HW 2

Kevin Lin

2/9/2026

1

(a) The gradient of hinge loss with respect to w:

$$\begin{aligned}\nabla_w \ell(w^T x, y) &= \nabla_w \max\{0, 1 - yw^T x\} \\ &= \begin{cases} 0 & \text{if } yw^T x \geq 1 \\ -yx & \text{if } yw^T x < 1 \end{cases}\end{aligned}$$

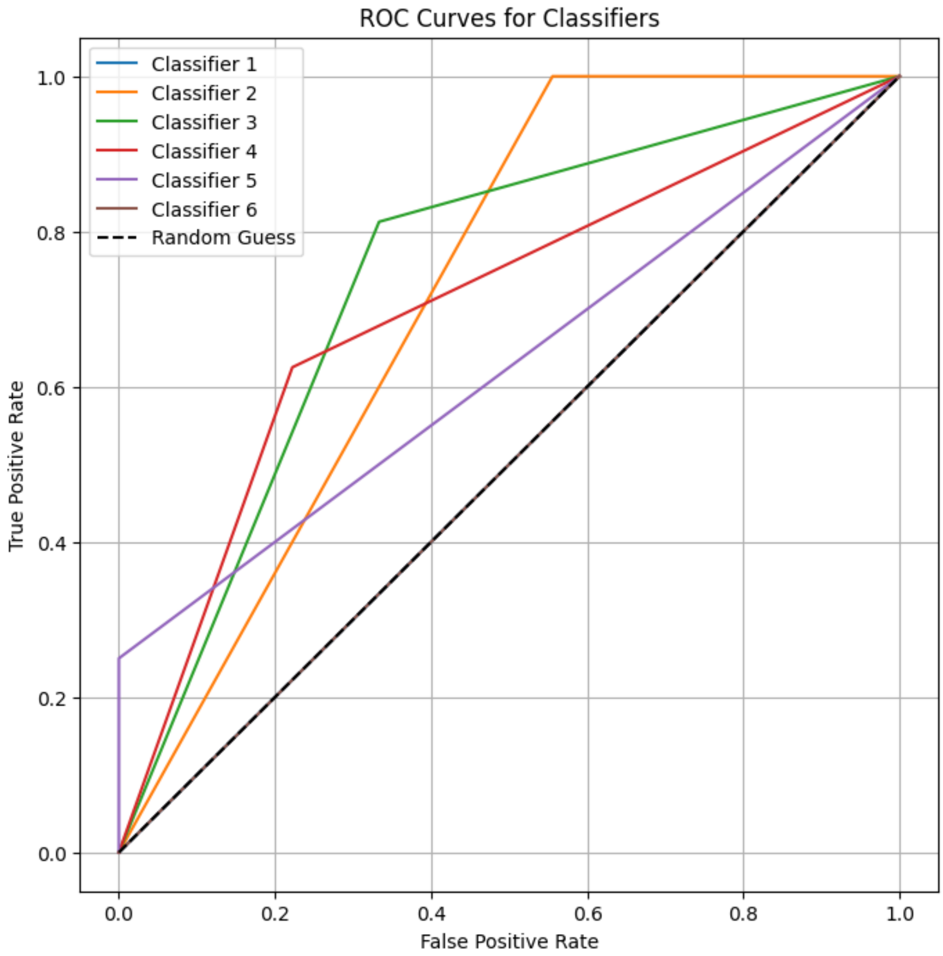
(b) The gradient of Perceptron loss w.r.t w:

$$\begin{aligned}\nabla_w \ell(w^T x, y) &= \nabla_w \max\{0, -yw^T x\} \\ &= \begin{cases} 0 & \text{if } yw^T x \geq 0 \\ -yx & \text{if } yw^T x < 0 \end{cases}\end{aligned}$$

(c) For hinge loss, w is updated only when the margin condition $yw^T x < 1$ is violated, meaning the prediction is not only incorrect but also not confident enough. However, for Perceptron loss, the update occurs whenever the prediction is incorrect (when $yw^T x < 0$). This means that hinge loss encourages a larger margin between classes while Perceptron loss focuses solely on correct classification.

2

(a) ROC plot (see Jupyter notebook for code):



(b) Classifier 2 has the highest accuracy of 0.8, while Classifier 6 has the lowest accuracy of 0.36. See Jupyter notebook for code.

(c) Classifier 5 has the highest precision of 1, while Classifier 1 has the lowest precision of 0.64. Classifier 6 has undefined precision as it has no true or false positives. See Jupyter notebook for code.

(d) Classifier 1 F1 score: 0.78
Classifier 2 F1 score: 0.86
Classifier 3 F1 score: 0.81
Classifier 4 F1 score: 0.71
Classifier 5 F1 score: 0.4
Classifier 6 F1 score: Undefined for the same reasons as precision.
See Jupyter notebook for code.

3

See Jupyter notebook for code and plots.

4

(a) We can derive the optimal $\mathbf{w}^*$ that minimizes $E_2\mathbf{w}$ as follows:

$$
\begin{aligned}
E_2(\mathbf{w}) &= \frac{1}{N}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_2^2 \\
&= \frac{1}{N}(\mathbf{X}\mathbf{w} - \mathrm{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y}) + \lambda\mathbf{w}^T\mathbf{w} \\
&= \frac{1}{N}(\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{y}^T\mathbf{y}) + \lambda\mathbf{w}^T\mathbf{w}
\end{aligned}
$$

Taking the gradient with respect to $\mathbf{w}$ and setting it to zero:

$$
\begin{aligned}
\nabla_{\mathbf{w}} E_2(\mathbf{w}) &= \frac{2}{N}\mathbf{X}^T\mathbf{X}\mathbf{w} - \frac{2}{N}\mathbf{X}^T\mathbf{y} + 2\lambda\mathbf{w} = 0 \\
&= (\mathbf{X}^T\mathbf{X} + N\lambda\mathbf{I})\mathbf{w} = \mathbf{X}^T\mathbf{y} \\
\mathbf{w}^* &= (\mathbf{X}^T\mathbf{X} + N\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}
\end{aligned}
$$

(b) This new objective function overcomes the singularity issue by adding the term $\lambda\|\mathbf{w}\|_2^2$, which effectively adds $\lambda\mathbf{I}$ to the matrix $\mathbf{X}^T\mathbf{X}$. This addition ensures that the matrix $\mathbf{X}^T\mathbf{X} + N\lambda\mathbf{I}$ is positive definite and invertible, even if $\mathbf{X}^T\mathbf{X}$ is singular. The regularization term penalizes large weights, promoting stability and preventing overfitting.

5