Generative Modelling

Christos Dimitrakakis

October 31, 2023

Outline

Generative modelling

general idea

- Data (x, y).
- Need to model $P(y|x)$.
- Model the complete data distribution: $P(x|y)$, $P(x)$, $P(y)$.
- Calculate $P(y|x) = \frac{P(x|y)P(x)}{P(y)}$.

Examples

- Naive Bayes classifier
- Gaussian Mixture Classifier

Modelling the data distribution

- Need to estimate the density $P(x|y)$ for each class y.
- Need to estimate $P(y)$

The basic graphical model

A discriminative classification model

Here $P(y|x)$ is given directly. $y \longleftarrow x$

A generative classification model

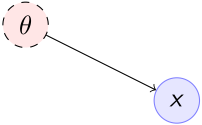
Here $P(y|x) = P(x|y)P(y)/P(x)$. $y \longrightarrow x$

A generative model

Here we just have $P(x)$. x

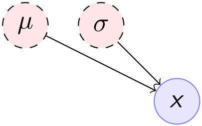
Adding parameters to the graphical model

A Bernoulli RV



Here, $x|\theta \sim \mathrm{Bernoulli}(\theta)$

A normally distributed variable



Here $x|\mu, \sigma \sim \mathrm{Normal}(\mu, \sigma^2)$

Classification: Naive Bayes Classifier

- Data (x, y)
- $x \in X$
- $y \in Y \subset \mathbb{N}$, N_i: amount of data from class i.

Separately model each class

- Assume each class data comes from a different normal distribution
- $x|y = i \sim \mathrm{Normal}(\mu_i, \sigma_i I)$
- For each class, calculate
  - Empirical mean $\hat{\mu}_i = \sum_{t: y_t = i} x_t / N_i$
  - Empirical variance $\hat{\sigma}_i$.

Decision rule

Use Bayes's theorem:

$$P(y|x) = P(x|y)P(y)/P(x),$$

choosing the y with largest posterior $P(y|x)$.

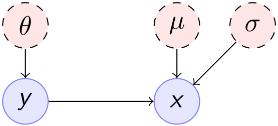
- $P(x|y = i) \propto \exp(-\|\hat{\mu}_i - x\|^2 / \hat{\sigma}_i^2)$

Graphical model for the Naive Bayes Classifier

When $x \in \mathbb{R}$

Assume k classes, then

- $\mu = (\mu_1, \ldots, \mu_k)$
- $\sigma = (\sigma_1, \ldots, \sigma_k)$
- $\theta = (\theta_1, \ldots, \theta_k)$



- $y \mid \theta \sim \mathrm{Multinomial}(\theta)$
- $x \mid y, \mu, \sigma \sim \mathrm{Normal}(\mu_y, \sigma_y^2)$

General idea

Parametric models

- Fixed histograms
- Gaussian Mixtures

Non-parametric models

- Variable-bin histograms
- Infinite Gaussian Mixture Model
- Kernel methods

Histograms

Fixed histogram

- Hyper-Parameters: number of bins
- Parameters: Number of points in each bin.

Variable histogram

- Hyper-parameters: Rule for constructing bins
- Generally $\sqrt{n}$ points in each bin.

Gaussian Mixture Model

Hyperparameters:

- Number of Gaussian k.

Parameters:

- Multinomial distribution $\boldsymbol{\theta}$ over Gaussians
- For each Gaussian i, center μ_i, covariance matrix Σ_i.

Model. For each point x_t:

- $c_t = i$ w.p. θ_i
- $x_t | c_t = i \sim \mathrm{Normal}(\mu_i, \Sigma_i)$.

Algorithms:

- Expectation Maximisation
- Gradient Ascent
- Variational Bayesian Inference (with appropriate prior)

Gradient ascent

Objective function

$L(\theta) = P(x|\theta)$

Marginalisation over latent variable

$L(θ) = ∑_z P(z, x | θ)

Gradient ascent

$\theta^{(n+1)} = \theta^{(n)} + \alpha \nabla_\theta L(\theta)$

Gradient calculation

Here we use the log trick: $\nabla \ln f(x) = \nabla f(x)/f(x)$.

$$
\begin{aligned}
\nabla_\theta L(\theta) &= \sum_z \nabla_\theta P(z, x \mid \theta) && (1)\\
&= \sum_z P(z, x \mid \theta) \nabla_\theta \ln P(z, x \mid \theta) && (2)\\
&= \sum_z P(x \mid z, \theta) P(z \mid \theta) \nabla_\theta \ln P(z, x \mid \theta) && (3)\\
&\approx \frac{1}{m} \sum_{i=1}^{m} P(x \mid z^{(i)}, \theta) \nabla_\theta \ln P(z^{(i)}, x \mid \theta) \quad z^{(i)} \sim P(z| \mid \theta) && (4)
\end{aligned}
$$

A lower bound on the likelihood

$$
\begin{aligned}
\ln P(x|\theta) &= \sum_z G(z)P(x|\theta) \\
&= \sum_z G(z)[\ln P(x,z|\theta) - \ln P(z|x,\theta)] \\
&= \sum_z G(z)\ln P(x,z|\theta) - \sum_z G(z)\ln P(z|x,\theta)] \\
&= \sum_z P(z|x,\theta^{(k)})\ln P(x,z|\theta) - \sum_z P(z|x,\theta^{(k)})\ln P(z|x,\theta) \\
&\geq \sum_z P(z|x,\theta^{(k)})\ln P(x,z|\theta) - \sum_z P(z|x,\theta^{(k)})\ln P(z|x,\theta^{(k)}) \\
&= Q(\theta \mid \theta^{(k)}) + \mathbb{H}(z \mid x = x, \theta = \theta^{(k)})
\end{aligned}
$$

The Gibbs Inequality

$D_{KL}(P\|Q) \geq 0$, or $\sum_x \ln P(x)P(x) \geq \sum_x \ln Q(x)P(x)$.

EM Algorithm (Dempster et al, 1977)

- Initial parameter $\boldsymbol{\theta}^{(0)}$, observed data x
- For $k = 0, 1, \ldots$

– Expectation step:

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(k)}) \triangleq \mathbb{E}_{z \sim P(z|x,\boldsymbol{\theta}^{(k)})}[\ln P(x, z|\boldsymbol{\theta})] = \sum_{z}[\ln P(x, z|\boldsymbol{\theta})]P(z \mid x, \boldsymbol{\theta}^{(k)})$$

– Maximisation step:

$$\boldsymbol{\theta}^{(k+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}).$$

See *Expectation-Maximization as lower bound maximization, Minka, 1998*

Minorise-Maximise

EM can be seen as a version of the minorise-maximise algorithm

- $f(\boldsymbol{\theta})$: Target function to maximise
- $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$: surrogate function

Q Minorizes f

This means surrogate is always a lower bound so that

$$f(\boldsymbol{\theta}) \geq Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}), \qquad f(\boldsymbol{\theta}^{(k)}) \geq Q(\boldsymbol{\theta}^{(k)}|\boldsymbol{\theta}^{(k)}),$$

Algorithm

- Calculate: $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$
- Optimise: $\boldsymbol{\theta}^{(k+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$.

GMM versus histogram

- Generate some data x from an arbitrary distribution in $\mathbb{R}$.
- Fit the data with a histogram for varying numbers of bins
- Fit a GMM with varying numbers of Gaussians
- What is the best fit? How can you measure it?

GMM Classifier

Base class: sklearn GaussianMixtureModel

- *fit()* only works for Density Estimaiton
- *predict()* only predicts cluster labels

Problem

- Create a GMMClassifier class
- *fit()* should take X, y, arguments
- *predict()* should predict class labels
- Hint: Use $predict_{proba}()$ and multiple GMM models