CSCI567 Machine Learning (Spring 2021)

Sirisha Rambhatla

University of Southern California

Feb 3, 2021

Outline

1 Logistics

3 Logistic regression

2 Review of Last Lecture

Outline

1 Logistics

3 Logistic regression

2 Review of Last Lecture

Logistics

- We'll be discussing the project today after the lecture.

Outline

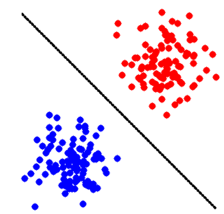
1. Logistics
3. Logistic regression
2. Review of Last Lecture

Summary

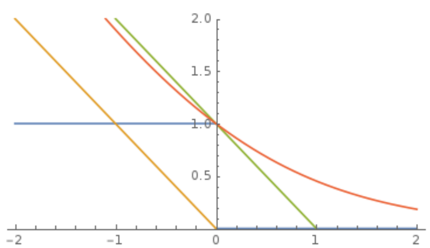
Linear models for **binary** classification:

Step 1. Model is the set of **separating hyperplanes**

$$\mathcal{F} = \{f(\boldsymbol{x}) = \mathsf{sgn}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}) \mid \boldsymbol{w} \in \mathbb{R}^{\mathsf{D}}\}$$



Step 2. Pick the **surrogate loss**



- perceptron loss $\ell_{\text{perceptron}}(z) = \max\{0, -z\}$ (used in Perceptron)
- hinge loss $\ell_{\text{hinge}}(z) = \max\{0, 1-z\}$(used in SVM and many others)
- logistic loss $\ell_{\text{logistic}}(z) = \log(1 + \exp(-z))$ (used in logistic regression)

Step 3. Find empirical risk minimizer (ERM):

$$\boldsymbol{w}^* = \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^{\mathsf{D}}} F(\boldsymbol{w}) = \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^{\mathsf{D}}} \frac{1}{N}\sum_{n=1}^{N} \ell(y_n \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n)$$

using

- **GD:** $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \nabla F(\boldsymbol{w})$
- **SGD:** $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \tilde{\nabla} F(\boldsymbol{w})$

Outline

1. Logistics
3. Logistic regression
   - A Probabilistic View
   - Optimization
2. Review of Last Lecture

A simple view

In one sentence: find the minimizer of

$$F(\boldsymbol{w}) = \frac{1}{N}\sum_{n=1}^{N} \ell_{\text{logistic}}(y_n \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_n)$$
$$= \frac{1}{N}\sum_{n=1}^{N} \ln(1 + e^{-y_n \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_n})$$

But why logistic loss? and why "regression"?

Predicting probability

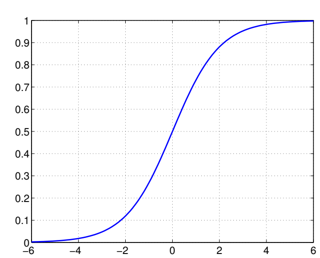
Instead of predicting a discrete label, can we *predict the probability of each label?* i.e. regress the probabilities

One way: **sigmoid function + linear model**

$$\mathbb{P}(y = +1 \mid \boldsymbol{x}; \boldsymbol{w}) = \sigma(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x})$$

where σ is the sigmoid function:

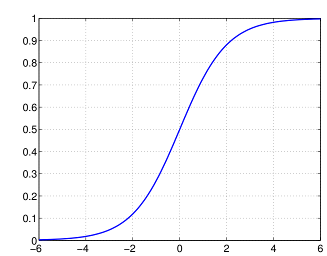
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



Properties

Properties of sigmoid $\sigma(z) = \frac{1}{1+e^{-z}}$

- between 0 and 1 (good as probability)
- $\sigma(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}) \geq 0.5 \Leftrightarrow \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x} \geq 0$, consistent with predicting the label with $\text{sgn}(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x})$
- larger $\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x} \Rightarrow$ larger $\sigma(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}) \Rightarrow$ higher *confidence* in label 1
- $\sigma(z) + \sigma(-z) = 1$ for all z



The probability of label -1 is naturally

$$1 - \mathbb{P}(y = +1 \mid \boldsymbol{x}; \boldsymbol{w}) = 1 - \sigma(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}) = \sigma(-\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x})$$

and thus

$$\mathbb{P}(y \mid \boldsymbol{x}; \boldsymbol{w}) = \sigma(y \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}) = \frac{1}{1 + e^{-y \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}}}$$

How to regress with discrete labels?

What we observe are labels, not probabilities.

Take a **probabilistic view**

- assume data is generated in this way by some $\boldsymbol{w}$
- perform Maximum Likelihood Estimation (MLE)

Specifically, what is the probability of seeing label $y_1, \cdots, y_n$ given $x_1, \cdots, x_n$, as a function of some $\boldsymbol{w}$?

$$P(\boldsymbol{w}) = \prod_{n=1}^{N} \mathbb{P}(y_n \mid \boldsymbol{x_n}; \boldsymbol{w})$$

MLE: find $\boldsymbol{w}^*$ that **maximizes the probability** $P(\boldsymbol{w})$

The MLE solution

$$\begin{aligned}
\boldsymbol{w}^* &= \operatorname*{argmax}_{\boldsymbol{w}} P(\boldsymbol{w}) = \operatorname*{argmax}_{\boldsymbol{w}} \prod_{n=1}^{N} \mathbb{P}(y_n \mid \boldsymbol{x_n}; \boldsymbol{w}) \\
&= \operatorname*{argmax}_{\boldsymbol{w}} \sum_{n=1}^{N} \ln \mathbb{P}(y_n \mid \boldsymbol{x_n}; \boldsymbol{w}) = \operatorname*{argmin}_{\boldsymbol{w}} \sum_{n=1}^{N} - \ln \mathbb{P}(y_n \mid \boldsymbol{x_n}; \boldsymbol{w}) \\
&= \operatorname*{argmin}_{\boldsymbol{w}} \sum_{n=1}^{N} \ln(1 + e^{-y_n \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x_n}}) = \operatorname*{argmin}_{\boldsymbol{w}} \sum_{n=1}^{N} \ell_{\text{logistic}}(y_n \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_n) \\
&= \operatorname*{argmin}_{\boldsymbol{w}} F(\boldsymbol{w})
\end{aligned}$$

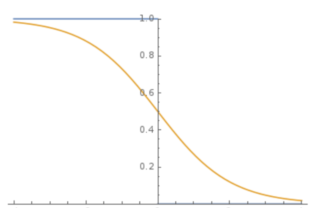
i.e. *minimizing logistic loss is exactly doing MLE for the sigmoid model!*

Let's apply SGD again

$$\begin{aligned}
\boldsymbol{w} &\leftarrow \boldsymbol{w} - \eta \tilde{\nabla} F(\boldsymbol{w}) \\
&= \boldsymbol{w} - \eta \nabla_{\boldsymbol{w}} \ell_{\text{logistic}}(y_n \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_n) \qquad (n \in [N] \text{ is drawn u.a.r.}) \\
&= \boldsymbol{w} - \eta \left(\frac{\partial \ell_{\text{logistic}}(z)}{\partial z} \Big|_{z = y_n \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_n} \right) y_n \boldsymbol{x}_n \\
&= \boldsymbol{w} - \eta \left(\frac{-e^{-z}}{1 + e^{-z}} \Big|_{z = y_n \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_n} \right) y_n \boldsymbol{x}_n \\
&= \boldsymbol{w} + \eta \sigma(-y_n \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_n) y_n \boldsymbol{x}_n \\
&= \boldsymbol{w} + \eta \mathbb{P}(-y_n \mid \boldsymbol{x}_n; \boldsymbol{w}) y_n \boldsymbol{x}_n
\end{aligned}$$

This is a *soft version of Perceptron!*

$\mathbb{P}(-y_n | \boldsymbol{x}_n; \boldsymbol{w})$ versus $\mathbb{I}[y_n \neq \text{sgn}(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_n)]$



A second-order method: Newton method

Recall the intuition of GD: we look at first-order **Taylor approximation**

$$F(\boldsymbol{w}) \approx F(\boldsymbol{w}^{(t)}) + \nabla F(\boldsymbol{w}^{(t)})^{\mathrm{T}} (\boldsymbol{w} - \boldsymbol{w}^{(t)})$$

What if we look at *second-order* Taylor approximation?

$$F(\boldsymbol{w}) \approx F(\boldsymbol{w}^{(t)}) + \nabla F(\boldsymbol{w}^{(t)})^{\mathrm{T}} (\boldsymbol{w} - \boldsymbol{w}^{(t)}) + \frac{1}{2} (\boldsymbol{w} - \boldsymbol{w}^{(t)})^{\mathrm{T}} \boldsymbol{H}_t (\boldsymbol{w} - \boldsymbol{w}^{(t)})$$

where $\boldsymbol{H}_t = \nabla^2 F(\boldsymbol{w}^{(t)}) \in \mathbb{R}^{\mathsf{D} \times \mathsf{D}}$ is the *Hessian* of F at $\boldsymbol{w}^{(t)}$, i.e.,

$$H_{t,ij} = \frac{\partial^2 F(\boldsymbol{w})}{\partial w_i \partial w_j} \Big|_{\boldsymbol{w} = \boldsymbol{w}^{(t)}}$$

(think "second derivative" when $D = 1$)

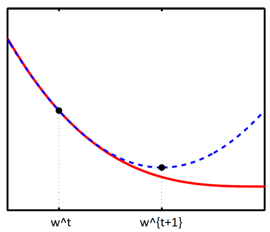
Deriving Newton method

If we minimize the second-order approximation (via "complete the square")

$$
\begin{aligned}
&F(\boldsymbol{w}) \\
&\approx F(\boldsymbol{w}^{(t)}) + \nabla F(\boldsymbol{w}^{(t)})^{\mathrm{T}}(\boldsymbol{w}-\boldsymbol{w}^{(t)}) + \frac{1}{2}(\boldsymbol{w}-\boldsymbol{w}^{(t)})^{\mathrm{T}}\boldsymbol{H}_t(\boldsymbol{w}-\boldsymbol{w}^{(t)}) \\
&= \frac{1}{2}\left(\boldsymbol{w}-\boldsymbol{w}^{(t)} + \boldsymbol{H}_t^{-1}\nabla F(\boldsymbol{w}^{(t)})\right)^{\mathrm{T}}\boldsymbol{H}_t\left(\boldsymbol{w}-\boldsymbol{w}^{(t)} + \boldsymbol{H}_t^{-1}\nabla F(\boldsymbol{w}^{(t)})\right) + \text{cnt.}
\end{aligned}
$$

for convex F (so H_t is *positive semidefinite*)
we obtain **Newton method**:

$$\boldsymbol{w}^{(t+1)} \leftarrow \boldsymbol{w}^{(t)} - \boldsymbol{H}_t^{-1}\nabla F(\boldsymbol{w}^{(t)})$$



Comparing GD and Newton

$$\boldsymbol{w}^{(t+1)} \leftarrow \boldsymbol{w}^{(t)} - \eta\nabla F(\boldsymbol{w}^{(t)}) \qquad \text{(GD)}$$

$$\boldsymbol{w}^{(t+1)} \leftarrow \boldsymbol{w}^{(t)} - \boldsymbol{H}_t^{-1}\nabla F(\boldsymbol{w}^{(t)}) \qquad \text{(Newton)}$$

Both are iterative optimization procedures, but Newton method

- has no learning rate η (*so no tuning needed!*)
- converges *super fast* in terms of #iterations needed
  - e.g. how many iterations needed when applied to a quadratic?
- requires **second-order** information and is *slow* each iteration (there are many ways to improve it though)

Applying Newton to logistic loss

$$\nabla_{\boldsymbol{w}}\ell_{\text{logistic}}(y_n\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n) = -\sigma(-y_n\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n)y_n\boldsymbol{x}_n$$

$$
\begin{aligned}
\nabla^2_{\boldsymbol{w}}\ell_{\text{logistic}}(y_n\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n) &= \left(\left.\frac{\partial\sigma(z)}{\partial z}\right|_{z=-y_n\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n}\right) y_n^2\boldsymbol{x}_n\boldsymbol{x}_n^{\mathrm{T}} \\
&= \left(\left.\frac{e^{-z}}{(1+e^{-z})^2}\right|_{z=-y_n\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n}\right)\boldsymbol{x}_n\boldsymbol{x}_n^{\mathrm{T}} \\
&= \sigma(y_n\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n)\left(1-\sigma(y_n\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n)\right)\boldsymbol{x}_n\boldsymbol{x}_n^{\mathrm{T}}
\end{aligned}
$$