Bayes Decision Theory

Alan Yuille

Feb 5 2024

Bayes Introduction

- This lecture introduces Bayes and Bayes Decision Theory
- Bayes Decision Theory
- Empirical Risk
- Critique of Bayes
- Bayes in the Big

Bayes decision theory

- Bayes decision theory (BDT) is a framework for making optimal decisions in the presence of uncertainty. .
- The theory contains three ingredients: (I) A *probability distribution* $P(x, y)$ over the input $x \in \mathcal{X}$ and output $y \in \mathcal{Y}$. (II) A set of *decision rules* $\{\alpha(); \alpha \in \mathcal{A}\}$ where $\alpha(x) \in \mathcal{Y}$. (III) A *loss function* $L(\alpha(x); y)$, which is the cost of making decision $\alpha(x)$ if the real decision should be y.
- The *risk* is specified by the expected loss function $R(\alpha) = \sum_{x,y} P(x, y)L(\alpha(x), y)$.
- The optimal decision is *Bayes rule* $\hat{\alpha} = \arg\min_{\alpha \in \mathcal{A}} R(\alpha)$ minimizing the risk yielding the *Bayes risk* $\min_\alpha R(\alpha) = R(\hat{\alpha})$ (Caveat).

Likelihoods, priors, and posteriors

- By basic probability theory we can re-express the *joint distribution* $P(x, y)$ in two different ways: (I) $P(x, y) = P(x|y)P(y)$, where $P(x|y)$ is the *conditional distribution* of x conditioned on y. (II) Similarly $P(x, y) = P(y|x)P(x)$. By equating $P(x|y)P(y) = P(y|x)P(x)$ we derive *Bayes Theorem* $P(y|x) = P(x|y)P(y)/P(x)$.
- The goal of BDP is to estimate y from x: (I) $P(x|y)$ is the *likelihood function* of y and specifies what we know about y given the observation x. (II) $P(y)$ specifies the prior knowledge of y independent of the observation. (III) $P(y|x)$ is the *posterior distribution* of y after making the observation x, combining the likelihood function and the prior/
- Priors are used if they can be estimated objectively. Sometimes they are criticized as subjective. We will return to this later.

Bayes Rule and special cases

- We can express the expected risk as $\sum_x P(x) \sum_y P(y|x) L(\alpha(x), y)$. Hence the Bayes rule $\hat{\alpha}(.)$ can be expressed as $\hat{\alpha}(x) = \arg\min \sum_y P(y|x) L(\alpha(x), y)$.
- First special case. Suppose the loss function penalizes all errors equally with $Ł(\alpha(x), y) = -\delta(y - \alpha(x))$, where $\delta()$ is the Dirac delta function, and $\hat{\alpha}(x) = \arg\max_y P(y|x)$. This is the *maximum a posteriori* (MAP) estimate of y.
- Second special case, Suppose in addition that the prior $P(y)$ is the uniform distribution. In this case, $\hat{\alpha}(x) = \arg\max_y P(x|y)$ which is the *maximum likelihood* (ML) estimate of y.

Bayes rule for binary decisions

- The binary case $y \in \{-1, 1\}$ illustrates the trade off between different types of errors. We call $y = 1$ the *target* and $y = -1$ as the *distractor*.
- For a decision rule $\alpha(x)$, we define (x, y) to be a *false positives* if $\alpha(x) = 1$ and $y = -1$, We define (x, y) to be a *false negative* if $\alpha(x) = -1$ and $y = 1$.
- In other word, a false positive occurs if we predict the input x to be the target when it is a distractor. A false negative occurs if the decision rule predicts the signal to be a distractor but instead it is a target.
- This situation comes up frequently in practice for example if we are trying to detect a disease. Ideally we would like a decision rule which is always correct and has either *true positive* — $\alpha(x) = 1$ and $y = 1$ – or true negatives $\alpha(x) = -1$ and $y = -1$. In practice we need to choose a loss function that trades offs the false negatives with the false positives.

Bayes rule for binary decisions

- For binary decision problems $y \in \{\pm 1\}$, the loss function is usually chosen to pay no penalty if the correct decision is made (i.e., $\alpha(x) = y$) but has a penalty F_p for *false positives*, where $y = -1$ but $\alpha(x) = 1$, and F_n for *false negatives*, where $y = 1$ but $\alpha(x) = -1$ is $y = -1$.
- It follows, see next slide, that we can express the Bayes rule in terms of a log-likelihood ratio test $\log \frac{P(x|y=1)}{P(x|y=-1)} > T$, where T depends on the prior $p(y)$ and the loss function $L(\alpha(x), y)$.
- This is why Bayesian edge detection (previous lecture) reduces to thresholding the log-likelihood ratio of the probabilities that the features are generated by edges $y = 1$ or background $y = -1$.

Bayes rule (III)

- More specifically, the Bayes risk is $R(\alpha) = \sum_x P(x) \sum_y L(\alpha(x), y) P(y|x)$. Then we divide the data (x, y) into four sets:
- (1) the *true positives* $\{(x, y) : \text{ s.t. } \alpha(x) = y = 1\}$;
- (2) the *true negatives* $\{(x, y) : \text{ s.t. } \alpha(x) = y = -1\}$;
- (3) the *false positives* $\{(x, y) : \text{ s.t. } \alpha(x) = 1, y = -1\}$;
- (4) the *false negatives* $\{(x, y) : \text{ s.t. } \alpha(x) = -1, y = 1\}$.
- These four cases correspond to loss function values $L(\alpha(x) = 1, y = 1) = T_p$, $L(\alpha(x) = -1, y = -1) = T_n$, $L(\alpha(x) = 1, y = -1) = F_p$, $L(\alpha(x) = -1, y = 1) = F_n$.
- Then the best decision rule $\hat{\alpha_T}(.)$ can be expressed as.

$$\log \frac{P(x|y=1)}{P(x|y=-1)} > \log \frac{T_n - F_p}{T_p - F_n} + \log \frac{P(y=-1)}{P(y=1)}.$$

- The intuition is that the evidence in the log-likelihood must be bigger than our prior biases while taking into account the penalties paid for different types of mistakes.

Bayes rule (IV)

The results on Bayesian edge detection and texture classification can be derived from decision theory.

For Bayesian edge detection, the prior $P(y)$ specify the probability that an image patch contains an edge (empirically $P(y = 1) \approx 0.05$ and $P(y = -1) \approx 0.95$). The prior probability that a pixel is an edge is very small. If the loss function penalizes false positives and false negatives equally the best decision rule is to estimate that every pixel is background, because this is successful ninety five percent of the time. The loss function must be selected so that failing to detect an edge is penalized much larger than misclassifying a background pixel as an edge.

The loss function should be chosen to specify the cost of making different types of mistakes. For texture classification, the variable y takes values in a set $\mathcal{Y}$, which is called a multiclass decision. The same theory applies to tasks for which we need to make a set of related but nonlocal decisions.

Signal detection theory (I)

We now show that an important special case of *signal detection theory* (Green & Swets, 1966) – often used as a framework to model how humans make decisions when performing visual, auditory, and other tasks – can be obtained as a special case of Bayes decision theory.

We consider the two class case, where $y \in \{\pm 1\}$, and suppose that the likelihood functions are specified by Gaussian distributions, $P(x|y) = \frac{1}{\sqrt{2\pi}\sigma_y} \exp\{-(x-\mu_y)^2/(2\sigma_y^2)\}$, which differ by their means (μ_1, μ_{-1}) and their variances $(\sigma_1^2, \sigma_{-1}^2)$. The Bayes rule can be expressed in terms of the log-likelihood ratio test:

$$\hat{\alpha}(x) = \arg\max_y y\{-(x-\mu_1)^2/(2\sigma_1^2) - \log\sigma_1 + (x-\mu_{-1})^2/(2\sigma_{-1}^2) + \log\sigma_2 - T\}.$$

Signal detection theory (II)

- This decision rule requires determining whether the data point x is above or below a quadratic polynomial curve in x. In the special case when the standard deviations are identical $\sigma_1^2 = \sigma_2^2$ (so we drop the subscripts $_{1,-1}$), the decision is based only on whether the data point x satisfies:

$$2x(\mu_1 - \mu_{-1}) + (\mu_1^2 - \mu_2^2) < 2T\sigma^2$$

- This special case, with $\sigma_1^2 = \sigma_{-1}^2$, is much studied in signal detection theory (Green & Swets, 1966). It means that the decision is based on a single function $d' = \frac{\mu_1 - \mu_{-1}}{\sigma}$. This quantity is used to quantify human performance for psychophysical tasks.
- Historically Signal Detection theory was one of the first scientific applications of Bayes Decision Theory, which was developed during WW2 for applications like decrypting codes (e.g., the Enigma machine) or for detecting the enemy using radar. There were big debates in that Statistics community about the value of Bayes decision theory. One of the strongest advocates for BDT was I.J. Good who have worked with Turing on decypting codes in WW2 and who, like Turing, wrote papers about AI.

Learning the Probability Distributions

- Bayes Decision Theory assumes that we know the probability distributions $P(x|y)$ and $P(y)$. Or, at least, the posterior distribution $P(y|x)$.
- These distributions should be learnt from data $\mathcal{X} = \{x_i : i = 1, ..., N\}$. BDT can be applied to learn these distributions $P(y), P(x|y)$. For example, to learn a probability distribution of x we select a parameterized probability distribution $P(x|\lambda)$ and then estimate the parameters λ.
- The ML estimate $\hat{\lambda}$ is given by $\arg\max_\lambda \prod_{i=1}^{N} P(x_i|\lambda)$ or, equivalently, by $\hat{\lambda} = \arg\min_\lambda (-1) \sum_{i=1}^{N} \log P(x_i|\lambda)$.
- We can also estimate $\hat{\lambda}$ by MAP by introducing a prior $P(\alpha)$. This reduces to $\hat{\lambda} = \arg\min_\lambda (-1)\{\log P(\lambda) + \sum_{i=1}^{N} \log P(x_i|\lambda)$. If the number N of training data is large then the prior will have little effect and can be ignored. If N is small them the prior can sometimes have a big effect.

The Empirical Risk

- An alternative approach, used in much of machine learning, is to learn the decision rule $\alpha(.)$ directly from the data $\mathcal{D}_N = \{(x_i, y_i) : i = 1, ..., N\}$. This differs from using the data to learn the probability distribution and then computing the decision rule by minimizing the Bayes risk.
- This approximates $R(\alpha) = \sum_{x,y} L(\alpha(x), y)P(x, y)$ by $R_{emp,\mathcal{D}_n}(\alpha) = 1/N \sum_{i=1}^{N} L(\alpha(x_i).y_i)$. In the limit as $N \mapsto \infty$ the empirical risk $R_{emp,\mathcal{D}_n} \mapsto R(\alpha)$. This assumes that the $\{(x_i, y_i)\}$ are identically independently distributed (iid) samples from $P(x, y)$. Then we estimate $\hat{\alpha}()$ by minimizing $R_{,\mathcal{D}_n}(\alpha)$.
- This is attractive because it avoids the need for learning the probability distributions of the data. If the final goal is to find the best decision rule then why not estimate it directly instead of first estimating the probability distributions? Note: my view is that learning the likelihood and the prior are better for more complex situations, as illustrated by domain transfer for edge detection.

The Empirical Risk: PAC theory

- This approach was used by Support Vector Machines (SVMs) which was the most popular ML used in computer vision before neural networks. SVM argues that using the data to estimates probabilities is wasteful and it is better to concentrate directly on the decision boundaries. For SVMs this meant using the data to learn the decision boundaries, specified by the *support vectors*.
- This approach comes with mathematical theories like Probably Approximate Correct (PAC) which gives upper bounds of the amount of data needed for the estimator to be close to $\arg\min R(\alpha)$ depending on the capacity of the decision rule. This guarantees that, with high probability, the decision rule will *generalize* to new data that the rule has not been learn from, provided it comes from the same source (i.e. iid from $P(x, y)$). But these theoretical bounds are not tight and rarely useful in practice.
- PAC theory, and more practical considerations, suggest that you need more training data than the capacity of the set of classifiers $\mathcal{A}$. But although the capacity is an important conceptual concept it is very hard to measure, expect for very simple decision rules. This is too complex to discuss here. For some types of decision rules, like some types of deep networks, the capacity is "elastic" and the decision rules generalize well if there is only a small amount of data and perform better when there is more (inconsistent with the idea that capacity is fixed). For other classes of rules, the capacity can be reduced by regularizing them,

The Empirical Risk and Learning Probabilities

- Suppose the decision rules can be expressed as $\hat{\alpha}(x) = \arg max_y P(y|x;\alpha)$ where $P(y|x;\alpha)$ is a family of probability distributions parameterized by α. This corresponds to convolutional neural networks.
- If the loss function $L(\alpha(x), y) = -\log P(y|x:\alpha)$, where α is the parameter of a distribution, then we obtain the loss used by convolutional neural networks (CNNs). This can also be obtained from probabilistic learning where we seek to learn the posterior distribution $P(y|x)$ directly, as will be discussed later in the course.
- Many (most) of the loss functions used to train CNNs relate directly to BDT. E.g., CNNs for edge detection use loss functions to penalize false negatives 9failure to detect an edge) for the same reasons they are used for Bayesian edge detection.

The Limits of BDP

- BDT measure performance by average case. Why not be more ambitious and measure it by worst case? Or characterize the stimuls space into regions where the decision rule works well and regions where it does not?
- Average case can be problematic particularly if the datasets are baised as datasets always are.
- Priors can be useful but can cause biases.
- All these issues will be discussed further later in the course.
- The key ideas of Bayes – generative distributions and priors – are very important. BDT is a good start but is not nearly enough.