Machine Learning Foundations

Mahesan Niranjan

School of Electronics and Computer Science
University of Southampton

Summer School 2020

Machine Learning as Data-driven Modelling

Single-slide overview of the subject and challenging questions

Data	$\{\boldsymbol{x}_n,\ \boldsymbol{y}_n\}_{n=1}^{N}$ $\quad\quad \{\boldsymbol{x}_n\}_{n=1}^{N}$
Function Approximator	$\boldsymbol{y} \ = \ f\left(\boldsymbol{x},\ \boldsymbol{\theta}\right) \ + \ v$
Parameter Estimation	$E_0 \ = \ \sum_{n=1}^{N} \left\{ \lvert\lvert\, \boldsymbol{y}_n \ - \ f\left(\boldsymbol{x}_n;\ \boldsymbol{\theta}\right)\ \rvert\rvert \right\}^2$
Prediction	$\hat{\boldsymbol{y}}_{N+1} \ = \ f\left(\boldsymbol{x}_{N+1},\ \hat{\boldsymbol{\theta}}\right)$
Regularization	$E_1 \ = \ \sum_{n=1}^{N} \left\{ \lvert\lvert\, \boldsymbol{y}_n \ - \ f\left(\boldsymbol{x}_n\right)\ \rvert\rvert \right\}^2 \ + \ g\left(\lvert\lvert\boldsymbol{\theta}\rvert\rvert\right)$
Modelling Uncertainty	$p\left(\boldsymbol{\theta} \mid \{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^{N}\right)$
Probabilistic Inference	$\boldsymbol{E}\left[g\left(\boldsymbol{\theta}\right)\right] = \int g\left(\boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right) d\boldsymbol{\theta} \ = \frac{1}{N_s} \sum_{n=1}^{N_s} g\left(\boldsymbol{\theta}^{(n)}\right)$
Sequential Estimation	$\boldsymbol{\theta}\left(n-1 \mid n-1\right) \longrightarrow \boldsymbol{\theta}\left(n \mid n-1\right) \longrightarrow \boldsymbol{\theta}\left(n \mid n\right)$ Kalman & Particle Filters; Reinforcement Learning

Bayesian Decision Theory

- Classes: $\omega_i, \; i = 1, ..., K$
- Prior Probabilities: $P[\omega_1], ..., P[\omega_K]$;
  $P[\omega_i] \geq 0, \quad \sum_{i=1}^{C} P[\omega_i] \; = \; 1$
- Likelihoods (class conditional probabilities): $p(\boldsymbol{x}|\omega_i), \; i = 1, .., K$
- Posterior Probability: $P[\omega_j \,|\, \boldsymbol{x}]$

$$P[\omega_j \,|\, \boldsymbol{x}] \;=\; \frac{p(\boldsymbol{x} \,|\, \omega_j) \; P[\omega_j]}{\sum_{i=1}^{K} p(\boldsymbol{x} \,|\, \omega_i) \; P[\omega_i]}$$

- From prior knowledge: $P[\omega_i]$; From training data: $p(\boldsymbol{x}|\omega_i)$
- Decision rule: Assign $\boldsymbol{x}$ to the class that maximizes posterior probability.
- The denominator is a constant; *i.e.* does not depend on ω_j
- Hence the decision rule becomes:

$$\boldsymbol{x} \in \max_j p(\boldsymbol{x} \,|\, \omega_j) \; P[\omega_j]$$

Bayes' Classifier for Gaussian Densities

Make assumptions, cancel common terms when making comparisons...

- Decision rule from: $p(\boldsymbol{x} \,|\, \omega_j) \; P[\omega_j]$
- Assume the two classes are Gaussian distributed with distinct means and identical covariance matrices
  $p(\boldsymbol{x} \,|\, \omega_j) \;=\; \mathcal{N}(\boldsymbol{m}_j, \, \boldsymbol{C})$
- Substitute into Bayes' classifier decision rule

$$\begin{aligned} P[\omega_1|\boldsymbol{x}] \quad &\lessgtr \quad P[\omega_2|\boldsymbol{x}] \\ p(\boldsymbol{x}|\omega_1) \; P[\omega_1] \quad &\lessgtr \quad p(\boldsymbol{x}|\omega_2) \; P[\omega_2] \end{aligned}$$

$$\frac{1}{(2\pi)^{p/2}(\det(C))^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{m}_1)^t \boldsymbol{C}^{-1}(\boldsymbol{x} - \boldsymbol{m}_1)\right\} P[\omega_1] \quad \lessgtr$$
$$\frac{1}{(2\pi)^{p/2}(\det(C))^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{m}_2)^t \boldsymbol{C}^{-1}(\boldsymbol{x} - \boldsymbol{m}_2)\right\} P[\omega_2]$$

Bayes' classifier for simple densities (cont'd)

Distinct Means; Equal, isotropic covariance matrix

- Suppose the densities are isotropic and priors are equal *i.e.* $\boldsymbol{C} = \sigma^2 \boldsymbol{I}$ and $P[\omega_1] = P[\omega_2]$
- The comparison simplifies to (see algebra on board):

$$\begin{aligned}(\boldsymbol{x} - \boldsymbol{m}_1)^t(\boldsymbol{x} - \boldsymbol{m}_1) &\lessgtr (\boldsymbol{x} - \boldsymbol{m}_2)^t(\boldsymbol{x} - \boldsymbol{m}_2) \\ |\boldsymbol{x} - \boldsymbol{m}_1| &\lessgtr |\boldsymbol{x} - \boldsymbol{m}_2|\end{aligned}$$

- The above is a simple *distance to mean* classifier
- Under the above simplistic assumptions, we only need to store one template per class (the means)!

Bayes' classifier for simple densities (cont'd)

Distinct Means; Common covariance matrix (but not isotropic)

- Cancel common terms and take log

$$(\boldsymbol{x} - \boldsymbol{m_1})^t \, \boldsymbol{C}^{-1} \, (\boldsymbol{x} - \boldsymbol{m_1}) \lessgtr (\boldsymbol{x} - \boldsymbol{m_2})^t \, \boldsymbol{C}^{-1} \, (\boldsymbol{x} - \boldsymbol{m_2}) \; - \; \log \left\{ \frac{P[\omega_1]}{P[\omega_2} \right\}$$

- Also simplifies to a $\boxed{\text{linear classifier}}$!

$$\boldsymbol{w}^t \, \boldsymbol{x} \, + \, b \lessgtr 0$$

$$\begin{aligned}\boldsymbol{w} &= 2\boldsymbol{C}^{-1} \, (\boldsymbol{m}_2 - \boldsymbol{m}_1) \\ b &= \left(\boldsymbol{m}_1^t \boldsymbol{C}^{-1} \boldsymbol{m}_1 \; - \; \boldsymbol{m}_2^t \boldsymbol{C}^{-1} \boldsymbol{m}_2 \right) \; - \; \log \left\{ \frac{P[\omega_1]}{P[\omega_2]} \right\}\end{aligned}$$

- Also a distance to template classifier, where the distance is

$$(\boldsymbol{x} - \boldsymbol{m_1})^t \, \boldsymbol{C}^{-1} \, (\boldsymbol{x} - \boldsymbol{m_1})$$

Known as Mahalanobis distance

Posterior probabilities for simple Gaussian cases

- Bayes classifier:

$$P\left[\omega_1|\boldsymbol{x}\right] = \frac{p\left(\boldsymbol{x}|\omega_1\right)P\left[\omega_1\right]}{p\left(\boldsymbol{x}|\omega_1\right)P\left[\omega_1\right] + p\left(\boldsymbol{x}|\omega_2\right)P\left[\omega_2\right]}$$

- Restrictive assumptions:
  - Gaussian $p\left(\boldsymbol{x}|\omega_j\right) = \mathcal{N}\left(\boldsymbol{m}_j, \boldsymbol{C}_j\right)$
  - Equal covariance matrices: $\boldsymbol{C}_1 = \boldsymbol{C}_2 = C$
- Substitute, divide through by numerator term and cancel common terms to get

$$P\left[\omega_1|\boldsymbol{x}\right] = \frac{1}{1 + \exp\left\{-\left(\boldsymbol{w}^t\boldsymbol{x} + w_0\right)\right\}}$$

- The functional form $1/(1+\exp(-\alpha))$ is known as sigmoid / logistic (See lab class for W3)

Bayesian decision theory is fundamental to machine learning; we started from a probabilistic setting in which we described the posterior probability of membership and derived several results starting from this premise. You need to be able to:

- Derive and demonstrate that under certain assumptions...
  - The class boundary is linear
  - The posterior probability has a sigmoidal shape
  - The optimal classifier reduces to a distance to template classifier
- ...and under certain other assumptions...
  - The best classifier is still a distance to template classifier, but instead of Euclidean distance we need to use Mahalanobis distance.
- ...and under certain other assumptions...
  - The optimal classifier is a quadratic classifier