



Bayesian Methods for Supervised, Semi-Supervised, and Unsupervised Learning

Week 6

February 26, 2024

Spring 24 | CUSP-GX 7033 – Machine Learning for Cities | Dr. Anton Rozhkov

Today's Outline

- Comparison of supervised, unsupervised, and semi-supervised learning
- Applications of clustering for modeling group structure
- Naïve Bayes (NB) classification
- Naive Bayes vs. logistic regression
- Expectation-maximization (EM) for clustering and semi-supervised NB classification
- Lab: NB and EM in Python

From Lecture 1:

ML problem paradigms represent a **functional** grouping of methods by what we're trying to accomplish. A related grouping is based on what the data looks like and, in particular, whether we have **labeled** or **unlabeled** data.

Supervised Learning

Data/input	Labels/output
------------	---------------

x_1	y_1
x_2	y_2
...	...
x_N	y_N

Learn dependence:

$$y = f(x)$$

Discrete y = classification
Continuous y = regression

Semi-supervised learning:

Only some data points are labeled; the goal is still typically prediction.

Unsupervised learning:

No labels, just input data x_i . Various goals including clustering, modeling, anomaly detection, etc.

From Lecture 1:

ML problem paradigms represent a **functional** grouping of methods by what we're trying to accomplish. A related grouping is based on what the data looks like and, in particular, whether we have **labeled** or **unlabeled** data.

Supervised Learning

In today's lecture, we will learn about a class of methods based on Bayesian probability that covers the entire range from **supervised** to **semi-supervised** to **unsupervised** learning, including both **classification** and **clustering**.

- Classification only (not regression).
- Labeled and/or unlabeled data.
- Can obtain class/cluster probabilities.

Learn dependence:

$$y = f(x)$$

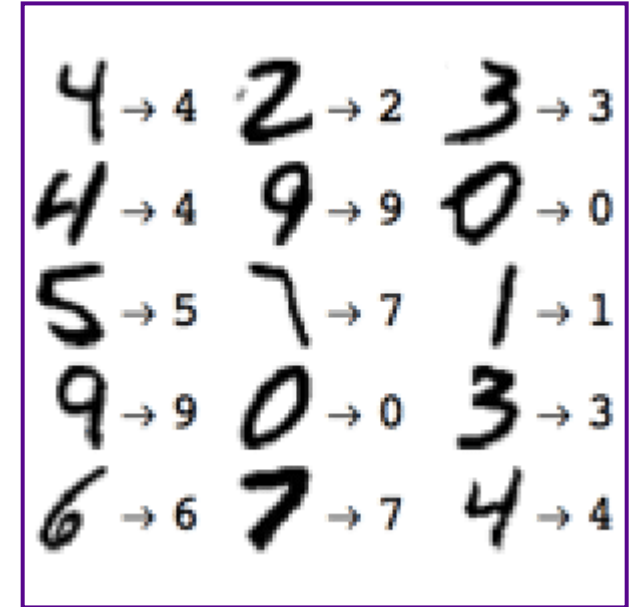
Discrete y = classification
Continuous y = regression

Unsupervised learning:

No labels, just input data x_i . Various goals including clustering, modeling, anomaly detection, etc.

Model-Based Classification

- Let's start with the **supervised** case, where we have labeled training data.
- We focus on classification and ask, "How likely is this data point to be in each class?"
- This question is important, e.g. for risk prediction, but hard to answer accurately using the methods that we've seen so far.



Solution: learn a **probabilistic model** for each class, then compute the class posterior probabilities using Bayes' Theorem.

Using Bayes' Theorem

- The posterior probability that data record x_i belongs to class C_j :

$$\Pr(C_j | x_i) = \frac{\Pr(x_i | C_j) \Pr(C_j)}{\sum_{C_k} \Pr(x_i | C_k) \Pr(C_k)}$$

Likelihood of record x_i 's attribute values if it belongs to class C_j

Prior probability of class C_j

Normalize probabilities (must sum to 1)

- Simple example: {has stripes, seen on land}

<u>Class</u>	<u>Prior</u>	<u>Likelihood</u>	<u>Unnormalized posterior</u>
Horse	$\Pr(\text{Horse}) = 0.4$	$\Pr(\text{stripes, land} \text{Horse}) = 0.050$	$0.4 \times 0.050 = 0.020$
Zebra	$\Pr(\text{Zebra}) = 0.1$	$\Pr(\text{stripes, land} \text{Zebra}) = 0.950$	$0.1 \times 0.950 = 0.095$
Fish	$\Pr(\text{Fish}) = 0.5$	$\Pr(\text{stripes, land} \text{Fish}) = 0.002$	$0.5 \times 0.002 = 0.001$
			<u>0.116</u>



Using Bayes' Theorem

- The posterior probability that data record x_i belongs to class C_j :

$$\Pr(C_j | x_i) = \frac{\Pr(x_i | C_j) \Pr(C_j)}{\sum_{C_k} \Pr(x_i | C_k) \Pr(C_k)}$$

Likelihood of record x_i 's attribute values if it belongs to class C_j

Prior probability of class C_j

Normalize probabilities (must sum to 1)

- Simple example: {has stripes, seen on land}

<u>Class</u>	<u>Prior</u>	<u>Likelihood</u>	<u>Normalized posterior</u>
Horse	$\Pr(\text{Horse}) = 0.4$	$\Pr(\text{stripes, land} \text{Horse}) = 0.050$	$0.020 / 0.116 = 0.172$
Zebra	$\Pr(\text{Zebra}) = 0.1$	$\Pr(\text{stripes, land} \text{Zebra}) = 0.950$	$0.095 / 0.116 = 0.819$
Fish	$\Pr(\text{Fish}) = 0.5$	$\Pr(\text{stripes, land} \text{Fish}) = 0.002$	$0.001 / 0.116 = 0.009$



Using Bayes' Theorem

- The posterior probability that data record x_i belongs to class C_j :

$$\Pr(C_j | x_i) = \frac{\Pr(x_i | C_j) \Pr(C_j)}{\sum_{C_k} \Pr(x_i | C_k) \Pr(C_k)}$$

Likelihood of record x_i 's attribute values if it belongs to class C_j

Prior probability of class C_j

Normalize probabilities (must sum to 1)

So how can we obtain the priors and likelihoods?

1. Prior knowledge (ask a domain expert).
2. **Learn** the priors and likelihoods from a representative "training" dataset.

200 horses, 50 zebras, 250 fish

$$\Pr(\text{horse}) = 200 / 500 = 0.4$$

$$\Pr(\text{zebra}) = 50 / 500 = 0.1$$

$$\Pr(\text{fish}) = 250 / 500 = 0.5$$

Using Bayes' Theorem

- The posterior probability that data record x_i belongs to class C_j :

$$\Pr(C_j | x_i) = \frac{\Pr(x_i | C_j) \Pr(C_j)}{\sum_{C_k} \Pr(x_i | C_k) \Pr(C_k)}$$

Likelihood of record x_i 's attribute values if it belongs to class C_j

Prior probability of class C_j

Normalize probabilities (must sum to 1)

So how can we obtain the priors and likelihoods?

- Prior knowledge (ask a domain expert).
- Learn** the priors and likelihoods from a representative “training” dataset.

Dataset of N records

$$\Pr(C_k) = \frac{\#\{\text{class} = C_k\}}{N}$$

(“Maximum likelihood estimation”)

Using Bayes' Theorem

- The posterior probability that data record x_i belongs to class C_j :

$$\Pr(C_j | x_i) = \frac{\Pr(x_i | C_j) \Pr(C_j)}{\sum_{C_k} \Pr(x_i | C_k) \Pr(C_k)}$$

Likelihood of record x_i 's attribute values if it belongs to class C_j

Prior probability of class C_j

Normalize probabilities (must sum to 1)

So how can we obtain the priors and likelihoods?

To learn the likelihoods $\Pr(x_i | C_k)$, we need a model of how the data is generated, for each class C_k .

Model-Based Classification

Dataset representation: records x_i , attributes A_j , values v_{ij}

Gender	BMI	Diabetes?	Heart attack risk
Male	26	Yes	???

To decide whether the patient's risk is high or low, we must first compute the likelihood of seeing his attribute values for both high-risk and low-risk groups:

$\Pr(\text{Gender} = \text{Male}, \text{BMI} = 26, \text{Diabetes} = \text{Yes} \mid \text{Risk} = \text{High})$

$\Pr(\text{Gender} = \text{Male}, \text{BMI} = 26, \text{Diabetes} = \text{Yes} \mid \text{Risk} = \text{Low})$

Naïve Bayes assumption: All attribute values are conditionally independent given the class.

$$\Pr(x_i \mid C_k) = \prod_{j=1..J} \Pr(A_j = v_{ij} \mid C = C_k)$$

Model-Based Classification

Dataset representation: records x_i , attributes A_j , values v_{ij}

Gender	BMI	Diabetes?	Heart attack risk
Male	26	Yes	???

To decide whether the patient's risk is high or low, we must first compute the likelihood of seeing his attribute values for both high-risk and low-risk groups:

$\Pr(\text{Gender} = \text{Male}, \text{BMI} = 26, \text{Diabetes} = \text{Yes} \mid \text{Risk} = \text{High})$

$\Pr(\text{Gender} = \text{Male}, \text{BMI} = 26, \text{Diabetes} = \text{Yes} \mid \text{Risk} = \text{Low})$

$\Pr(\text{Gender}, \text{BMI}, \text{Diabetes} \mid \text{Risk}) = \Pr(\text{Gender} \mid \text{Risk}) \Pr(\text{BMI} \mid \text{Risk}) \Pr(\text{Diabetes} \mid \text{Risk})$

<u>Class</u>	$\Pr(A_j = v_{ij} \mid C = C_k)$			$\Pr(x_i \mid C = C_k)$
	<u>Gender = Male</u>	<u>BMI = 26</u>	<u>Diabetes = Yes</u>	<u>Total likelihood</u>
Risk = Low	0.44	0.01	0.03	1.32×10^{-4}
Risk = High	0.53	0.02	0.45	4.77×10^{-3}

↑ $\Pr(\text{Gender} = \text{Male} \mid \text{Risk} = \text{High}) = 0.53$

Model-Based Classification

Dataset representation: records x_i , attributes A_j , values v_{ij}

Gender	BMI	Diabetes?	Heart attack risk
Male	26	Yes	???

To decide whether the patient's risk is high or low, we must first compute the likelihood of seeing his attribute values for both high-risk and low-risk groups:

$\Pr(\text{Gender} = \text{Male}, \text{BMI} = 26, \text{Diabetes} = \text{Yes} \mid \text{Risk} = \text{High})$

$\Pr(\text{Gender} = \text{Male}, \text{BMI} = 26, \text{Diabetes} = \text{Yes} \mid \text{Risk} = \text{Low})$

$\Pr(\text{Gender}, \text{BMI}, \text{Diabetes} \mid \text{Risk}) = \Pr(\text{Gender} \mid \text{Risk}) \Pr(\text{BMI} \mid \text{Risk}) \Pr(\text{Diabetes} \mid \text{Risk})$

Now we can use Bayes' Theorem:

<u>Class</u>	<u>Total likelihood</u>	<u>Prior</u>	<u>Unnormalized posterior</u>	<u>Posterior</u>
Risk = Low	1.32×10^{-4}	0.9	1.19×10^{-4}	$1.19 / 5.96 = 0.2$
Risk = High	4.77×10^{-3}	0.1	4.77×10^{-4}	$4.77 / 5.96 = 0.8$
			5.96×10^{-4}	

Model-Based Classification

Dataset representation: records x_i , attributes A_j , values v_{ij}

Gender	BMI	Diabetes?	Heart attack risk
Male	26	Yes	???

To decide whether the patient's risk is high or low, we must first compute the likelihood of seeing his attribute values for both high-risk and low-risk groups:

$\Pr(\text{Gender} = \text{Male}, \text{BMI} = 26, \text{Diabetes} = \text{Yes} \mid \text{Risk} = \text{High})$

$\Pr(\text{Gender} = \text{Male}, \text{BMI} = 26, \text{Diabetes} = \text{Yes} \mid \text{Risk} = \text{Low})$

$\Pr(\text{Gender}, \text{BMI}, \text{Diabetes} \mid \text{Risk}) = \Pr(\text{Gender} \mid \text{Risk}) \Pr(\text{BMI} \mid \text{Risk}) \Pr(\text{Diabetes} \mid \text{Risk})$

Question 1: How to estimate the conditional probability of a discrete attribute?

$$\Pr(\text{Diabetes} = \text{Yes} \mid \text{Risk} = \text{High}) = \frac{\#(\text{Diabetes} = \text{Yes AND Risk} = \text{High})}{\#\{\text{Risk} = \text{High}\}}$$

For example, if there were 500 training records with Risk = High, and 265 of these also had Diabetes = Yes, our probability estimate would be $265 / 500 = 0.53$.

Model-Based Classification

Dataset representation: records x_i , attributes A_j , values v_{ij}

Gender	BMI	Diabetes?	Heart attack risk
Male	26	Yes	???

To decide whether the patient's risk is high or low, we must first compute the likelihood of seeing his attribute values for both high-risk and low-risk groups:

$\Pr(\text{Gender} = \text{Male}, \text{BMI} = 26, \text{Diabetes} = \text{Yes} \mid \text{Risk} = \text{High})$

$\Pr(\text{Gender} = \text{Male}, \text{BMI} = 26, \text{Diabetes} = \text{Yes} \mid \text{Risk} = \text{Low})$

$\Pr(\text{Gender}, \text{BMI}, \text{Diabetes} \mid \text{Risk}) = \Pr(\text{Gender} \mid \text{Risk}) \Pr(\text{BMI} \mid \text{Risk}) \Pr(\text{Diabetes} \mid \text{Risk})$

Question 2: How to estimate the conditional probability* of a real-valued attribute?


$$f(\text{BMI} = 26 \mid \text{Risk} = \text{High}) = ???$$

Solution: learn a Gaussian distribution from all of the training records with Risk = High, and use this distribution to estimate the probability.

← This is called Gaussian Naïve Bayes classification.

*Technically, we are estimating the probability density $f(x)$, not the probability of drawing x .

Naïve Bayes Classifiers

[Install](#) [User Guide](#) [API](#) [Examples](#) [Community](#) [More ▾](#)

[Prev](#) [Up](#) [Next](#)

scikit-learn 1.4.1
[Other versions](#)

Please [cite us](#) if you use the software.

1.9. Naive Bayes

- [1.9.1. Gaussian Naive Bayes](#)
- [1.9.2. Multinomial Naive Bayes](#)
- [1.9.3. Complement Naive Bayes](#)
- [1.9.4. Bernoulli Naive Bayes](#)
- [1.9.5. Categorical Naive Bayes](#)
- [1.9.6. Out-of-core naive Bayes model fitting](#)

1.9. Naive Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Bayes' theorem states the following relationship, given class variable y and dependent feature vector x_1 through x_n :

$$P(y | x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n | y)}{P(x_1, \dots, x_n)}$$

Using the naive conditional independence assumption that

$$P(x_i | y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | y),$$

for all i , this relationship is simplified to

$$P(y | x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i | y)}{P(x_1, \dots, x_n)}$$

Since $P(x_1, \dots, x_n)$ is constant given the input, we can use the following classification rule:

$$P(y | x_1, \dots, x_n) \propto P(y) \prod_{i=1}^n P(x_i | y)$$
$$\Downarrow$$
$$\hat{y} = \arg \max_y P(y) \prod_{i=1}^n P(x_i | y),$$

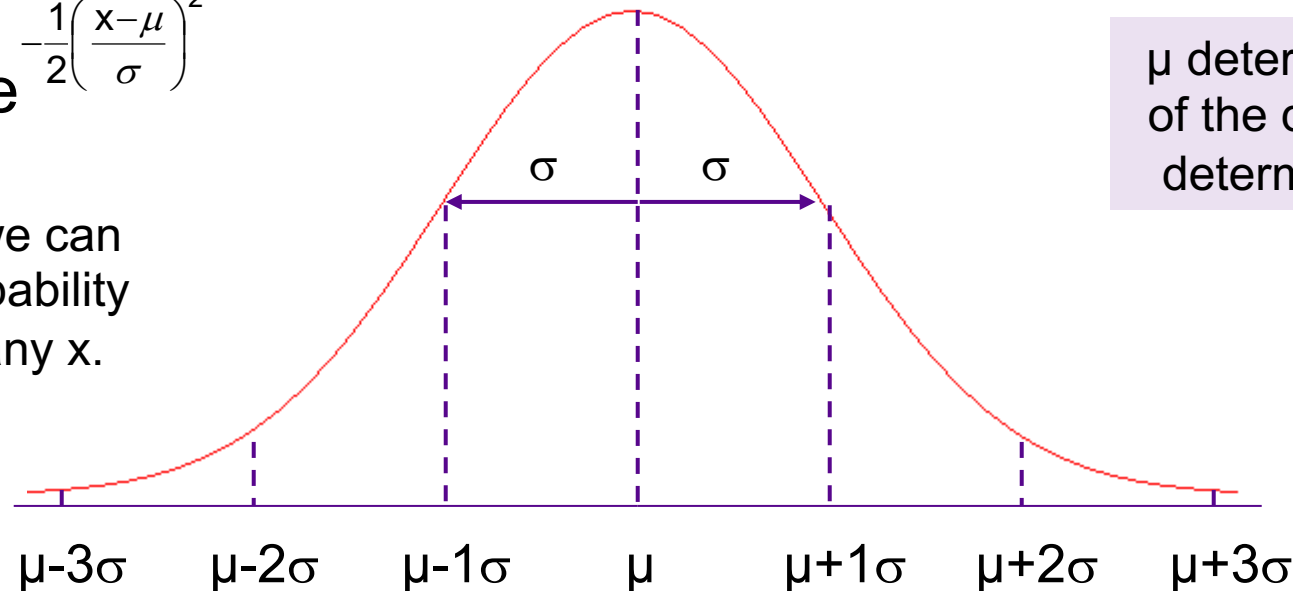
and we can use Maximum A Posteriori (MAP) estimation to estimate $P(y)$ and $P(x_i | y)$; the former is then the relative frequency of class y in the training set.

Review of the Gaussian Distribution

- Also called “normal distribution” or “bell curve”.
- A good approximation for many real-world distributions.
 - Central Limit Theorem: The sum (or mean) of sufficiently many i.i.d. samples from any distribution is approximately Gaussian.
- A symmetric, unimodal distribution $N(\mu, \sigma)$, determined by its mean μ and standard deviation σ :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Given μ and σ , we can compute the probability density $f(x)$ for any x .



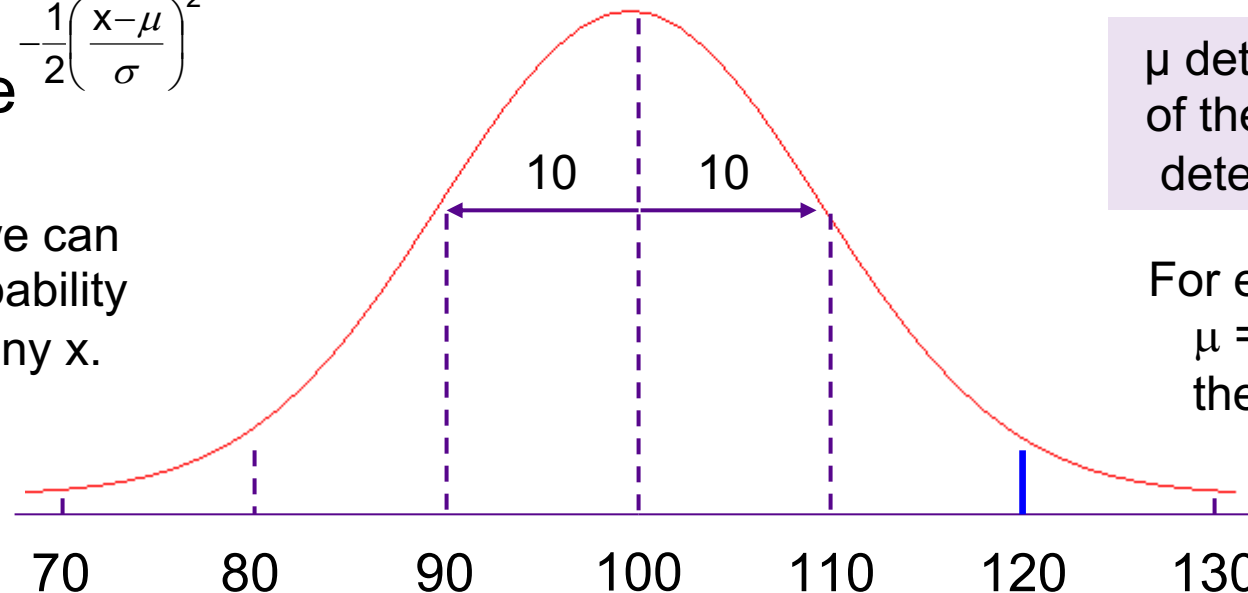
μ determines the center of the distribution and σ determines its spread.

Review of the Gaussian Distribution

- Also called “normal distribution” or “bell curve”.
- A good approximation for many real-world distributions.
 - Central Limit Theorem: The sum (or mean) of sufficiently many i.i.d. samples from any distribution is approximately Gaussian.
- A symmetric, unimodal distribution $N(\mu, \sigma)$, determined by its mean μ and standard deviation σ :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Given μ and σ , we can compute the probability density $f(x)$ for any x .



μ determines the center of the distribution and σ determines its spread.

For example, if we have $\mu = 100$ and $\sigma = 10$, then $f(120) = .0054$.

Learning Gaussian Distributions

Gaussians are simple to learn from training data:

μ = sample mean of x_i

σ = sample standard deviation of x_i

$$\mu = \frac{\sum x_i}{N} \quad \sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N-1}}$$

For example, consider the distribution of body mass index (BMI) for patients with low and high heart attack risks respectively:

Low risk (4500 patients): mean = 20.0, standard deviation = 2.6

High risk (500 patients): mean = 32.0, standard deviation = 3.1

Probability density of BMI = 26 for Low-risk group:

$$f(x = 26 \mid N(20, 2.6)) = .01$$

Probability density of BMI = 26 for High-risk group:

$$f(x = 26 \mid N(32, 3.1)) = .02$$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$



Summary of Naïve Bayes

Step 1: Learn a **class-conditional model** for each attribute for each class, using the training data.

Discrete-valued attribute: $\Pr(A_j = v_{ij} \mid C = C_k)$

Real-valued attribute: μ and σ of A_j for $C = C_k$

	<u>Gender</u>	<u>BMI</u>	<u>Diabetes</u>
Risk = Low	$\Pr(\text{Male}) = 0.44$ $\Pr(\text{Female}) = 0.56$	$m = 20$ $s = 2.6$	$\Pr(\text{Diabetes}) = 0.03$ $\Pr(\text{No diabetes}) = 0.97$
Risk = High	$\Pr(\text{Male}) = 0.53$ $\Pr(\text{Female}) = 0.47$	$m = 32$ $s = 3.1$	$\Pr(\text{Diabetes}) = 0.45$ $\Pr(\text{No diabetes}) = 0.55$

Summary of Naïve Bayes

Step 1: Learn a **class-conditional model** for each attribute for each class, using the training data.

Discrete-valued attribute: $\Pr(A_j = v_{ij} \mid C = C_k)$

Real-valued attribute: μ and σ of A_j for $C = C_k$

	<u>Gender</u>	<u>BMI</u>	<u>Diabetes</u>
Risk = Low	$\Pr(\text{Male}) = 0.44$ $\Pr(\text{Female}) = 0.56$	$m = 20$ $s = 2.6$	$\Pr(\text{Diabetes}) = 0.03$ $\Pr(\text{No diabetes}) = 0.97$
Risk = High	$\Pr(\text{Male}) = 0.53$ $\Pr(\text{Female}) = 0.47$	$m = 32$ $s = 3.1$	$\Pr(\text{Diabetes}) = 0.45$ $\Pr(\text{No diabetes}) = 0.55$

Step 2: To classify a given test record, first compute the likelihood of each of its attributes given each class, and multiply to obtain the total likelihood.

	<u>Gender = Male</u>	<u>BMI = 26</u>	<u>Diabetes = Yes</u>	<u>Total likelihood</u>
Risk = Low	0.44	0.01	0.03	1.32×10^{-4}
Risk = High	0.53	0.02	0.45	4.77×10^{-3}

Summary of Naïve Bayes

Step 3: Then obtain the prior probability of each class from the training data and combine this with the likelihood using Bayes' Theorem.

	<u>Total likelihood</u>	<u>Prior</u>	<u>Unnormalized posterior</u>	<u>Posterior</u>
Risk = Low	1.32×10^{-4}	0.9	1.19×10^{-4}	0.2
Risk = High	4.77×10^{-3}	0.1	4.77×10^{-4}	0.8

When to Use Naïve Bayes

- We have a dataset with some class we want to predict.
 - We can only do classification, not regression
 - Datasets with lots of attributes and/or lots of records are okay.
 - Datasets with discrete or real values, or both, are okay.
 - We can predict the posterior probability of each class and use these probabilities to make decisions.

When to Use Naïve Bayes

- We have a dataset with some class we want to predict.
 - We can only do classification, not regression
 - Datasets with lots of attributes and/or lots of records are okay.
 - Datasets with discrete or real values, or both, are okay.
 - We can predict the posterior probability of each class and use these probabilities to make decisions.
- We want to create an interpretable model of each class and understand how each attribute affects our predictions.
 - For a discrete attribute, which values are common for each class?
 - For a real attribute, what are μ and σ for each class?
 - For a given test record, compare the class-conditional likelihoods for each attribute.

When to Use Naïve Bayes

- We have a dataset with some class we want to predict.
 - We can only do classification, not regression
 - Datasets with lots of attributes and/or lots of records are okay.
 - Datasets with discrete or real values, or both, are okay.
 - We can predict the posterior probability of each class and use these probabilities to make decisions.
- We want to create an interpretable model of each class and understand how each attribute affects our predictions.
 - For a discrete attribute, which values are common for each class?
 - For a real attribute, what are μ and σ for each class?
 - For a given test record, compare the class-conditional likelihoods for each attribute.
- Performance is better when the given model makes sense.
 - Naïve Bayes assumes all attributes are conditionally independent given the class. This assumption does well most of the time...
 - We can also use a more complicated model for each class, such as a Bayesian network (to be discussed in a future lectures).

Naïve Bayes vs Logistic Regression

- For binary classification, NB and LR can both be written in the form:

$$\log\left(\frac{\Pr(y = 0 | x)}{\Pr(y = 1 | x)}\right) = w_0 + \sum_{i=1..M} w_i x_i$$

This sum is over the attributes (columns), for a single data record.

- For Gaussian Naïve Bayes, we have:

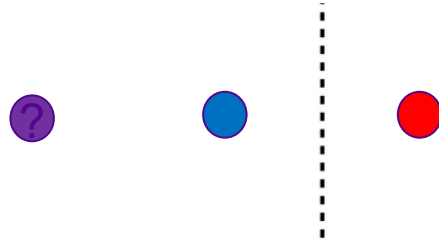
$$w_i = \frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2} \quad w_0 = \ln \frac{1 - \pi}{\pi} + \sum_i \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma_i^2}$$

See Mitchell (2020) for the derivation.

- For LR, the parameters w_i and w_0 are learned jointly to maximize the likelihood of the training data $\Pr(y | x_1..x_M)$. (“discriminative learning”)
- For Bayesian classification, the parameters w_i and w_0 are learned to maximize $\Pr(x_1..x_M | y)$ and $\Pr(y)$. (“generative learning”)
- For NB, we assume conditional independence and each w_i is learned separately by maximum likelihood.
- It can be shown that, if the NB assumption is true, NB and LR converge to the same w_i values but NB converges much more quickly, leading to improved performance for small amounts of training data.
- If the NB assumption is false, with enough training data LR will eventually outperform NB. See Mitchell (2020) for additional details.

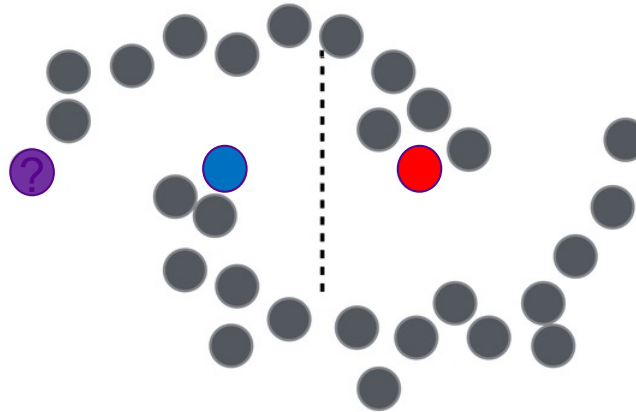
From Supervised to Semi-Supervised...

- If you have a small amount of labeled data and a large amount of unlabeled data (a very common situation in the real world...), the unlabeled data can often be used to improve prediction performance.



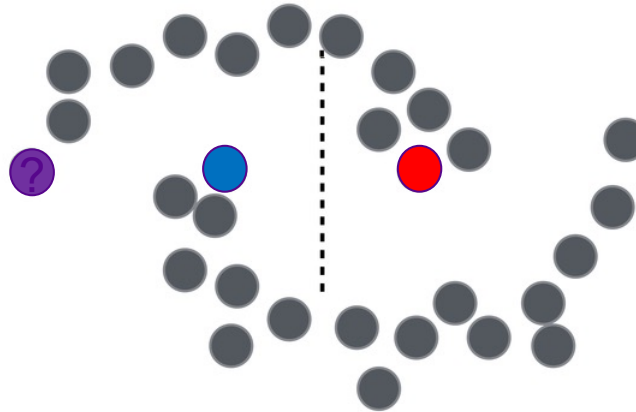
From Supervised to Semi-Supervised...

- If you have a small amount of labeled data and a large amount of unlabeled data (a very common situation in the real world...), the unlabeled data can often be used to improve prediction performance.



From Supervised to Semi-Supervised...

- If you have a small amount of labeled data and a large amount of unlabeled data (a very common situation in the real world...), the unlabeled data can often be used to improve prediction performance.



- For unlabeled data to improve performance, we must assume some structure to the underlying data distribution. Typical assumption of **smoothness**: data points near each other tend to belong to the same class.
- For example, can dramatically improve the performance of Naïve Bayes text classification using EM.

K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

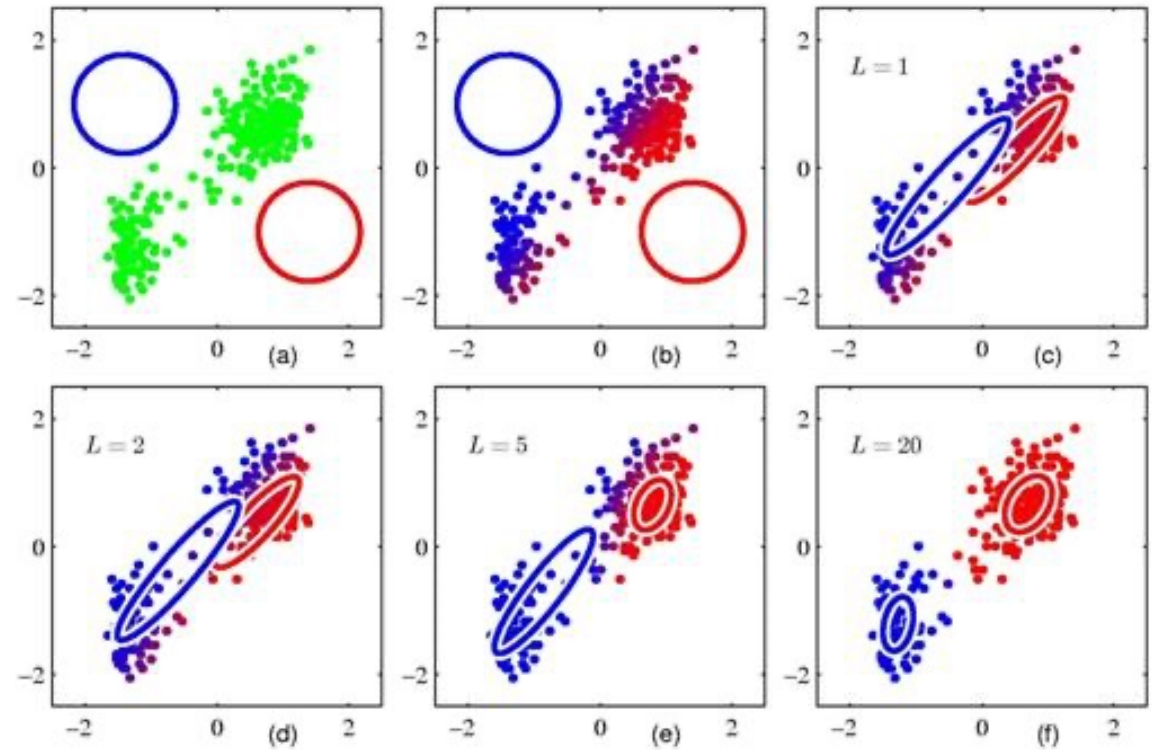
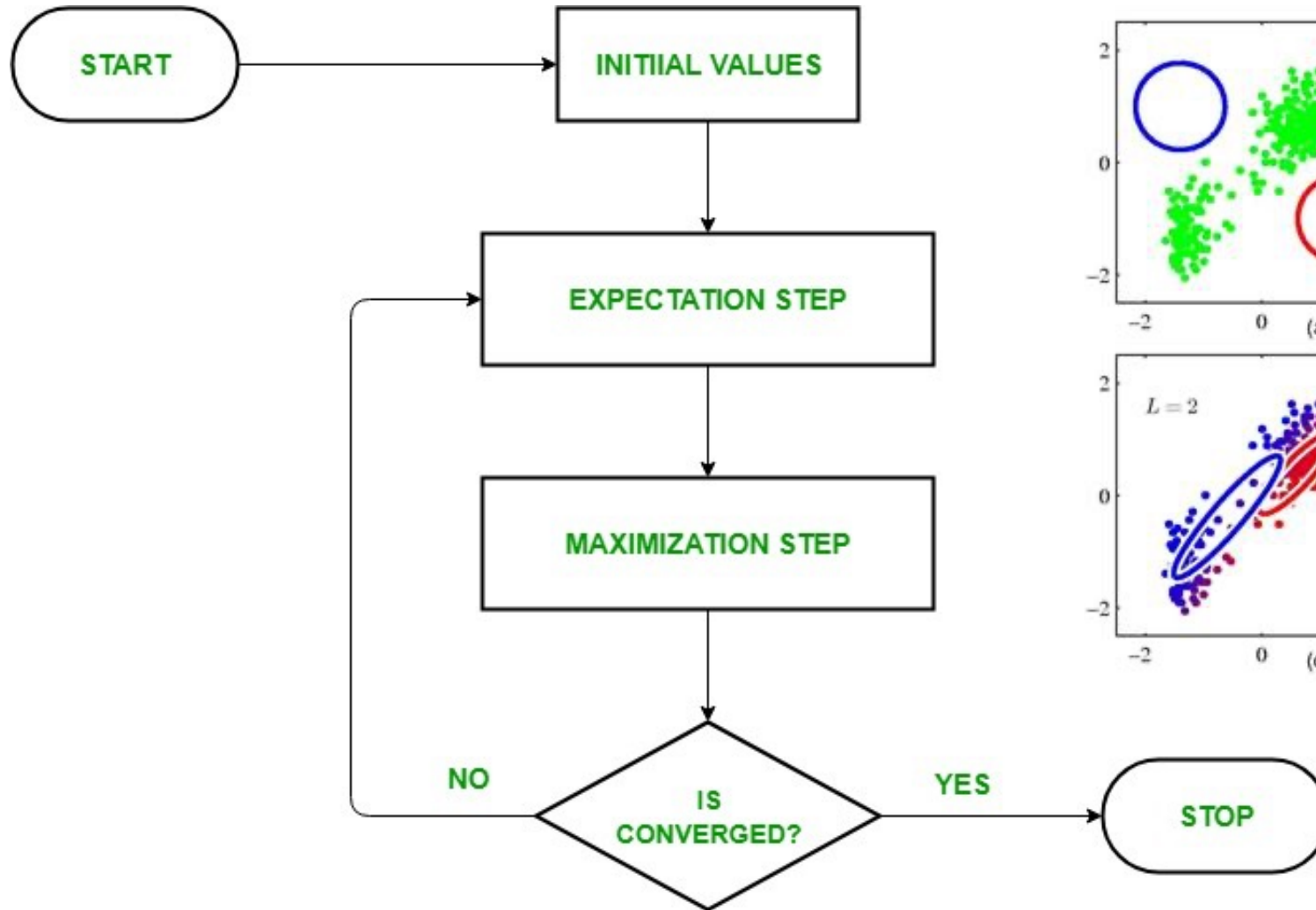
Semi-Supervised Naïve Bayes

- Let's represent each data record $i, i = 1..N$, by (x_i, y_i) where $x_i = (x_i^1..x_i^M)$. However, some y_i are unknown (unlabeled data).
- Initialization step: based only on the **labeled** data, set $t = 0$ and estimate parameters θ^t consisting of class-conditional models $\Pr(x^j = v \mid y = k)$ for discrete-valued attributes, Gaussian mean/var $((\mu_j, \sigma_j^2) \mid y = k)$ for real-valued attributes, and priors $\Pr(y = k)$.
 - Maximum likelihood estimation **just like training NB**: proportion or sample mean/var.
- Expectation (E) step: assume discrete random variable \hat{y}_i for each record. $\hat{y}_i = y_i$ if known; otherwise, we **compute class probabilities** given the current parameters θ^t , **just like classifying with NB**:

$$\Pr(\hat{y}_i = k) = \frac{\Pr(y = k \mid \theta^t) \prod_j \Pr(x^j = v_{ij} \mid y = k, \theta^t)}{\sum_{k'} (\Pr(y = k' \mid \theta^t) \prod_j \Pr(x^j = v_{ij} \mid y = k', \theta^t))}$$

- Maximization (M) step: compute new parameters θ^{t+1} consisting of class-conditional models $\Pr(x^j = v \mid y = k)$ for discrete-valued attributes, Gaussian mean/var $((\mu_j, \sigma_j^2) \mid y = k)$ for real-valued attributes, and priors $\Pr(y = k)$.
- Iterate between E and M steps until convergence. This process is typically called EM (Expectation-Maximization).

EM Algorithm



Semi-Supervised Naïve Baves

The maximization step is **just like training NB**, except that each data record with unknown y_i is treated as a partial observation of each class, weighted proportionally to the estimated class probabilities.

$$\Pr(y = k) = \frac{1}{N} \sum_{i=1..N} \Pr(\hat{y}_i = k)$$

$$\Pr(x^j = v \mid y = k) = \frac{\sum_{i=1..N} \Pr(\hat{y}_i = k) 1\{x_i^j = v\}}{\sum_{i=1..N} \Pr(\hat{y}_i = k)}$$

$$(\mu_j \mid y = k) = \frac{\sum_{i=1..N} \Pr(\hat{y}_i = k) x_i^j}{\sum_{i=1..N} \Pr(\hat{y}_i = k)} \quad (\sigma_j^2 \mid y = k) = \frac{\sum_{i=1..N} \Pr(\hat{y}_i = k) (x_i^j - (\mu_j \mid y = k))^2}{\sum_{i=1..N} \Pr(\hat{y}_i = k)}$$

- Maximization (M) step: compute new parameters θ^{t+1} consisting of class-conditional models $\Pr(x^j = v \mid y = k)$ for discrete-valued attributes, Gaussian mean/var $((\mu_j, \sigma_j^2) \mid y = k)$ for real-valued attributes, and priors $\Pr(y = k)$.
- Iterate between E and M steps until convergence. This process is typically called EM (Expectation-Maximization).

Semi-Supervised Naïve Bayes

The maximization step is **just like training NB**, except that each data record with unknown y_i is treated as a partial observation of each class, weighted proportionally to the estimated class probabilities.

$$\Pr(y = k) = \frac{1}{N} \sum_{i=1..N} \Pr(\hat{y}_i = k)$$

$$\Pr(x^j = v \mid y = k) = \frac{\sum_{i=1..N} \Pr(\hat{y}_i = k) 1\{x_i^j = v\}}{\sum_{i=1..N} \Pr(\hat{y}_i = k)}$$

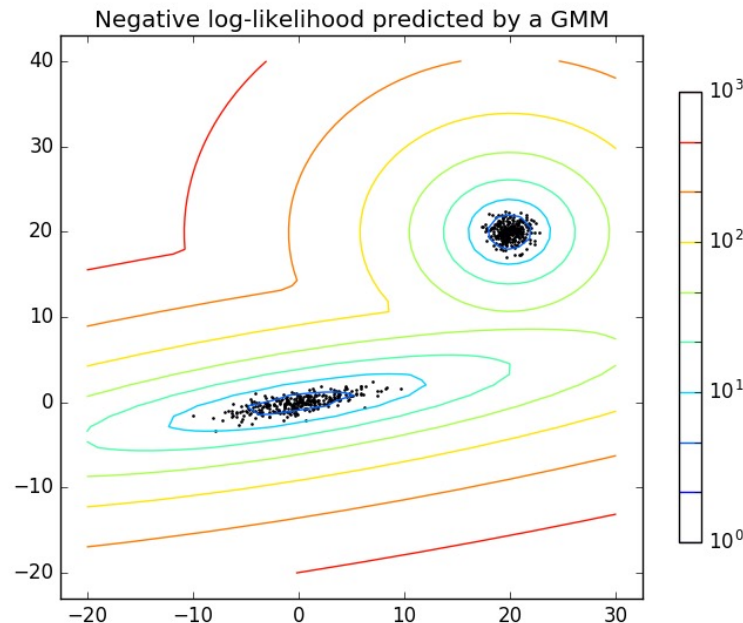
$$(\mu_j \mid y = k) = \frac{\sum_{i=1..N} \Pr(\hat{y}_i = k) x_i^j}{\sum_{i=1..N} \Pr(\hat{y}_i = k)} \quad (\sigma_j^2 \mid y = k) = \frac{\sum_{i=1..N} \Pr(\hat{y}_i = k) (x_i^j - (\mu_j \mid y = k))^2}{\sum_{i=1..N} \Pr(\hat{y}_i = k)}$$

The log-likelihood of the data increases monotonically at each step.

$$L(\theta) = \sum_i \log \left(\sum_k \Pr(\hat{y}_i = k) \prod_j \Pr(x_i^j = v_{ij} \mid y_i = k) \right)$$

From Semi-Supervised to Unsupervised...

- We can use the same EM algorithm (with the conditional independence assumption of NB) to perform **clustering** when the data is entirely unlabeled (all y_i are unknown).
- Choose number of clusters K . We will then partition the N data points into K clusters ($K \ll N$) to maximize the data log-likelihood.
 - Each point x_i gets a probabilistic cluster assignment, $\Pr(\hat{y}_i = k)$ for each cluster k .
- The resulting clusters capture the natural **grouping** of the data (i.e., similar points are placed in the same cluster), but may or may not be useful for predicting any particular output y_i .



From Semi-Supervised to Unsupervised...

- We can use the same EM algorithm (with the conditional independence assumption of NB) to perform **clustering** when the data is entirely unlabeled (all y_i are unknown).
- Choose number of clusters K . We will then partition the N data points into K clusters ($K \ll N$) to maximize the data log-likelihood.
 - Each point x_i gets a probabilistic cluster assignment, $\Pr(\hat{y}_i = k)$ for each cluster k .
- The resulting clusters capture the natural **grouping** of the data (i.e., similar points are placed in the same cluster), but may or may not be useful for predicting any particular output y_i .
- Can follow same E and M steps as in the semi-supervised case. Only difference is initialization (since no labeled points to start from):
 - Simplest approach: pick k points uniformly at random to assign cluster means.
 - Can start with equal cluster priors and variances (but not equal means: why?)
- This use of EM for clustering, with real-valued attributes and assumed Gaussian distributions, is called Gaussian mixture models (GMM).
- EM has many other uses, potentially any time you fit parameters by maximum likelihood but have latent (unobserved) variables.
- Lots more on clustering in the next lecture; for now, we've focused mostly on the supervised and semi-supervised cases.

Lab Time

For the Next Week (Week 7)

1. Check readings (optional) and review them
2. Assignment 2 (SVMs, Naïve Bayes, EM)

Due: March 6, 2024 (11:59pm)

References

Naïve Bayes:

1. Scikit-learn documentation for Naïve Bayes (supervised)
2. http://scikit-learn.org/stable/modules/naive_bayes.html
3. T.M. Mitchell (2020). Generative and discriminative classifiers: naïve Bayes and logistic regression. Ch. 3 of upcoming book, Machine Learning, 2nd ed. <http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf>
4. A.W. Moore. Probability for data miners, available at: <https://www.cs.cmu.edu/~awm/tutorials/prob.html>
5. Duda, Hart, and Stork, Ch. 2, 3, and 10.

Expectation-Maximization (EM):

1. Scikit-learn documentation for Expectation Maximization (unsupervised- Gaussian mixture model)
2. <http://scikit-learn.org/stable/modules/mixture.html>
3. A.P. Dempster, N.M. Laird, and D.B. Rubin (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, Series B. 39(1): 1–38.
4. A.W. Moore. Clustering with Gaussian mixtures, available at: <https://www.cs.cmu.edu/~awm/tutorials/gmm.html>