COMP 3105 Introduction to Machine Learning
Lecture 1: Linear Regression

Junfeng Wen

Carleton University

Fall 2025

Regression Examples: Estimate Real Numbers

Estimate fish weight

- Shape (length/width) $\mapsto$ weight

Annual production of a corn farm

- Rainfall/sunshine/pest levels $\mapsto$ production

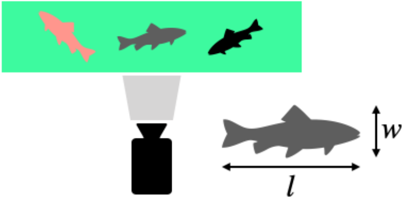
Glucose level of patients with diabetes

- Insulin injection, patient info $\mapsto$ glucose level

Market price of a new product

- Cost/quality/transportation $\mapsto$ price

An Example: Estimate Fish Weight



Every fish is measured by [length, width]

- E.g., [70 cm, 18 cm]
- Captured by camera on the conveyor belt

Want to estimate its *weight*

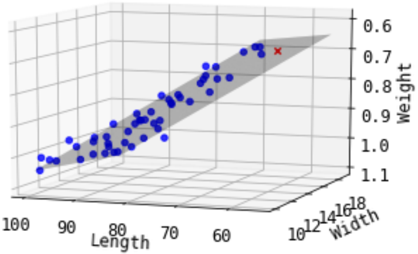
An Example: Estimate Fish Weight (Cont.)

One possible formulation

- Represent each fish as a (column) *vector* $\mathbf{x} = \begin{bmatrix} 1 \\ 70 \\ 18 \end{bmatrix}$ & $y = 0.7$
- Plot what we know (blue points; three axes)
- Find a "plane" (with augmented $\mathbf{x}$):

$$h(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 = \underset{1\times 3}{\mathbf{x}^\top} \underset{3\times 1}{\mathbf{w}}$$

- For new fish $\mathbf{x}_0$, predict $\widehat{y}_0 = h(\mathbf{x}_0)$



General Framework: Supervised Learning

$$\begin{array}{c}\text{Training data}\\ (x_1, y_1)\\ \vdots\\ (x_n, y_n)\end{array} \longrightarrow \boxed{\text{Learner}} \longrightarrow \text{Model } h : \mathcal{X} \mapsto \mathcal{Y}$$

x_i input in domain $\mathcal{X}$ (e.g., length and width $\mathbb{R}^2$)

- A.k.a. features, attributes, variables...
- Numeric (Length: 0.2, 1.3...)
- Categorical (Blood type: A/B/AB/O)
- Ordinal (Difficulty: easy/normal/hard; Discrete but rankable)

y_i output in range $\mathcal{Y}$ (e.g., weight $\mathbb{R}$)

- A.k.a. ground-truth label (or simply label), target variable...
- Supervised: training data have both x <u>and y</u>

Goal: Find h that predicts well for <u>novel test</u> $x \in \mathcal{X}$

The I.I.D. Assumption

ML requires certain assumption to work

Assume $\{(x_i, y_i)\}_{i=1}^{n}$ are <u>independently and identically distributed</u>

- Independently: $\forall i, (x_i, y_i)$ is freshly (independently) drawn from a probability distribution $P(x, y)$
- Identical: All examples are drawn from the same $P(x, y)$

The same applies to the <u>novel test</u> data (drawn from $P(x, y)$)

Linear Regression Formulation

Suppose

- Input vector $\mathbf{x}_i \in \mathcal{X} = \mathbb{R}^d$
- Output scalar $y_i \in \mathcal{Y} = \mathbb{R}$ (regression)
- Linear model $h_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}$ for some *parameters* $\mathbf{w} \in \mathbb{R}^d$

Goal

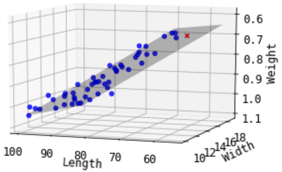
- Find a good $\mathbf{w}$ so that $\widehat{y} \triangleq h_{\mathbf{w}}(\mathbf{x})$ is "close" to y

Closeness? Loss function $L(\widehat{y}, y)$

- Absolute loss (L_1 loss) $L_1(\widehat{y}, y) = |\widehat{y} - y|$
- Squared loss (L_2 loss) $L_2(\widehat{y}, y) = \frac{1}{2}(\widehat{y} - y)^2$

Objective function: Empirical risk minimization (ERM)

$$\min_{\mathbf{w} \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} L(\mathbf{x}_i^\top \mathbf{w}, y_i)$$



Matrix/Vector Notations

The input matrix and output vector

$$X \triangleq \begin{bmatrix} - & \mathbf{x}_1^\top & - \\ - & \mathbf{x}_2^\top & - \\ & \vdots & \\ - & \mathbf{x}_n^\top & - \end{bmatrix} \in \mathbb{R}^{n \times d} \text{ and } \mathbf{y} \triangleq \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$$

Then the prediction

$$\widehat{\mathbf{y}} = X\mathbf{w} = \begin{bmatrix} - & \mathbf{x}_1^\top & - \\ - & \mathbf{x}_2^\top & - \\ & \vdots & \\ - & \mathbf{x}_n^\top & - \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} \in \mathbb{R}^n$$

Want $\widehat{\mathbf{y}} \approx \mathbf{y}$

Linear Regression Objective

$$\min_{\mathbf{w} \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} L(\mathbf{x}_i^\top \mathbf{w}, y_i)$$

Matrix/Vector notations

$$X \triangleq \begin{bmatrix} - & \mathbf{x}_1^\top & - \\ - & \mathbf{x}_2^\top & - \\ & \vdots & \\ - & \mathbf{x}_n^\top & - \end{bmatrix} \in \mathbb{R}^{n \times d} \text{ and } \mathbf{y} \triangleq \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$$

- Want $\widehat{\mathbf{y}} = X\mathbf{w} \approx \mathbf{y}$

With L_2 loss $L_2(\widehat{y}_i, y_i) = \frac{1}{2}(\widehat{y}_i - y_i)^2 = \frac{1}{2}(\mathbf{x}_i^\top \mathbf{w} - y_i)^2$

- Recall (squared) L_2 norm $\|\mathbf{a}\|_2^2 = a_1^2 + a_2^2 + \cdots + a_n^2 = \mathbf{a}^\top \mathbf{a}$

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2n} \|X\mathbf{w} - \mathbf{y}\|_2^2$$

How to Solve It? Direct Approach

$$\min_{\mathbf{w}\in\mathbb{R}^d} J(\mathbf{w}) \triangleq \frac{1}{2n}\|X\mathbf{w}-\mathbf{y}\|_2^2 = \frac{1}{2n}(X\mathbf{w}-\mathbf{y})^\top(X\mathbf{w}-\mathbf{y})$$

Note that this objective function J is <u>convex</u>

Compute the gradient

$$\begin{aligned}
\nabla J(\mathbf{w}) &= \frac{1}{2n}\nabla\left[(X\mathbf{w}-\mathbf{y})^\top(X\mathbf{w}-\mathbf{y})\right] && \text{if } f(u)=(au-b)^2,\ f'(u)\text{? (chain rule)}\\
&= \frac{1}{n}(X\mathbf{w}-\mathbf{y})\ \times_?\ \nabla(X\mathbf{w}-\mathbf{y}) && \text{if } f(u)=au-b,\ f'(u)\text{?}\\
&= \frac{1}{n}(\underset{n\times d}{X}\ \underset{d\times 1}{\mathbf{w}} - \underset{n\times 1}{\mathbf{y}})\ \times_?\ \underset{n\times d}{X} && \text{Dimension matched?}\\
&= \frac{1}{n}\underset{d\times n}{X^\top}\underset{n\times 1}{(X\mathbf{w}-\mathbf{y})} && \textcolor{red}{\text{Grad shares the same dim as the params}}
\end{aligned}$$

Set it to zero to find the best $\mathbf{w}$. Assume that $X^\top X$ is invertible

$$X^\top(X\mathbf{w}-\mathbf{y}) = \mathbf{0}_d \iff X^\top X\mathbf{w} = X^\top\mathbf{y} \iff \mathbf{w}^* = (X^\top X)^{-1}X^\top\mathbf{y}$$

How to Solve It? Indirect Approach

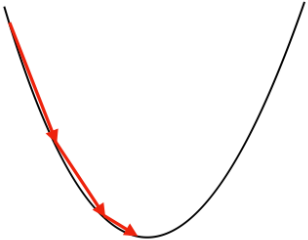
$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2n} \|X\mathbf{w} - \mathbf{y}\|_2^2 = (X^\top X)^{-1} X^\top \mathbf{y}$$

Direct approach: compute the gradient and set to zero

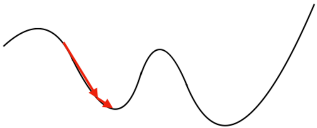
- Not always possible

Indirect approach: iterative algorithm (gradient descent)

- Initial $\mathbf{w}^{(0)}$ (e.g. $\mathbf{w}^{(0)} = \mathbf{0}_d$, a vector of all zeros)
- $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla J(\mathbf{w}^{(t)})$ with step size $\eta > 0$
- That's why convexity could be important



Convex

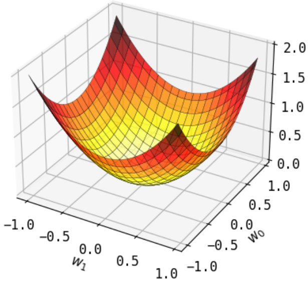


Non-convex

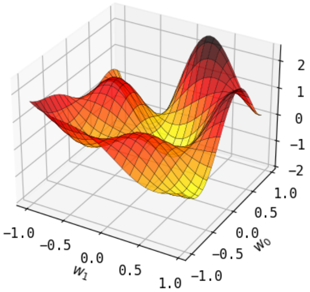
How to Solve It? Indirect Approach

Gradient descent 3D visualization

- $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla J(\mathbf{w}^{(t)})$ with step size $\eta > 0$



Convex

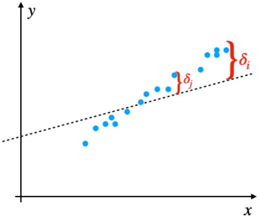


Non-convex

How about Other Losses? L_1 Loss

L_1 loss minimization

$$\min_{\mathbf{w}\in\mathbb{R}^d} \quad \frac{1}{n}\sum_{i=1}^{n} |\mathbf{x}_i^\top \mathbf{w} - y_i|$$



- Convex but not *smooth* (abs not differentiable at 0)
- Direct approach is not possible

Other solvers?

- For point $(\mathbf{x}_i, y_i)$, defined a "tolerance" $\delta_i \geq 0$

$$|\widehat{y}_i - y_i| \leq \delta_i \iff -\delta_i \leq \widehat{y}_i - y_i \leq \delta_i$$
$$\iff \mathbf{x}_i^\top \mathbf{w} - y_i \leq \delta_i \ \underline{\text{and}} \ y_i - \mathbf{x}_i^\top \mathbf{w} \leq \delta_i$$

- Minimize sum of tolerances $\min_{\mathbf{w},\delta_i} \ \frac{1}{n}\sum_{i=1}^{n} \delta_i$

L_1 Loss Minimization in Matrix Form

Using matrix/vector notations

- Recall L_1 norm $\|\mathbf{a}\|_1 = |a_1| + |a_2| + \cdots + |a_n|$

$$\min_{\mathbf{w}\in\mathbb{R}^d} \quad \frac{1}{n}\|X\mathbf{w} - \mathbf{y}\|_1$$

Let $\boldsymbol{\delta} \in \mathbb{R}^n_+$ be the "tolerance" between $\widehat{\mathbf{y}}$ and $\mathbf{y}$

- Always non-negative $\boldsymbol{\delta} \succeq \mathbf{0}_n$

$$\begin{aligned}
\min_{\mathbf{w},\boldsymbol{\delta}} \quad & \boldsymbol{\delta}^\top \mathbf{1}_n = \sum_{i=1}^{n} \delta_i \ \text{ Want tolerance sum to be small} \\
\text{s.t.} \quad & \boldsymbol{\delta} \succeq \mathbf{0}_n \\
& X\mathbf{w} - \mathbf{y} \preceq \boldsymbol{\delta} \quad \text{Sandwich the difference} \\
& \mathbf{y} - X\mathbf{w} \preceq \boldsymbol{\delta} \qquad \text{on both sides} \Leftrightarrow |X\mathbf{w} - \mathbf{y}| \preceq \boldsymbol{\delta}
\end{aligned}$$

Linear programming (LP) problem (Why? Linear in both $\mathbf{w}$ and $\boldsymbol{\delta}$), which can be solved efficiently

How about Other Losses? L_∞ Loss

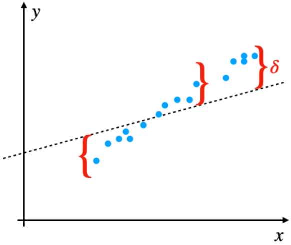
In general, the L_p loss: $\|X\mathbf{w} - \mathbf{y}\|_p^p, 0 \le p \le \infty$

L_∞ loss: $\|X\mathbf{w} - \mathbf{y}\|_\infty \triangleq \max_{i=1,\dots,n} \; |\mathbf{x}_i^\top \mathbf{w} - y_i|$

- Control the maximum gap (worse point)

Let $\delta \ge 0$ be the <u>scalar</u> "tolerance" (such that $|\mathbf{x}_i^\top \mathbf{w} - y_i| \le \delta$)

$$
\begin{aligned}
\min_{\mathbf{w},\delta} \quad & \delta \\
\text{s.t.} \quad & \delta \ge 0 \\
& X\mathbf{w} - \mathbf{y} \preceq \delta \cdot \mathbf{1}_n \\
& \mathbf{y} - X\mathbf{w} \preceq \delta \cdot \mathbf{1}_n
\end{aligned}
$$

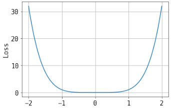


Again, an LP problem

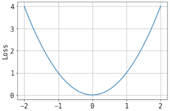
Which Loss to Use?

Different p values give different models $\mathbf{w}^*$

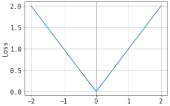
- $1 \leq p \leq \infty$: objective function is convex
- $1 < p < \infty$: objective function is smooth and differentiable
- $0 < p < 1$: *not* convex, but grow slowly as $\hat{y}$ deviates from y
  - More robust, not affected much by outliers



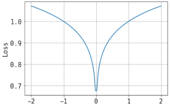
$p = 5$



$p = 2$



$p = 1$

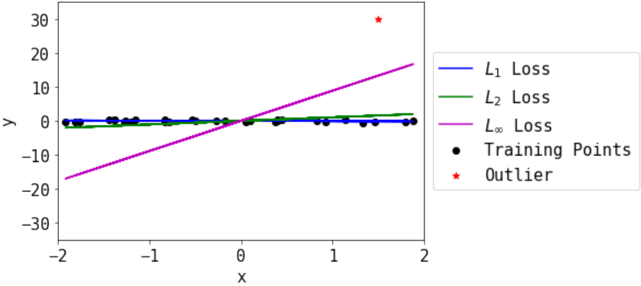


$p = 0.1$

Loss with different p values for one data point (x-axis: $\hat{y} - y$)

Which Loss to Use?

An example: Learning with outliers



- Large p (L_∞) chases the outlier (focus on the worse error)
- Small p (L_1) is robust (treat every error equally)

Always use small p? No, because...

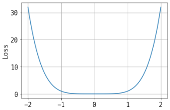
- $p < 1$ is not convex and difficult to minimize

Which Loss to Use?

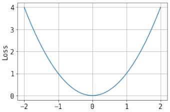
No free lunch

Pros and cons of convex versus robust losses

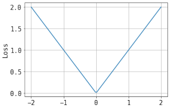
Loss type	Pros	Cons
Convex loss $p \in [1, \infty]$	Efficient optimization	Sensitive to outlier
Robust loss $p \in [0, 1)$	Slow growth, robust	Difficult to optimize



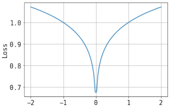
$p = 5$



$p = 2$



$p = 1$



$p = 0.1$

Recap

This lecture: linear regression models $\mathbf{y} \approx X\mathbf{w}$

- L_2 loss $\rightarrow$ analytic solution
- L_1 & L_∞ losses $\rightarrow$ linear programming (LP)
- Different p: convex versus robust losses

References

Bishop and Nasrabadi (2006, Sec.3.1)
Hastie et al. (2009, Sec.2.3.1 & Sec.3.2)

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Appendix: CVXOPT LP Formulation for L_1 Loss

$$\begin{aligned} \min_{\mathbf{w},\boldsymbol{\delta}} \quad & \boldsymbol{\delta}^\top \mathbf{1}_n \\ \text{s.t.} \quad & \boldsymbol{\delta} \succeq \mathbf{0}_n \\ & X\mathbf{w} - \mathbf{y} \preceq \boldsymbol{\delta} \\ & \mathbf{y} - X\mathbf{w} \preceq \boldsymbol{\delta} \end{aligned} \qquad\qquad \begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & G\mathbf{x} \preceq \mathbf{h} \end{aligned}$$

$\mathbf{x}$ in CVXOPT means unknowns (not a training input vector)

$$\mathbf{x} = \begin{bmatrix} \underset{d\times 1}{\mathbf{w}} \\ \underset{n\times 1}{\boldsymbol{\delta}} \end{bmatrix} \in \mathbb{R}^{d+n}$$

Want $\mathbf{c}^\top \mathbf{x} = \boldsymbol{\delta}^\top \mathbf{1}_n = \sum_{i=1}^n \delta_i$. What is $\mathbf{c} \in \mathbb{R}^{d+n}$ then?

$$\mathbf{c}^\top \mathbf{x} = \left[\underbrace{0, 0, \ldots, 0}_{d \text{ zeros}}, \underbrace{1, 1, \ldots, 1}_{n \text{ ones}},\right] \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\delta} \end{bmatrix} = \boldsymbol{\delta}^\top \mathbf{1}_n$$

Appendix: CVXOPT LP Formulation for L_1 Loss

$$\begin{aligned} \text{s.t.} \quad & \boldsymbol{\delta} \succeq \mathbf{0}_n \\ & X\mathbf{w} - \mathbf{y} \preceq \boldsymbol{\delta} \\ & \mathbf{y} - X\mathbf{w} \preceq \boldsymbol{\delta} \end{aligned} \qquad\qquad \text{s.t.} \quad G\mathbf{x} \preceq \mathbf{h}$$

Constraints in CVXOPT

$$G\mathbf{x} = \begin{bmatrix} - & G_{1:} & - \\ - & G_{2:} & - \\ & \vdots & \\ - & G_{k:} & - \end{bmatrix} \mathbf{x} \preceq \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_k \end{bmatrix} = \mathbf{h}$$

Each row is a constraint

$$G_{i:}\mathbf{x} \leq h_i \quad i = 1, 2, \ldots, k$$

Appendix: CVXOPT LP Formulation for L_1 Loss

$$\begin{aligned} \text{s.t.} \quad & \boldsymbol{\delta} \succeq \mathbf{0}_n \iff -\boldsymbol{\delta} \preceq \mathbf{0}_n \qquad \text{s.t.} \quad G\mathbf{x} \preceq \mathbf{h} \\ & X\mathbf{w} - \mathbf{y} \preceq \boldsymbol{\delta} \\ & \mathbf{y} - X\mathbf{w} \preceq \boldsymbol{\delta} \end{aligned}$$

Three sets of constraints (color coded)

$$\underset{3n\times(d+n)}{G} \cdot \underset{(d+n)\times 1}{\mathbf{x}} = \begin{bmatrix} \underset{n\times(d+n)}{G^{(1)}} \\ \underset{n\times(d+n)}{G^{(2)}} \\ \underset{n\times(d+n)}{G^{(3)}} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\delta} \end{bmatrix} \preceq \begin{bmatrix} \underset{n\times 1}{\mathbf{h}^{(1)}} \\ \underset{n\times 1}{\mathbf{h}^{(2)}} \\ \underset{n\times 1}{\mathbf{h}^{(3)}} \end{bmatrix} = \mathbf{h}$$

Want $\underset{n\times(d+n)}{G^{(1)}} \cdot \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\delta} \end{bmatrix} \preceq \underset{n\times 1}{\mathbf{h}^{(1)}} \Leftrightarrow -\boldsymbol{\delta} \preceq \mathbf{0}_n$. What are $G^{(1)}$ and $\mathbf{h}^{(1)}$?

$$\left[\underset{n\times d}{G^{(11)}} \ \underset{n\times n}{G^{(12)}}\right] \cdot \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\delta} \end{bmatrix} = \underset{n\times d}{G^{(11)}} \underset{d\times 1}{\mathbf{w}} + \underset{n\times n}{G^{(12)}} \underset{n\times 1}{\boldsymbol{\delta}} = -\boldsymbol{\delta} \preceq \mathbf{0}_n = \mathbf{h}^{(1)}$$

So $G^{(11)} = \mathbf{0}_{n\times d}$, $G^{(12)} = -I_n$ and $\mathbf{h}^{(1)} = \mathbf{0}_n$

Appendix: CVXOPT LP Formulation for L_1 Loss

$$\text{s.t.} \quad \boldsymbol{\delta} \succeq \mathbf{0}_n \iff -\boldsymbol{\delta} \preceq \mathbf{0}_n \qquad \text{s.t.} \quad G\mathbf{x} \preceq \mathbf{h}$$
$$X\mathbf{w} - \mathbf{y} \preceq \boldsymbol{\delta}$$
$$\mathbf{y} - X\mathbf{w} \preceq \boldsymbol{\delta}$$

Three sets of constraints (color coded)

$$\underset{3n\times(d+n)}{G} \cdot \underset{(d+n)\times 1}{\mathbf{x}} = \begin{bmatrix} \underset{n\times(d+n)}{G^{(1)}} \\ \underset{n\times(d+n)}{G^{(2)}} \\ \underset{n\times(d+n)}{G^{(3)}} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\delta} \end{bmatrix} \preceq \begin{bmatrix} \underset{n\times 1}{\mathbf{h}^{(1)}} \\ \underset{n\times 1}{\mathbf{h}^{(2)}} \\ \underset{n\times 1}{\mathbf{h}^{(3)}} \end{bmatrix} = \mathbf{h}$$

Want $\underset{n\times(d+n)}{G^{(2)}} \cdot \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\delta} \end{bmatrix} \preceq \underset{n\times 1}{\mathbf{h}^{(2)}} \Leftrightarrow X\mathbf{w} - \boldsymbol{\delta} \preceq \mathbf{y}$. What are $G^{(2)}, \mathbf{h}^{(2)}$?

$$\left[\underset{n\times d}{G^{(21)}} \;\; \underset{n\times n}{G^{(22)}}\right] \cdot \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\delta} \end{bmatrix} = \underset{n\times d}{G^{(21)}} \underset{d\times 1}{\mathbf{w}} + \underset{n\times n}{G^{(22)}} \underset{n\times 1}{\boldsymbol{\delta}} = X\mathbf{w} - \boldsymbol{\delta} \preceq \mathbf{y} = \mathbf{h}^{(2)}$$

So $G^{(21)} = X$, $G^{(22)} = -I_n$ and $\mathbf{h}^{(2)} = \mathbf{y}$

Appendix: CVXOPT LP Formulation for L_1 Loss

$$\text{s.t.} \quad \boldsymbol{\delta} \succeq \mathbf{0}_n \iff -\boldsymbol{\delta} \preceq \mathbf{0}_n \qquad \text{s.t.} \quad G\mathbf{x} \preceq \mathbf{h}$$

$$X\mathbf{w} - \mathbf{y} \preceq \boldsymbol{\delta}$$

$$\mathbf{y} - X\mathbf{w} \preceq \boldsymbol{\delta}$$

Three sets of constraints (color coded)

$$\underset{3n\times(d+n)}{G} \cdot \underset{(d+n)\times 1}{\mathbf{x}} = \begin{bmatrix} \underset{n\times(d+n)}{G^{(1)}} \\ \underset{n\times(d+n)}{G^{(2)}} \\ \underset{n\times(d+n)}{G^{(3)}} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\delta} \end{bmatrix} \preceq \begin{bmatrix} \underset{n\times 1}{\mathbf{h}^{(1)}} \\ \underset{n\times 1}{\mathbf{h}^{(2)}} \\ \underset{n\times 1}{\mathbf{h}^{(3)}} \end{bmatrix} = \mathbf{h}$$

Want $\underset{n\times(d+n)}{G^{(3)}} \cdot \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\delta} \end{bmatrix} \preceq \underset{n\times 1}{\mathbf{h}^{(3)}} \Leftrightarrow -X\mathbf{w} - \boldsymbol{\delta} \preceq -\mathbf{y}$. Then $G^{(3)}, \mathbf{h}^{(3)}$?

$$\left[\underset{n\times d}{G^{(31)}} \ \underset{n\times n}{G^{(32)}}\right] \cdot \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\delta} \end{bmatrix} = \underset{n\times d}{G^{(31)}} \underset{d\times 1}{\mathbf{w}} + \underset{n\times n}{G^{(32)}} \underset{n\times 1}{\boldsymbol{\delta}} = -X\mathbf{w} - \boldsymbol{\delta} \preceq -\mathbf{y} = \mathbf{h}^{(3)}$$

So $G^{(31)} = -X$, $G^{(32)} = -I_n$ and $\mathbf{h}^{(3)} = -\mathbf{y}$

Appendix: CVXOPT LP Formulation for L_1 Loss

To summarize

$$\min_{\mathbf{x}} \quad \underbrace{\left[0, 0, \ldots, 0, 1, 1, \ldots, 1\right]}_{\mathbf{c}^\top} \cdot \underbrace{\begin{bmatrix} \mathbf{w} \\ \boldsymbol{\delta} \end{bmatrix}}_{\mathbf{x}}$$

$$\text{s.t.} \quad \underbrace{\begin{bmatrix} \mathbf{0}_{n \times d} & -I_n \\ X & -I_n \\ -X & -I_n \end{bmatrix}}_{G} \cdot \underbrace{\begin{bmatrix} \mathbf{w} \\ \boldsymbol{\delta} \end{bmatrix}}_{\mathbf{x}} \preceq \underbrace{\begin{bmatrix} \mathbf{0}_n \\ \mathbf{y} \\ -\mathbf{y} \end{bmatrix}}_{\mathbf{h}}$$

After solving $\mathbf{x} = \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\delta} \end{bmatrix}$

- $\mathbf{w}$ model parameters
- $\boldsymbol{\delta}$ tolerances

Exercise/Assignment: L_∞ loss