CS229: Study Guide - Loss Functions

This study guide covers the fundamental concepts of loss functions in supervised learning, drawing from our lecture notes. Understanding loss functions is crucial as they dictate how our models learn and what constitutes a "good" prediction.

1. Introduction to Loss Functions

In supervised learning, our goal is to learn a mapping from input data x to targets y. This process generally involves three steps:

1. **Choosing a representation** for our problem (e.g., linear models like $\theta^T x$).

2. **Choosing a loss function** L that quantifies the penalty for incorrect predictions.

3. **Minimizing the loss** over a training set to find the optimal model parameters θ.

A loss function $L(z, y)$ measures the "loss" or "error" we incur when predicting z (often $\theta^T x$) for a true target y.

2. Loss Functions for Different Supervised Learning Problems

The choice of loss function depends heavily on the type of problem (regression, binary classification, multiclass classification) and the nature of the target variable Y.

Linear Regression (Y = R)

Prediction: $z = \theta^T x$

Loss Function: Squared Error Loss

$L(z, y) = 1/2 * (z - y)^2$

$L(\theta^T x, y) = 1/2 * (\theta^T x - y)^2$

Note: The L2 loss (squared error) is analytically convenient and widely used, leading to the conditional mean as the optimal predictor.

Binary Classification (Y = {-1, 1})

Prediction: $z = \theta^T x$ (often interpreted as a "score" or "margin")

Loss Function: Logistic Loss

$L(z, y) = \log(1 + e^{(-yz)})$

$L(\theta Tx, y) = \log(1 + e^{(-y\theta Tx)})$

Note: This is the foundation of Logistic Regression.

Multiclass Classification (Y = {1, 2, ..., k})

Prediction: $z = [\theta 1Tx, ..., \theta kTx]$ (a vector of scores for each class)

Loss Function: A variant of **Cross-Entropy Loss** (often called Softmax Loss)

$L(z, y) = \log(\Sigma_{i=1 \text{ to } k} \exp(zi - zy))$

$L(\Theta Tx, y) = \log(\Sigma_{i=1 \text{ to } k} \exp(xT(\theta i - \theta y)))$

Goal: To have the score for the true class y (θyTx) be greater than the scores for all other classes $i \neq y$ (θiTx).

3. Challenges with Simple Loss Functions: The Zero-One Loss

A seemingly intuitive loss for classification is the **Zero-One Loss**:

* $L(z, y) = 1$ if z is misclassified (e.g., $z \leq 0$ for y=1 or $z > 0$ for y=-1)

* $L(z, y) = 0$ if z is correctly classified

While this directly measures the average number of mistakes, it has significant drawbacks:

* **Discontinuous:** Jumps from 0 to 1, making gradient-based optimization difficult.

* **Non-convex:** Has many local minima, making global minimization hard.

* **NP-hard to minimize:** Computationally intractable for large problems.

Due to these issues, we prefer "surrogate" loss functions that are continuous and convex, even if they don't perfectly align with the zero-one error.

4. Desired Properties of Loss Functions

For effective optimization, especially with gradient-based methods, we generally seek loss functions that are:

* **Convex:** Ensures that any local minimum is also a global minimum.

* **Continuous and Differentiable (or sub-differentiable):** Allows for the use of gradient descent and related optimization algorithms.

Furthermore, for margin-based classifiers (where $z = yx^T\theta$ represents the "margin" of correctness):

* $\varphi(z) \to 0$ as $z \to \infty$: As the prediction becomes very confident and correct, the loss should approach zero.

* $\varphi(z) \to \infty$ as $z \to -\infty$: As the prediction becomes very confident and incorrect, the loss should grow infinitely large.

5. Margin-Based Loss Functions (for Binary Classification, $y \in \{-1, 1\}$)

These losses are defined in terms of the **margin $z = yx^T\theta$**, which indicates how "correct" and "confident" a prediction is. A positive margin means a correct classification, and a larger positive margin means higher confidence.

Logistic Loss:

$\varphi_{logistic}(z) = \log(1 + e^{(-z)})$

Characteristics: Smooth, convex, approaches 0 for large positive z, and grows linearly for large negative z.

Associated Algorithm: Logistic Regression.

Hinge Loss:

$\varphi_{hinge}(z) = [1 - z]_+ = \max\{1 - z, 0\}$

Characteristics: Convex, continuous but not differentiable at z=1. It penalizes misclassifications ($z < 1$) and has zero loss for correctly classified points with a margin greater than or equal to 1.

Associated Algorithm: Support Vector Machines (SVMs).

Exponential Loss:

φexp(z) = e^(-z)

Characteristics: Convex, smooth, penalizes misclassifications very aggressively (exponentially).

Associated Algorithm: Boosting (e.g., AdaBoost).

Figure 2 in the notes visually compares these three margin-based loss functions.

6. Other Noteworthy Loss Functions

L1 Loss (Mean Absolute Error - MAE):

E|Y - f(X)|

Characteristics: Leads to the conditional median as the optimal predictor. More robust to outliers than L2 loss.

Drawbacks: Discontinuities in its derivatives can hinder optimization.

Zero-One Loss (for Categorical Output G):

Represented by a K x K matrix L, where L(k, ■) is the cost of classifying an observation belonging to class Gk as G■.

Most commonly, L(k, ■) = 0 if k=■ and 1 if k≠■ (all misclassifications cost 1 unit).

Expected Prediction Error (EPE): E[L(G, ■(X))]

Note: While conceptually simple, its practical minimization is difficult due to its non-convex and discontinuous nature, as discussed in Section 3.

Key Takeaways

The choice of loss function is a fundamental design decision in supervised learning, directly influencing the model's behavior and the type of errors it prioritizes.

Different problem types (regression, classification) require different families of loss functions.

While the zero-one loss is intuitive for classification, its non-convexity and discontinuity make it impractical for optimization.

Convex and continuous (or sub-differentiable) loss functions are preferred for their amenability to efficient optimization algorithms.

Margin-based losses (Logistic, Hinge, Exponential) are crucial for binary classification, each leading to distinct and powerful machine learning algorithms.