

Exercises

§6.2,6.7: Cross-Entropy

Python: There is an optional starter script that supports Python-based problem(s) on this sheet. Use it as scaffolding: read the comments, complete the lines marked **TODO**, and compare the output with the mathematics requested in the problem(s).

Problem 1 Compute the binary cross-entropy loss for: (a) $y = 1, p = 0.8$, (b) $y = 1, p = 0.2$, (c) $y = 0, p = 0.8$, (d) $y = 0, p = 0.2$. Which predictions are correct at a 0.5 threshold?

Problem 2 Let the true class-1 probability be $q = 0.7$. The expected cross-entropy of a prediction p is $C(p) = -[0.7 \log p + 0.3 \log(1 - p)]$. Differentiate and verify that $p = 0.7$ is the unique minimizer.

Problem 3 Let $p = \sigma(f)$. Starting from $\ell(y, p) = -[y \log p + (1 - y) \log(1 - p)]$, derive $\frac{\partial \ell}{\partial f} = p - y$.

Problem 4 Suppose one observation has $\tilde{\mathbf{x}} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$, $y = 1$, and the model predicts $p = 0.25$. Find the gradient contribution $(p - y)\tilde{\mathbf{x}}$.

Problem 5 For $X = \begin{bmatrix} 1 & -1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$, and $\mathbf{w} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, compute $\mathbf{p}$ and $\nabla g = \frac{1}{3}X^T(\mathbf{p} - \mathbf{y})$. Then compute one gradient-descent step with $\alpha = 0.3$.

Problem 6 Show that the cross-entropy written in terms of the score satisfies $\ell(y, f) = \log(1 + e^f) - yf$ for $y \in \{0, 1\}$. Then show that this equals $\log(1 + e^{-sf})$ under the label conversion $s = 2y - 1$.

Problem 7 A classifier predicts class 1 for all three of the following probabilities: 0.51, 0.80, 0.99. If the observed class is 1, compute the cross-entropy for each. Why can cross-entropy distinguish models that accuracy cannot distinguish on this observation?

Problem 8 Use Python to implement

```
def cross_entropy_from_scores(w, X, y):
    ...
```

using np. For

```
X = np.array([[1., -1.],
               [1., 0.],
               [1., 1.]])
```

```
y = np.array([0., 1., 1.])
```

```
w = np.array([0.5, 1.0])
```

compute the mean cross-entropy and the gradient.