Machine Learning

Linear Models

Fabio Vandin

November 4^{th}, 2022

Feature normalization

Given the training set, we have "normalized" each feature x_i, $i=1,\ldots,d$ so that:

- the average of each feature across the training set is 0
- the standard deviation of each feature is 1

Data normalization is important:

- stability of the computation
- interpretability of linear models (weight is high $\Rightarrow$ feature is important)

If you build a model from normalized data $\Rightarrow$ the same normalization function must be applied to the data on which you make prediction.

Logistic Regression

Learn a function h from $\mathbb{R}^d$ to $[0, 1]$.

What can this be used for?

Classification!

Example: binary classification ($\mathcal{Y} = \{-1, 1\}$) - $h(\mathbf{x}) =$ *probability* that label of $\mathbf{x}$ is 1.

For simplicity of presentation, we consider binary classification with $\mathcal{Y} = \{-1, 1\}$, but similar considerations apply for multiclass classification.

Logistic Regression: Model

Hypothesis class $\mathcal{H}$: $\phi_{\text{sig}} \circ L_d$, where $\phi_{\text{sig}} : \mathbb{R} \rightarrow [0, 1]$ is *sigmoid function*

linear model

Logistic Regression: Model

Hypothesis class $\mathcal{H}$: $\phi_{\text{sig}} \circ L_d$, where $\phi_{\text{sig}} : \mathbb{R} \to [0, 1]$ is *sigmoid function*

Sigmoid function = "S-shaped" function

For logistic regression, the sigmoid ϕ_{sig} used is the *logistic regression*:

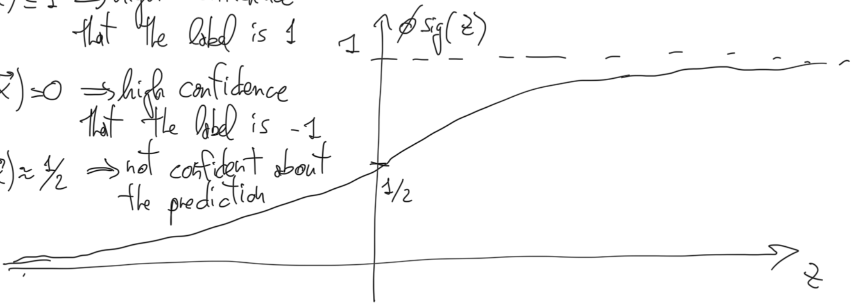
output of linear model $(\langle \vec{x}, \vec{w} \rangle)$

$$\phi_{\text{sig}}(z) = \frac{1}{1 + e^{-z}}$$

$h(\vec{x}) = 1 \Rightarrow$ high confidence that the label is 1

$h(\vec{x}) = 0 \Rightarrow$ high confidence that the label is -1

$h(\vec{x}) \approx 1/2 \Rightarrow$ not confident about the prediction



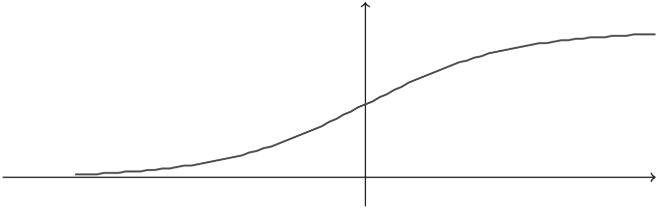
Logistic Regression: Model

Hypothesis class $\mathcal{H}$: $\phi_{\text{sig}} \circ L_d$, where $\phi_{\text{sig}} : \mathbb{R} \to [0, 1]$ is *sigmoid function*

Sigmoid function = "S-shaped" function

For logistic regression, the sigmoid ϕ_{sig} used is the *logistic regression*:

$$\phi_{\text{sig}}(z) = \frac{1}{1 + e^{-z}}$$



Therefore

$$H_{\mathsf{sig}} = \phi_{\mathsf{sig}} \circ L_d = \{\mathbf{x} \to \phi_{\mathsf{sig}}(\langle \mathbf{w}, \mathbf{x} \rangle) : \mathbf{w} \in \mathbb{R}^d\}$$

and $h_{\mathbf{w}}(\mathbf{x}) \in H_{\mathsf{sig}}$ is:

$$h_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1 + e^{-\langle \mathbf{w}, \mathbf{x} \rangle}}$$

Therefore

$$H_{\text{sig}} = \phi_{\text{sig}} \circ L_d = \{\mathbf{x} \rightarrow \phi_{\text{sig}}(\langle \mathbf{w}, \mathbf{x} \rangle) : \mathbf{w} \in \mathbb{R}^d\}$$

and $h_{\mathbf{w}}(\mathbf{x}) \in H_{\text{sig}}$ is:

$$h_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1 + e^{-\langle \mathbf{w}, \mathbf{x} \rangle}}$$

Main difference with binary classification with halfspaces: when $\langle \mathbf{w}, \mathbf{x} \rangle \approx 0$

- halfspace prediction is deterministically 1 or -1
- $\phi_{\text{sig}}(\langle \mathbf{w}, \mathbf{x} \rangle) \approx 1/2 \Rightarrow$ uncertainty in predicted label

Loss Function

probability that the label is 1

Need to define how bad it is to predict $h_{\mathbf{w}}(\mathbf{x}) \in [0, 1]$ given that true label is $y = \pm 1$

what do we want?

- if $y = +1 \Rightarrow h_{\vec{w}}(\vec{x})$ large
- if $y = -1 \Rightarrow h_{\vec{w}}(\vec{x})$ small

$\Rightarrow 1 - h_{\vec{w}}(\vec{x})$ large

Loss Function

Need to define how bad it is to predict $h_{\mathbf{w}}(\mathbf{x}) \in [0,1]$ given that true label is $y = \pm 1$

Desiderata

- $h_{\mathbf{w}}(\mathbf{x})$ "large" if $y = 1$
- $1 - h_{\mathbf{w}}(\mathbf{x})$ "large" if $y = -1$

Note that

$$\begin{aligned} 1 - h_{\mathbf{w}}(\mathbf{x}) &= 1 - \frac{1}{1 + e^{-\langle \mathbf{w}, \mathbf{x} \rangle}} \\ &= \frac{e^{-\langle \mathbf{w}, \mathbf{x} \rangle}}{1 + e^{-\langle \mathbf{w}, \mathbf{x} \rangle}} \cdot \frac{e^{\langle \vec{w}, \vec{x} \rangle}}{e^{\langle \vec{w}, \vec{x} \rangle}} \\ &= \frac{1}{1 + e^{\langle \mathbf{w}, \mathbf{x} \rangle}} \end{aligned}$$

Then *reasonable* loss function: increases monotonically with

$$\frac{1}{1 + e^{y\langle \mathbf{w}, \mathbf{x} \rangle}}$$

$\Rightarrow$ *reasonable* loss function: increases monotonically with

$$1 + e^{-y\langle \mathbf{w}, \mathbf{x} \rangle}$$

Loss function for logistic regression:

$$\ell(h_{\mathbf{w}}, (\mathbf{x}, y)) = \log\left(1 + e^{-y\langle \mathbf{w}, \mathbf{x} \rangle}\right)$$

Therefore, given training set $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$ the ERM problem for logistic regression is:

$$\arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{m}\sum_{i=1}^{m} \underbrace{\log\left(1 + e^{-y_i\langle \mathbf{w},\mathbf{x}_i\rangle}\right)}_{\ell(h_{\vec{w}}, (\vec{x}_i, y_i))}$$

Therefore, given training set $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$ the ERM problem for logistic regression is:

$$\arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{m}\sum_{i=1}^{m} \log\left(1 + e^{-y_i\langle \mathbf{w}, \mathbf{x}_i\rangle}\right)$$

Notes: logistic loss function is a *convex function* $\Rightarrow$ ERM problem can be solved efficiently

Definition may look a bit arbitrary: actually, ERM formulation is the same as the one arising from *Maximum Likelihood Estimation*