MATH 565: Lecture 12 (02/19/2026)

Today:
* Newton method for L_2-SVM
* Newton for Logistic regression
* Challenges with Newton

Recall: L_2-SVM

$$\nabla J_{L_2\text{-svm}}(\bar{w}) = D^T \Delta_{\bar{w}} (D\bar{w} - \bar{y}) \qquad \Delta_{\bar{w}} = [\operatorname{diag}(\delta(1 - y_i(\bar{w}^T \bar{x}_i) > 0)]$$

$$HJ_{L_2\text{-svm}}(\bar{w}) = D^T \Delta_{\bar{w}} D$$

$\Delta_{\bar{w}}$ drops points that do not meet margin condition; only (nearly) misclassified instances contribute to H.

Newton Update $\quad (\bar{w} \leftarrow \bar{w} - H^{-1} \nabla J)$

Here,
$$\bar{w} \leftarrow \bar{w} - (D^T \Delta_{\bar{w}} D)^{-1} (D^T \Delta_{\bar{w}} (D\bar{w} - \bar{y}))$$
$$= \underbrace{\bar{w} - \bar{w}}_{=\bar{0}} + (D^T \Delta_{\bar{w}} D)^{-1} D^T \Delta_{\bar{w}} \bar{y}$$

i.e., set $\bar{w} = (D^T \Delta_{\bar{w}} D)^{-1} D^T \Delta_{\bar{w}} \bar{y}$ — for linear regression, we got $\bar{w} = (D^T D)^{-1} D^T \bar{y}$

Line Search (Note: $J_{L_2\text{-svm}}$ is not quadratic)

$$\bar{w} \leftarrow \bar{w} - \alpha_t (D^T \Delta_{\bar{w}} D)^{-1} (D^T \Delta_{\bar{w}} (D\bar{w} - \bar{y}))$$

$$\Rightarrow \quad \bar{w} \leftarrow \bar{w}(1 - \alpha_t) + \alpha_t (D^T \Delta_{\bar{w}} D)^{-1} D^{-1} \Delta_{\bar{w}} \bar{y}$$

$\alpha_t > 1$ is possible here.

We can derive similar expressions for $J_{L_2\text{-svm}}$ with regularization term.

Newton Method for Logistic Regression SVM

$$J_{LR}(\bar{w}) = \sum_{i=1}^{n} \log\left(1+e^{-y_i(\bar{w}^T\bar{x}_i)}\right)$$

the function in general is $f(z) = \log(1+e^{-y_i z})$

$$= \sum_{i=1}^{n} J_i(\bar{w})$$

$$\nabla J_{LR} = \sum_{i=1}^{n} \frac{-y_i e^{-y_i(\bar{w}^T\bar{x}_i)}}{(1+e^{-y_i(\bar{w}^T\bar{x}_i)})}\bar{x}_i$$

$= p_i$ (see next page)

Quick aside on logistic regression

(linear) regression: $y = \beta_0 + \beta_1 w_1 + \cdots + \beta_d w_d \to \bar{y} = \bar{\beta}^T \bar{w}'$ for $\bar{\beta} = \begin{bmatrix}\beta_0\\ \beta_1\\ \vdots \\ \beta_d\end{bmatrix}$, $\bar{w}' = \begin{bmatrix}1\\ \bar{w}\end{bmatrix}$.

In logistic regression, we first compute the logit

$$z = \beta_0 + \beta_1 w_1 + \cdots + \beta_d w_d$$

and convert z to a probability:

$$P(y=1 \mid \bar{x}) = \frac{1}{1+e^{-z}}.$$

note: $0 < P < 1$ for all $z \in \mathbb{R}$.

We then convert this probability to a a 0/1 (binary) prediction:

If $P \geq 0.5$, predict 1
else if $P < 0.5$, predict 0.

With $f(z) = \log(1+e^{-y_i z})$, $f'(z) = \dfrac{-y_i e^{-y_i z}}{(1+e^{-y_i z})} = \dfrac{-y_i}{(1+e^{y_i z})}$.

We can interpret $\dfrac{1}{(1+e^{y_i z})}$ as a probability of misclassification of i^{th} point.

In detail, we interpret as the probability of correct prediction $p_i^c = \frac{1}{1+e^{-y_i(\bar{w}^T \bar{x}_i)}}$ and as the probability of wrong prediction $p_i = 1 - p_i^c = \frac{1}{1+e^{y_i(\bar{w}^T \bar{x}_i)}}$.

Check: $p_i^c + p_i = \frac{1}{1+e^{-y_i(\bar{w}^T \bar{x}_i)}} + \frac{1}{1+e^{y_i(\bar{w}^T \bar{x}_i)}}$

$$= \frac{2 + e^{y_i(\bar{w}^T \bar{x}_i)} + e^{-y_i(\bar{w}^T \bar{x}_i)}}{2 + e^{y_i(\bar{w}^T \bar{x}_i)} + e^{-y_i(\bar{w}^T \bar{x}_i)}} = 1.$$

Back to ∇J_{LR} ...

$$\nabla J_{LR}(\bar{w}) = -\sum_{i=1}^{n} y_i p_i \bar{x}_i \quad \text{where}$$

p_i = probability of misclassifying (or making a mistake on) the i^{th} point

We can write this expression as

$$\nabla J_{LR}(\bar{w}) = -D^T P_{\bar{w}} \bar{y}$$

where $P_{\bar{w}} = [\mathrm{diag}(p_i)]$, the diagonal $n \times n$ matrix of probabilities of mistakes.

$P_{\bar{w}}$ is a "soft" version of $\Delta_{\bar{w}}$ matrix in L_2-SVM, which removes well-classified instances.

with $\nabla J_{LR}(\bar{w}) = -\sum_{i=1}^{n} y_i p_i \bar{x}_i$ for $p_i = \frac{1}{1+e^{y_i(\bar{w}^T \bar{x}_i)}}$,

$$HJ_{LR} = \sum_{i=1}^{n} -y_i \bar{x}_i \left[\frac{\partial p_i}{\partial \bar{w}}\right]$$

$$\left[\frac{\partial p_i}{\partial \bar{w}}\right] = \frac{-y_i \cdot \bar{x}_i^T e^{y_i(\bar{w}^T \bar{x}_i)}}{\left(1+e^{y_i(\bar{w}^T \bar{x}_i)}\right)^2} = -y_i \underbrace{\frac{1}{\left(1+e^{y_i(\bar{w}^T\bar{x}_i)}\right)}}_{p_i} \cdot \underbrace{\frac{1}{\left(1+e^{-y_i(\bar{w}^T\bar{x}_i)}\right)}}_{(1-p_i)} \bar{x}_i^T$$

$$\Rightarrow HJ_{LR} = \sum_{i=1}^{n} \underbrace{(y_i)^2}_{=1} p_i(1-p_i) \underbrace{\bar{x}_i \bar{x}_i^T}_{\text{outer product}}$$

$$\Rightarrow HJ_{LR} = D^T U_{\bar{w}} D \quad \text{where}$$

$U_{\bar{w}} = \left[\text{diag}\left(p_i(1-p_i)\right)\right]$ is the diagonal matrix of uncertainties on classifying $\bar{x}_i$ (i^{th} instance).

When $p_i \approx 0$ or $p_i \approx 1$ $(0<p_i<1)$, the product $p_i(1-p_i) \approx 0$.
$p_i(1-p_i)$ is largest when $p_i = \frac{1}{2}$.

Recall, in L_2-SVM, $HJ_{L_2\text{-SVM}} = D^T \Delta_{\bar{w}} D$, which drops correctly classified points. In contrast, logistic regression SVM gives a soft weight to each point based on the level of uncertainty in its classification.

Newton Update

$$\bar{w} \longleftarrow \bar{w} + \alpha \left(D^T U_{\bar{w}} D\right)^{-1} D^{-1} P_{\bar{w}} \bar{y}$$

with line search

Newton Method for ML: Summary

Problem	Basic Update	Update with line search
Linear regression & Least squares classification	$\bar{w} = (D^T D)^{-1} D^T \bar{y}$	N/A
L_2 – SVM	$\bar{w} = \left(D^T \Delta_{\bar{w}} D\right)^{-1} D^T \Delta_{\bar{w}} y$	$\bar{w} \Leftarrow (1-\alpha_t)\bar{w} + \alpha_t \left(D^T \Delta_{\bar{w}} D\right)^{-1} D^T \Delta_{\bar{w}} y$
Logistic Regression SVM	$\bar{w} = \left(D^T U_{\bar{w}} D\right)^{-1} D^T P_{\bar{w}} \bar{y}$	$\bar{w} \leftarrow \bar{w} + \alpha_t \left(D^T U_{\bar{w}} D\right)^{-1} D^T P_{\bar{w}} \bar{y}$

Note the similarity is the form of all three updates, and how L_2–SVM cuts off well-classified instances but logistic regression SVM deals with all instances softly.

Newton Method: Challenges

1. Ill-conditioned Hessians

H is the sum of outer products $\bar{x}_i \bar{x}_i^T$ of marginally or incorrectly classified points.

Each $\bar{x}_i \bar{x}_i^T$ has rank 1, and H needs to full rank (d) for H^{-1} to exist, which usually happens when we have at least d points contributing. But this may not occur when we are close to an optimal $\bar{w}$.

Regularization $\left(\frac{\lambda}{2}\|\bar{w}\|^2\right)$ helps!

— more in the next lecture...