Lecture 6: Advanced Classification, SVMs, and Deep Learning

Yonghao Lee

December 7, 2025

1 Advanced Logistic Regression

1.1 Recap: The Logit Score

In linear classification, the hypothesis is a hyperplane defined by $\mathbf{w} \cdot \mathbf{x} + b = 0$. The signed distance from a sample to this hyperplane is the score (logit) $s = \mathbf{w} \cdot \mathbf{x} + b$. To predict probabilities, we pass this score through the sigmoid function $\sigma(s) = \frac{1}{1+e^{-s}}$.

1.2 Multi-Class Classification (Softmax)

To extend this to K classes, we learn a specific weight vector $\mathbf{w}^k$ for each class k. We compute scores $z^k = \mathbf{w}^k \cdot \mathbf{x}$ for all classes. To obtain a valid probability distribution (summing to 1), we apply the **Softmax function**:

$$P(y = k|\mathbf{x}) = \text{Softmax}(z^k) = \frac{e^{z^k}}{\sum_{j=1}^{K} e^{z^j}} \tag{1}$$

The optimization involves finding the weights $\mathbf{W} = \{\mathbf{w}^1, \ldots, \mathbf{w}^K\}$ that minimize the negative log-likelihood of the true class labels.

1.3 Ridge Classification

An alternative to Logistic Regression is Ridge Classification, where we treat the binary labels as regression targets (e.g., $+1, -1$) and apply Ridge Regression (MSE loss with L2 regularization).

$$L(S_{train}, \mathbf{w}) = \frac{1}{2N} \sum (y - \mathbf{w} \cdot \mathbf{x})^2 + \frac{\alpha}{2} \|\mathbf{w}\|^2 \tag{2}$$

While simpler and possessing a closed-form solution, it requires tuning α on a validation set.

2 Optimization: Stochastic Gradient Descent

Computing the gradient over the entire training set for every step is computationally expensive for large datasets.

- **Stochastic Gradient Descent (SGD):** Compute the gradient and update weights using only a single random sample. It is faster but noisy.
- **Mini-Batch GD:** Compute the gradient on a small batch of samples (e.g., 32 or 64). This offers a tradeoff: it is less noisy than SGD and computationally more efficient than full GD.

3 Support Vector Machines (SVM)

Logistic regression is sensitive to "extreme" examples far from the boundary. The Maximum Margin principle suggests focusing on the **decision region**.

3.1 Hard-SVM

We define two parallel hyperplanes separating the classes: $P_1 : \mathbf{w} \cdot \mathbf{x} + b = 1$ and $P_2 : \mathbf{w} \cdot \mathbf{x} + b = -1$. The distance (margin) between them is derived as:

$$\text{Distance} = \frac{2}{\|\mathbf{w}\|} \tag{3}$$

The objective is to maximize this distance, which is equivalent to minimizing $\|\mathbf{w}\|^2$ subject to the constraint that all points are correctly classified outside the margin.

3.2 Soft-SVM

When data is not linearly separable, Hard-SVM fails. Soft-SVM relaxes the constraints by introducing a cost for violating the margin. It minimizes:

$$L(\mathbf{w}, b) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i + b)) \tag{4}$$

The second term is known as the Hinge Loss.

4 Performance Metrics

Accuracy is often insufficient, especially for imbalanced datasets. We use the confusion matrix to derive better metrics:

- **Recall (Sensitivity):** $TP/(TP + FN)$. The percentage of actual positives correctly identified.
- **Precision:** $TP/(TP + FP)$. The percentage of predicted positives that are actually positive.
- **F1 Score:** The harmonic mean of Precision and Recall.

4.1 Evaluation Curves

Since metrics depend on the decision threshold, we use curves to visualize performance across all thresholds:

- **ROC Curve:** Plots True Positive Rate vs. False Positive Rate. The Area Under Curve (AUC) summarizes performance.
- **PR Curve:** Plots Precision vs. Recall.