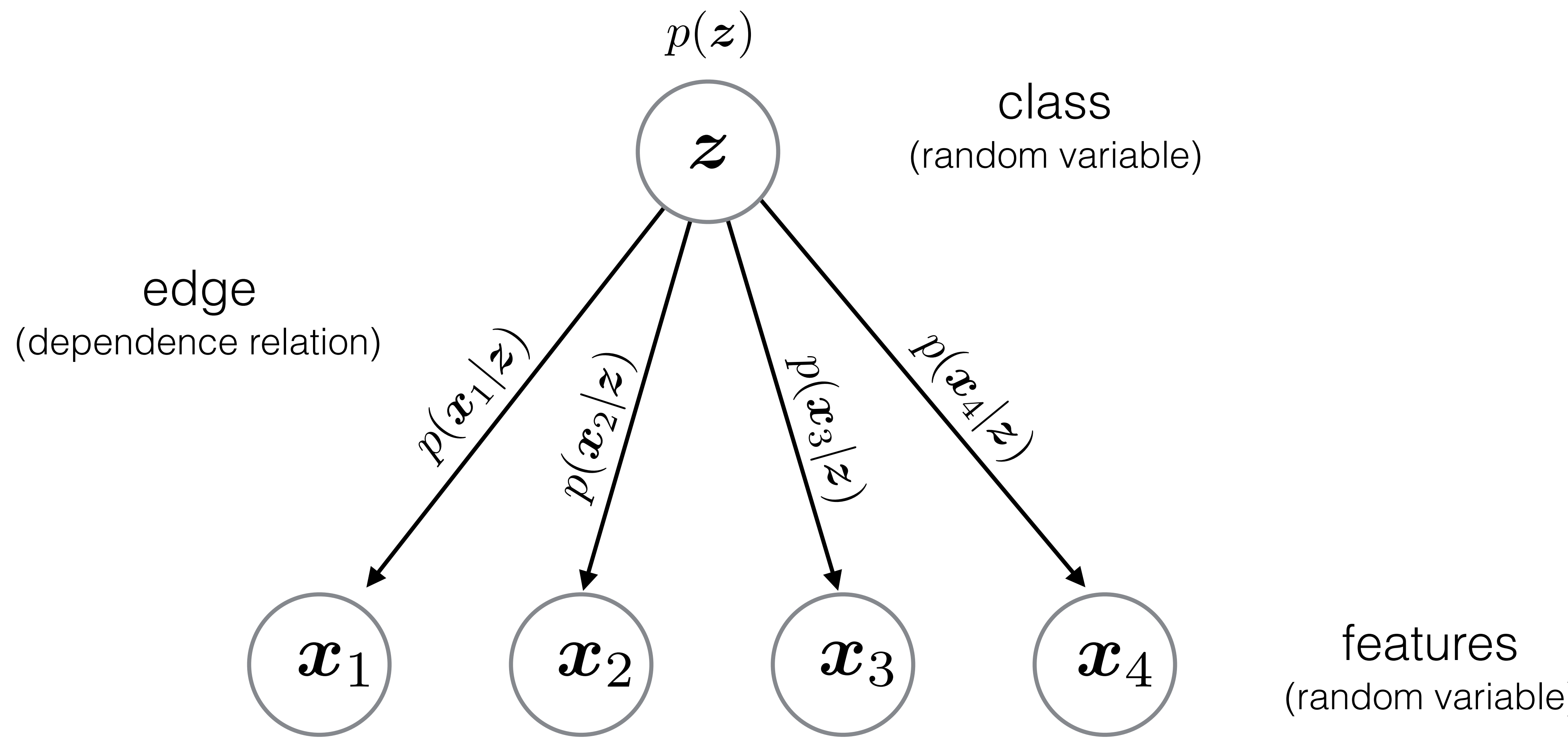


Instead of this. Do this …

Naive Bayes assumption: If you assume conditional independence of the observations…

$$
\begin{aligned}
p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N | \boldsymbol{z}) &= p(\boldsymbol{x}_1 | \boldsymbol{z}) p(\boldsymbol{x}_2, \ldots, \boldsymbol{x}_N | \boldsymbol{z}) \\
&= p(\boldsymbol{x}_1 | \boldsymbol{z}) p(\boldsymbol{x}_2 | \boldsymbol{z}) p(\boldsymbol{x}_3, \ldots, \boldsymbol{x}_N | \boldsymbol{z}) \\
&= p(\boldsymbol{x}_1 | \boldsymbol{z}) p(\boldsymbol{x}_2 | \boldsymbol{z}) \cdots p(\boldsymbol{x}_N | \boldsymbol{z})
\end{aligned}
$$

You can visualize the distribution using a graphical model:



To compute the MAP estimate

In order to solve this:

$$\hat{z} = \arg\max_{z \in \boldsymbol{z}} p(z|\boldsymbol{X})$$

Given (1) a set of known parameters

$$p(\boldsymbol{z}) \quad p(\boldsymbol{x}|\boldsymbol{z})$$

(2) observations

$$\{x_1, x_2, \ldots, x_N\}$$

In order to solve this:

$$\hat{z} = \arg\max_{z \in \boldsymbol{z}} p(z|\boldsymbol{X})$$

To compute the MAP estimate

Given (1) a set of known parameters

$$p(\boldsymbol{z}) \quad p(\boldsymbol{x}|\boldsymbol{z})$$

(2) observations

$$\{x_1, x_2, \ldots, x_N\}$$

Compute which z has the largest probability

$$\hat{z} = \arg\max_{z \in \boldsymbol{z}} p(z) \prod_n p(x_n|z)$$

To compute the MAP estimate

In order to solve this:

$$\hat{z} = \arg\max_{z \in \boldsymbol{z}} p(z|\boldsymbol{X})$$

Given (1) a set of known parameters

$$p(\boldsymbol{z}) \quad p(\boldsymbol{x}|\boldsymbol{z})$$

(2) observations

$$\{x_1, x_2, \ldots, x_N\}$$

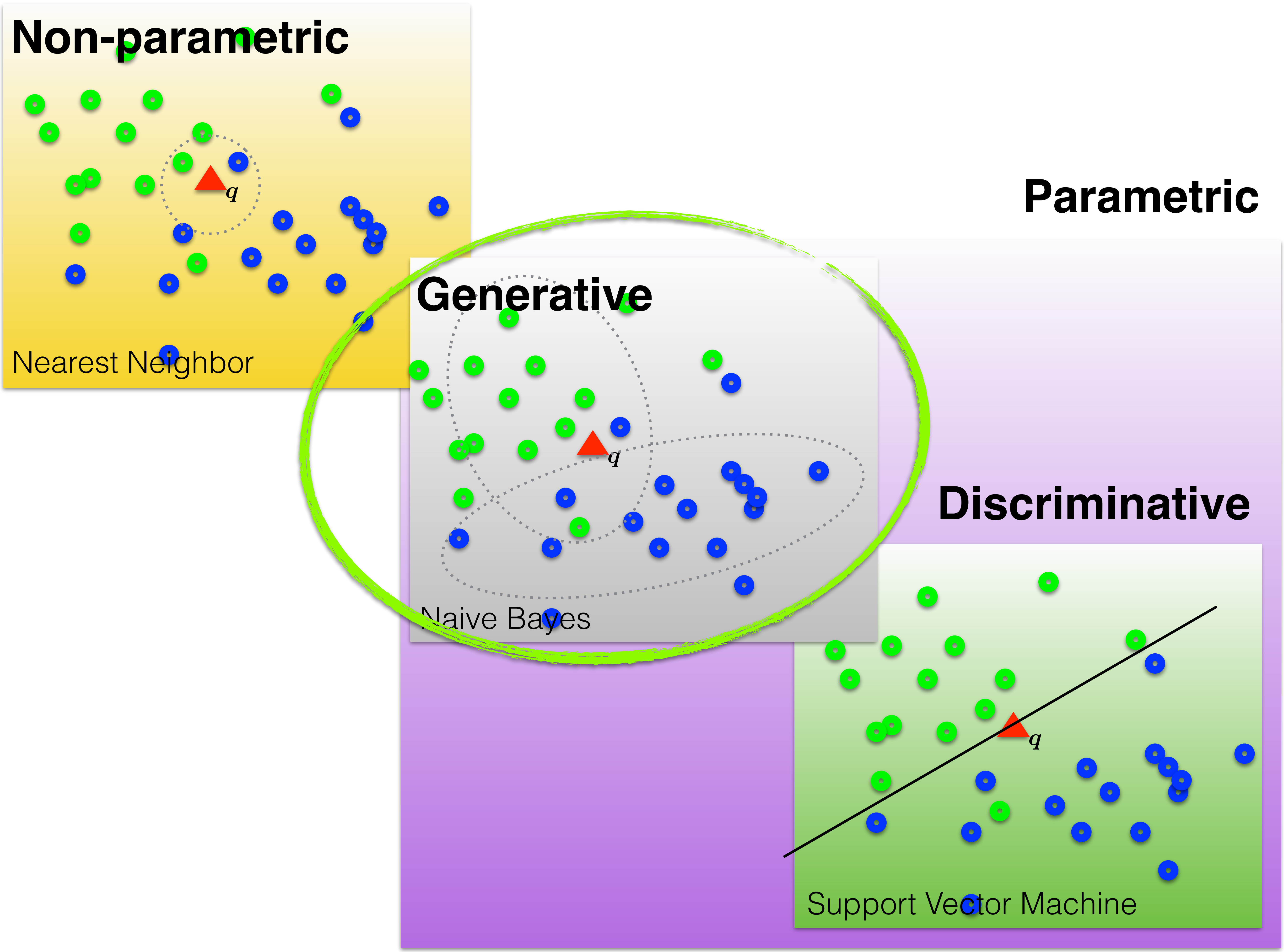
Compute which z has the largest probability

$$\hat{z} = \arg\max_{z \in \boldsymbol{z}} p(z) \prod_n p(x_n|z)$$

This process is called ‘**inference**’

(you can learn more about ‘learning’ in machine learning)

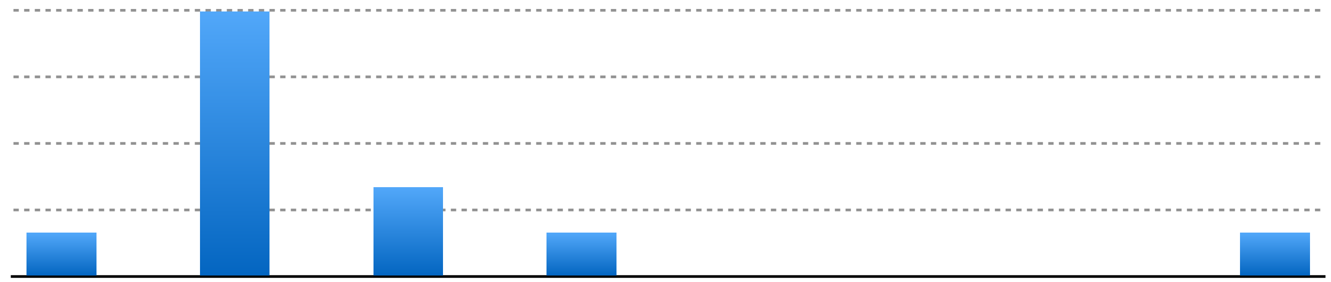
(see concrete example at the end of slides)



Non-parametric
q
Nearest Neighbor
Parametric
Generative
q
Naive Bayes
Discriminative
q
Support Vector Machine

Concrete Example:
Decomposing the likelihood
for the bag of words model


The Newsp
DARPA Selects Carnegie Me



count	1	6	2	1	0	0	0	1
word	Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor
p(x|z)	0.09	0.55	0.18	0.09	0.0	0.0	0.0	0.09

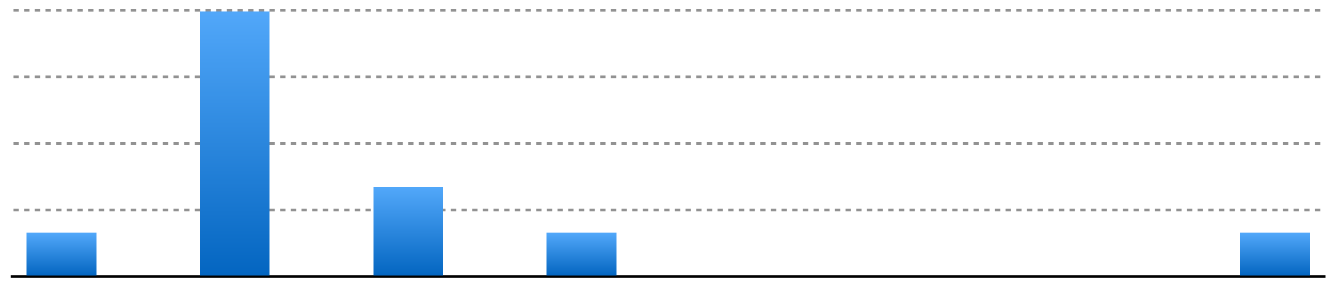
z= ‘grand challenge’

What is the likelihood of this document?

$$p(X|z)$$

The Newsp

DARPA Selects Carnegie Me

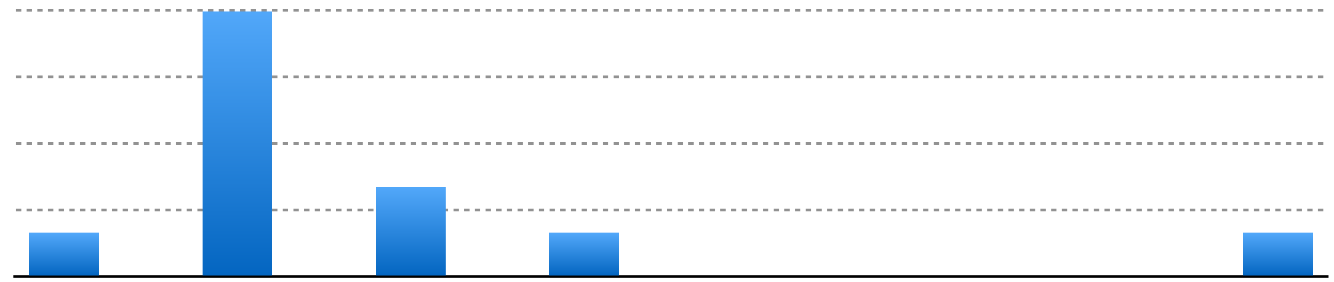


count	1	6	2	1	0	0	0	1
word	Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor
p(x|z)	0.09	0.55	0.18	0.09	0.0	0.0	0.0	0.09

$$p(X|z) = \prod_{v} p(x_v|z)^{c(w_v)}$$

$$= (0.09)^1 (0.55)^6 \cdots (0.09)^1$$


The Newspa
DARPA Selects Carnegie Me



count	1	6	2	1	0	0	0	1
word	Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor
p(x|z)	0.09	0.55	0.18	0.09	0.0	0.0	0.0	0.09

$$p(X|z) = \prod_v p(x_v|z)^{c(w_v)}$$

$$= (0.09)^1 (0.55)^6 \cdots (0.09)^1$$

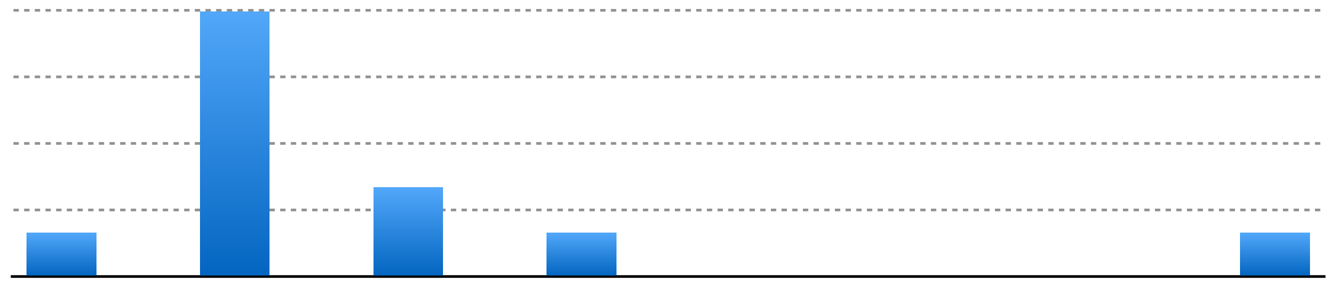
Numbers get really small so use log probabilities

$$\log p(X|z = \text{‘grandchallenge’}) = -2.42 - 3.68 - 3.43 - 2.42 - 0.07 - 0.07 - 0.07 - 2.42 = -14.58$$

* typically add pseudo-counts (0.001)
** this is an example for computing the likelihood, need to multiply times **prior** to get posterior

The Newsp

DARPA Selects Carnegie Me



count	1	6	2	1	0	0	0	1
word	Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor
p(x|z)	0.09	0.55	0.18	0.09	0.0	0.0	0.0	0.09

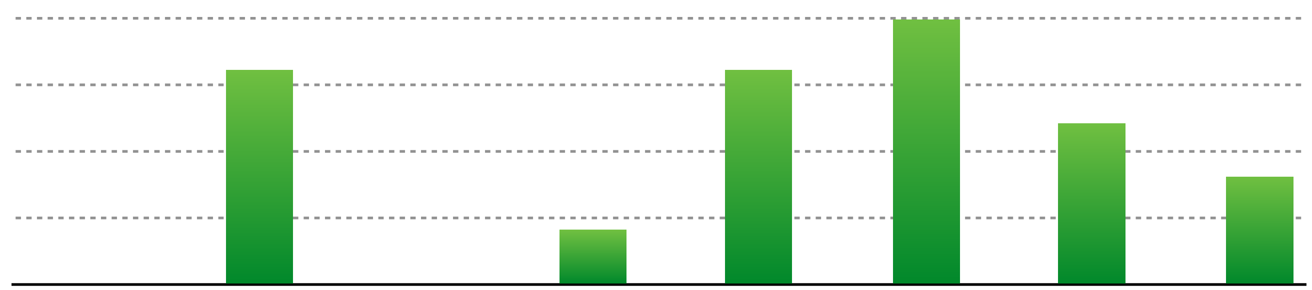
log p(X|z=grand challenge) = **- 14.58**
log p(X|z=bio inspired) = - 37.48

z= ‘grand challenge’

Tartan Tim

Bio-Inspired Robotic Device

http://www.fodey.com/generators/newspaper/snippet.asp



count	0	4	0	1	4	5	3	2
word	Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor
p(x|z)	0.0	0.21	0.0	0.05	0.21	0.26	0.16	0.11

log p(X|z=grand challenge) = - 94.06
log p(X|z=bio inspired) = **- 32.41**

z= ‘bio-inspired’

* typically add pseudo-counts (0.001)
** this is an example for computing the likelihood, need to multiply times prior to get posterior