CSC 461: Machine Learning

Fall 2024

Logistic regression (part II)

Prof. Marco Alvarez, Computer Science
University of Rhode Island

Solving logistic regression

Maximum likelihood estimation (MLE)

- MLE principle
  - statistical method used to estimate the parameters of a probability distribution (e.g., mean and standard deviation for normal distributions)
    - if we observe $\mathscr{D}$, choose the parameters that make $\mathscr{D}$ most probable
  - many machine learning algorithms follow this principle
- The **likelihood** function
  - probability of observing the data given some parameters:

$$\mathscr{L}(\mathbf{w}) = P(X; \mathbf{w})$$

  - for independent and identically distributed observations:

$$\mathscr{L}(\mathbf{w}) = P(\mathbf{x_1}; \mathbf{w}) \cdot P(\mathbf{x_2}; \mathbf{w}) \cdot \ldots \cdot P(\mathbf{x_n}; \mathbf{w})$$

MLE and logistic regression

- MLE objective
  - find $\mathbf{w}$ that **maximizes** the likelihood function

$$w^* = \arg\max_{\mathbf{w}} \mathscr{L}(\mathbf{w})$$

- Logistic regression
  - conditional data likelihood:

$$\mathscr{L}(\mathbf{w}) = \prod_{i=1}^{n} P\left(y^{(i)} \mid \mathbf{x}^{(i)}; \mathbf{w}\right)$$

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \prod_{i=1}^{n} P(y^{(i)} \mid \mathbf{x}^{(i)}; \mathbf{w})$$

Solving logistic regression

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \prod_{i=1}^{n} P\left(y^{(i)} \mid \mathbf{x}^{(i)}; \mathbf{w}\right)$$

$$= \arg\max_{\mathbf{w}} \log\left(\prod_{i=1}^{n} \frac{1}{1+e^{-y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)}}}\right)$$

$$= \arg\max_{\mathbf{w}} -\sum_{i=1}^{n} \log\left(1+e^{-y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)}}\right)$$

$$= \arg\max_{\mathbf{w}} -\sum_{i=1}^{n} \log\left(1+e^{-y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)}}\right)$$

negative log likelihood

$$= \arg\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \boxed{\log\left(1+e^{-y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)}}\right)}$$

error (loss) $e\left(h(\mathbf{x}), y\right)$

Objective function

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \boxed{\frac{1}{n}\sum_{i=1}^{n} \log\left(1+e^{-y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)}}\right)}$$

cross-entropy loss
(over a dataset)

- the loss function (objective) is convex
  - however, no closed-form solution
  - can use **gradient descent** or second-order methods
    - coming soon …

How to classify new data?

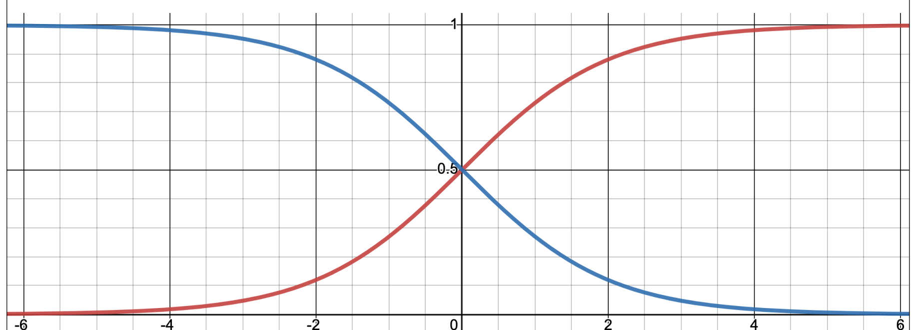
- Once the final hypothesis $h(\mathbf{x})$ is known …

$$h(\mathbf{x}) = P(y=+1 \mid \mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$$

- predict label $+1$ to input instance $\mathbf{x}$
  - if $p(+1 \mid \mathbf{x}) \geq 0.5$
- predict label -1 to input instance $\mathbf{x}$
  - if $p(+1 \mid \mathbf{x}) < 0.5$

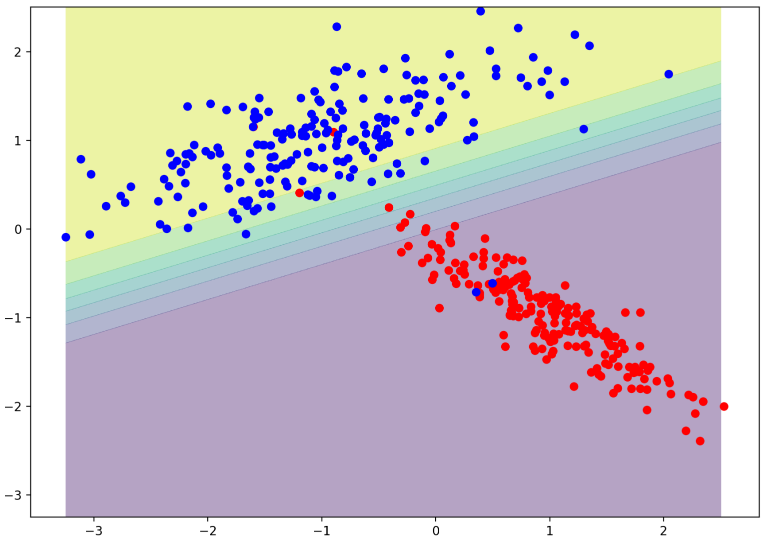
Decision boundary

$$\underbrace{P(y=+1 \mid \mathbf{x})}_{\sigma(\mathbf{w}^T\mathbf{x})} = \underbrace{P(y=-1 \mid \mathbf{x})}_{\sigma(-\mathbf{w}^T\mathbf{x})} = 0.5$$



Logistic regression finds a linear decision boundary with $\mathbf{w}^T\mathbf{x} = 0$

Decision boundary



Final remarks

- Simple classifier with **probabilistic outputs**
- Loss function is convex
  - guaranteed global minimum
- Robust to overfitting
  - use regularization (coming soon)
- Offers interpretability to weights
  - feature importance
- Decision boundary is still linear