IntroML - Lecture Notes Week 11

Ruben Schenk, ruben.schenk@inf.ethz.ch

July 19, 2022

0.1 Bayesian Decision Theory

0.2 Introduction

We have seen how we can interpret supervised learning as *fitting probabilistic models* $p(y \mid x)$ of the data. Next, we'll see how we can use the estimated models to inform decisions.

Suppose we have estimated a logistic regression model (e.g. for spam filtering), and obtain $p(Y = \text{spam} \mid X)$. Furthermore suppose we have three actions: Spam, NotSpam and AskUser.

Bayesian decision theory, a.k.a. maximum expected utility principle, works as follows. Given:

1. Conditional distribution over labels $p(y \mid x)$ for $y \in Y$
2. Set of actions A
3. Cost function $C : Y \times A \to \mathbb{R}$

The Bayesian decision theory recommends to pick the action that *minimizes the expected cost:*

$$a^* = \arg\min_{a \in \mathcal{A}} \mathbb{E}_y[C(y,\, a) \mid x]$$

If we had access to the true distribution, this decision implements the Bayesian optimal decision (i.e. minimizes the expected cost over all decision rules). In practice, we can only estiamte it, e.g. via logistic regression $p(y \mid x)$.

Recall the logistic regression:

- *Learning:*

$$\begin{aligned}\hat{w} &= \arg\min_w \sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i)) + \lambda ||w||_2^2 \\ &= \arg\max_w P(w \mid x_1, ..., x_n,\, y_1, ..., y_n)\end{aligned}$$

- *Classification:*

$$P(y \mid x,\, w) = \frac{1}{1 + \exp(-y \hat{w}^T x)}$$

0.3 Optimal Decisions for Classification

Consider the following setup:

- Est. conditional dist.: $p(y \mid x) = \text{Ber}(y;\, \sigma(f(x)))$, e.g. $f(x) = w^T x$
- Action set: $A = \{+1,\, -1\}$
- Cost function: $C(y,\, a) = [y \neq a]$

Then the action that minimizes the expected cost $a^* = \arg\min_{a \in A} \mathbb{E}_y[C(y,\, a) \mid x]$ is the most likely class:

$$a^* = \arg\max_y p(y \mid x) = \text{sign}(f(x))$$

0.4 Asymmetric Costs

Consider the same setup as above, but with an **asymmetric cost,** i.e. the cost function is given by:

$$C(y,\, a) = \begin{cases} c_{FP} & \text{if } y = -1 \text{ and } a = +1 \\ c_{FN} & \text{if } y = +1 \text{ and } a = -1 \\ 0 & \text{otherwise} \end{cases}$$

Then the action that minimizes the expected cost $a^* = \arg\min_{a \in A} \mathbb{E}_y[C(y,\, a) \,|\, x]$ is given by the following costs:

- $c_+ = \mathbb{E}_y[c(y,\, +1) \,|\, x] = p(y = -1 \,|\, x) \cdot c_{FP} + p(y = +1 \,|\, x) \cdot 0 = (1 - p) \cdot c_{FP}$
- $c_- = \mathbb{E}_y[c(y,\, -1) \,|\, x] = p \cdot c_{FN} + (1 - p) \cdot 0 = p \cdot c_{FN}$

Therefore, we predict $+1$ if:

$$c_+ \leq c_- \Rightarrow (1 - p) \cdot c_{FP} \leq p \cdot c_{FN} \Rightarrow p \geq \frac{c_{FP}}{c_{FP} \cdot c_{FN}}$$

0.5 Classification with Abstention

Consider the same setup as above, but with an **abstention,** i.e. the action set is given by $A = \{+1,\, -1,\, D\}$ and the cost function by:

$$C(y,\, a) = \begin{cases} [y \neq a] & \text{if } a \in \{+1,\, -1\} \\ c & \text{if } a = D \end{cases}$$

Then the action that minimizes the expected cost $a^* = \arg\min_{a \in A} \mathbb{E}_y[C(y,\, a) \,|\, x]$ is given by:

$$a^* = \begin{cases} y & \text{if } P(y \,|\, x) \geq 1 - c \\ D & \text{otherwise} \end{cases}$$

In other words, we pick the most likely class only if it's confident enough.

0.6 Optimal Decisions for LS Regression

Consider the following setup:

- Est. conditional dist.: $p(y,\, |\, x) = \mathcal{N}(y;\, f(x),\, \sigma^2)$, e.g. $f(x) = w^T x$
- Action set: $A = \mathbb{R}$
- Cost function: $C(y,\, a) = (y - a)^2$

Then the action that minimizes the expected cost $a^* = \arg\min_{a \in A} \mathbb{E}_y[c(y,\, a) \,|\, x]$ is the conditional mean:

$$a^* = \mathbb{E}[y \,|\, x] = \int p(y \,|\, x)\, dy = f(x)$$

Example: We might also consider an asymmetric cost for regression. The setup is the same with a different cost function: $c(y,\, a) = c_1 \max(y - a,\, a) + c_2 \max(a - y,\, 0)$. Then the action that minimizes the expected cost is:

$$a^* = f(x) + \sigma \cdot \Phi^{-1}(\frac{c_1}{c_1 + c_2}),$$

where $\Phi(z) = \int_{-\infty}^{\infty} \mathcal{N}(z;\, 0,\, 1)\, dz$.

0.7 Active Learning: Uncertainty Sampling

In general, labels are expensive (since we need to ask an expert). For this reason, we somehow want to minimize the number of labels. One simple strategy is to always pick the example that we are most uncertain about (**uncertainty sampling**):

Given a pool of unlabeled examples $D_U = \{x_1, ..., x_n\}$, we also maintain a labeled dataset D_L which is initially empty. For $t = 1, 2, 3, ...$:

1. Estimate $p(y \mid x)$ given the current data D_L
2. Pick the unlabeled example that we are most uncertain about (highest entropy)
$$i_t \in \arg\min_{x \in D_U} H(p(y \mid x))$$
3. Query the label y_{i_t} and set $D_L \leftarrow D_L \cup \{(x_{i_t}, y_{i_t})\}$

It is important to note that active learning *violates i.i.d. assumption!* We can get stuck with bad models.

0.8 Summary

Bayesian decision theory provides a principled way to derive decision rules from conditional distributions. Standard rules arise as special cases:

- Linear regression: $w^T x$
- Logistic regression: $\text{sign}(w^T x)$

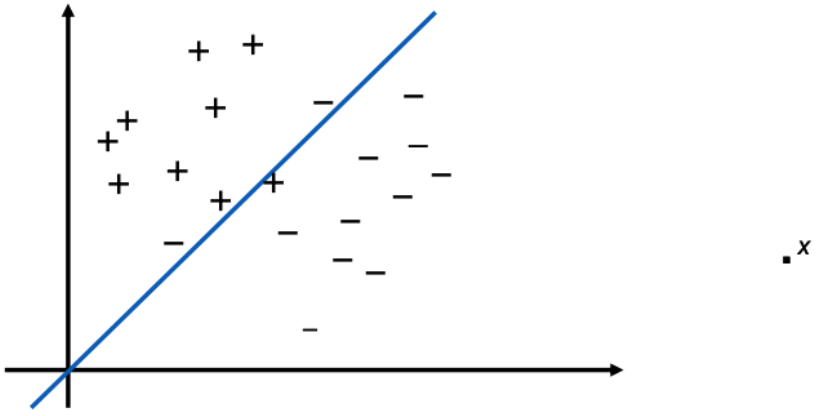
Finally, in summary for learning through MAP inference:

1. Start with statistical assumptions on data: Data points modeled as i.i.d.
2. Choose a likelihood function, e.g. Gaussian, student-t, logistic, exponential, etc. This provides the loss function
3. Choose a prior, e.g. Gaussian, Laplace, etc. This provides the regularizer.
4. Optimize for MAP parameters
5. Choose hyperparameters (i.e. variance, etc.) through cross-validation.
6. Make predictions via Bayesian decision theory

1 Generative Modeling

1.1 Introduction

Consider the following case:



What will logistic regression predict for data point x? The problem is that logistic regression can be *overconfident about labels for outliers.*

We can compare discrimiantive and generative models:

- *Discriminative models* aim to estimate $p(y \mid x)$
- *Generative models* aim to estiamte the joint distribution $p(y,\, y)$

We can derive a conditional from a joint distribution, but not vice versa!

1.2 General Approach

The general approach to **generative modeling** for classification is:

- Estimate prior on labels: $p(y)$
- Estimate conditional distribution for each class y: $p(x \mid y)$
- Obtain predictive distribution using Bayes' rule: $p(y \mid x) = \frac{1}{Z} p(y) p(x \mid y)$

Some notes on generative modeling:

1. Generative modeling attempts to infer the process, according tow hich examples are generated
2. The first generated class label is $p(y)$
3. Then, generate features given the class $p(x \mid y)$

1.3 Naive Bayes

For example, we might consider the **Naive Bayes Model.** In this model, we model class labels as generated from the categorical variable:

$$P(Y = y) = p_y \quad y \in \mathcal{Y} = \{1, ..., c\}$$

We model features as conditionally independent given Y,

$$P(X_1, ..., X_d \mid Y) = \prod_{i=1}^{d} P(X_1 \mid Y),$$

i.e. given the class label, each features is generated independently of the other features. However, we still need to specify the feature distributions $P(X_i \mid Y)$.

In order to predict label y for a new point x, we use:

$$P(y \mid x) = \frac{1}{Z} P(y) P(x \mid y) \quad Z = \sum_{y} P(y) P(x \mid y)$$

1. 1. PRedict using Bayesian decision theory.
2. E.g. in order to minimize misclassification error, we predict:

$$y = \arg\max_{y'} P(y' \mid x)$$

The **Gaussian Naive Bayes Classifiers** work as follows. We do the learning given some data $D = \{(x_1,\, y_i), ..., (x_n,\, y_n)\}$:

1. MLE for class prior: $P(Y = y) = \hat{p}_y = \frac{\text{Count}(Y=y)}{n}$

2. MLE for feature distribution: $P(x_i \mid y) = \mathcal{N}(x_i;\, \hat{\mu}_{y,i},\, \sigma^2_{y,i})$

$$\hat{\mu}_{y,i} = \frac{1}{\text{Count}(Y = y)} \sum_{j; y_j = y} x_{j,i}$$

$$\sigma^2_{y,i} = \frac{1}{\text{Count}(Y = y)} \sum_{j: y_i = y} (x_{j,i} - \hat{\mu}_{y,i})^2$$

Then, the prediction given some new point x is:

$$y = \arg\max_y P(y' \mid x) = \arg\max_{y'} P(y') \prod_{i=1}^{d} P(x_i \mid y')$$

1.4 Decision Rules for Binary Classification

We want to predict $y = \arg\max_{y'} P(y' \mid x)$. For binary tasks, i.e. $c = 2$, $y \in \{+1, -1\}$, this is equivalent to:

$$y = \text{sign}\Big(\log \frac{P(Y = +1) \mid x}{P(Y = -1) \mid x}\Big)$$

The function $f(x) = \log \frac{P(Y=+1) \mid x}{P(Y=-1) \mid x}$ is called **discriminant function.**

Example: Let us consider the special case of Gaussian Naive Bayes with $c = 2$ and shared variance. We are given $p(x \mid y) = \prod_i \mathcal{N}(x_i;\, \mu_{y,i},\, \sigma^2)$. We want $f(x) = \log \frac{P(Y=+1) \mid x}{P(Y=-1) \mid x}$.

In case of shared variance, GNB produces a linear classifier:

$$f(x) = w^T x + w_0$$

Hereby:

$$w_0 = \log \frac{\hat{p}_+}{1 - \hat{p}_+} + \sum_{i=1}^{d} \frac{\hat{\mu}^2_{-,i} - \hat{\mu}^2_{+,i}}{2\hat{\sigma}^2_i}$$

$$w_i = \frac{\mu_{+,i} - \mu_{-,i}}{\sigma^2_i}$$

The corresponding class distribution

$$P(Y = 1 \mid x) = \frac{1}{1 + \exp(-f(x))} = \sigma(w^T x + w_0)$$

has the *same form as logistic regression.* If model assumptions are met, GNB will make the same predictions as logistic regressions.

Issues with Naive Bayes models:

- Conditional independence assumption means that features are generated independently given the class label
- If there is conditional correlation between class labels, then this assumption is violated
- Due to conditional independence assumption, predictions can become overconfident (very close to 0 or 1)
- This might be fine if we care about most likely class only, but not if we want to use probabilities for making decisions (e.g. asymmetric losses etc.)

1.5 Gaussian Bayes Classifiers

Lets consider general Gaussian Bayes classifiers:

- Model class label as generated from categorical variable: $P(Y = y) = p_y$ and $y \in \mathcal{Y} = \{1, ..., c\}$
- Modeal features as generated by multivariate Gaussian: $P(x \mid y) = \mathcal{N}(x;\, \mu_y,\, \Sigma_y)$

The MLE for Gaussian Bayes classifiers with given data set $D = \{(x_1,\, y_1), ..., (x_n,\, y_n)\}$ is given by:

- MLE for class label distribution

$$P(Y = y) = \hat{p}_y \quad \hat{p}_y = \frac{\text{Count}(Y = y)}{n}$$

- MLE for feature distribution

$$P(x \mid y) = \mathcal{N}(x;\, \hat{\mu}_y,\, \hat{\Sigma}_y)$$

$$\hat{\mu}_y = \frac{1}{\text{Count}(Y = y)} \sum_{i:y_i=y} x_i$$

$$\hat{\sigma}_y = \frac{1}{\text{Count}(Y = y)} \sum_{i:y_i=y} (x_i \hat{\mu}_y)(x_i - \hat{\mu}_y)^T$$

1.6 Fisher's Linear Discriminant Analysis LDA

Suppose we fix $p = 0.5$ and that the covariances are equal, i.e. $\hat{\Sigma}_- = \hat{\Sigma}_+ = \hat{\Sigma}$. Then the discriminant function simplifies to:

$$f(x) = x^T \hat{\Sigma}^{-1}(\hat{\mu}_+ - \hat{\mu}_-) + \frac{1}{2}(\hat{\mu}_-^T \Sigma^{-1} \hat{\mu}_- - \hat{\mu}_+^T \Sigma^{-1} \hat{\mu}_+) = x^T w + w_0$$

Under these assumptions, we predict:

$$y = \text{sign}(f(x)) = \text{sign}(w^T x + w_0)$$

This linear classifier is called **Fisher's linear discriminant analysis.**