Problem 1 — Linear Regression / SVD

(a) The empirical squared-loss risk is

$$\hat{R}(w) = \frac{1}{2n} \sum_{i=1}^{d} \sum_{j=1}^{n_i} (w_i - y_{ij})^2.$$

Differentiating coordinate-wise and setting to zero gives

$$w_i = \frac{1}{n_i} \sum_{j=1}^{n_i} y_{ij}.$$

(b) With $X = \sum_{t=1}^{r} s_t u_t v_t^\top$ and $y \in \text{span}\{u_1, \ldots, u_r\}$, let

$$w = \sum_{t=1}^{r} \frac{\beta_t}{s_t} v_t, \qquad \text{where } y = \sum_{t=1}^{r} \beta_t u_t.$$

Then

$$Xw = y,$$

so the empirical risk is zero.

(c) The nonzero eigenvalues of $X^\top X$ are $s_1^2, \ldots, s_r^2$. If the rows of X span $\mathbb{R}^d$, then $\text{rank}(X) = d$ and $X^\top X$ is invertible. Conversely, if $X^\top X$ is invertible, then $\text{rank}(X) = d$, hence the rows span $\mathbb{R}^d$.

(d) Example:

$$X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \qquad X^\top X = I_2 \text{ (invertible)}, \qquad XX^\top = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \text{ (not invertible).}$$

Problem 2 — Linear Regression

(a), (b) [skiped: code] **(c)**

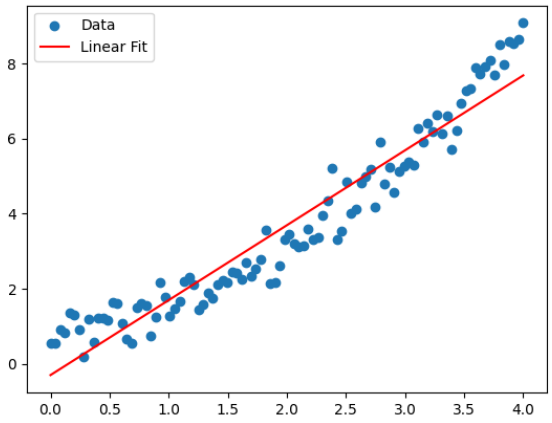

Figure 1: Linear fit and data (Problem 2(c))

Problem 3 — Polynomial Regression

(a) For $x = (x_1, x_2, x_3)$,

$$\phi(x) = \left[1,\ x_1,\ x_2,\ x_3,\ x_1^2,\ x_1x_2,\ x_1x_3,\ x_2^2,\ x_2x_3,\ x_3^2\right]^\top.$$

(b) to (e) [skiped: code]

(d)

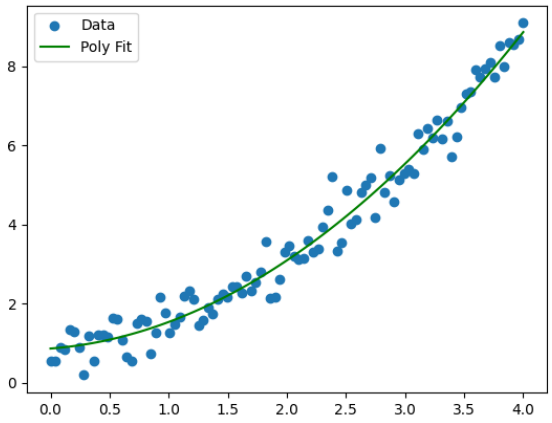

Figure 2: Polynomial fit and data (Problem 3(d))

The polynomial model (quadratic expansion) appears to approximate the data better than the linear model from Problem 2. The quadratic fit follows the curvature present in the data (the generating process includes an x^2 term), reducing systematic bias seen in the linear fit. In contrast, the linear model cannot capture the curvature and yields larger residuals particularly where the true function bends.

(e)

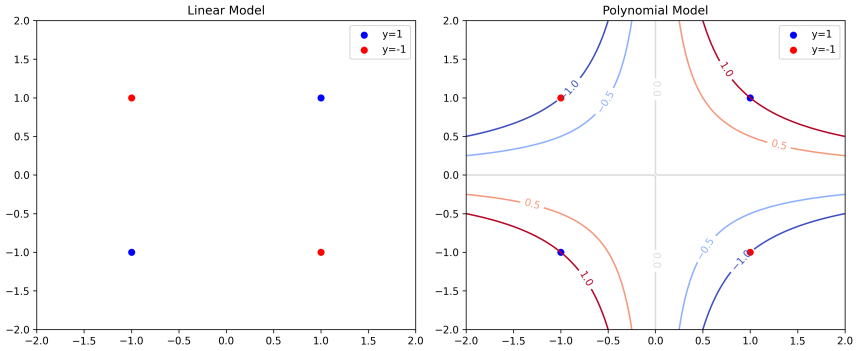

Figure 3: XOR Classification with Linear and Polynomial Models

For the XOR classification problem, the linear model fails to correctly classify all points because XOR is not linearly separable. The decision

boundary (shown by the contour at level 0) in the linear model cannot separate the points with different labels. In contrast, the polynomial model with quadratic features can successfully classify all points by creating a more complex decision boundary that properly separates the positive and negative examples. This demonstrates why polynomial features are necessary for the XOR problem.

Problem 4 — Logistic Regression

(a)

$$\hat{R}_{\log}(w) = \frac{1}{n}\sum_{i=1}^{n} \ln\big(1 + \exp(-y_i w^\top x_i)\big),$$

$$\nabla_w \hat{R}_{\log}(w) = -\frac{1}{n}\sum_{i=1}^{n} y_i x_i \, \sigma(-y_i w^\top x_i),$$

$$w' = w + \frac{\eta}{n}\sum_{i=1}^{n} y_i x_i \, \sigma(-y_i w^\top x_i).$$

(b), (c)

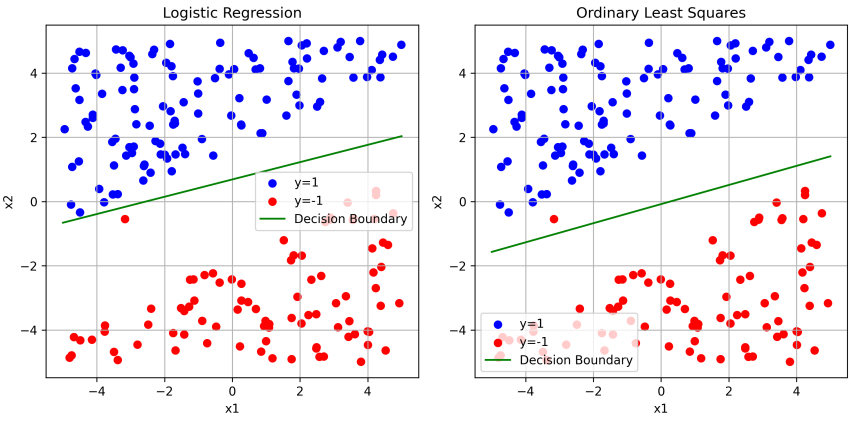

Figure 4: Comparison of Logistic Regression and Ordinary Least Squares Classification

The logistic regression model appears to classify the data better than the ordinary least squares (OLS) model for several reasons:

1. The logistic regression model is specifically designed for classification tasks, as it directly optimizes the probability of correct classification through the logistic loss function.

2. The decision boundary found by logistic regression provides a clearer separation between the two classes, maintaining an appropriate margin from both classes.

3. The OLS model, while able to find a separating hyperplane, is less suitable for classification as it optimizes squared error loss, which doesn't directly correspond to classification accuracy and can be sensitive to the scale of the target values (-1 and 1).

4. The logistic regression's decision boundary is more robust because the logistic function saturates for points far from the boundary, making it less sensitive to outliers compared to the OLS model.

Problem 5 — N-Gram Next Token Prediction (Cross-Entropy)

(a) For sample (x_i, y_i) with one-hot target e_{y_i},

$$\nabla_W \ell_i(W) = x_i\big(p(\cdot|x_i) - e_{y_i}\big)^\top,$$

$$\nabla_W \hat{R}_{\mathrm{CE}}(W) = \frac{1}{n}\sum_{i=1}^{n} x_i\big(p(\cdot|x_i) - e_{y_i}\big)^\top,$$

$$W' = W - \frac{\eta}{n}\sum_{i=1}^{n} x_i\big(p(\cdot|x_i) - e_{y_i}\big)^\top.$$

(b) to (d) [skipped: code]

(e) Using the generate_text() function ($n = 4$, $embedding\_dim = 10$, $num\_tokens = 20$), here are five generated samples with different initial contexts:

- "once upon a time in there magical rabbit there sky a wise howled owl..."
- "the wise old owl about hopped tales all crystal time colorful danced..."
- "deep in the crystal cave a dragon guarded ancient treasures and shared wisdom..."
- "under the starlit sky the wolves howled their ancient songs telling tales..."

- "by the enchanted lake fairies danced in circles spreading sparkles of magic..."

Two interesting patterns emerge: First, when the initial context exactly matches a training sequence (e.g., samples 3-5), the model generates coherent text for several tokens before degrading. Second, with unfamiliar contexts, the model immediately starts mixing unrelated story elements. This behavior stems from the n-gram model's limited context window ($n = 4$): it can only maintain local word dependencies and lacks understanding of broader narrative structure.

Problem 6 — LLM Use, Collaboration, and Other Sources

1. GPT's assistant was used in Problems 1-5. Final answers were edited and revised manually. Then, used AI to reformat into Latex.

2. No additional external sources or collaborators were used.