Announcements

- Quiz 2 **due tomorrow (Thursday), September 22 at 8pm**
- Quiz 3 will be posted tonight
- Homework 2 posted (**Due Monday, October 3 at 8pm**)

Recap: Maximum Likelihood Estimation

- **Compare to loss function minimization:**
  - **Before:** $y_i \approx f_\beta(x_i)$
  - **Now:** $y_i \sim p_\beta(\cdot \mid x_i; \beta)$
- **Intuition the difference:**
  - $f_\beta(x_i)$ just provides a point that y_i should be close to
  - $p_\beta(\cdot \mid x_i; \beta)$ provides a score for each possible y_i
- Maximum likelihood estimation combines the **loss function** and **model family** design decisions

Recap: Maximum Likelihood Estimation

- **Model family is the most likely label:**

$$f_\beta(x) = \arg\max_y p_\beta(y \mid x)$$

- **Loss function is the negative log likelihood (NLL)**:

$$\ell(\beta; Z) = -\log L(\beta; Z) = -\sum_{i=1}^{n} \log p_\beta(y_i \mid x_i)$$

Recap: MLE for Linear Regression

- **Design decision:** We choose the likelihood to be

$$p_{\beta}(y \mid x) = N(y; \beta^{\top} x, 1) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{(\beta^{\top} x - y)^2}{2}}$$

Recap: MLE for Linear Regression

- **Model family:**

$$f_\beta(x) = \arg\max_y p_\beta(y \mid x) = \beta^\top x$$

- **Negative log likelihood:**

$$\ell(\beta; Z) = -\sum_{i=1}^{n} \log p_\beta(y_i \mid x_i) = \frac{n \log(2\pi)}{2} + \sum_{i=1}^{n} (\beta^\top x_i - y_i)^2$$

Recap: MLE for Logistic Regression

- **Design decision:** We choose the likelihood to be

$$p_\beta(Y = 1 \mid x) = \frac{1}{1 + e^{-\beta^\top x}} = \sigma(\beta^\top x)$$

$$p_\beta(Y = 0 \mid x) = 1 - \sigma(\beta^\top x)$$

Recap: MLE for Logistic Regression

- **Model family:**

$$f_\beta(x) = \arg\max_y p_\beta(y \mid x) = 1(\beta^\top x \geq 0)$$

- **Negative log likelihood:**

$$\begin{aligned}\ell(\beta; Z) &= -\sum_{i=1}^{n} \log p_\beta(y_i \mid x_i) \\ &= -\sum_{i=1}^{n} y_i \log\big(\sigma(\beta^\top x_i)\big) + (1 - y_i) \log\big(1 - \sigma(\beta^\top x_i)\big)\end{aligned}$$

Maximum Likelihood View of ML

- **Two design decisions**
  - **Likelihood:** Probability $p_\beta(\, y \mid x \,)$ of data (x, y) given parameters β
  - **Optimizer:** How do we optimize the NLL? (E.g., gradient descent)

- **Corresponding Loss Minimization View:**
  - **Model family:** Most likely label $f_\beta(x) = \arg\max_y p_\beta(\, y \mid x \,)$
  - **Loss function:** Negative log likelihood (NLL) $\ell(\beta; Z) = -\sum_{i=1}^{n} \log p_\beta(\, y_i \mid x_i \,)$

Lecture 6: Logistic Regression (Part 2)

CIS 4190/5190

Fall 2022

Classification Metrics

- While we minimize the NLL, we often evaluate using **accuracy**

- However, even accuracy isn't necessarily the "right" metric
    - If 99% of labels are negative (i.e., $y_i = 0$), accuracy of $f_\beta(x) = 0$ is 99%!
    - For instance, very few patients test positive for most diseases
    - "Imbalanced data"

- What are alternative metrics for these settings?

Classification Metrics

- **Classify test examples as follows:**
  - **True positive (TP):** Actually positive, predictive positive
  - **False negative (FN):** Actually positive, predicted negative
  - **True negative (TN):** Actually negative, predicted negative
  - **False positive (FP):** Actually negative, predicted positive

- Many metrics expressed in terms of these; for example:

$$\text{accuracy} = \frac{TP + TN}{n} \qquad \text{error} = 1 - \text{accuracy} = \frac{FP + FN}{n}$$

Confusion Matrix

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

Confusion Matrix

		Predicted Class	
		Yes	No
Actual Class	Yes	3 TP	4 FN
	No	6 FP	37 TN

Accuracy = 0.8

Classification Metrics

- For imbalanced metrics, we roughly want to disentangle:
  - Accuracy on “positive examples”
  - Accuracy on “negative examples”

- Different definitions are possible (and lead to different meanings)!

Sensitivity & Specificity

- **Sensitivity:** What fraction of **actual positives** are **predicted positive**?
  - **Good sensitivity:** If you have the disease, the test correctly detects it
  - Also called **true positive rate**
- **Specificity:** What fraction of **actual negatives** are **predicted negative**?
  - **Good specificity:** If you do not have the disease, the test says so
  - Also called **true negative rate**
- Commonly used in medicine

Sensitivity & Specificity

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

$$\text{sensitivity} = \frac{TP}{TP + FN}$$

$$\text{specificty} = \frac{TN}{TN + FP}$$

Sensitivity & Specificity

		Predicted Class	
		Yes	No
Actual Class	Yes	3 TP	4 FN
	No	6 FP	37 TN

$$\text{sensitivity} = \frac{TP}{TP + FN}$$

$$\text{specificity} = \frac{TN}{TN + FP}$$

Sensitivity & Specificity

Actual Class \ Predicted Class	Yes	No	
Yes	3 TP	4 FN	$\text{sensitivity} = 3/7$
No	6 FP	37 TN	$\text{specificity} = 37/43$

Precision & Recall

- **Recall:** What fraction of **actual positives** are **predicted positive**?
  - **Good recall:** If you have the disease, the test correctly detects it
  - Also called the **true positive rate** (and sensitivity)
- **Precision:** What fraction of **predicted positives** are **actual positives**?
  - **Good precision:** If the test says you have the disease, then you have it
  - Also called **positive predictive value**
- Used in information retrieval, NLP

Precision & Recall

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

$$\text{recall} = \frac{TP}{TP + FN}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

Precision & Recall

		Predicted Class	
		Yes	No
Actual Class	Yes	3 TP	4 FN
	No	6 FP	37 TN

$$\text{recall} = \frac{TP}{TP + FN}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

Precision & Recall

		Predicted Class	
		Yes	No
Actual Class	Yes	3 TP	4 FN
	No	6 FP	37 TN

$\text{recall} = 3/7$

$\text{precision} = 3/9$

Classification Metrics

- **How to obtain a single metric?**
  - Combination, e.g., F_1 score $= \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$ is the harmonic mean
  - More on this later

- **How to choose the “right” metric?**
  - No generally correct answer
  - Depends on the goals for the specific problem/domain

Optimizing a Classification Metric

- We are training a model to minimize NLL, but we have a different “true” metric that we actually want to optimize

- Two strategies (can be used together):
  - **Strategy 1:** Optimize prediction threshold threshold
  - **Strategy 2:** Upweight positive (or negative) examples

Optimizing Prediction Threshold

- Consider hyperparameter τ for the threshold:

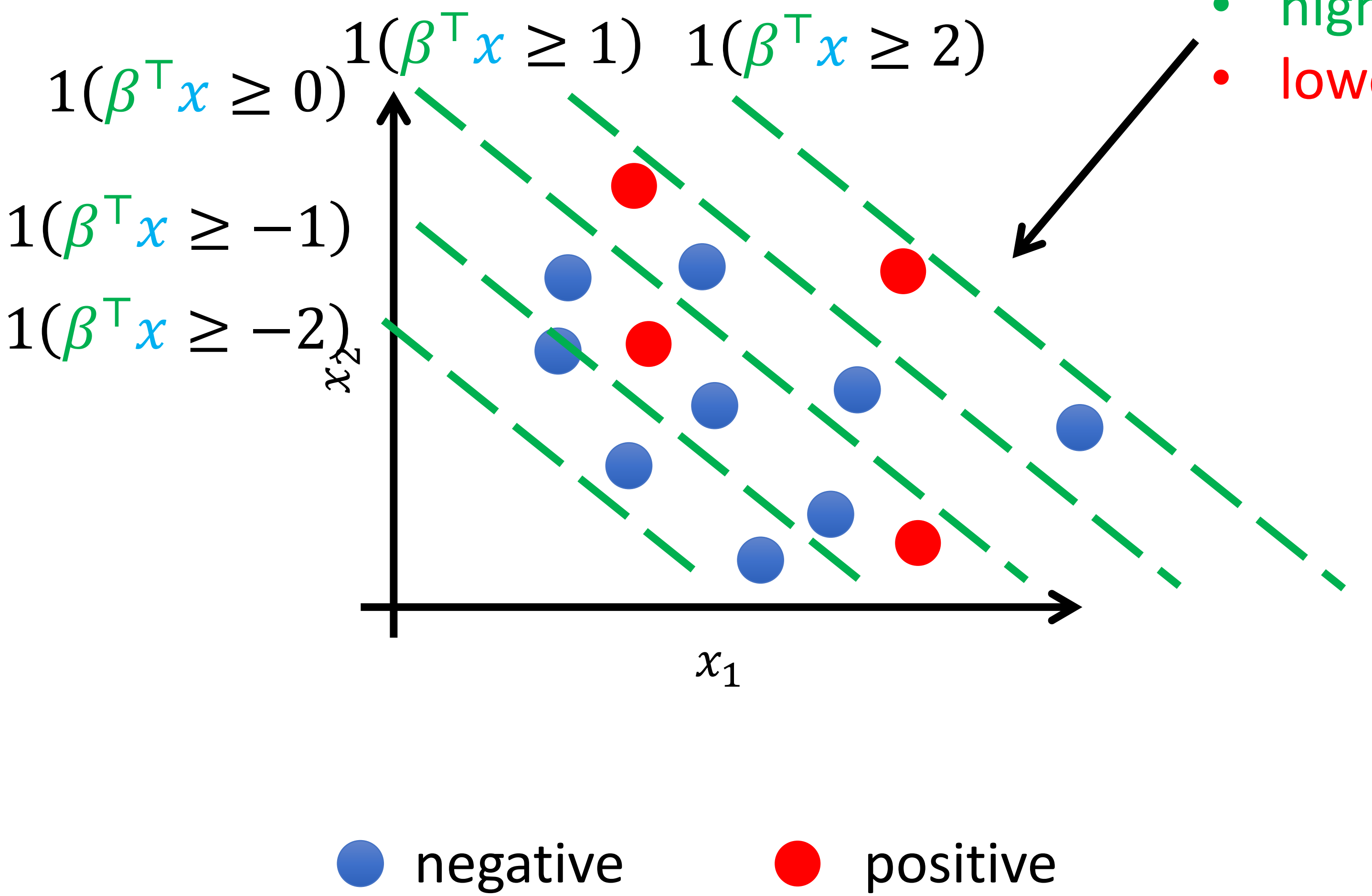
$$f_\beta(x) = 1(\beta^\top x \geq 0)$$

Optimizing Prediction Threshold

- Consider hyperparameter τ for the threshold:

$$f_\beta(x) = 1(\beta^\top x \geq \tau)$$

Optimizing Prediction Threshold

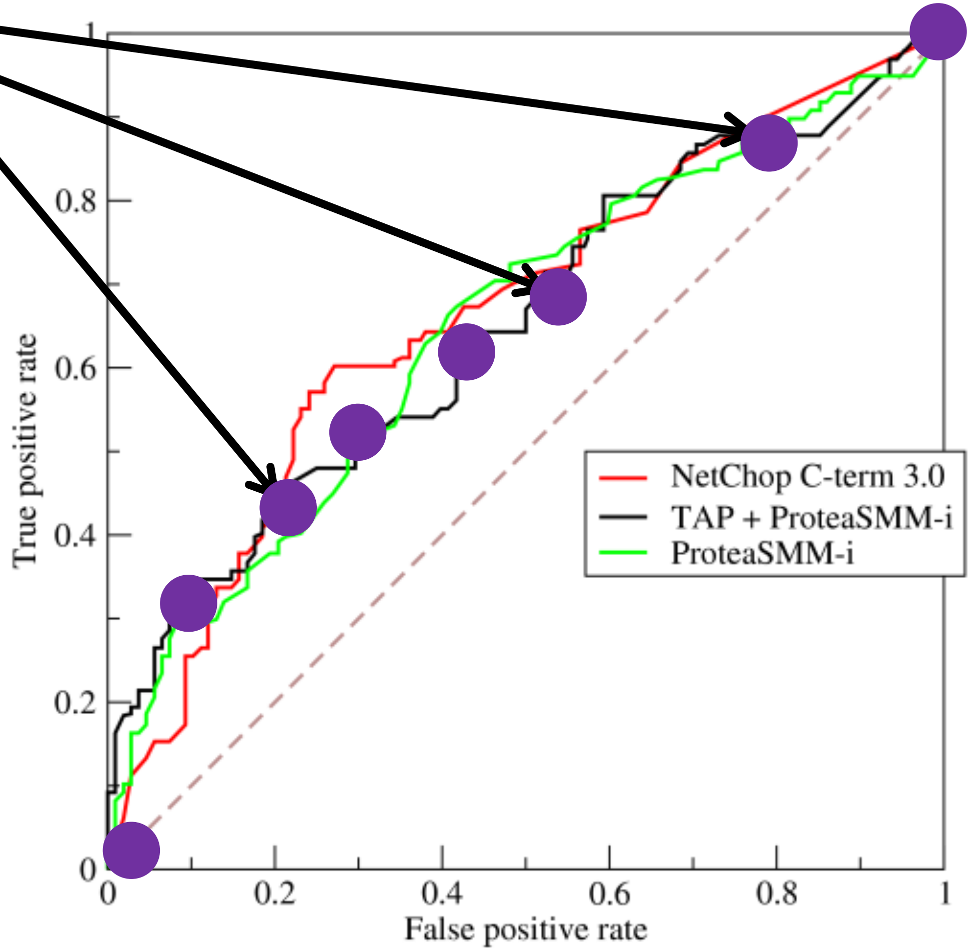


- higher sensitivity
- lower specificity

Visualization: ROC Curve

Each point on this curve corresponds to a choice of τ

(a.k.a sensitivity)



(a.k.a $1 -$ specificity)

Aside: Area under ROC curve is another metric people consider when evaluating $\hat{\beta}(Z)$

Optimizing Prediction Threshold

- Consider hyperparameter τ for the threshold:

$$f_\beta(x) = 1(\beta^\top x \geq \tau)$$

- Unlike most hyperparameters, we choose this one **after** we have already fit the model on the training data
  - Then, choose the value of τ that optimizes the desired metric
  - Fit using validation data (training data is OK if needed)

Optimizing Prediction Threshold

- **Step 1:** Compute the optimal parameters $\hat{\beta}(Z_{\text{train}})$
  - Using gradient descent on NLL loss over the training dataset
  - **Resulting model:** $f_{\hat{\beta}(Z_{\text{train}})}(x) = 1\left(\hat{\beta}(Z_{\text{train}})^\top x \geq 0\right)$

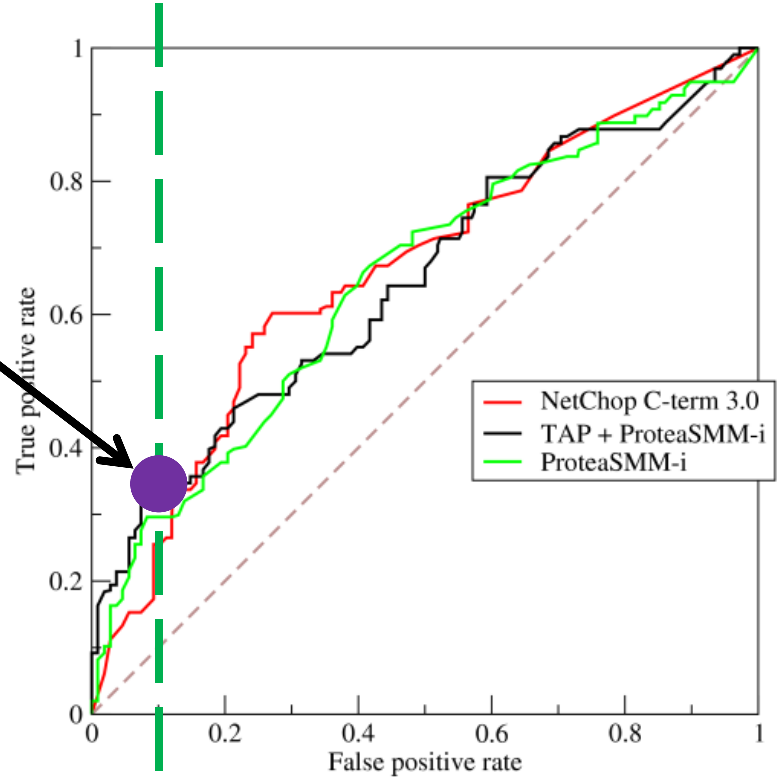
- **Step 2:** Modify threshold τ in model to optimize desired metric
  - Search over a fixed set of τ on the validation dataset
  - **Resulting model:** $f_{\hat{\beta}(Z_{\text{train}}),\hat{\tau}(Z_{\text{val}})}(x) = 1\left(\hat{\beta}(Z_{\text{train}})^\top x \geq \hat{\tau}(Z_{\text{val}})\right)$

- **Step 3:** Evaluate desired metric on test set

Choice of Metric Revisited

- **Common strategy:** Optimize one metric at fixed value of another

Choose τ corresponding to model at this point



$$\text{specificity} = 0.9$$

Optimizing a Classification Metric

- We are training a model to minimize NLL, but we have a different "true" metric that we actually want to optimize

- Two strategies (can be used together):
  - **Strategy 1:** Optimize prediction threshold threshold
  - **Strategy 2:** Upweight positive (or negative) examples

Class Re-Weighting

- **Weighted NLL:** Include a class-dependent weight w_y:

$$\ell(\beta; Z) = -\sum_{i=1}^{n} w_{y_i} \cdot \log p_\beta(y_i \mid x_i)$$

- **Intuition:** Tradeoff between accuracy on negative/positive examples
  - To improve sensitivity (true positive rate), upweight positive examples
  - To improve specificity (true negative rate), upweight negative examples

- Can use this strategy to learn β, and the first strategy to choose τ

Classification Metrics

- NLL isn’t usually the “true” metric
  - Instead, frequently used due to good computational properties

- Many choices with different meanings

- Typical strategy:
  - Learn β by minimizing the NLL loss
  - Choose class weights w_y and threshold τ to optimize desired metric