CSC 411: Lecture 08: Generative Models for Classification

Richard Zemel, Raquel Urtasun and Sanja Fidler

University of Toronto

Today

- Classification – Bayes classifier
- Estimate probability densities from data
- Making decisions: Risk

Classification

- Given inputs **x** and classes y we can do classification in several ways. How?

(features)

$\mathbf{x}$

e.g:
- height
- weight
- color

(class label)

y

elefant

bear

Discriminative Classifiers

- **Discriminative** classifiers try to either:
  - learn mappings directly from the space of inputs $\mathcal{X}$ to class labels $\{0, 1, 2, \dots, K\}$

(features)

$\mathbf{x}$

e.g:

- height
- weight
- color

$\longrightarrow$

(class label)

y

elefant

bear

For example:

- Linear regression (as a classifier)
- Neural Networks

Discriminative Classifiers

- **Discriminative** classifiers try to either:
  - or try to learn $p(y|\mathbf{x})$ directly

(features)

$\mathbf{x}$

e.g:
- height
- weight
- color

$\longrightarrow$

(class probability)

$p(y|\mathbf{x})$

elefant

bear

For example:
- Logistic Regression

Generative Classifiers

How about this approach: build a model of "how data for a class looks like"

- **Generative** classifiers try to model $p(\mathbf{x}|y)$
- Classification via Bayes rule (thus also called Bayes classifiers)

(prob. of features given label) (class label)

$p(\mathbf{x}|y)$ $\longleftarrow$ y

e.g:
- height
- weight
- color

elefant

bear

Tries to model:
- How does data look like for a class?

Classification (How?)

Generative vs Discriminative

Two approaches to classification:

- Discriminative classifiers estimate parameters of decision boundary/class separator directly from labeled examples
  - learn $p(y|\mathbf{x})$ directly (logistic regression models)
  - learn mappings from inputs to classes (least-squares, neural nets)

- Generative approach: model the distribution of inputs characteristic of the class (Bayes classifier)
  - Build a model of $p(\mathbf{x}|y)$
  - Apply Bayes Rule

Bayes Classifier

- Aim to diagnose whether patient has diabetes: classify into one of two classes (yes C=1; no C=0)
- Run battery of tests on the patients, get $\mathbf{x}$ for each patient
- Given patient's results: $\mathbf{x} = [x_1, x_2, \cdots, x_d]^T$ we want to compute class probabilities using Bayes Rule:

$$p(C|\mathbf{x}) = \frac{p(\mathbf{x}|C)p(C)}{p(\mathbf{x})}$$

- More formally

$$\text{posterior} = \frac{\text{Class likelihood} \times \text{prior}}{\text{Evidence}}$$

- How can we compute $p(\mathbf{x})$ for the two class case?

$$p(\mathbf{x}) = p(\mathbf{x}|C=0)p(C=0) + p(\mathbf{x}|C=1)p(C=1)$$

- To compute $p(C|\mathbf{x})$ we need: $p(\mathbf{x}|C)$ and $p(C)$

Classification: Diabetes Example

- Let's start with the simplest case where the input is only 1-dimensional, for example: white blood cell count (this is our x)
- We need to choose a probability distribution $p(x|C)$ that makes sense

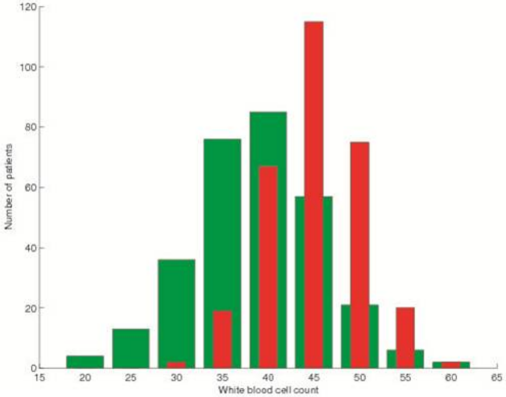


Figure : Our example (showing counts of patients for input value): What distribution to choose?

Gaussian Discriminant Analysis (Gaussian Bayes Classifier)

- Our first generative classifier assumes that $p(\mathbf{x}|y)$ is distributed according to a multivariate normal (Gaussian) distribution
- This classifier is called Gaussian Discriminant Analysis
- Let's first continue our simple case when inputs are just 1-dim and have a Gaussian distribution:

$$p(x|C) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu_C)^2}{2\sigma_C^2}\right)$$

  with $\mu \in \Re$ and $\sigma^2 \in \Re^+$
- Notice that we have different parameters for different classes
- How can I fit a Gaussian distribution to my data?

MLE for Gaussians

- Let's assume that the class-conditional densities are Gaussian

$$p(x|C) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu_C)^2}{2\sigma_C^2}\right)$$

with $\mu \in \Re$ and $\sigma^2 \in \Re^+$

- How can I fit a Gaussian distribution to my data?
- We are given a set of training examples $\{x^{(n)}, t^{(n)}\}_{n=1,\cdots,N}$ with $t^{(n)} \in \{0,1\}$ and we want to estimate the model parameters $\{(\mu_0, \sigma_0), (\mu_1, \sigma_1)\}$
- First divide the training examples into two classes according to $t^{(n)}$, and for each class take all the examples and fit a Gaussian to model $p(x|C)$
- Let's try maximum likelihood estimation (MLE)

MLE for Gaussians

(note: we are dropping subscript C for simplicity of notation)

- We assume that the data points that we have are independent and identically distributed

$$p(x^{(1)}, \cdots, x^{(N)}|C) = \prod_{n=1}^{N} p(x^{(n)}|C) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x^{(n)} - \mu)^2}{2\sigma^2}\right)$$

- Now we want to maximize the likelihood, or minimize its negative (if you think in terms of a loss)

$$\begin{aligned}
\ell_{log-loss} &= -\ln p(x^{(1)}, \cdots, x^{(N)}|C) = -\ln\left(\prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x^{(n)} - \mu)^2}{2\sigma^2}\right)\right) \\
&= \sum_{n=1}^{N} \ln(\sqrt{2\pi}\sigma) + \sum_{n=1}^{N} \frac{(x^{(n)} - \mu)^2}{2\sigma^2} = \frac{N}{2} \ln\left(2\pi\sigma^2\right) + \sum_{n=1}^{N} \frac{(x^{(n)} - \mu)^2}{2\sigma^2}
\end{aligned}$$

- How do we minimize the function?

Computing the Mean

- (let's try to find a) Closed-form solution: Write $\frac{d\ell_{log-loss}}{d\mu}$ and $\frac{d\ell_{log-loss}}{d\sigma^2}$ and equal it to 0 to find the parameters μ and σ^2

$$
\begin{aligned}
\frac{\partial \ell_{log-loss}}{\partial \mu} &= \frac{\partial \left(\frac{N}{2} \ln \left(2\pi\sigma^2\right) + \sum_{n=1}^{N} \frac{(x^{(n)}-\mu)^2}{2\sigma^2} \right)}{\partial \mu} = \frac{d \left(\sum_{n=1}^{N} \frac{(x^{(n)}-\mu)^2}{2\sigma^2} \right)}{d\mu} \\
&= \frac{-\sum_{n=1}^{N} 2(x^{(n)}-\mu)}{2\sigma^2} = -\sum_{n=1}^{N} \frac{(x^{(n)}-\mu)}{\sigma^2} = \frac{N\mu - \sum_{n=1}^{N} x^{(n)}}{\sigma^2}
\end{aligned}
$$

- And equating to zero we have

$$
\frac{d\ell_{log-loss}}{d\mu} = 0 = \frac{N\mu - \sum_{n=1}^{N} x^{(n)}}{\sigma^2}
$$

Thus

$$
\boxed{\mu = \frac{1}{N} \sum_{n=1}^{N} x^{(n)}}
$$

Computing the Variance

- And for σ^2:

$$
\begin{aligned}
\frac{d\ell_{log-loss}}{d\sigma^2} &= \frac{d\left(\frac{N}{2}\ln\left(2\pi\sigma^2\right) + \sum_{n=1}^{N}\frac{(x^{(n)}-\mu)^2}{2\sigma^2}\right)}{d\sigma^2} \\
&= \frac{N}{2}\frac{1}{2\pi\sigma^2}2\pi + \frac{\sum_{n=1}^{N}(x^{(n)}-\mu)^2}{2}\left(\frac{-1}{\sigma^4}\right) \\
&= \frac{N}{2\sigma^2} - \frac{\sum_{n=1}^{N}(x^{(n)}-\mu)^2}{2\sigma^4}
\end{aligned}
$$

- And equating to zero we have

$$
\frac{d\ell_{log-loss}}{d\sigma^2} = 0 = \frac{N}{2\sigma^2} - \frac{\sum_{n=1}^{N}(x^{(n)}-\mu)^2}{2\sigma^4} = \frac{N\sigma^2 - \sum_{n=1}^{N}(x^{(n)}-\mu)^2}{2\sigma^4}
$$

- Thus:

$$
\boxed{\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x^{(n)}-\mu)^2}
$$

MLE of a Gaussian

- In summary, we can compute the parameters of a Gaussian distribution in closed form for each class by taking the training points that belong to that class

MLE estimates of parameters for a Gaussian distribution:

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x^{(n)}$$

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N} (x^{(n)} - \mu)^2$$

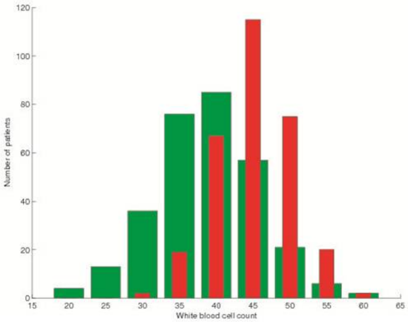
Posterior Probability

- We now have $p(x|C)$
- In order to compute the posterior probability:

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

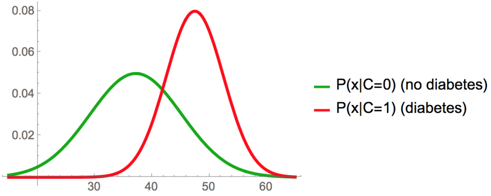
  given a new observation, we still need to compute the **prior**
- Prior: In the absence of any observation, what do I know about the problem?

Diabetes Example



- Doctor has a prior $p(C = 0) = 0.8$, how?
- A new patient comes in, the doctor measures $x = 48$
- Does the patient have diabetes?

Diabetes Example



- Compute $p(x = 48|C = 0)$ and $p(x = 48|C = 1)$ via our estimated Gaussian distributions
- Compute posterior $p(C = 0|x = 48)$ via Bayes rule using the prior (how can we get $p(C = 1|x = 48)$?)
- How can we decide on diabetes/non-diabetes?

Bayes Classifier

- Use Bayes classifier to classify new patients (unseen test examples)
- Simple Bayes classifier: estimate posterior probability of each class
- What should the decision criterion be?
- The optimal decision is the one that minimizes the expected number of mistakes

Risk of a Classifier

- Risk (expected loss) of a C-class classifier $y(\mathbf{x})$:

$$\begin{aligned} R(y) &= E_{\mathbf{x},t}[L(y(\mathbf{x}),t)] \\ &= \int_{\mathbf{x}} \sum_{c=1}^{C} L(y(\mathbf{x}),t)p(\mathbf{x},t=c)d\mathbf{x} \\ &= \int_{\mathbf{x}} \left[\sum_{c=1}^{C} L(y(\mathbf{x}),t)p(t=c|\mathbf{x}) \right] p(\mathbf{x})d\mathbf{x} \end{aligned}$$

- Clearly, its enough to minimize the conditional risk for any $\mathbf{x}$:

$$R(y|\mathbf{x}) = \sum_{c=1}^{C} L(y(\mathbf{x}),t)p(t=c|x)$$

Conditional Risk of a Classifier

- We have assumed a zero-one loss:

$$L(y(\mathbf{x}), t) = \begin{cases} 0 & \text{if } y(\mathbf{x}) = t \\ 1 & \text{if } y(\mathbf{x}) \neq t \end{cases}$$

- **Conditional risk:**

$$\begin{aligned} R(y|\mathbf{x}) &= \sum_{c=1}^{C} L(y(\mathbf{x}), t) p(t = c|\mathbf{x}) \\ &= 0 \cdot p(t = y(\mathbf{x})|\mathbf{x}) + 1 \cdot \sum_{c \neq y} p(t = c|\mathbf{x}) \\ &= \sum_{c \neq y} p(t = c|\mathbf{x}) = 1 - p(t = y(\mathbf{x})|\mathbf{x}) \end{aligned}$$

- To minimize conditional risk given x, the classifier must decide

$$y(\mathbf{x}) = arg\max p(t = c|x)$$

Log-odds Ratio

- Optimal rule $y = arg\max_c p(t = c|x)$ is equivalent to

$$\begin{aligned} y = c \quad &\Leftrightarrow \quad \frac{p(t=c|x)}{p(t=j|x)} \geq 1 \quad \forall j \neq c \\ &\Leftrightarrow \quad \log \frac{p(t=c|x)}{p(t=j|x)} \geq 0 \quad \forall j \neq c \end{aligned}$$

- For the binary case

$$y = 1 \quad \Leftrightarrow \quad \log \frac{p(t=1|x)}{p(t=0|x)} \geq 0$$

- Where have we used this rule before?

Gaussian Discriminant Analysis

- Consider the 2-class case
- Interesting: When $\sigma_0 = \sigma_1$, then the posterior takes the following form:

$$p(t = 1|x) = \frac{1}{1 + e^{-w \cdot x}}$$

  where w is some appropriate function of $\phi, \mu_0, \mu_1, \sigma_0$, where we denoted the prior with $p(t) = \phi^t (1 - \phi)^{(1-t)}$ (Bernoulli distribution). Prove this!
- In this case the GDA and Logistic Regression are equivalent
- When would you choose one over the other?
- GDA makes strong modeling assumptions (data has Gaussian distribution)
- If data really had Gaussian distribution, then GDA will find a better fit
- Logistic Regression is more robust and less sensitive to incorrect modeling assumptions

[Credit: A. Ng]