Decision Theory and Loss Fuction

Khanh Nguyen

June 04, 2015

What is decision theory? In what contexts does it apply to?

Imagine the nature chooses an unknown label y, which is unknown to us, and only presents to us the observation x generated from that label. We are asked to take an action a (e.g. guess the label of the observation). Each action we will incur a loss, calculated from loss function $L(y, a)$. This function measures how good is our action (thus zero loss is best). For example, we can use a zero-one loss $I\{y \neq a\}$, or a quadratic loss $(y - a)^2$.

Eventually, after scoring the actions, we have to pick one based on their losses. This is where *decision theory* comes in, defines a policy to choose an optimal action based on a loss function. In general, the policy is to

1 Decision theory

This section presents decision theory of views about statistics: Bayesian and frequentist. The fundamental difference between these two is that Bayesian statistics views the model parameters as random variables and tries to model their distribution conditional on observed data, whereas frequentist statistics considers model parameters as fixed unknown quantities and estimate them by drawing samples of data.

1.1 Bayes decision theory

Our goal is to choose the action that minimizes the expected loss:

$$\hat{a} = arg \min_a E_{\sim P(y|x)}[L(y, a)] = \sum_y L(y, a) P(y|x)$$

Zero-one loss

If $L(y, a) = I\{y \neq a\}$ (as in accuracy measurements), then:

$$\hat{a} = arg \min_a \sum_y I\{y \neq a\} P(y|x) = arg \min_a [1 - P(a|x)] = arg \max_a P(a|x)$$

The optimal action is to return the most probable class.

Supervised learning

In this setting, the action is a parameter distribution estimator, δ, of the model parameters, θ [1]. Denote by $\delta(x)$, the label of the observation x inferred from the estimate. We define the loss function as follows:

$$L(\theta, \delta) = E_{\sim P(x,y|\theta)}\left[l(y, \delta(x))\right] = \sum_x \sum_y l(y, \delta(x))p(x, y|\theta)$$

We want to minimize this posterior expected loss:

$$E_{\sim P(\theta|D)}\left[L(\theta, \hat{\theta})\right] = \int P(\theta|D)L(\theta, \hat{\theta})d\theta$$

1.2 Frequentist decision theory

As mentioned, frequentists treat the model paramters as an unknown fixed value. We denote this quantity by θ^* to distinguish from the Bayesian view. The action is defined as an estimator of the model parameters from a dataset $\delta(D)$

The frequentist expected loss (or risk) is defined as follows:

$$R(\theta^*, \delta) = E_{\sim P(D|\theta^*)}\left[L(theta^*, \delta(D))\right] = \int L(\theta^*, \delta(D))p(D|\theta^*)dD$$

This is totally contrary to the Bayesian posterior expected loss, which integrates over θ instead of D. Notice that the frequentist expected loss cannot be computed since θ^* is unknown. One solution for this problem is to put a prior on θ^*, resulting in a *Bayes estimator*:

$$R_B(\delta) = E_{\sim P(\theta^*)]}\left[R(\theta^*, \delta)\right] = \int R(\theta^*, \delta)p(\theta^*)d\theta^*$$

Theorem: *A Bayes estimator can be obtained by minimizing the posterior expected loss for each x.*

This is saying that choosing an action based on minimizing a Bayes estimator is equivalent to choosing an action based on minimizing a posterior expected loss.

In *supervised learning*, the frequentist expected risk becomes [2] :

$$R(\theta^*, \delta) = E_{\ P(x,y|\theta^*)}\left[L(y, \delta(x)\right] = \sum_x \sum_y L(y, \delta(x))P(x, y|\theta^*)$$

Computing the expected risk is intractable since we have to sum over all possible pairs (x, y) from an unknown distribution. One way to approximate it is to calculate the *empirical risk* on an observed data D [3] :

[1] Remember that this is a random variable.

[2] Although it looks very similar to the Bayesian supervised learning's loss function, it computes a quantity whereas the Bayesian one is a function of θ.

[3] For unsupervised learning, since there is no label y, the empirical risk is defined as $\frac{1}{|D|}\sum_{x_i \in D} L(x_i, \delta(x_i))$

$$R(\theta^*, \delta) \approx R_{emp}(D) = \frac{1}{|D|} \sum_{(x_i, y_i) \in D} L(y_i, \delta(x_i)) \tag{1}$$

Notice that if the loss function is the negative log-likelihood $-P(y|x, \theta)$, minimizing the risk turns into maximizing the likelihood of the observed data, an idea that is widely adopted in machine learning.

2 Loss functions for energy-based models

This section is based on "A Tutorial on Energy-Based Learning", written by Yan Lecunn, appeared in "Predicting Structured data" (G. Bakir, T. Hofman, B. Schölkopf, A. Smola, B. Taskar), MIT Press, 2006. It is written for the self-studying purpose only.

2.1 Energy-based models

Energy-based models capture dependencies between variables by associating an energy level to each configuration of the variables. Inspired by Physics, we would prefer low energy configurations since they are more "stable". Hence, learning in these models involves finding an energy function that associates low energies to correct configurations of the free variables, and higher energies to incorrect configurations. A loss functional is required to measure the quality of values of the energy function.

The energy function usually has the form $E(w, x, y)$ where x is an observation, y is the corresponding label (can be continuous), and w is the interaction strengths between the variables within the models (the size of w varies depending on the model).

Let Y denote the set of labels, the (Gibbs) distribution of an energy-based model is written as:

$$P(y|x) = \frac{e^{-\beta E(w,x,y)}}{\int_{y' \in Y} e^{-\beta E(w,x,y')}}$$

where β is a positive constant.

A loss functional, $\mathcal{L}(w, x, y)$, for most cases, is defined as follows:

$$\mathcal{L}(w, x, y) = \frac{1}{N} \sum_{i=1}^{n} L(y_i, E(w, x_i, Y)) + R(w) = \frac{1}{N} \sum_{i=1}^{n} \sum_{y \in Y} L(y_i, E(w, x_i, y)) + R(w)$$

Except for the regularizer term, $R(w)$, this loss functional is a special case of Equation 1. We will discuss different forms of the loss function $L(y, E(w, x, Y))$.

2.2 Loss functions

Energy loss

$$L_{energy}(y, E(w,x,Y)) = E(w,x,y)$$

This loss only pushes down the energy of the desired label but does not push up other labels' energies. It only work with models that are designed in a way such that pushing down on the desired answer will automatically push up others' energies (e.g. mean squared error).

Generalized perceptron loss

$$L_{perception}(y, E(w,x,Y)) = E(w,x,y) - \min_{y' \in Y} E(w,x,y')$$

This loss is always positive, since the second term is a lower bound on the first term. It pushes down on the data label, while pulling up on the energy of the label inferred by the model. Its major drawback is that it cannot create a gap between the correct answer and the incorrect ones like the margin loss, which will be discussed next.

Generalized margin loss

Define the *most offending incorrect answer*, $\bar{y}$, to be the answer that has the lowest energy among all answers that are incorrect:

$$\bar{y} = \arg\min_{y' \in Y, y' \neq y} E(w,x,y')$$

where y is the correct answer.

Margin losses separates the true answer from the lower bound of the higher energy answers.

Hinge loss

$$L_{hinge}(w,x,y) = \max(0, m + E(w,x,y) - E(w,x,\bar{y})$$

The difference between the energies of the most offending answer and the correct answer is penalized when larger than m.

Log loss

$$L_{log}(w,x,y) = \log\left(1 + e^{E(w,x,y) - E(w,x,\bar{y})}\right)$$

This loss encourages the gap between the energies of the correct answer and the most offending answer to be as small as possible.

Negative log-likelihood loss

$$L_{nll}(w,x,y) = E(w,x,y) + \frac{1}{\beta}\log\int_{y' \in Y} e^{-\beta E(w,x,y')}$$

The motivation is to maximize the likelihood of the observed data. The negative log-likelihood combines the energies of all values of Y in its contrastive term (second term). The contrastive term pulls up on the energy of each answer with a force proportional to the likelihood of that answer according to the model. Unfortunately, it is quite hard to compute in complex models. A solution is to approximate it using Monte Carlo sampling methods.

Interestingly, the negative log-likelihood loss reduces to the generalized perceptron loss when $\beta \to \infty$, and reduces to log loss when Y has two elements.

2.3 "Good" loss functions

As discussed previously, loss functions such as generalized perception loss can lead to flat-surfaced energy function, which assigns the same value for any configuration of the variables. We have to find functions that are immune to that condition.

Energy loss is a bad loss function in general but certain form can be good. *Generalized perceptron loss* has a margin of zero and thus can lead to a collapsed energy surface. However, this is not always the case. The probability of having a collapsed surface is quite small since the set of collapsed solutions is a small subspace in the solution space .Generalized margin loss such as hinge loss and log loss are generally good. *Negative log-likelihood loss* is also a safe choice.