DS 4400

Machine Learning and Data Mining I

Alina Oprea
Associate Professor
Khoury College of Computer Science
Northeastern University

October 13 2020

Outline

- Logistic regression
  - Cross-entropy objective
  - Gradient descent for logistic regression
- Project discussion
- Evaluation of classifiers
  - Metrics
  - ROC curves
- Linear Discriminant Analysis (LDA)

Logistic Regression

- Setup
  - Training data: $\{x_i, y_i\}$, for $i = 1, \ldots, N$
  - Labels: $y_i \in \{0,1\}$
- Goals
  - Learn $\boxed{P(Y = 1 | X = x)}$

LINEAR CLASSIFIER

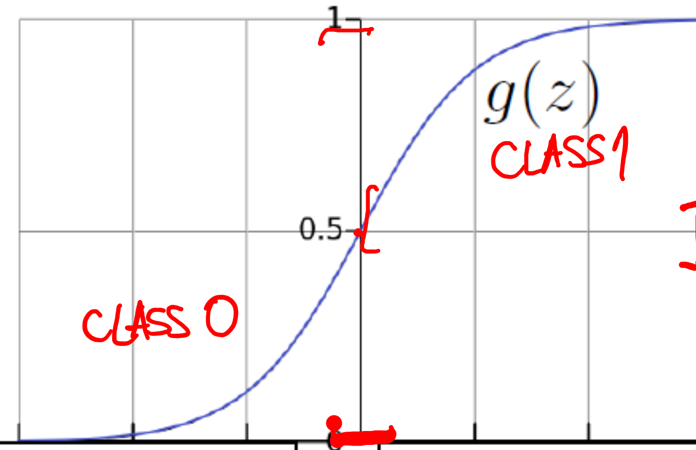
- Highlights
  - Probabilistic output
  - At the basis of more complex models (e.g., neural networks)
  - Supports regularization (Ridge, Lasso)
  - Can be trained with Gradient Descent

Logistic Regression

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boxed{g\left(\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{x}\right)}$$

SIGMOID $$g(z) = \frac{1}{1+e^{-z}}$$

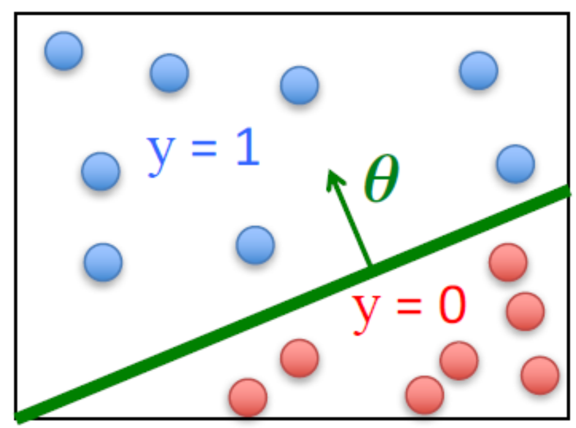
$\in [0,1]$



$\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{x}$ should be large negative values for negative instances

$\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{x}$ should be large positive values for positive instances

- Assume a threshold and...
  - Predict $Y = 1$ if $\boxed{h_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq 0.5}$
  - Predict $Y = 0$ if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) < 0.5$



$\theta^T x > 0$

Logistic Regression is a linear classifier!

Cross-Entropy Objective

$$P(Y = y_i | X = x_i; \theta) = \underbrace{\boxed{h_\theta(x_i)}}_{} ^{y_i} \underbrace{(1 - h_\theta(x_i))}_{P[Y=0|X=x_i]}^{1-y_i}$$

$= P[Y=1 | X=x_i]$

$y_i \in \{0,1\}$

MLE

$\theta = \text{argmax } L(\theta)$

$\theta = \text{argmax } \log L(\theta)$

$$\theta_{MLE} = \text{argmax}_\theta \sum_{i=1}^{N} \log P[Y = y_i | X = x_i; \theta]$$

$$= \text{argmax}_\theta \sum_{i=1}^{N} \underbrace{y_i \log h_\theta(x_i) + (1 - y_i) \log(1 - h_\theta(x_i))}_{\text{for } (x_i, y_i)}$$

Logistic regression objective

$$\min_\theta J(\theta)$$

$$\boxed{J(\theta)} = -\sum_{i=1}^{N} [\underbrace{y_i \log h_\theta(x_i) + (1 - y_i) \log(1 - h_\theta(x_i))}_{}]$$

Gradient Descent for Logistic Regression

$$J(\theta) = -\sum_{i=1}^{N}[y_{\mathrm{i}}\log h_\theta(x_i) + (1 - y_i)\log\left(1 - h_\theta(x_i)\right)]$$

Want $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \boxed{\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})}$$

simultaneous update
for j = 0 ... d

Gradient Computation

Gradient Descent for Logistic Regression

$$J(\theta) = -\sum_{i=1}^{N}[y_{\mathrm{i}} \log h_\theta(x_i) + (1 - y_i)\log\left(1 - h_\theta(x_i)\right)]$$

Want $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence (simultaneous update for j = 0 ... d)

$$\theta_j \leftarrow \theta_j - \alpha \sum_{i=1}^{N} \left(h_\theta(x_i) - y_i\right) x_{ij}$$

Gradient Descent for Logistic Regression

Want $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence (simultaneous update for j = 0 ... d)

$$\theta_0 \leftarrow \theta_0 - \alpha \sum_{i=1}^{N} (h_\theta(x_i) - y_i)$$

$$\theta_j \leftarrow \theta_j - \alpha \sum_{i=1}^{N} (h_\theta(x_i) - y_i) x_{ij}$$

This looks IDENTICAL to Linear Regression!

- However, the form of the model is very different:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^\mathsf{T}\boldsymbol{x}}}$$

Regularized Logistic Regression

$$J(\theta) = -\sum_{i=1}^{N}[y_{\mathrm{i}}\log h_\theta(x_i) + (1-y_i)\log\left(1-h_\theta(x_i)\right)]$$

- We can regularize logistic regression exactly as before:

$$J_{\text{regularized}}(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \boxed{\lambda \sum_{j=1}^{d} \theta_j^2}$$

$$= J(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}_{[1:d]}\|_2^2$$

RIDGE

$$J_{LASSO}(\theta) = J(\theta) + \boxed{\lambda \sum_{j=1}^{d} |\theta_j|}$$

Outline

- Logistic regression
  - Cross-entropy objective
  - Gradient descent for logistic regression
- Project discussion
- Evaluation of classifiers
  - Metrics
  - ROC curves
- Linear Discriminant Analysis (LDA)

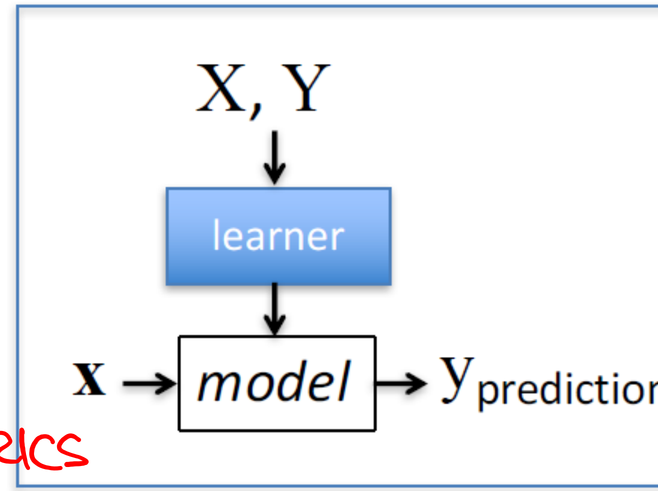
Classifier Evaluation

- Classification is a supervised learning problem
  - Prediction is binary or multi-class
- Classification techniques
  - Linear classifiers
    - Perceptron (online or batch mode)
    - Logistic regression (probabilistic interpretation)
  - Instance learners
    - kNN: need to store entire training data
- Cross-validation should be used for parameter selection and estimation of model error

Evaluation of classifiers

Given: labeled training data $X, Y = \{\langle \boldsymbol{x}_i, y_i \rangle\}_{i=1}^{n}$

- Assumes each $\boldsymbol{x}_i \sim \mathcal{D}(\mathcal{X})$



Train the model:

model ← *classifier*.train(X, Y)

CROSS-VALIDATION

TRAIN

VALIDATION → METRICS

Apply the model to new data:

- Given: new unlabeled instance $\boldsymbol{x} \sim \mathcal{D}(\mathcal{X})$

$y_{prediction}$ ← *model*.predict($\mathbf{x}$)

Classification Metrics

$$\boxed{\text{accuracy}} = \frac{\#\text{ correct predictions}}{\#\text{ test instances}} \in [0,1]$$

$$\boxed{\text{error}} = 1 - \text{accuracy} = \frac{\#\text{ incorrect predictions}}{\#\text{ test instances}} \in [0,1]$$

- Training set accuracy and error
- Testing set accuracy and error

Confusion Matrix

BINARY CLASSIFICATION

Given a dataset of P positive instances and N negative instances:

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

Positive

Negative

$P = TP + FN$

$N = TN + FP$

$$\text{ACCURACY} = \frac{TP + TN}{P + N}$$

$$\text{ERROR} = \frac{FP + FN}{P + N}$$

Confusion Matrix

- Given a dataset of P positive instances and N negative instances:

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

$$\text{accuracy} = \frac{TP + TN}{P + N}$$

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

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

Why One Metric is Not Enough

Assume that in your training data, Spam email is 1% of data, and Ham email is 99% of data

CLASS IMBALANCE

- Scenario 1
  - Have classifier always output HAM!
  - What is the accuracy?

ACC = 99%, ERR = 1%

PRECISION = 0

- Scenario 2
  - Predict one SPAM email as SPAM, all other emails as legitimate
  - What is the precision?

$\text{PREC} = \frac{TP}{TP+FP} = 1$; $\text{ACC} \simeq 99\%$

$\text{RECALL} = \frac{TP}{TP+FN} = \frac{1}{\#\text{SPAM}}$

- Scenario 3
  - Output always SPAM!
  - What is the recall?

RECALL = 1

ACC = 0.1

Precision & Recall

Precision

- the fraction of positive predictions that are correct
- P(is pos|predicted pos)

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

Recall

- fraction of positive instances that are identified
- P(predicted pos|is pos)

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

- You can get high recall (but low precision) by only predicting positive
- Recall is a non-decreasing function of the # positive predictions
- Typically, precision decreases as either the number of positive predictions or recall increases
- Precision & recall are widely used in information retrieval

F-Score

- Combined measure of precision/recall tradeoff

$$\boxed{F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}}$$

  - This is the harmonic mean of precision and recall
  - In the F_1 measure, precision and recall are weighted evenly
  - Can also have biased weightings that emphasize either precision or recall more (F_2 = 2 × recall; $F_{0.5}$ = 2 × precision)

- Limitations:
  - F-measure can exaggerate performance if balance between precision and recall is incorrect for application
    - Don't typically know balance ahead of time

A Word of Caution

- Consider binary classifiers A, B, C:

		A	.	B	.	C	.
TRUE LABELS		1	0	1	0	1	0
Predictions	1	0.9	0.1	0.8	0	0.78	0
	0	0	0	0.1	0.1	0.12	0.1

A Word of Caution

- Consider binary classifiers A, B, C:

		A	.	B	.	C	.
		1	0	1	0	1	0
Predictions	1	0.9	0.1	0.8	0	0.78	0
	0	0	0	0.1	0.1	0.12	0.1

- Clearly A is useless, since it always predicts 1
- B is slightly better than C
  - less probability mass wasted on the off-diagonals
- But, here are the performance metrics:

Metric	A	B	C
Accuracy	0.9	0.9	0.88
Precision	0.9	1.0	1.0
Recall	1.0	0.888	0.8667
F-score	0.947	0.941	0.9286

Acknowledgements

- Slides made using resources from:
  - Andrew Ng
  - Eric Eaton
  - David Sontag
- Thanks!