Statistical Machine Learning (STAT W4400)

Homework 1

Linan Qiu
lq2137

September 29, 2015

1 Naive Bayes

1.1 Gaussian Assumption into a Naive Bayes Classifier

$$\begin{aligned}\hat{y}_{\text{new}} &= f(\mathbf{x}_{\text{new}}) \\ &= \arg\max_{y \in k} P(y|\mathbf{x}_{\text{new}}) \\ &= \arg\max_{y \in k} \frac{P(\mathbf{x}_{\text{new}}|y)P(y)}{P(\mathbf{x}_{\text{new}})} \\ &= \arg\max_{y \in k} P(\mathbf{x}_{\text{new}}|y)P(y)\end{aligned}$$

Naive bayes assumes $p(\mathbf{x}|y) = \prod_{j=1}^{d} p(x_j|y)$

Then,

$$\begin{aligned}\hat{y}_{\text{new}} &= \arg\max_{y \in k} \left(\prod_{j=1}^{d} p(x_j|y) \right) p(y) \\ &= \arg\max_{y \in k} \left(\prod_{j=1}^{d} \frac{1}{\sqrt{2\pi}} \exp - \frac{(x - \mu_k)^2}{2} \right) p(y)\end{aligned}$$

1.2 Parameter Estimation

1.2.1 Parameter Estimation for Class-Conditional Distribution

Covariance matrix is given as the identity matrix $\mathbb{I}$. Then, we only have to estimate μ_k for each class. Using a MLE estimator, we find

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i=1}^{n} x_i \left[\mathbb{1}(y_i == C_k)\right]$$

where n_k is the number of training datum in C_k.

1.2.2 Parameter Estimation for Class Prior

$p(y)$ can be estimated from the training data as follows:

$$\hat{p}(y_k) = \frac{n_k}{n}$$

where n_k is the number of training datum in C_k.

1.3 Performance of Naive Bayes Classifier

Naive bayes only performs well with low overlap. Since the distributions are assumed to be spherical gaussian with unit covariance, it will only work well if the means μ_k are far apart from each other. Otherwise, the data will be only marginally separable. One can calculate the standard deviations for which we can classify data reliable 95% of the time, but I'm tired.

2 Maximum Likelihood Estimation

2.1 General Analytic Procedure

The maximum likelihood estimator is defined as

$$\hat{\theta}_{\mathrm{ML}} := \arg\max_{\theta \in T} p(x_1, ..., x_n|\theta)$$

the parameter which maximizes the joint density of the data.

Now if the data was i.i.d, $p(x_1, ..., x_n) = \prod_{i=1}^{n} p(x_i|\theta) = l(\theta)$. Then the maximum likelihood estimator would be

$$\nabla_\theta \left(\prod_{i=1}^{n} p(x_i|\theta) \right) = 0$$

Then,

$$\begin{aligned}
\hat{\theta}_{\text{ML}} &= \arg\max_\theta \prod_{i=1}^{n} p(x_i|\theta) \\
&= \arg\max_\theta \log \left(\prod_{i=1}^{n} p(x_i|\theta) \right) \\
&= \arg\max_\theta \sum_{i=1}^{n} \log p(x_i|\theta)
\end{aligned}$$

Then the analytic maximality criterion is

$$0 = \sum_{i=1}^{n} \nabla_\theta \log p(x_i|\theta) = \sum_{i=1}^{n} \frac{\nabla_\theta p(x_i|\theta)}{p(x_i|\theta)}$$

2.2 ML Estimate for Location Parameter μ

It will be easier to start with the log maximality criterion instead of proceeding to the un-log-ed one.

$$
\begin{aligned}
&\sum_{i=1}^{n} \nabla_{\hat{\mu}} \log p(x_i|\theta) \\
=&\sum_{i=1}^{n} \nabla_{\hat{\mu}} \log \left(\frac{v}{\hat{\mu}}\right)^{v} \frac{x^{v-1}}{\Gamma(v)} \exp\left(\frac{-vx_i}{\hat{\mu}}\right) \\
=&\sum_{i=1}^{n} \nabla_{\hat{\mu}} \left[v \log \left(\frac{v}{\hat{\mu}}\right) + \log\left(\frac{x^{v-1}}{\Gamma(v)}\right) - \frac{vx_i}{\hat{\mu}}\right] \\
=&\sum_{i=1}^{n} \left[\frac{v\hat{\mu}}{v}\frac{-v}{\hat{\mu}^2} + \frac{vx_i}{\hat{\mu}^2}\right] \\
=&\sum_{i=1}^{n} \left[-\frac{v}{\hat{\mu}} + \frac{vx_i}{\hat{\mu}^2}\right] \\
=&-\frac{nv}{\hat{\mu}} + \sum_{i=1}^{n} \frac{vx_i}{\hat{\mu}^2} = 0
\end{aligned}
$$

Simplifying the last equation,

$$\frac{1}{n}\sum_{i=1}^{n} x_i = \hat{\mu}$$

Then,

$$\hat{\mu}_{\mathrm{ML}} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

2.3 ML Estimate for Shape Parameter

Following the same procedure as above but for the parameter v,

$$
\begin{aligned}
&\sum_{i=1}^{n} \nabla_{\hat{v}} \log \left[\left(\frac{\hat{v}}{\mu} \right)^{\hat{v}} \frac{x^{\hat{v}-1}}{\Gamma(\hat{v})} \exp\left(-\frac{\hat{v}x}{\mu} \right) \right] \\
=&\sum_{i=1}^{n} \nabla_{\hat{v}} \left[\hat{v} \log \frac{\hat{v}}{\mu} + (\hat{v}-1)\log x - \log \Gamma(\hat{v}) - \frac{\hat{v}x}{\mu} \right] \\
=&\sum_{i=1}^{n} \log \frac{\hat{v}}{\mu} + \hat{v}\left(\frac{\mu}{\hat{v}} \right)\left(\frac{1}{\mu} \right) + \log x - \nabla_{\hat{v}} \log\left(\Gamma(\hat{v}) \right) - \frac{x}{\mu} \\
=&\sum_{i=1}^{n} \log\left(\frac{\hat{v}x}{\mu} \right) - \left(\frac{x}{\mu} - 1 \right) - \nabla_{\hat{v}} \log\left(\Gamma(\hat{v}) \right) \\
=&\sum_{i=1}^{n} \log\left(\frac{\hat{v}x}{\mu} \right) - \left(\frac{x}{\mu} - 1 \right) - \phi(\hat{v})
\end{aligned}
$$

Proven.

3 Bayes-Optimal Classifier

The Bayes-optimal classifier given is defined as

$$f_0(\mathbf{x}) := \arg\max_{y \in [\mathbf{K}]} P(y|\mathbf{x})$$

Since $P(y|\mathbf{x}) = P(\mathbf{x}, y)P(\mathbf{x})$ and $P(\mathbf{x}, y)$ is the true joint density, then $f_0(\mathbf{x})$ must pick y correctly given $\mathbf{x}$. Now

$$R(f|\mathbf{x}) := \sum_{y \in [\mathbf{K}]} L^{0-1}(y, f(\mathbf{x}))P(y|\mathbf{x})$$

Let's say we have a data point $\mathbf{x}_0$. For the classifier given, the 0-1 loss function L^{0-1} will evaluate to 0 for the right y_0 and 1 for all other inputs.

Furthermore, $P(y_0|\mathbf{x}_0) = \max_{y \in [\mathbf{K}]} P(y|\mathbf{x}_0)$. That means for any $\mathbf{x}$, its corresponding $P(y|\mathbf{x})$ is the largest among the possible $P(y|\mathbf{x}) \forall y \in [\mathbf{K}]$. Since the loss function evaluates to 0 for this particular conditional, the largest $P(y|\mathbf{x})$ is excluded for every $\mathbf{x}$.

Since a classifier can only exclude 1 of the $P(y|\mathbf{x})$, and the largest one is excluded by the Bayes-optimal classifier, it minimizes $R(f|\mathbf{x})$. This is further so since any other classifier f would exclude a non-maximum term, leading to a higher $R(f|\mathbf{x})$ for the particular $\mathbf{x}$, and would not minimize $R(f|\mathbf{x})$.

Since the Bayes-optimal classifier minimizes $R(f|\mathbf{x})$ for every $\mathbf{x}$, the result for $R(f)$ follows by monotonicty of the integral.

4 Risk

4.1 Functions Required to Calculate Risk

$$R(f) := E_p[L(y, f(\mathbf{x}))] = \int L(y, f(\mathbf{x}))p(\mathbf{x}, y)d\mathbf{x}dy$$

We need the following functions

- Labelled training data $(\mathbf{x}_i, y_i)$. Needed in loss function to calculate risk.
- A classifier $f(x)$. Needed to predicts label $\hat{y}_i$ for $\mathbf{x}_i$
- A loss function L. Needed to measure error of classifiers.
- Join distribution $p(\mathbf{x}, y)$, needed to calculate probability of error to weigh errors happening more frequently more expensively.

4.2 Risk $R(f)$ or Empirical Risk $\hat{R}_n(f)$

In most if not all cases the exact joint distribution $p(\mathbf{x}, y)$ is not known, hence empirical risk is used as a plug-in estimate of $R(f)$.

4.3 $\lim_{n\to\infty} \left|R(f) - \hat{R}_n(f)\right|$ for iid Draws from True Underlying Distribution

$\hat{R}_n(f)$ is a sample of the true distribution. As $n \to \infty$, by the law of large numbers, the empirical risk approaches the true $R(f)$.

4.4 Range of $R(f)$

$R(f) \in [0, 1]$. $R(f)$ is the probability of the classifier making a mistake (over the training data).

4.5 $E(R(f^1))$ under 01 Loss

Random choice of f^1 causes 0-1 loss $R(f^1)$ to be uniformly distributed over $[0, 1]$, which in turn causes $E(R(f^1)) = 0.5$.

4.6 $E(R(f^2))$ under Perceptron

A better classifier makes fewer errors. That means L will produce 0s more often. Then, $E(R(f^2)) < 0.5$, hence smaller than $E(R(f^1))$.