Designing asymmetric losses

Manfred K. Warmuth

UC Santa Cruz

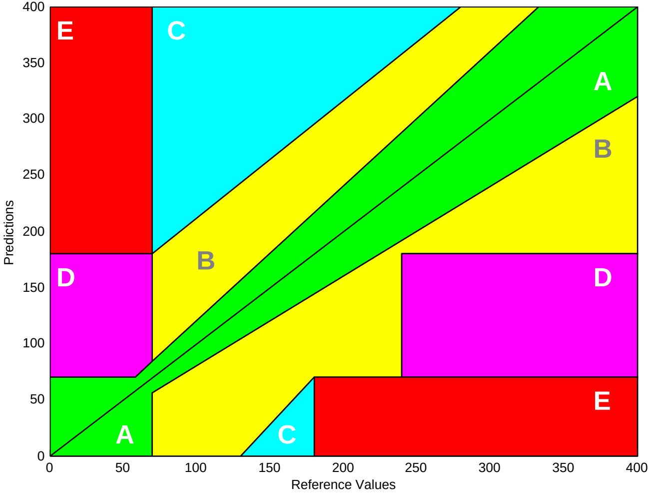
October 17, 2017, CMPS 242, UCSC

Joint work with Maya Hristakeva

Typical loss functions

Square loss, absolute loss, hinge loss

Clarke Grid - our running example loss

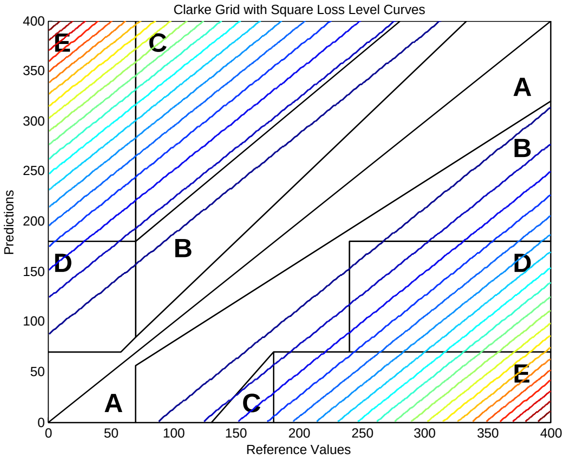


Designing Loss for Clarke Grid

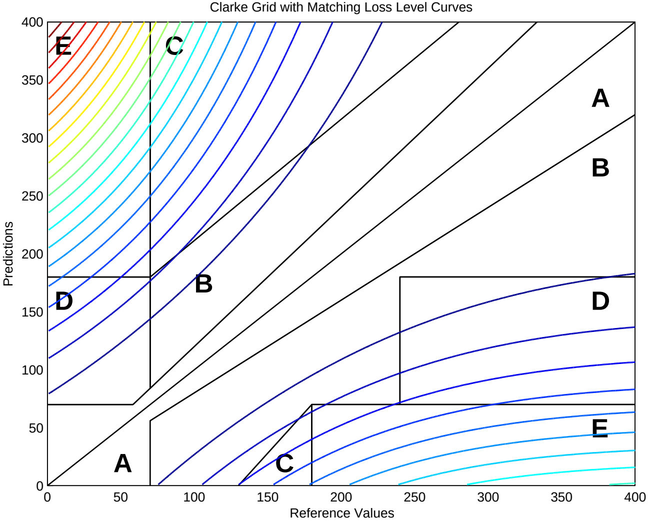
Goal: Accurately predict glucose levels of people with diabetes

- Asymmetry is needed ...
- Low concentrations more important than high ones
- Clarke Grid is the standard loss for this domain
- How do you optimize such a loss?

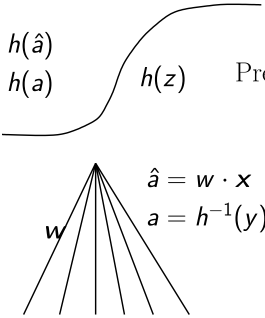
Square loss is bad fit



Clarke Grid versus our loss



Two setups based on a single neuron



Post example (x, y) Pre example (x, a)

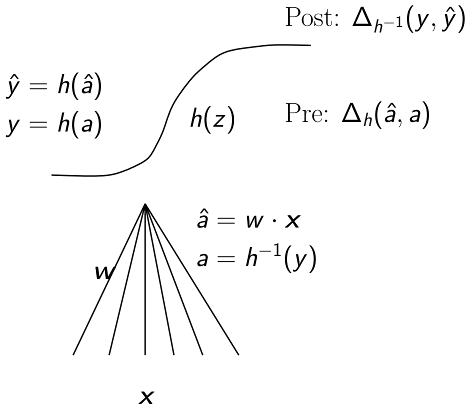
Two setups continued

- **Regression**
  - Pre examples are tuples $(\boldsymbol{x}_t, a_t)$
    - $\boldsymbol{x}_t \in \boldsymbol{R}^n$: data point (example)
    - $a_t \in \boldsymbol{R}$: true concentration (activity)
  - Linear activation label estimate: $\hat{a}_t = \boldsymbol{w} \cdot \boldsymbol{x}_t$
  - Loss between $\hat{a}_t$ and a_t
- **Classification**
  - Post examples are tuples $(\boldsymbol{x}_t, y_t)$
    - $\boldsymbol{x}_t \in \boldsymbol{R}^n$: data point (example)
    - $y_t \in [0, 1]$: true probability (label)
  - Probability label estimate: $\hat{y}_t = h(\hat{a}_t)$
  - Loss between y_t and $\hat{y}_t$

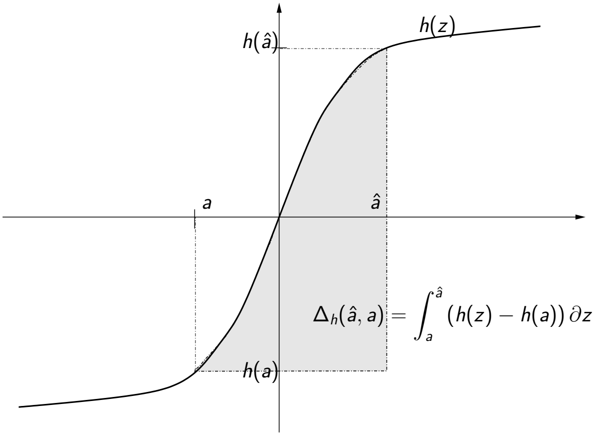
Why are we doing this?

- Want framework for designing asymmetric loss functions
- Loss functions should be steep in important areas and flat in unimportant areas
- Need flexible method for designing loss functions
- Running example: Clarke Grid for measuring Glucose

Single neuron again



Pre Matching Loss



$$\Delta_h(\hat{a}, a) = \int_a^{\hat{a}} (h(z) - h(a))\, \partial z$$

Pre Matching Loss Examples

$$\Delta_h(\hat{a}, a) \;=\; \int_a^{\hat{a}} (h(z) - h(a))\, dz$$

- Square Loss: $h(z) = z$

$$\Delta_h(\hat{a}, a) = \frac{1}{2}(\hat{a} - a)^2$$

- Pre Logistic Loss: $h(z) = \frac{e^z}{1+e^z}$

$$\Delta_h(\hat{a}, a) = \ln(1 + e^{\hat{a}}) - \ln(1 + e^a) - (\hat{a} - a)\underbrace{\frac{e^a}{1 + e^a}}_{y}$$

Post Matching Loss Examples

$$\Delta_{h^{-1}}(y,\hat{y}) \;=\; \int_{\hat{y}}^{y} \left(h^{-1}(p) - h^{-1}(\hat{y})\right) dp$$

- Square Loss: $h(z) = h^{-1}(z) = z$

$$\Delta_{h^{-1}}(y,\hat{y}) = \frac{1}{2}(\hat{y} - y)^2 = \frac{1}{2}(\hat{a} - a)^2 = \Delta_h(\hat{a}, a)$$

- Logistic Loss: $\hat{y} = h(z) = \frac{e^z}{1+e^z}$ and $h^{-1}(p) = \ln \frac{p}{1-p}$

$$\Delta_{h^{-1}}(y,\hat{y}) = y \ln \frac{y}{\hat{y}} + (1-y) \ln \frac{1-y}{1-\hat{y}}$$

Dual View of Matching Loss

$$
\begin{aligned}
\Delta_h(\hat{a}, a) &= \int_a^{\hat{a}} (h(z) - h(a))\, dz && \textbf{Pre} \\
&\overset{\substack{h(z)=p \quad h^{-1}(p)=z \\ dz=(h^{-1}(p))'\, dp}}{=} \int_{h(a)}^{h(\hat{a})} (p - h(a))\, (h^{-1}(p))'\, dp \\
&\overset{\text{Integ. by parts}}{=} \Big|_{h(a)}^{h(\hat{a})} (p - h(a)) h^{-1}(p) - \int_{h(a)}^{h(\hat{a})} (h^{-1}(p)) dp \\
&\overset{y=h(a) \quad \hat{y}=h(\hat{a})}{=} (\hat{y} - y) h^{-1}(\hat{y}) - \int_y^{\hat{y}} (h^{-1}(p)) dp \\
&= \int_{\hat{y}}^{y} \left(h^{-1}(p) - h^{-1}(\hat{y})\right) dp \\
&= \Delta_{h^{-1}}(y, \hat{y}) && \textbf{Post}
\end{aligned}
$$

Two domains

- **Pre domain:**
  - Examples: $(\boldsymbol{x}, a)$, for $a \in \mathcal{R}$
  - Prediction: $\hat{a} = \boldsymbol{x} \cdot \boldsymbol{w}$
  - Loss:
$$\Delta_h(\hat{a}, a) = \int_a^{\hat{a}} (h(z) - h(a))\, dz$$

- **Post domain:**
  - Examples: $(\boldsymbol{x}, y)$, for $y \in [0, 1]$
  - Prediction: $\hat{y} = h(\hat{a})$
  - Loss:
$$\Delta_{h^{-1}}(y, \hat{y}) = \int_{\hat{y}}^{y} \left(h^{-1}(p) - h^{-1}(\hat{y})\right) dp$$

Why are we doing this?

- Want to design good matching losses given a problem
- Post Domain:
  - Shifting and scaling $\boldsymbol{w}$ results in use of different part of transfer function
  - Shifting and scaling transfer function can be undone by shifting and scaling $\boldsymbol{w}$
- Pre Domain:
  - Shifting and scaling transfer function cannot be undone by shifting and scaling $\boldsymbol{w}$
  - Loss is "fixed" by choosing transfer function
  - Allows for design of "fancy" losses

Scaling and Shifting the Sigmoid

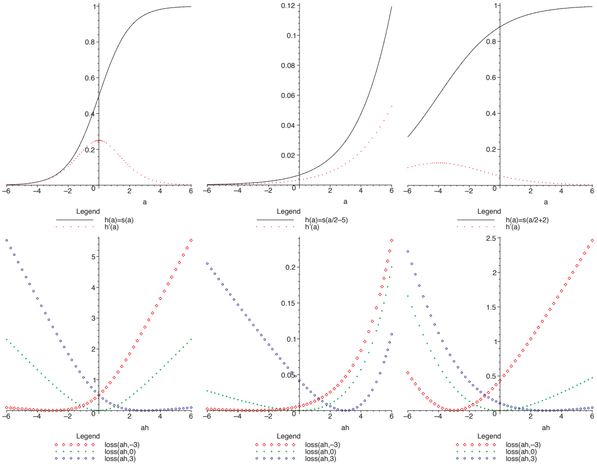
Define the transfer function $h(\hat{a})$ as

$$h(\hat{a}) = \frac{e^{\alpha(\boldsymbol{w}\cdot\boldsymbol{x}+\beta)}}{1 + e^{\alpha(\boldsymbol{w}\cdot\boldsymbol{x}+\beta)}},$$

where α scales the sigmoid and β shifts it

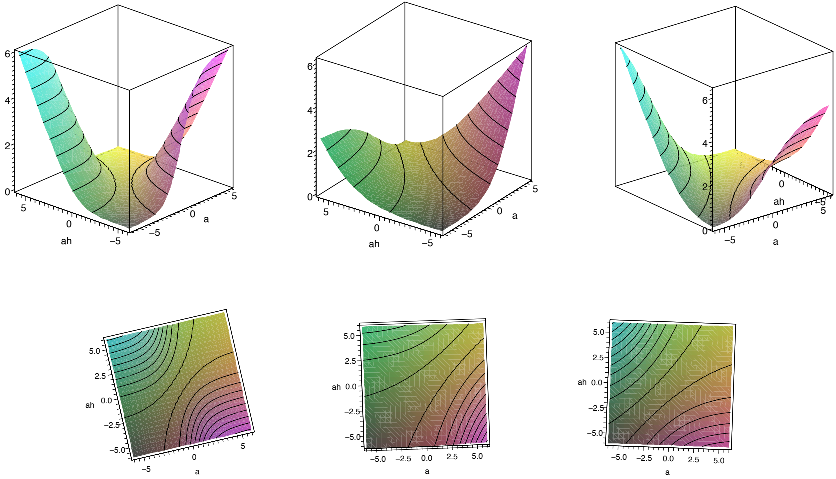
For Clarke Grid use piece of sigmoid that is steep on the small activations and then flattens out

Different Parts of Sigmoid



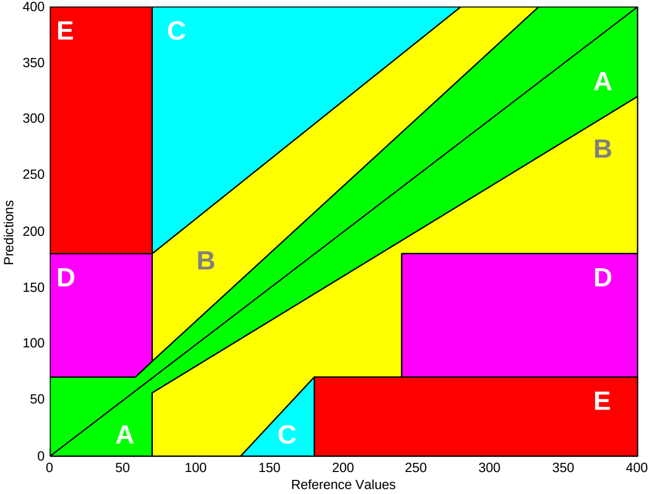
In the bottom row we plot the $\Delta(\hat{a}, a)$ as a function of the estimate $\hat{a}$ for fixed activities $a = -3, 0, 3$. Note that locally the losses are quadratic and the steepness of the bowl is determined by $h'(a)$.

3D View of the Loss



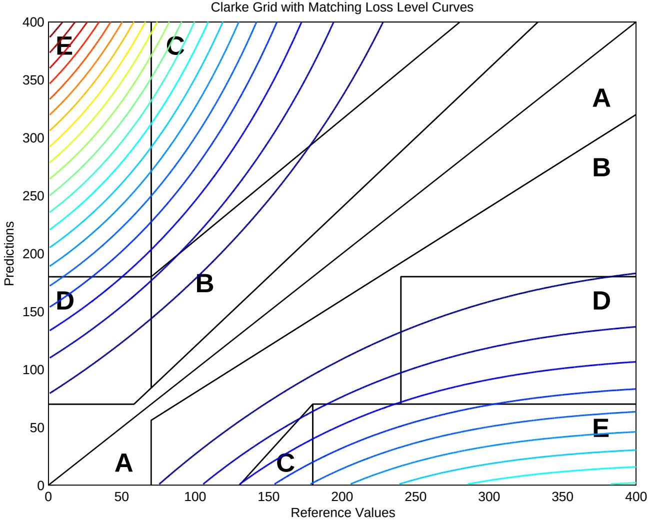
Regular Sigmoid, Left Piece of Sigmoid, Right Piece of Sigmoid

Clarke Grid

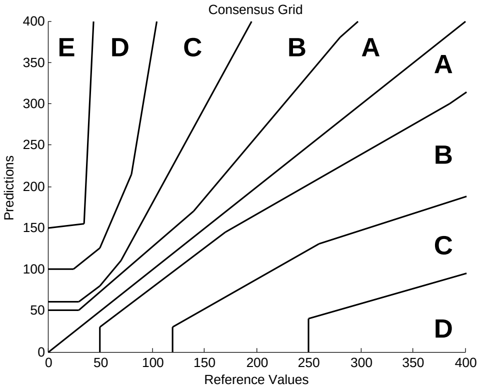


E
C
A
B
B
D
D
E
A
C
400
350
300
250
200
150
100
50
0
Predictions
0
50
100
150
200
250
300
350
400
Reference Values

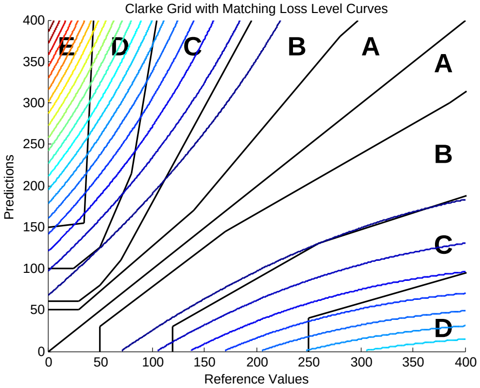
Overlay - Left piece - Low values are important



Improving on Clark
- a smooth more realistic version



Overlay with our loss - better fit



Non-convexity

Again - why are we doing this?

- Design good loss functions for given problem
- What is best loss?
- Square loss and logistic loss bad gold standards

Many chicken and egg problems

- Algorithms trained w/ one loss
  - performance measured on another
- Need nomenclature for asymmetric losses
  - training and performance measured on same loss
- **Use sigmoid**
  - identify piece of the sigmoid by stretch and shift

Motivation for my talk here

- Need a important problem with clear biological motivation
  - with asymmetric loss encoded in the problem
  - using asymmetric loss makes a big difference
- Public datasets
- Break the logjam !