Optimizing ROC Curves with a Sort-Based Surrogate Loss for Binary Classification and Changepoint Detection

Toby Dylan Hocking — toby.hocking@nau.edu
joint work with my student Jonathan Hillman
Machine Learning Research Lab — `http://ml.nau.edu`

Come to SICCS! Graduate Research Assistantships available!

Problem Setting and Related Work

Proposed algorithm

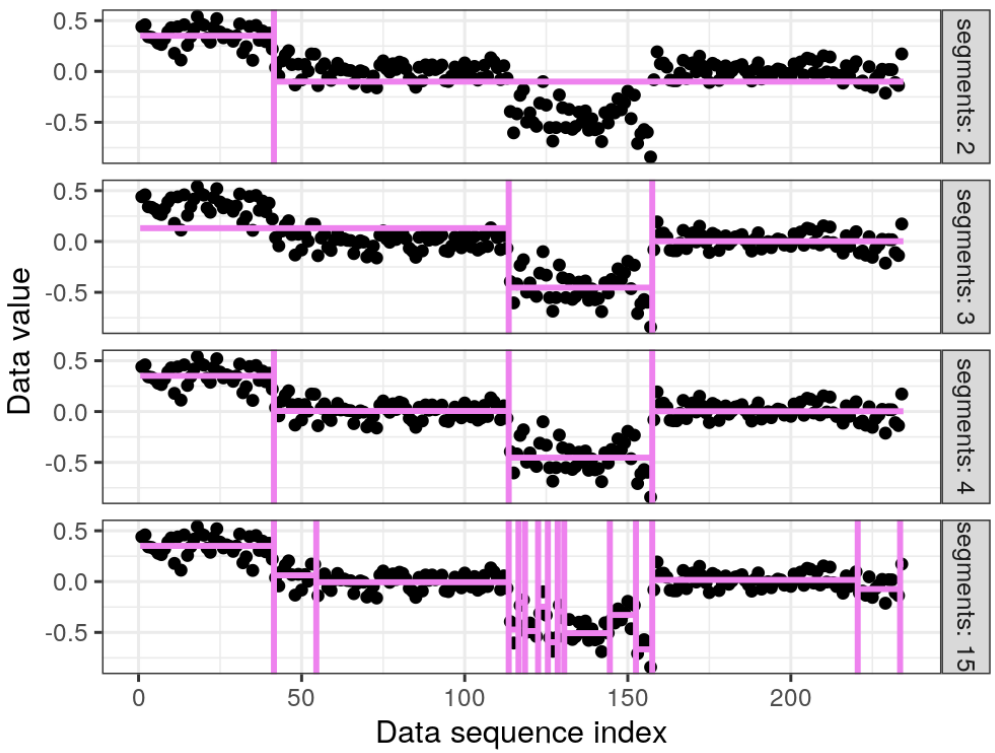
Results

Discussion and Conclusions

Problem: unsupervised changepoint detection

- We are given a data sequence $z_1, \ldots, z_T$ measured at T points over time/space.
- Ex: DNA copy number data for cancer diagnosis, $z_t \in \mathbb{R}$.
- The penalized changepoint problem is

$$\arg\min_{u_1,\ldots,u_T \in \mathbb{R}} \sum_{t=1}^{T} (u_t - z_t)^2 + \lambda \sum_{t=2}^{T} I[u_{t-1} \neq u_t].$$

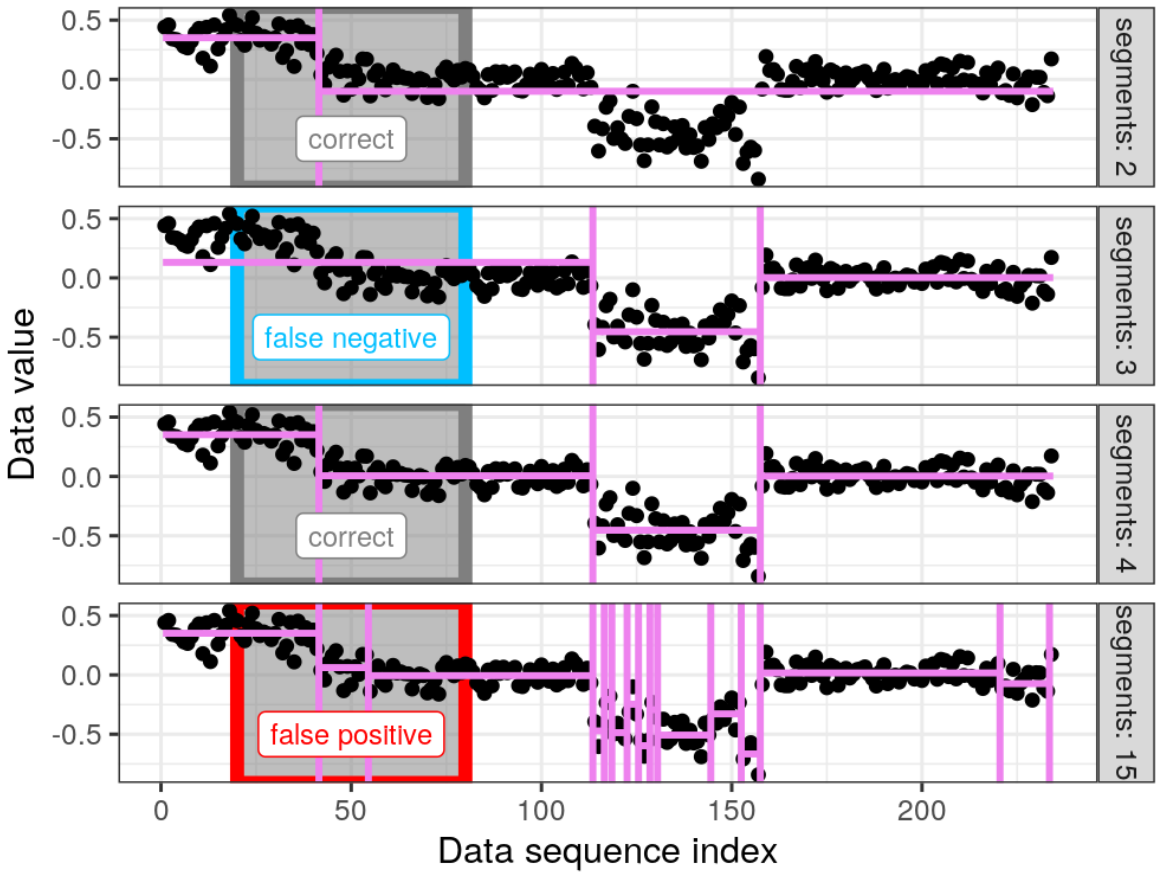


Larger penalty λ results in fewer changes/segments.

Smaller penalty λ results in more changes/segments.

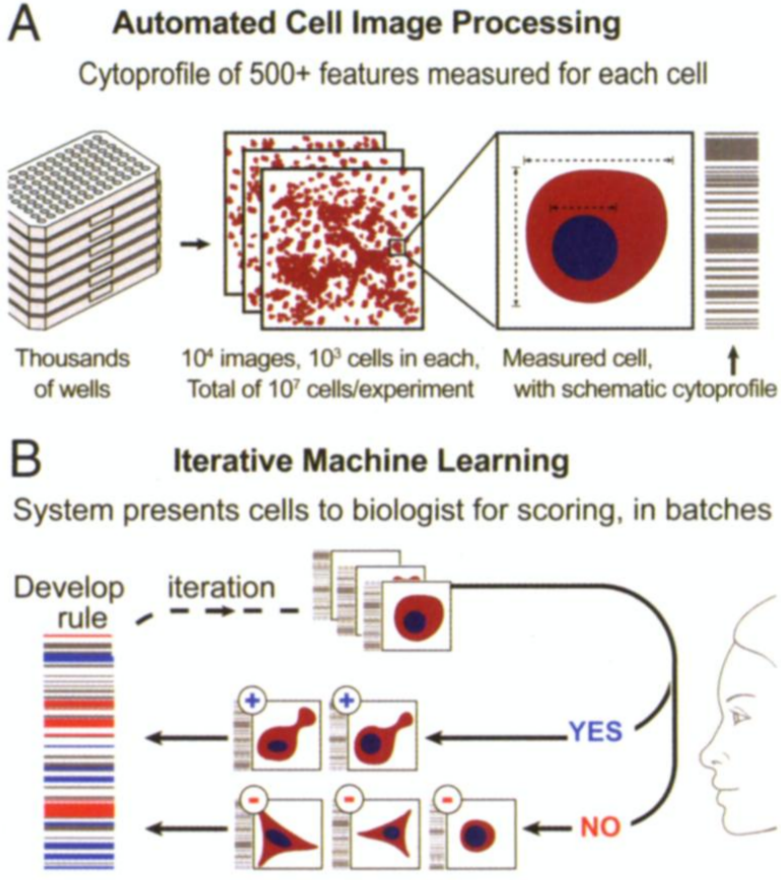
Problem: weakly supervised changepoint detection

- We are given a data sequence $\mathbf{z}$ with labeled regions L.
- We compute features $\mathbf{x} = \phi(\mathbf{z}) \in \mathbf{R}^p$ and want to learn a function $f(\mathbf{x}) = \log \lambda \in \mathbf{R}$ that minimizes label error.

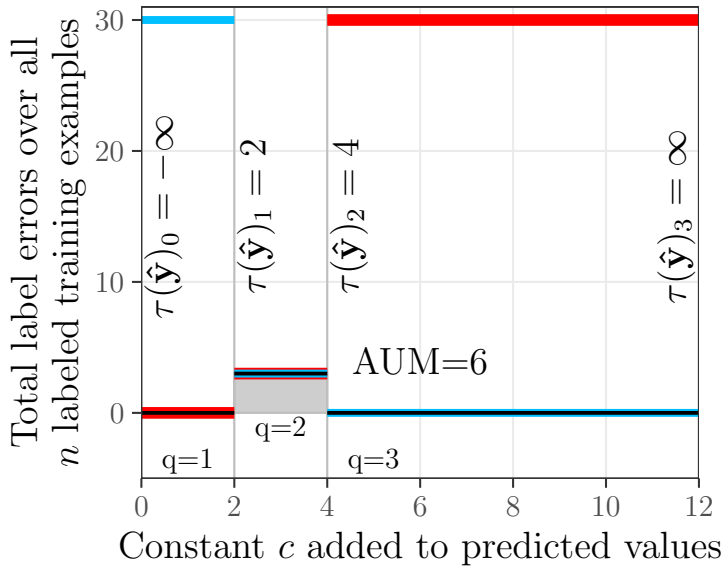


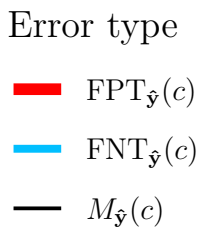
Problem: supervised binary classification

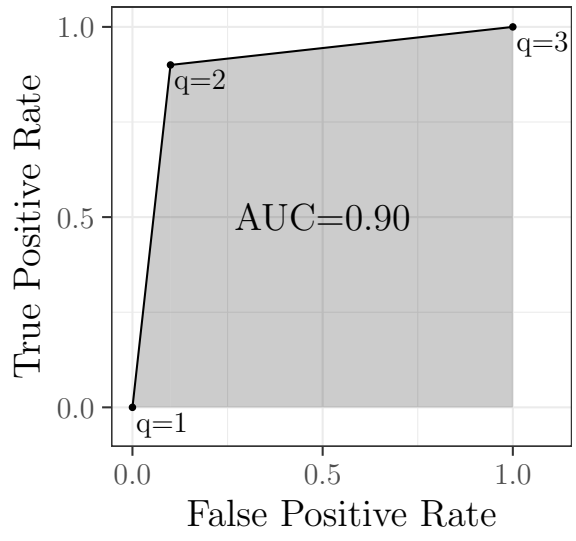
- Given pairs of inputs $\mathbf{x} \in \mathbb{R}^p$ and outputs $y \in \{0, 1\}$ can we learn $f(\mathbf{x}) = y$?
- Example: email, $\mathbf{x}$ =bag of words, y =spam or not.
- Example: images. Jones *et al.* PNAS 2009.



Looping ROC curve, simple synthetic example







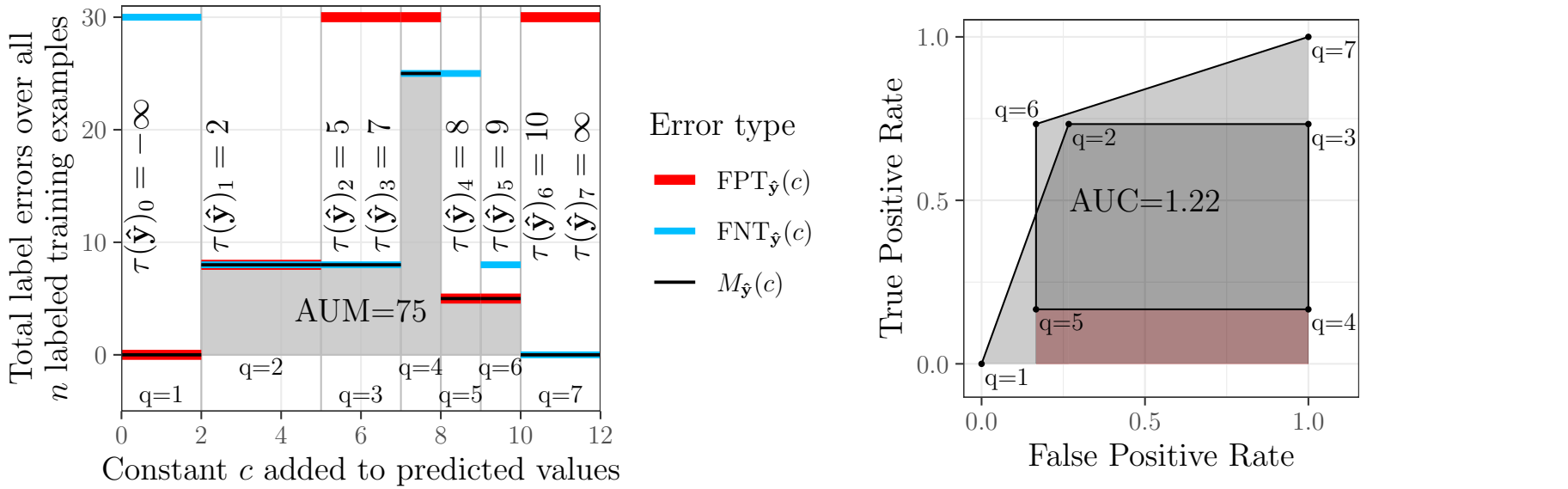
Problem Setting and Related Work

Proposed algorithm

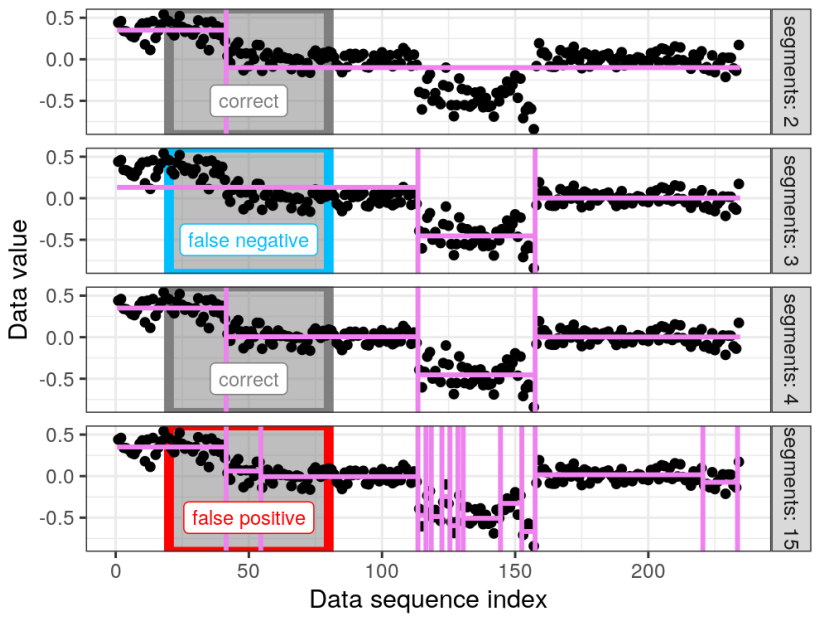
Results

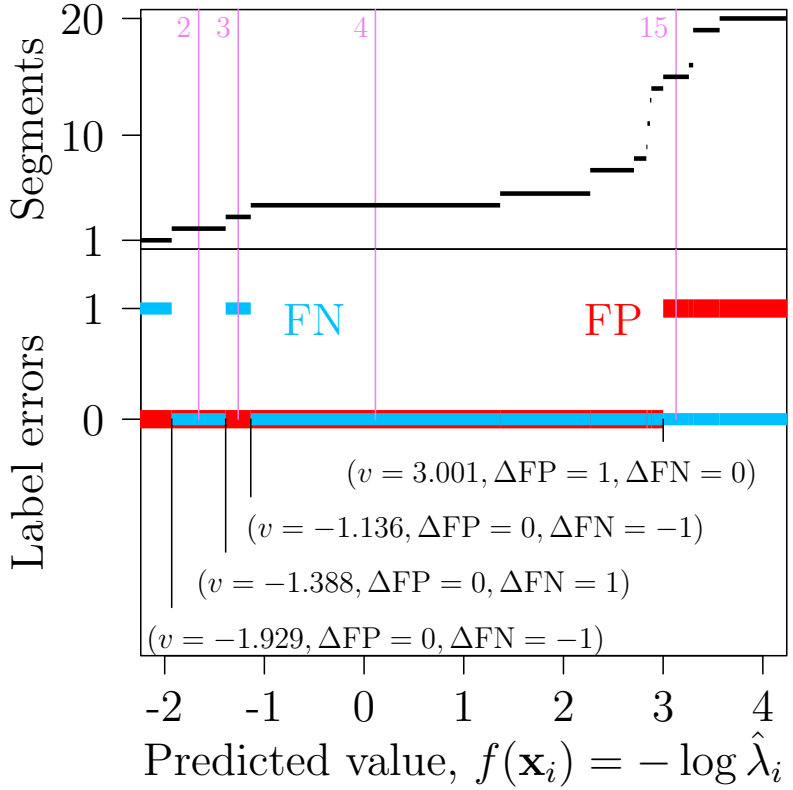
Discussion and Conclusions

Looping ROC curve, simple synthetic example



Real data example with non-monotonic label error





Notation

Let there be a total of B breakpoints in the error functions over all n labeled training examples, where each breakpoint $b \in \{1, \dots, B\}$ is represented by the tuple $(v_b, \Delta\text{FP}_b, \Delta\text{FN}_b, \mathcal{I}_b)$. The $\mathcal{I}_b \in \{1, \dots, n\}$ is an example index, so there are changes $\Delta\text{FP}_b, \Delta\text{FN}_b$ at predicted value $v_b \in \mathbb{R}$ in the error functions $\text{FP}_{\mathcal{I}_b}, \text{FN}_{\mathcal{I}_b}$ (Figure **??**). For example in binary classification, there are $B = n$ breakpoints (same as the number of labeled training examples); for each breakpoint $b \in \{1, \dots, B\}$ we have $v_b = 0$ and $\mathcal{I}_b = b$. For breakpoints b with positive labels $y_b = 1$ we have $\Delta\text{FP} = 0, \Delta\text{FN} = -1$, and for negative labels $y_b = -1$ we have $\Delta\text{FP} = 1, \Delta\text{FN} = 0$. In changepoint detection we have more general error functions, which may have more than one breakpoint per example.

Proposed algorithm

Input: Predictions $\hat{\mathbf{y}} \in \mathbb{R}^n$, breakpoints in error functions $v_b, \Delta\text{FP}_b, \Delta\text{FN}_b, \mathcal{I}_b$ for all $b \in \{1, \dots, B\}$.
Zero the AUM $\in \mathbb{R}$ and directional derivatives $\mathbf{D} \in \mathbb{R}^{n \times 2}$.
$t_b \leftarrow v_b - \hat{y}_{\mathcal{I}_b}$ for all b.
$s_1, \dots, s_B \leftarrow \text{SortedIndices}(t_1, \dots, t_B)$.
Compute $\underline{\text{FP}}_b, \overline{\text{FP}}_b, \underline{\text{FN}}_b, \overline{\text{FN}}_b$ for all b using $s_1, \dots, s_B$.
for $b \in \{2, \dots, B\}$ **do**
    $\text{AUM} \mathrel{+}= (t_{s_b} - t_{s_{b-1}}) \min\{\underline{\text{FP}}_b, \overline{\text{FN}}_b\}$.
for $b \in \{1, \dots, B\}$ **do**
    $\mathbf{D}_{\mathcal{I}_b,1} \mathrel{+}= \min\{\overline{\text{FP}}_b, \overline{\text{FN}}_b\} - \min\{\overline{\text{FP}}_b - \Delta\text{FP}_b, \overline{\text{FN}}_b - \Delta\text{FN}_b\}$
    $\mathbf{D}_{\mathcal{I}_b,2} \mathrel{+}= \min\{\underline{\text{FP}}_b + \Delta\text{FP}_b, \underline{\text{FN}}_b + \Delta\text{FN}_b\} - \min\{\underline{\text{FP}}_b, \underline{\text{FN}}_b\}$
Output: AUM and matrix $\mathbf{D}$ of directional derivatives.

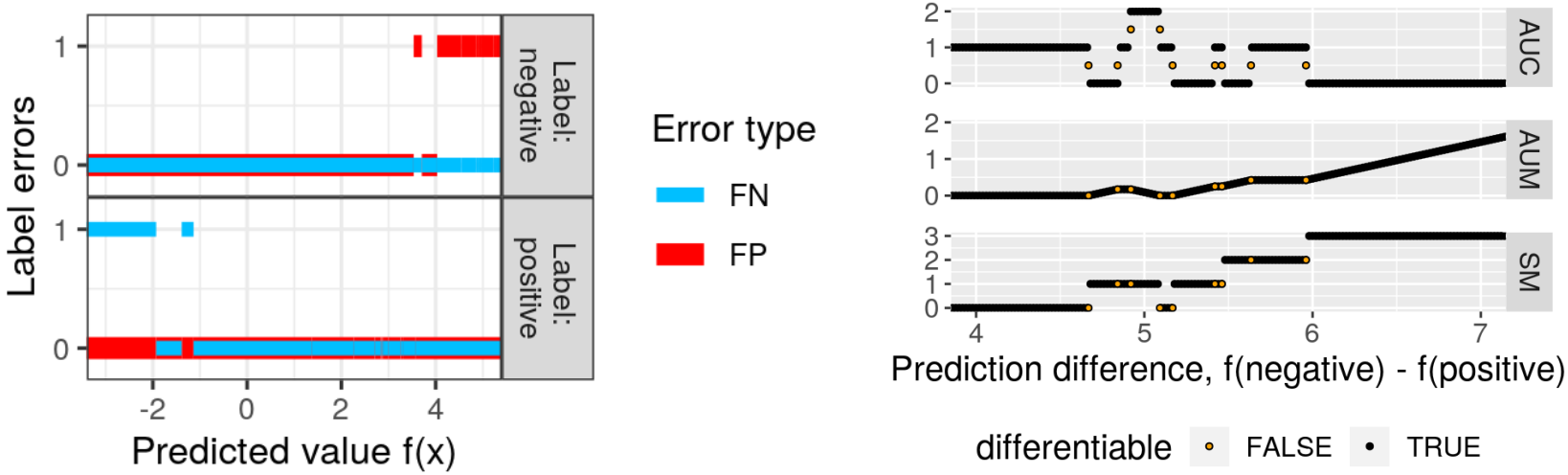
Problem Setting and Related Work

Proposed algorithm

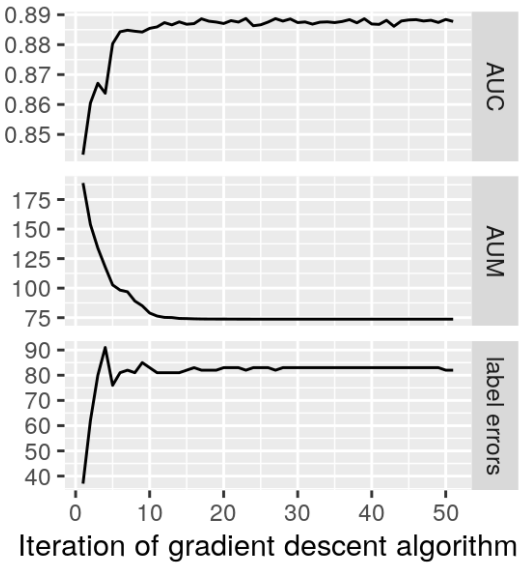
Results

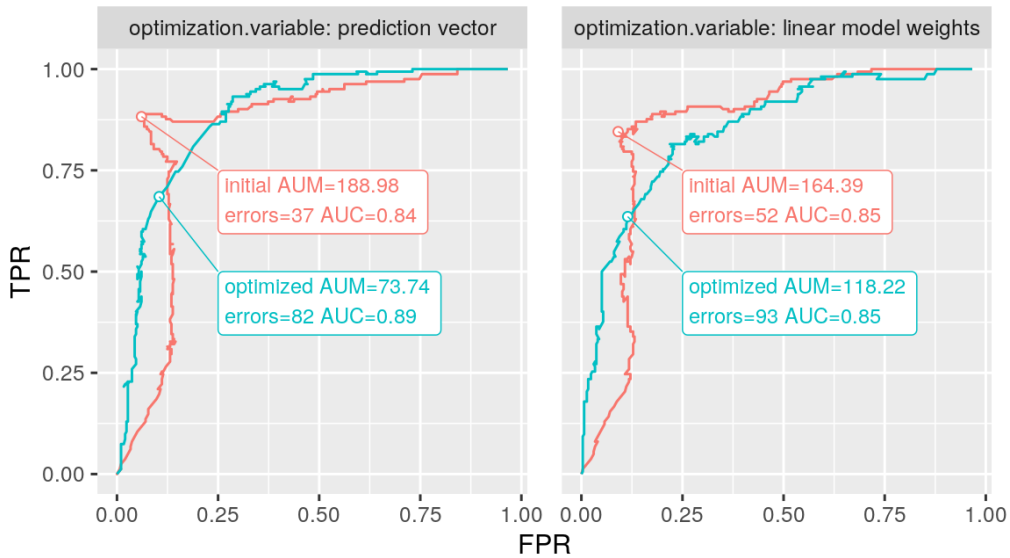
Discussion and Conclusions

Real data example with AUC greater than one

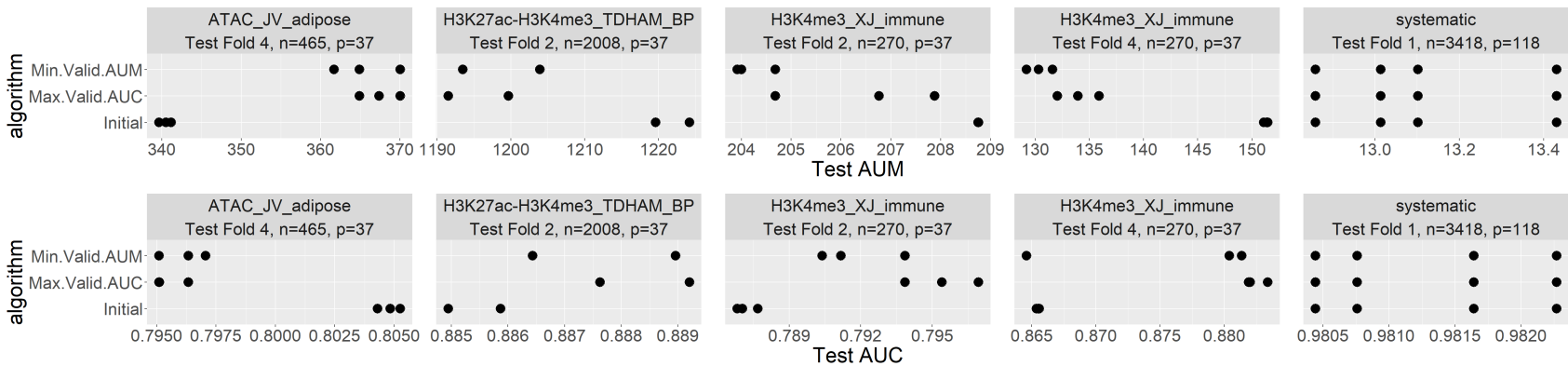


Train set ROC curves for a real changepoint problem

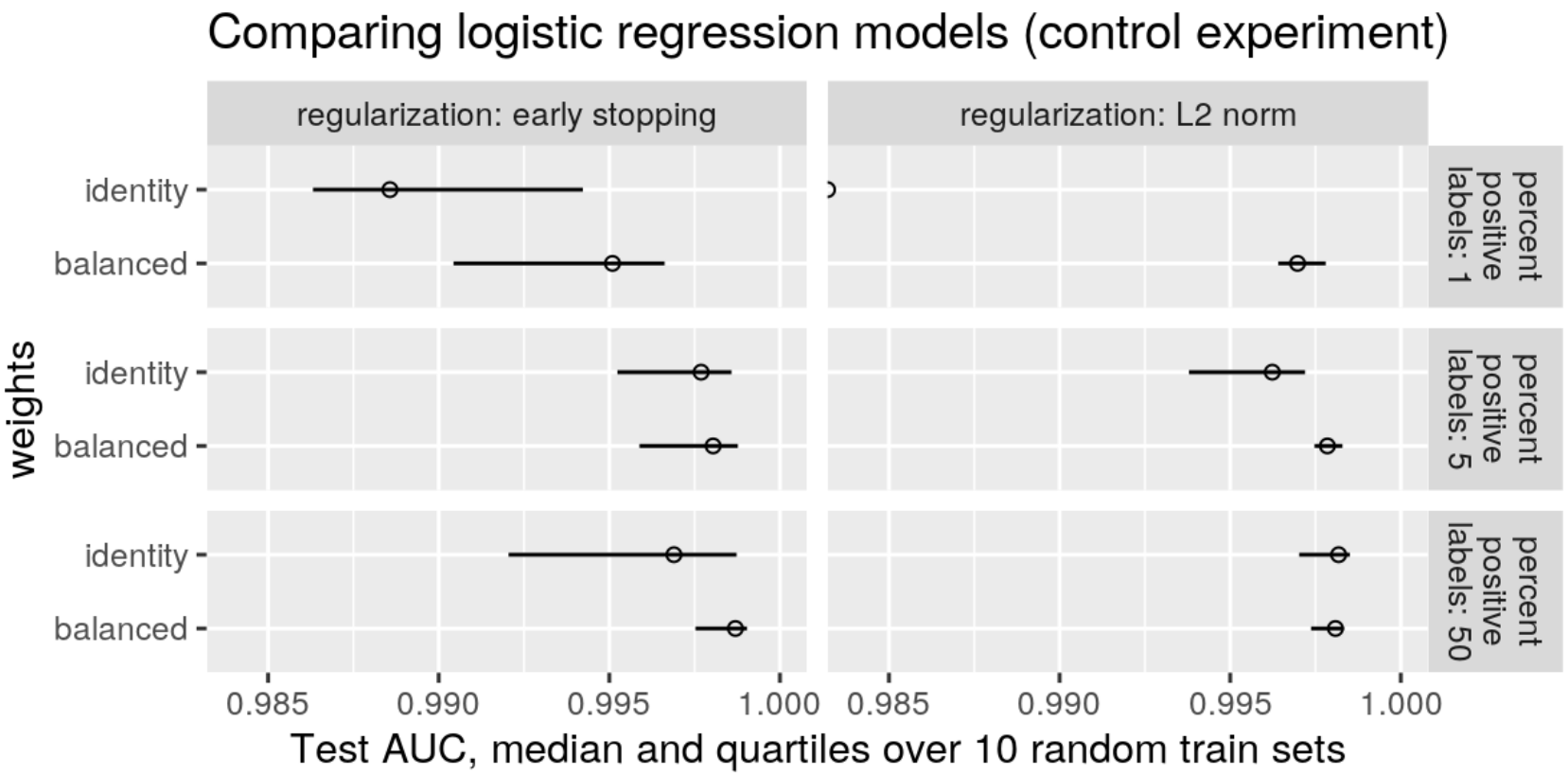




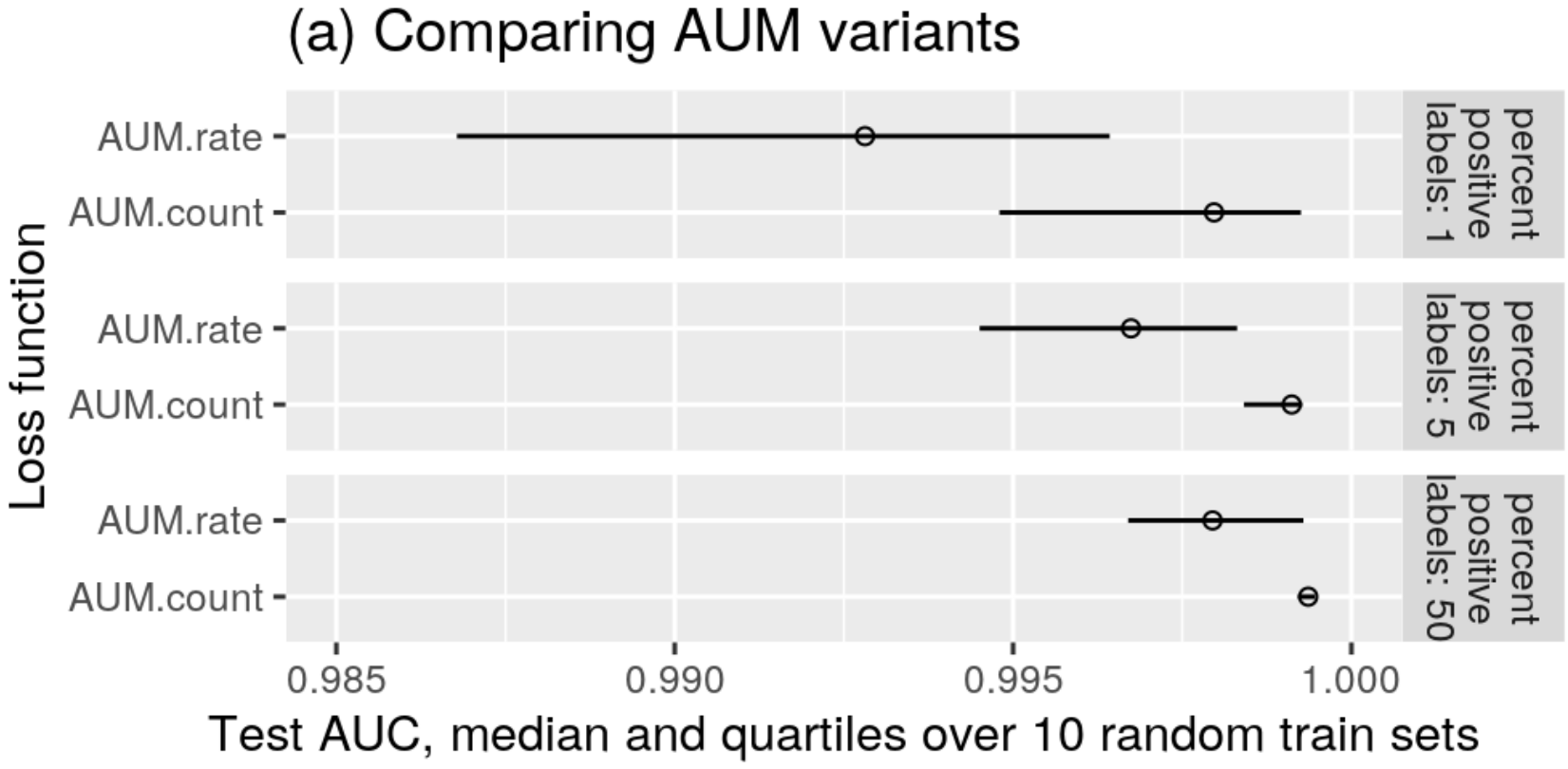
Learning algorithm results in better test AUC/AUM for changepoint problems



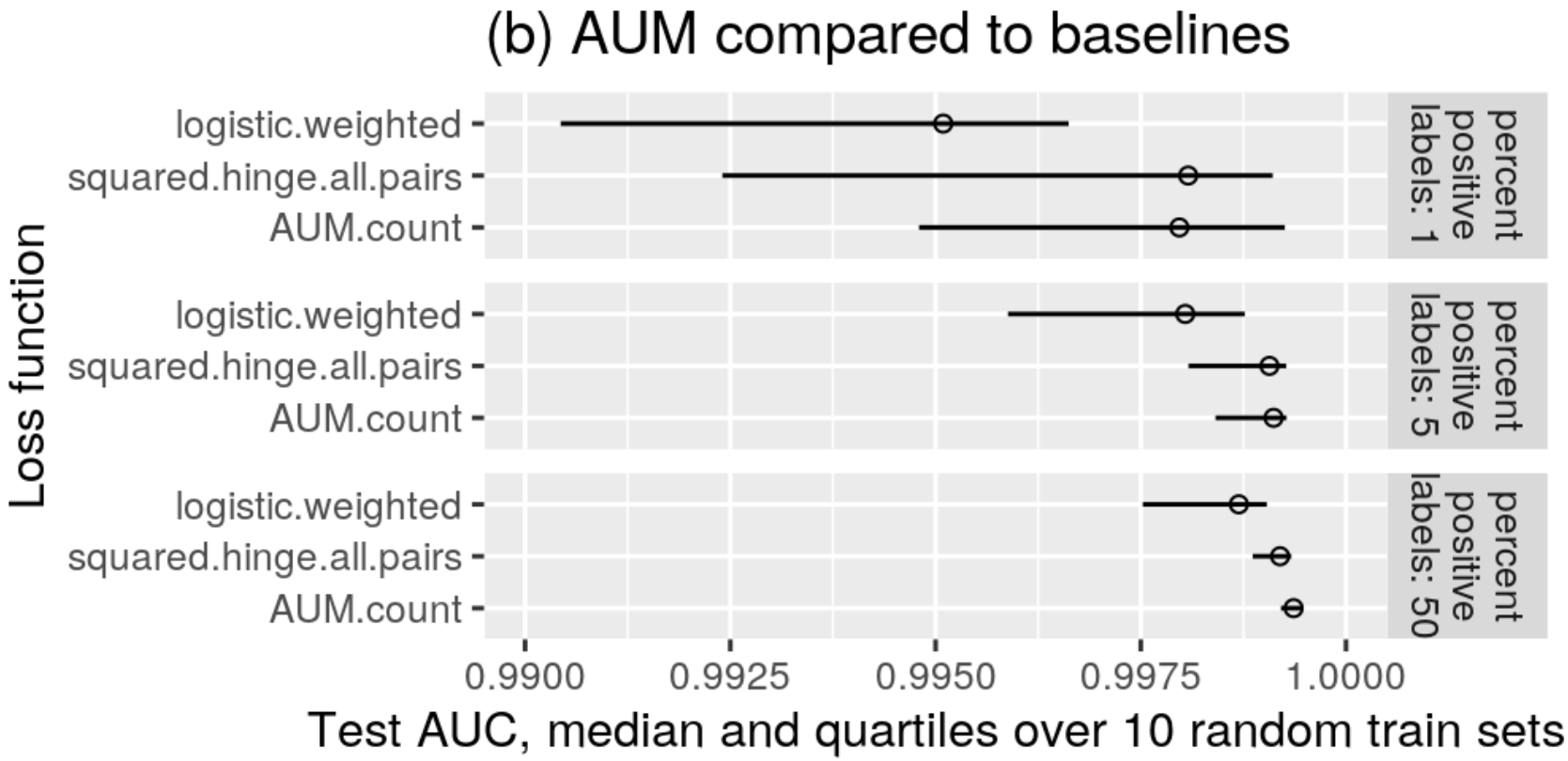
Standard logistic loss fails for highly imbalanced labels



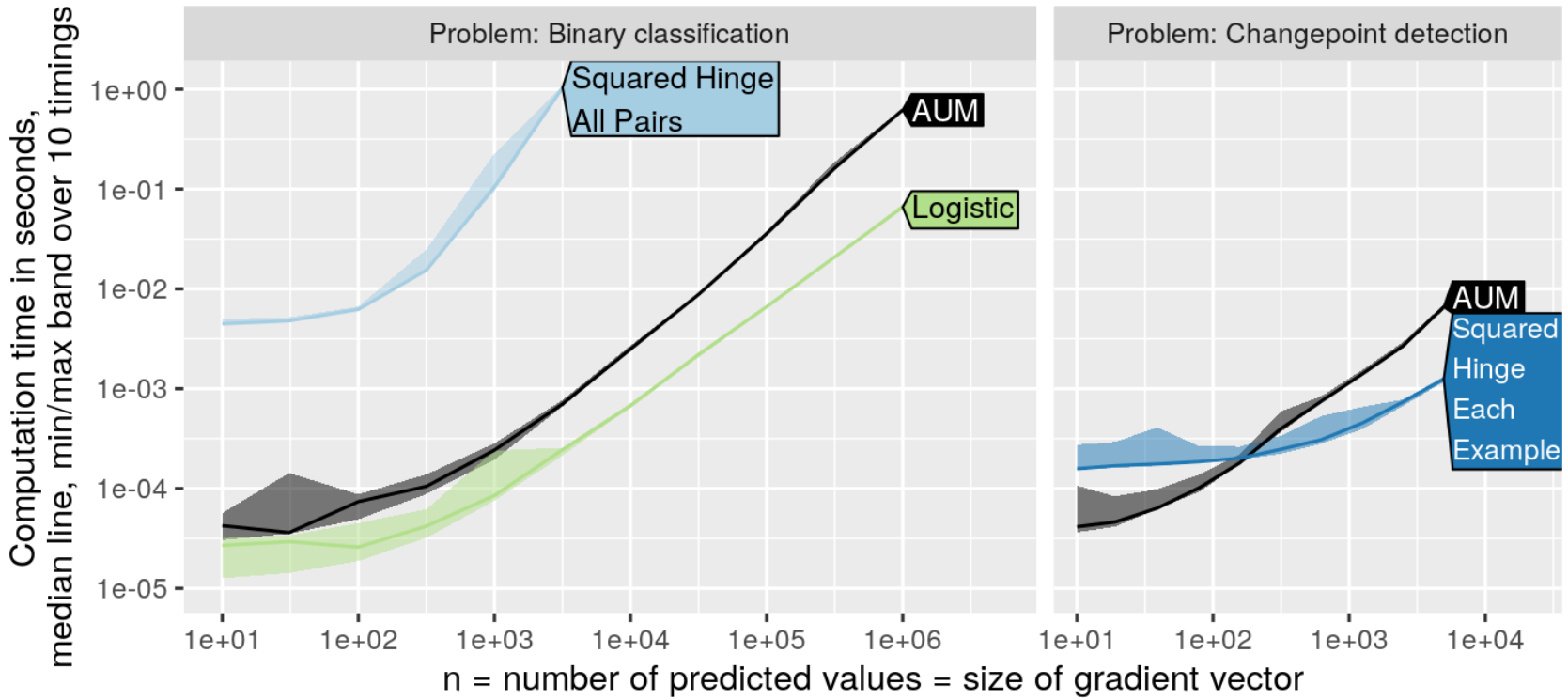
Error rate loss is not as useful as error count loss



Learning algorithm competitive for unbalanced binary classification



Comparable computation time to other loss functions



Problem Setting and Related Work

Proposed algorithm

Results

Discussion and Conclusions

Conclusions

TODO

More notation

First let $\{(\text{fpt}(\hat{\mathbf{y}})_q, \text{fnt}(\hat{\mathbf{y}})_q, \tau(\hat{\mathbf{y}})_q)\}_{q=1}^{Q}$ be a sequence of Q tuples, each of which corresponds to a point on the ROC curve (Figure **??**, right). The fpt/fnt are false positive/negative totals whereas τ are values such there is a change/threshold at $M_{\hat{\mathbf{y}}}(\tau)$. As shown in Figure **??** we assume these values are increasing, $-\infty = \tau(\hat{\mathbf{y}})_0 < \cdots < \tau(\hat{\mathbf{y}})_Q = \infty$. For each $q \in \{1, \ldots, Q\}$ there is a corresponding interval of values c between $\tau(\hat{\mathbf{y}})_{q-1}$ and $\tau(\hat{\mathbf{y}})_q$ such that $\text{FPT}_{\hat{\mathbf{y}}}(c) = \text{fpt}(\hat{\mathbf{y}})_q$ and $\text{FNT}_{\hat{\mathbf{y}}}(c) = \text{fnt}(\hat{\mathbf{y}})_q$ for all $c \in (\tau(\hat{\mathbf{y}})_{q-1}, \tau(\hat{\mathbf{y}})_q)$ (Figure **??**, left). Then we define $m(\hat{\mathbf{y}})_q = \min\{\text{fpt}(\hat{\mathbf{y}})_q, \text{fnt}(\hat{\mathbf{y}})_q\}$ and so since $m(\hat{\mathbf{y}})_1 = m(\hat{\mathbf{y}})_Q = 0$ the area under those intervals is zero, and the AUM can be computed by summing over all of the other intervals,

L1 relaxation interpretation

Our proposed loss function is

$$\text{AUM}(\hat{\mathbf{y}}) = \sum_{q=2}^{Q-1} [\tau(\hat{\mathbf{y}})_q - \tau(\hat{\mathbf{y}})_{q-1}] m(\hat{\mathbf{y}})_q.$$

It is an L1 relaxation of the following non-convex **S**um of **M**in(FP,FN) function,

$$\text{SM}(\hat{\mathbf{y}}) = \sum_{q=2}^{Q-1} I[\tau(\hat{\mathbf{y}})_q \neq \tau(\hat{\mathbf{y}})_{q-1}] m(\hat{\mathbf{y}})_q = \sum_{q:\tau(\hat{\mathbf{y}})_q \neq \tau(\hat{\mathbf{y}})_{q-1}} m(\hat{\mathbf{y}})_q.$$

FP and FN counts before/after each threshold

$$
\begin{aligned}
\underline{\mathrm{FP}}_b &= \sum_{j:t_j<t_b} \Delta\mathrm{FP}_j, \\
\overline{\mathrm{FP}}_b &= \sum_{j:t_j\leq t_b} \Delta\mathrm{FP}_j, \\
\underline{\mathrm{FN}}_b &= \sum_{j:t_j\geq t_b} -\Delta\mathrm{FN}_j, \\
\overline{\mathrm{FN}}_b &= \sum_{j:t_j>t_b} -\Delta\mathrm{FN}_j.
\end{aligned}
$$

Directional derivatives

Theorem

The AUM directional derivatives for a particular example $i \in \{1, \ldots, n\}$ can be computed using the following equations.

$$\nabla_{\mathbf{v}(-1,i)}\text{AUM}(\hat{\mathbf{y}}) = \sum_{b:\mathcal{I}_b=i} \min\{\overline{\text{FP}}_b, \overline{\text{FN}}_b\} - \min\{\overline{\text{FP}}_b - \Delta\text{FP}_b, \overline{\text{FN}}_b - \Delta\text{FN}_b\},$$

$$\nabla_{\mathbf{v}(1,i)}\text{AUM}(\hat{\mathbf{y}}) = \sum_{b:\mathcal{I}_b=i} \min\{\underline{\text{FP}}_b + \Delta\text{FP}_b, \underline{\text{FN}}_b + \Delta\text{FN}_b\} - \min\{\underline{\text{FP}}_b, \underline{\text{FN}}_b\}.$$