CS 6140 — Homework 2

Ashish Dasu

Due: February 22, 2026

Problem 1: Probability

1.1

Show that $P(X_1)P(X_2|X_1)P(X_3|X_1,X_2) = P(X_1,X_2,X_3)$.

Applying $P(A|B) = P(A,B)/P(B)$ to each conditional term:

$$P(X_1)\,P(X_2|X_1)\,P(X_3|X_1,X_2) = P(X_1)\cdot\frac{P(X_1,X_2)}{P(X_1)}\cdot\frac{P(X_1,X_2,X_3)}{P(X_1,X_2)} = P(X_1,X_2,X_3). \qquad \blacksquare$$

1.2

Assume $P(X|Y) = P(X)$. *Show that* $P(Y|X) = P(Y)$.

By Bayes' theorem:

$$P(Y|X) = \frac{P(X|Y)\,P(Y)}{P(X)} = \frac{P(X)\,P(Y)}{P(X)} = P(Y). \qquad \blacksquare$$

Problem 2: Logistic Regression Derivatives

Let $\sigma(z) = \dfrac{1}{1+e^{-z}}$.

2.1

Show that $\partial\sigma(z)/\partial z = \sigma(z)(1-\sigma(z))$.

Differentiating $\sigma(z) = (1+e^{-z})^{-1}$ via the chain rule:

$$\frac{\partial\sigma(z)}{\partial z} = \frac{e^{-z}}{(1+e^{-z})^2} = \frac{1}{1+e^{-z}}\cdot\frac{(1+e^{-z})-1}{1+e^{-z}} = \sigma(z)\big(1-\sigma(z)\big). \qquad \blacksquare$$

2.2

Compute $\partial\log(\sigma(z))/\partial z$.

$$\frac{\partial\log\sigma(z)}{\partial z} = \frac{1}{\sigma(z)}\cdot\sigma(z)(1-\sigma(z)) = 1-\sigma(z).$$

2.3

Let $z = w^\top \phi(x)$. Compute $\partial \log(\sigma(w^\top \phi(x)))/\partial w$.

By the chain rule:

$$\frac{\partial \log \sigma(w^\top \phi(x))}{\partial w} = \underbrace{\frac{\partial \log \sigma(z)}{\partial z}}_{1-\sigma(z)} \cdot \underbrace{\frac{\partial z}{\partial w}}_{\phi(x)} = \big(1 - \sigma(w^\top \phi(x))\big)\, \phi(x).$$

Problem 3: MLE

Given i.i.d. samples from $p_\lambda(X_i = x_i) = \lambda^{x_i} \times e^{-\lambda^2}$ ($\lambda > 0$, normalization constant removed), derive the MLE for λ.

3a)

$$L(\lambda) = \prod_{i=1}^{N} \lambda^{x_i} e^{-\lambda^2} = \lambda^{\sum x_i}\, e^{-N\lambda^2},$$

$$\ell(\lambda) = \Big(\textstyle\sum_{i=1}^{N} x_i\Big) \log \lambda - N\lambda^2.$$

3b)

Differentiating and setting to zero:

$$\frac{d\ell}{d\lambda} = \frac{\sum x_i}{\lambda} - 2N\lambda = 0 \implies \sum x_i = 2N\lambda^2 \implies \hat{\lambda}_{\text{MLE}} = \sqrt{\frac{\bar{x}}{2}}.$$

The second derivative $d^2\ell/d\lambda^2 = -(\sum x_i)/\lambda^2 - 2N < 0$ confirms this is a maximum. Since $\bar{x} > 0$, the constraint $\lambda > 0$ is satisfied.

Problem 4: MLE for Gaussian (μ only)

Given i.i.d. samples from $\mathcal{N}(\mu, \sigma^2)$, derive the MLE for μ.

The log-likelihood is:

$$\ell(\mu, \sigma^2) = -\frac{N}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{N} (x_i - \mu)^2.$$

Only the second term depends on μ. Differentiating and setting to zero:

$$\frac{\partial \ell}{\partial \mu} = \frac{1}{\sigma^2} \sum_{i=1}^{N} (x_i - \mu) = 0 \implies \hat{\mu}_{\text{MLE}} = \frac{1}{N} \sum_{i=1}^{N} x_i = \bar{x}.$$

The second derivative $\partial^2\ell/\partial\mu^2 = -N/\sigma^2 < 0$ confirms this is a maximum.

Problem 5: MAP Estimation for Classification

Assume a Gaussian prior $p(w) = (1/\sqrt{2\pi\sigma^2})^d \cdot \exp(-\|w\|_2^2/2\sigma^2)$*. Derive the MAP estimate for* w *and relate it to the regularized logistic regression solution.*

5.1

MAP maximizes the log-posterior:

$$\log p(w|\mathcal{D}) \propto \log p(\mathcal{D}|w) + \log p(w) = -\mathrm{BCE}_w(\mathcal{D}) - \frac{\|w\|_2^2}{2\sigma^2}.$$

Equivalently:

$$\hat{w}_{\mathrm{MAP}} = \arg\min_w \left[\mathrm{BCE}_w(\mathcal{D}) + \frac{1}{2\sigma^2}\|w\|_2^2\right].$$

5.2

With $\lambda = 1/(2\sigma^2)$, this becomes $\arg\min_w \left[\mathrm{BCE}_w(\mathcal{D}) + \lambda\|w\|_2^2\right]$, which is L_2-regularized logistic regression, a standard approach for controlling overfitting. So a zero-mean Gaussian prior on w is equivalent to adding an L_2 penalty, with the prior variance σ^2 controlling regularization strength.

Problem 6: Convex Functions and Sets

6.1 $S = \{x \in \mathbb{R}^n : Bx = a\}$ is convex: True

Show that the set $S = \{x \in \mathbb{R}^n : Bx = a\}$ *is convex.*

Let $x_1, x_2 \in S$ and $\theta \in [0, 1]$:

$$B(\theta x_1 + (1-\theta)x_2) = \theta Bx_1 + (1-\theta)Bx_2 = \theta a + (1-\theta)a = a.$$

So $\theta x_1 + (1-\theta)x_2 \in S$, and S is convex. ■

6.2 $f(x) = c_1 e^{-\alpha x} + c_2 x^4 + c_3 \log(x)$ is convex: True

$c_1, c_2 \geq 0$*,* $c_3 \leq 0$*, domain* $x \in (0, +\infty)$*.*

Computing the second derivative:

$$f''(x) = c_1\alpha^2 e^{-\alpha x} + 12c_2x^2 - \frac{c_3}{x^2}.$$

Each term is non-negative on $(0, +\infty)$ given $c_1, c_2 \geq 0$ and $c_3 \leq 0$. Therefore $f''(x) \geq 0$ on the domain, so f is convex. ■

6.3 $f(x) = 0.2x^3 + 30\sin(x)$ is convex: False

$f''(x) = 1.2x - 30\sin(x)$. At $x = \pi/2$:

$$f''(\pi/2) = 0.6\pi - 30 \approx -28.1 < 0.$$

Since $f''(\pi/2) < 0$, f is not convex. ■

Problem 7: Handwritten Digits

Part 1: Multi-Class Softmax Classification

Trained a softmax logistic regression model on standardized pixel features with L_2 regularization ($C = 1.0$, `solver='lbfgs'`, `multi_class='multinomial'`). Training accuracy: 100%, test accuracy: 97.78%.

Figure 1 shows the coefficient distributions for class 0 and class 7. The two distributions look fairly similar, which is expected. L2 regularization pushes most weights toward zero regardless of class, so the overall shape is dominated by that effect. The meaningful differences are in the tails: which specific pixels carry large positive or negative weights determines how each class is recognized, and that spatial information is not visible in the histogram alone. Notably, both classes achieve 100% test accuracy despite the similar-looking distributions, which reinforces this point: the discriminative information lies in the spatial arrangement of those tail weights, not the marginal distribution.

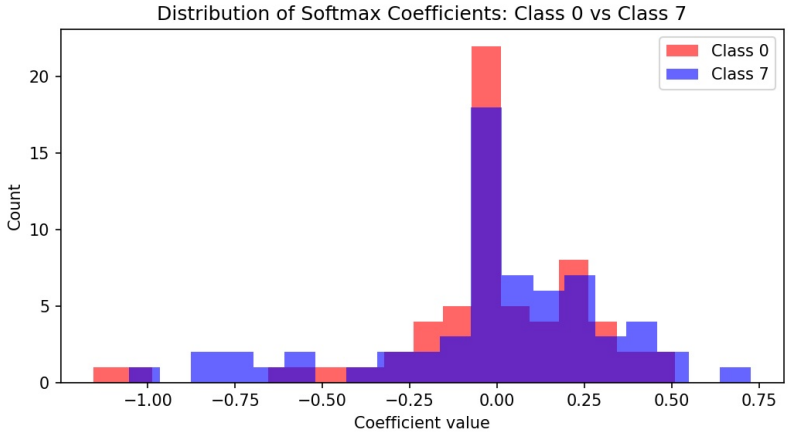

Figure 1: Coefficient distributions for class 0 (red) and class 7 (blue).

Per-class test accuracies are reported below. Classes 1 and 8 have the lowest accuracy, most likely due to visual similarity with other digits. 1 can look like 7, and 8 shares structure with both 9 and 3.

Class	Test Accuracy
0	100.00%
1	93.48%
2	100.00%
3	100.00%
4	100.00%
5	97.83%
6	97.78%
7	100.00%
8	93.02%
9	95.56%

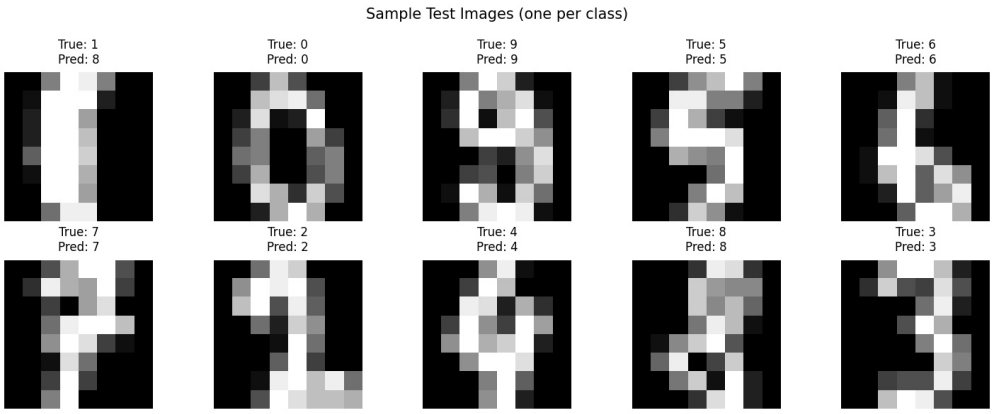

Figure 2: Sample test images with true and predicted labels.

Part 2: Multi-Label Classification from Raw Pixels

Three binary logistic regression models were trained independently, one per label. Features are the same standardized pixel vectors from Part 1.

Property	Train Accuracy	Test Accuracy
is_even	92.80%	91.33%
is_greater_than_5	90.57%	86.44%
is_prime	97.03%	95.11%

`is_prime` gets the best accuracy. The prime digits (2, 3, 5, 7) exclude the round digits (0, 6, 8, 9) and certain structured ones (1, 4), so pixel-level cues like closed loops or dominant vertical strokes become useful proxies for the boundary, even if the model never explicitly reasons about shape. `is_greater_than_5` is the hardest because it draws an arbitrary boundary in class space that has no geometric correlate in pixel space. Unlike `is_even`, where digits like 0 and 2 share some curved structure, the group $\{6, 7, 8, 9\}$ has no common visual pattern, so there is nothing in the raw pixels that naturally separates them from $\{0, 1, 2, 3, 4, 5\}$.

Part 3: Hierarchical Bridge

For each example, the 10-class softmax probability vector $p(x) \in \mathbb{R}^{10}$ from Part 1 was computed and used as the input feature for three new binary classifiers.

Property	Train Accuracy	Test Accuracy
is_even	100.00%	98.67%
is_greater_than_5	100.00%	98.67%
is_prime	100.00%	99.56%

Discussion. The accuracy jump from Part 2 to Part 3 makes sense once you think about what $p(x)$ actually is. All three properties are just functions of the digit class, so if the model already has a good estimate of which digit it is, predicting is_even or is_prime is basically a lookup. Raw pixels do not give the model that directly, so it has to infer class membership from scratch across 64 features, which is noisier and harder. The small gap from 100% in Part 3 is just error from Part 1 propagating through.