

CSCI567 Machine Learning (Spring 2021)

Sirisha Rambhatla

University of Southern California

Feb 3, 2021

Outline

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

Outline

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

Logistics

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

Outline

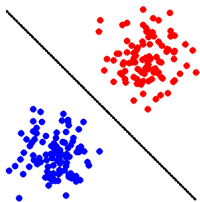
- 1 Logistics
- 2 Review of Last Lecture
- 3 Logistic regression

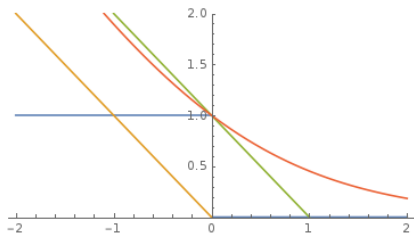
Summary

Linear models for **binary** classification:

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

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



Step 2. Pick the **surrogate loss**

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

Step 3. Find empirical risk minimizer (ERM):

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

using

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

Outline

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

A simple view

In one sentence: find the minimizer of

$$\begin{aligned} F(\mathbf{w}) &= \frac{1}{N} \sum_{n=1}^N \ell_{\text{logistic}}(y_n \mathbf{w}^T \mathbf{x}_n) \\ &= \frac{1}{N} \sum_{n=1}^N \ln(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}) \end{aligned}$$

A simple view

In one sentence: find the minimizer of

$$\begin{aligned} F(\mathbf{w}) &= \frac{1}{N} \sum_{n=1}^N \ell_{\text{logistic}}(y_n \mathbf{w}^T \mathbf{x}_n) \\ &= \frac{1}{N} \sum_{n=1}^N \ln(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}) \end{aligned}$$

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

Predicting probability

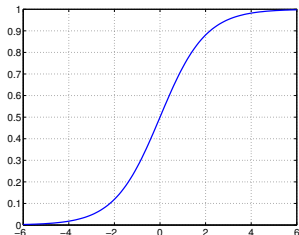
Instead of predicting a discrete label, can we *predict the probability of each label?* i.e. regress the probabilities

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

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

where σ is the sigmoid function:

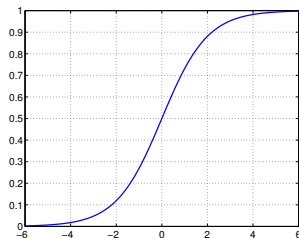
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



Properties

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

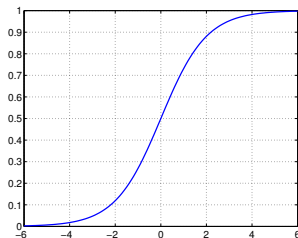
- between 0 and 1 (good as probability)



Properties

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

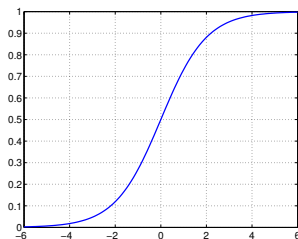
- between 0 and 1 (good as probability)
- $\sigma(\mathbf{w}^T \mathbf{x}) \geq 0.5 \Leftrightarrow \mathbf{w}^T \mathbf{x} \geq 0$, consistent with predicting the label with $\text{sgn}(\mathbf{w}^T \mathbf{x})$



Properties

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

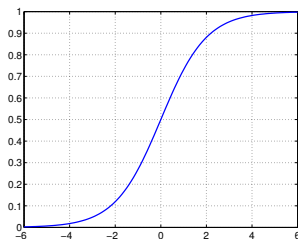
- between 0 and 1 (good as probability)
- $\sigma(\mathbf{w}^T \mathbf{x}) \geq 0.5 \Leftrightarrow \mathbf{w}^T \mathbf{x} \geq 0$, consistent with predicting the label with $\text{sgn}(\mathbf{w}^T \mathbf{x})$
- larger $\mathbf{w}^T \mathbf{x} \Rightarrow$ larger $\sigma(\mathbf{w}^T \mathbf{x}) \Rightarrow$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



Properties

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

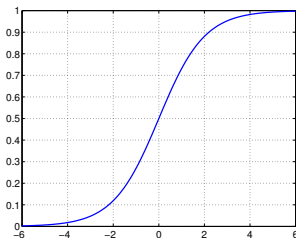
- between 0 and 1 (good as probability)
- $\sigma(\mathbf{w}^T \mathbf{x}) \geq 0.5 \Leftrightarrow \mathbf{w}^T \mathbf{x} \geq 0$, consistent with predicting the label with $\text{sgn}(\mathbf{w}^T \mathbf{x})$
- larger $\mathbf{w}^T \mathbf{x} \Rightarrow$ larger $\sigma(\mathbf{w}^T \mathbf{x}) \Rightarrow$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



Properties

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

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



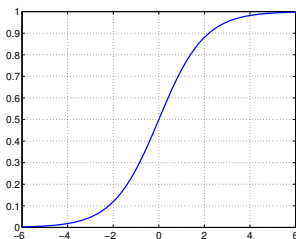
The probability of label -1 is naturally

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

Properties

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

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



The probability of label -1 is naturally

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

and thus

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

How to regress with discrete labels?

What we observe are labels, not probabilities.

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 w
- perform Maximum Likelihood Estimation (MLE)

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 w
- perform Maximum Likelihood Estimation (MLE)

Specifically, what is the probability of seeing label y_1, \dots, y_n given x_1, \dots, x_n , as a function of some w ?

$$P(w) = \prod_{n=1}^N \mathbb{P}(y_n \mid \mathbf{x}_n; w)$$

MLE: find w^* that **maximizes the probability** $P(w)$

The MLE solution

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmax}} P(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{n=1}^N \mathbb{P}(y_n \mid \mathbf{x}_n; \mathbf{w})$$

The MLE solution

$$\begin{aligned}\mathbf{w}^* &= \operatorname{argmax}_{\mathbf{w}} P(\mathbf{w}) = \operatorname{argmax}_{\mathbf{w}} \prod_{n=1}^N \mathbb{P}(y_n \mid \mathbf{x}_n; \mathbf{w}) \\ &= \operatorname{argmax}_{\mathbf{w}} \sum_{n=1}^N \ln \mathbb{P}(y_n \mid \mathbf{x}_n; \mathbf{w})\end{aligned}$$

The MLE solution

$$\begin{aligned}\mathbf{w}^* &= \operatorname{argmax}_{\mathbf{w}} P(\mathbf{w}) = \operatorname{argmax}_{\mathbf{w}} \prod_{n=1}^N \mathbb{P}(y_n \mid \mathbf{x}_n; \mathbf{w}) \\ &= \operatorname{argmax}_{\mathbf{w}} \sum_{n=1}^N \ln \mathbb{P}(y_n \mid \mathbf{x}_n; \mathbf{w}) = \operatorname{argmin}_{\mathbf{w}} \sum_{n=1}^N -\ln \mathbb{P}(y_n \mid \mathbf{x}_n; \mathbf{w})\end{aligned}$$

The MLE solution

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

The MLE solution

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

The MLE solution

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

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

Let's apply SGD again

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \tilde{\nabla} F(\mathbf{w})$$

Let's apply SGD again

$$\begin{aligned}\mathbf{w} &\leftarrow \mathbf{w} - \eta \tilde{\nabla} F(\mathbf{w}) \\ &= \mathbf{w} - \eta \nabla_{\mathbf{w}} \ell_{\text{logistic}}(y_n \mathbf{w}^T \mathbf{x}_n) \quad (n \in [N] \text{ is drawn u.a.r.})\end{aligned}$$

Let's apply SGD again

$$\begin{aligned}\mathbf{w} &\leftarrow \mathbf{w} - \eta \tilde{\nabla} F(\mathbf{w}) \\ &= \mathbf{w} - \eta \nabla_{\mathbf{w}} \ell_{\text{logistic}}(y_n \mathbf{w}^T \mathbf{x}_n) \quad (n \in [N] \text{ is drawn u.a.r.}) \\ &= \mathbf{w} - \eta \left(\left. \frac{\partial \ell_{\text{logistic}}(z)}{\partial z} \right|_{z=y_n \mathbf{w}^T \mathbf{x}_n} \right) y_n \mathbf{x}_n\end{aligned}$$

Let's apply SGD again

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

Let's apply SGD again

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

Let's apply SGD again

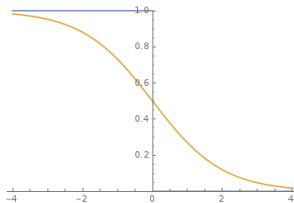
$$\begin{aligned}\mathbf{w} &\leftarrow \mathbf{w} - \eta \tilde{\nabla} F(\mathbf{w}) \\ &= \mathbf{w} - \eta \nabla_{\mathbf{w}} \ell_{\text{logistic}}(y_n \mathbf{w}^T \mathbf{x}_n) \quad (n \in [N] \text{ is drawn u.a.r.}) \\ &= \mathbf{w} - \eta \left(\left. \frac{\partial \ell_{\text{logistic}}(z)}{\partial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

Let's apply SGD again

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

This is a *soft version of Perceptron!*

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



A second-order method: Newton method

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

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

A second-order method: Newton method

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

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

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

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

A second-order method: Newton method

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

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

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

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

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

$$H_{t,ij} = \left. \frac{\partial^2 F(\mathbf{w})}{\partial w_i \partial w_j} \right|_{\mathbf{w}=\mathbf{w}^{(t)}}$$

(think “second derivative” when $D = 1$)

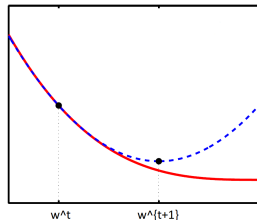
Deriving Newton method

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

$$F(\mathbf{w})$$

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

$$= \frac{1}{2} \left(\mathbf{w} - \mathbf{w}^{(t)} + \mathbf{H}_t^{-1} \nabla F(\mathbf{w}^{(t)}) \right)^T \mathbf{H}_t \left(\mathbf{w} - \mathbf{w}^{(t)} + \mathbf{H}_t^{-1} \nabla F(\mathbf{w}^{(t)}) \right) + \text{cnt.}$$



Deriving Newton method

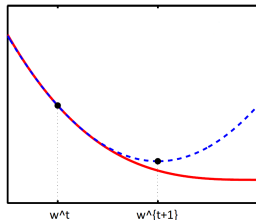
If we minimize the second-order approximation (via “complete the square”)

$$F(\mathbf{w})$$

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

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

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



Comparing GD and Newton

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

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

Both are iterative optimization procedures,

Comparing GD and Newton

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

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

Both are iterative optimization procedures, but Newton method

- has no learning rate η (*so no tuning needed!*)

Comparing GD and Newton

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

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

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

Comparing GD and Newton

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

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

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?

Comparing GD and Newton

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

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

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

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

Applying Newton to logistic loss

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

$$\nabla_{\mathbf{w}}^2 \ell_{\text{logistic}}(y_n \mathbf{w}^T \mathbf{x}_n) = \left(\frac{\partial \sigma(z)}{\partial z} \Big|_{z=-y_n \mathbf{w}^T \mathbf{x}_n} \right) y_n^2 \mathbf{x}_n \mathbf{x}_n^T$$

Applying Newton to logistic loss

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

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

Applying Newton to logistic loss

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

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