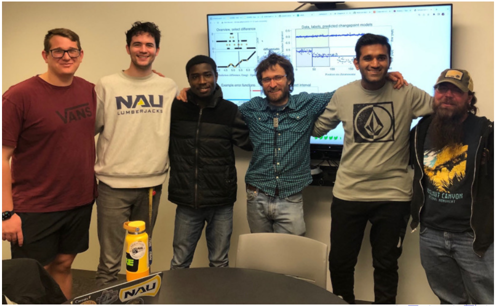
Efficient line search optimization of penalty functions in supervised changepoint detection

Toby Dylan Hocking — toby.hocking@nau.edu
joint work with my student Jadon Fowler
Machine Learning Research Lab — `http://ml.nau.edu`
School of Informatics, Computing and Cyber Systems
Northern Arizona University, USA



Problem Setting 1: ROC curves for evaluating supervised binary classification algorithms

Problem setting 2: ROC curves for evaluating supervised changepoint algorithms

ROC curve optimization using gradient descent

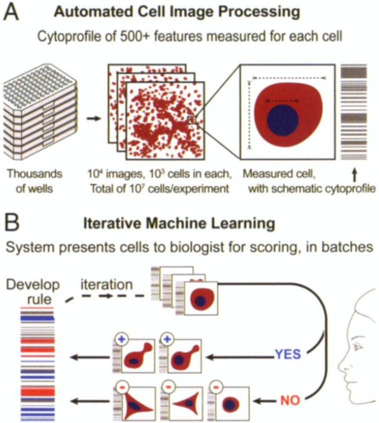
Proposed complete line search algorithm for surrogate loss: Area Under Min{FP,FN} (AUM)

Empirical results: increased speed and comparable accuracy using proposed complete line search

Discussion and Conclusions

Problem: supervised binary classification

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



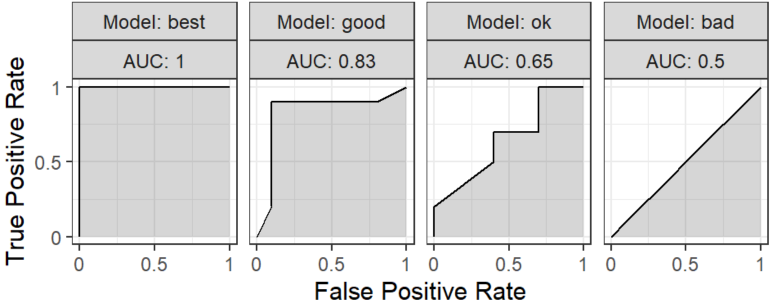
Most algorithms (SVM, Logistic regression, etc) minimize a differentiable surrogate of zero-one loss = sum of:

False positives: $f(\mathbf{x}) > 0$ but $y = 0$ (predict budding, but cell is not).

False negatives: $f(\mathbf{x}) < 0$ but $y = 1$ (predict not budding but cell is).

Receiver Operating Characteristic (ROC) Curves

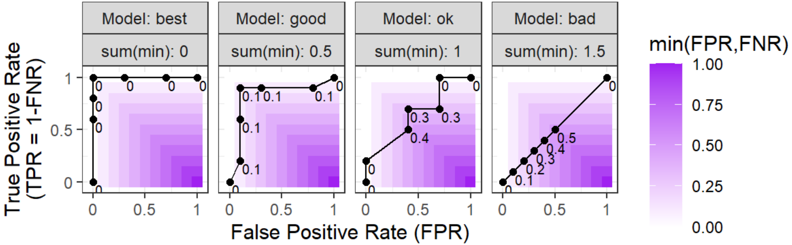
- Classic evaluation method from the signal processing literature (Egan and Egan, 1975).
- ROC curve of learned f is plot of True Positive Rate vs False Positive Rate: each point on the ROC curve is a different constant $c \in \mathbb{R}$ added to the predicted values: $f(\mathbf{x}) + c$.
- $c = \infty$ means always predict positive label (FPR=TPR=1).
- $c = -\infty$ means always predict negative label (FPR=TPR=0).
- Best classifier has a point near upper left (TPR=1, FPR=0), with large Area Under the Curve (AUC).



Research question and new idea

Can we learn a binary classification function f which directly optimizes the ROC curve?

- Most algorithms involve minimizing a differentiable surrogate of the zero-one loss, which is not the same.
- The Area Under the ROC Curve (AUC) is piecewise constant (gradient zero almost everywhere), so can not be used with gradient descent algorithms.
- We propose to encourage points to be in the upper left of ROC space, using a loss function which is a differentiable surrogate of the sum of min(FP,FN).



Problem Setting 1: ROC curves for evaluating supervised binary classification algorithms

Problem setting 2: ROC curves for evaluating supervised changepoint algorithms

ROC curve optimization using gradient descent

Proposed complete line search algorithm for surrogate loss: Area Under Min{FP,FN} (AUM)

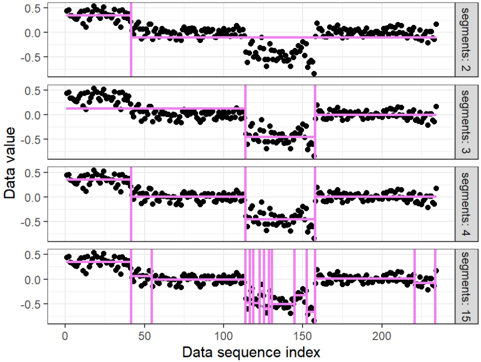
Empirical results: increased speed and comparable accuracy using proposed complete line search

Discussion and Conclusions

Problem: unsupervised changepoint detection

- Data sequence $z_1, \ldots, z_T$ at T points over time/space.
- Ex: DNA copy number data for cancer diagnosis, $z_t \in \mathbb{R}$.
- The penalized changepoint problem (Maidstone *et al.* 2017)

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

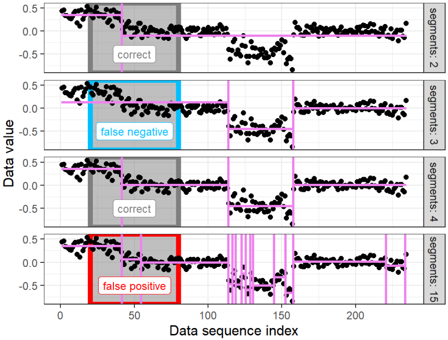


Larger penalty λ results in fewer changes/segments.

Smaller penalty λ results in more changes/segments.

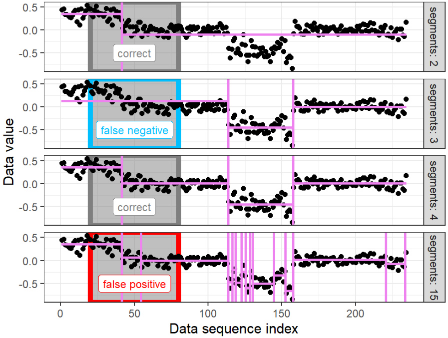
Problem: weakly supervised changepoint detection

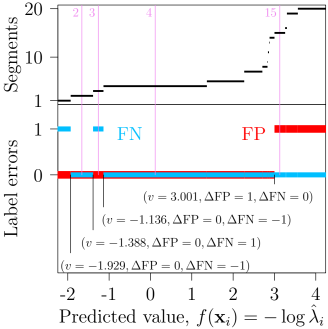
- First described by Hocking *et al.* ICML 2013.
- We are given a data sequence $\mathbf{z}$ with labeled regions L.



Problem: weakly supervised changepoint detection

- First described by Hocking *et al.* ICML 2013.
- We are given a data sequence $\mathbf{z}$ with labeled regions L.

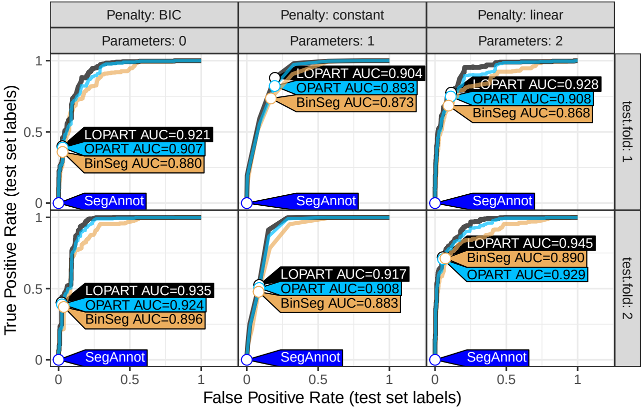




We compute features $\mathbf{x} = \phi(\mathbf{z}) \in \mathbf{R}^p$ and want to learn a function $f(\mathbf{x}) = -\log \lambda \in \mathbf{R}$ that minimizes label error (sum of false positives and false negatives), or maximizes AUC, Hocking, Hillman, *Journal of Machine Learning Research* (2023).

Comparing changepoint algorithms using ROC curves

Hocking TD, Srivastava A. Labeled Optimal Partitioning. Computational Statistics (2022).



LOPART algorithm (R package LOPART) has consistently larger test AUC than previous algorithms.

Problem Setting 1: ROC curves for evaluating supervised binary classification algorithms

Problem setting 2: ROC curves for evaluating supervised changepoint algorithms

ROC curve optimization using gradient descent

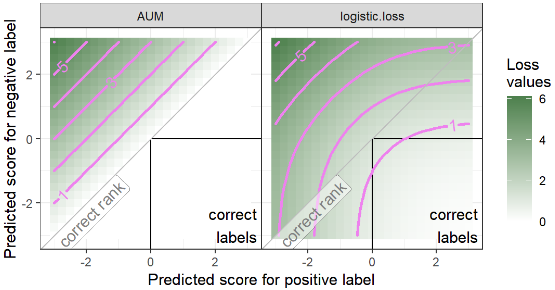
Proposed complete line search algorithm for surrogate loss: Area Under Min{FP,FN} (AUM)

Empirical results: increased speed and comparable accuracy using proposed complete line search

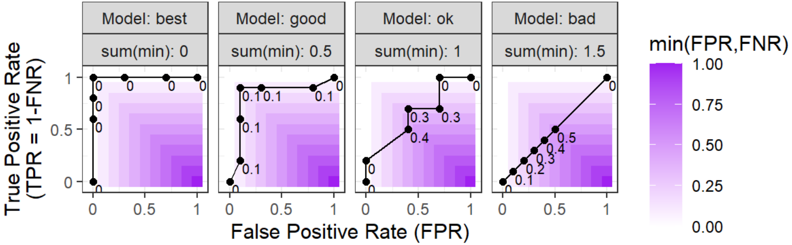
Discussion and Conclusions

Comparing logistic loss and rank/AUM loss

- Classic baselines: hinge and logistic loss, sum over samples, $\ell[yf(x)]$.
- Bamber (1975) proved ROC-AUC relation to Mann-Whitney U statistic (double sum over all pairs of positive and negative samples).
- Recently: SVM$^{\text{struct}}$ (Joachims 2005) and X-risk (Yang 2022) sum loss over pairs of positive and negative samples, $\ell[f(x^+) - f(x^-)]$.
- Proposed: AUM loss (sum over points on ROC curve).
- Figure below: loss for two samples: one positive, one negative.



L1 relaxation



Let $\hat{y}$ be the N-vector of predicted scores, making Q points on the ROC curve, with thresholds τ, and min(FPR,FNR) values m. Then the sum of the min over all ROC points is:

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

The AUM can be interpreted as an L1 relaxation of the following non-convex **S**um of **M**in(FP,FN) function,

$Q-1$

Problem Setting 1: ROC curves for evaluating supervised binary classification algorithms

Problem setting 2: ROC curves for evaluating supervised changepoint algorithms

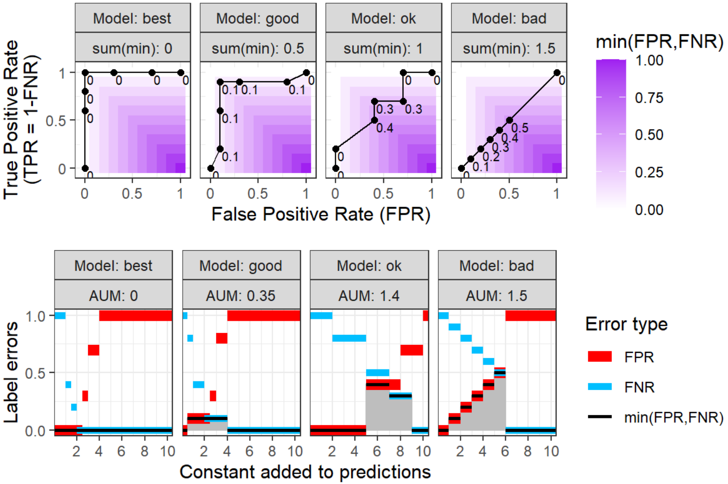
ROC curve optimization using gradient descent

Proposed complete line search algorithm for surrogate loss: Area Under Min{FP,FN} (AUM)

Empirical results: increased speed and comparable accuracy using proposed complete line search

Discussion and Conclusions

Large AUC ≈ small Area Under Min(FP,FN) (AUM)



Barr, Hocking, Morton, Thatcher, Shaw, *TransAI* (2022).
Hocking, Hillman, *Journal of Machine Learning Research* (2023).
Proposal: track how thresholds in error plot change with step size.

Using thresholds to compute AUM

Hillman and Hocking, JMLR 2023, showed that AUM for the current predictions, can be computed efficiently, as a function of $T_1 < \cdots < T_B$, thresholds of the min label error M_b,

$$\text{AUM} = \sum_{b=2}^{B} [T_b - T_{b-1}] M_b$$

This contribution: when learning a linear model, $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$, we can update the weights $\mathbf{w}$ using AUM gradient descent. We compute an exact representation of thresholds $T_b(s)$ and min error $M_b(s)$ as a function of step size s, which results in a complete piecewise linear representation of

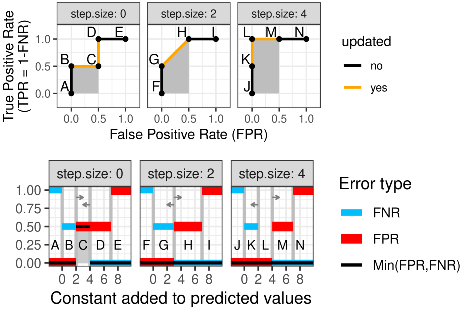
$$\text{AUM}(s) = \sum_{b=2}^{B} [T_b(s) - T_{b-1}(s)] M_b(s).$$

Simple example, proposed line search with 4 binary labels

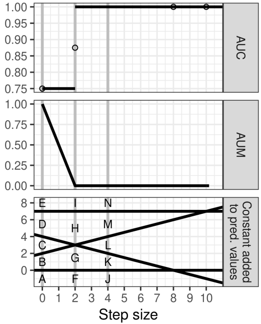
Theorem: when learning a linear model, $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$,

- AUC is piecewise constant, and
- AUM is piecewise linear,

as a function of step size in gradient descent.



ROC curves and error functions



Proposed line search

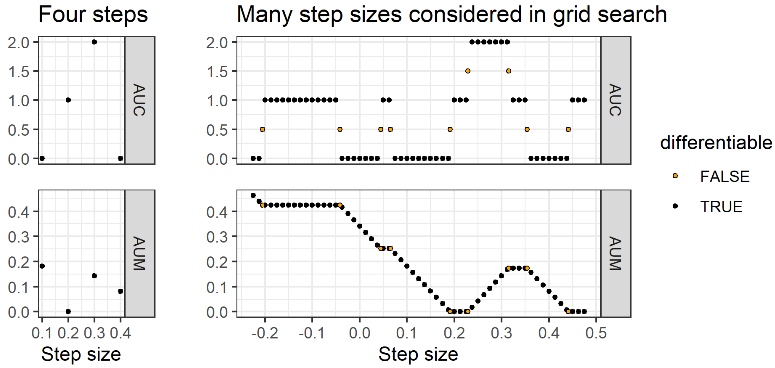
Letters show correspondence between points on the ROC curve, and intervals of constants added to predicted values.

Change-point example, comparison with grid search

Theorem: when learning a linear model, $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$,

- AUC is piecewise constant, and
- AUM is piecewise linear,

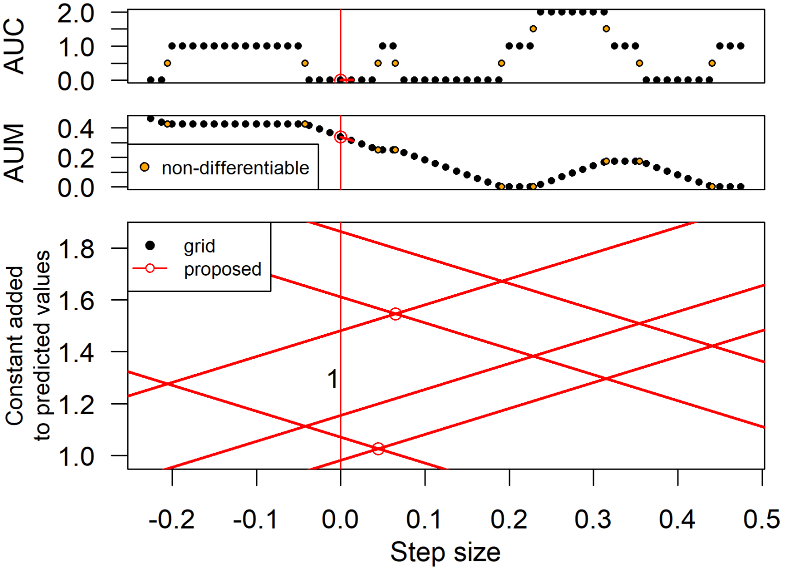
as a function of step size in gradient descent.



Proposed line search algorithm computes updates when there are possible changes in slope of AUM / values of AUC (orange dots).

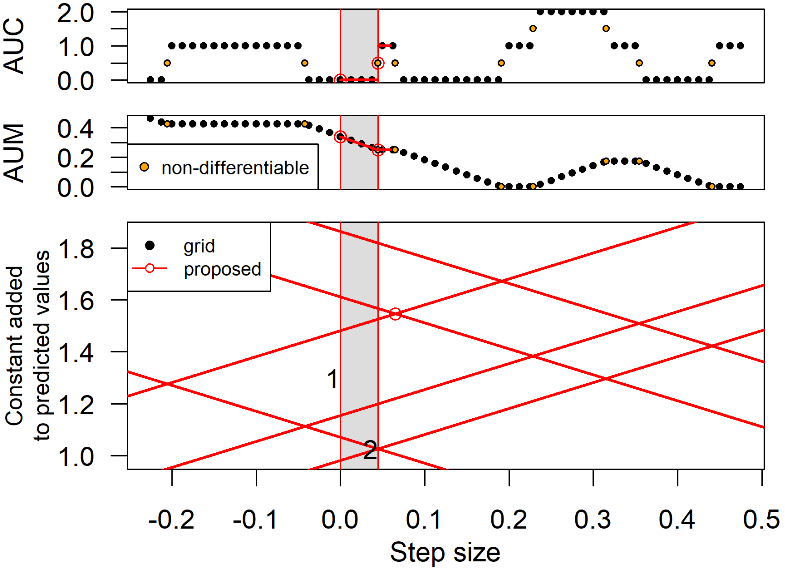
Proposed complete AUC/AUM line search, iteration 1

AUC/AUM values known only at red vertical line.



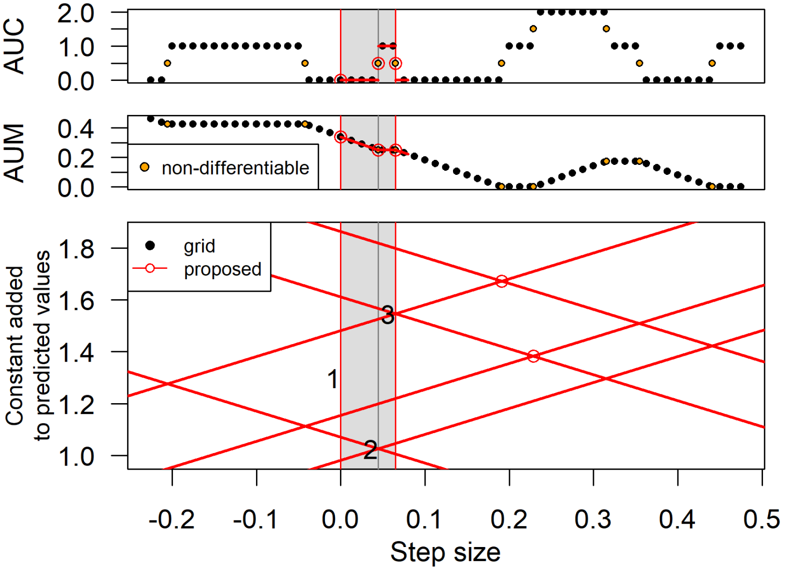
Proposed complete AUC/AUM line search, iteration 2

AUC/AUM values completely known within shaded grey region.



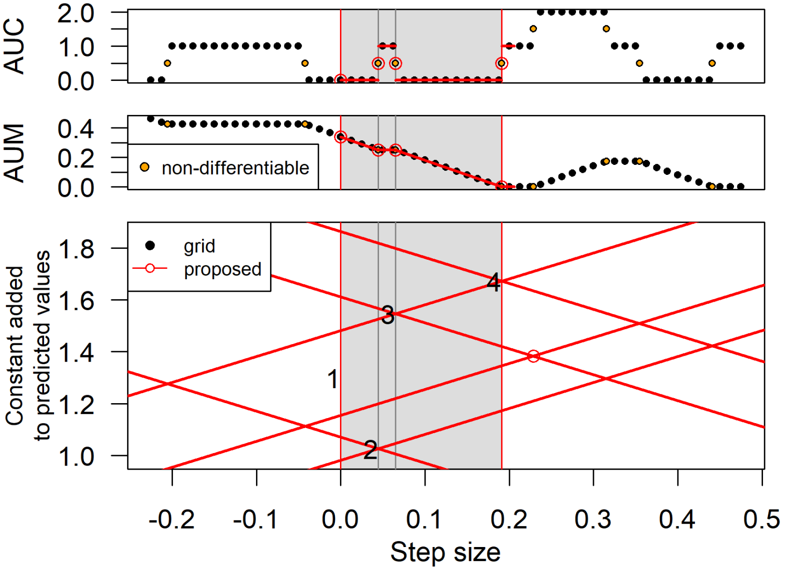
Proposed complete AUC/AUM line search, iteration 3

AUC/AUM values completely known within shaded grey region.



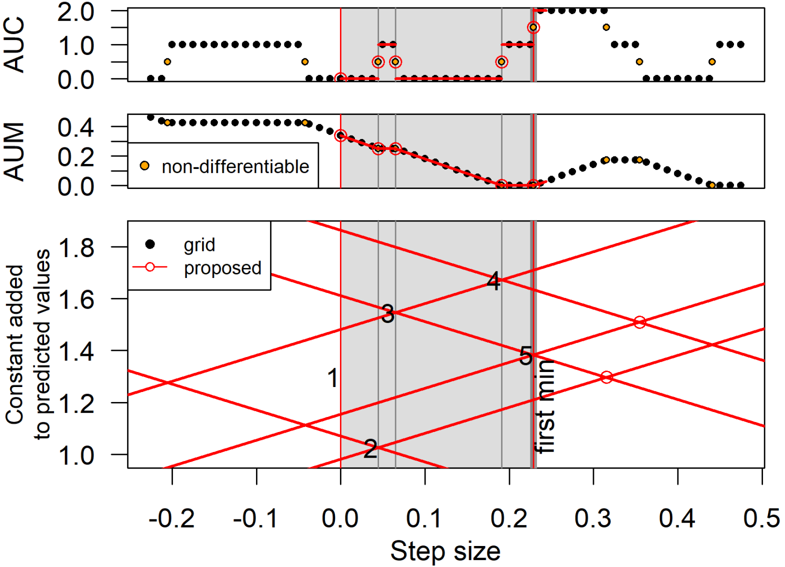
Proposed complete AUC/AUM line search, iteration 4

AUC/AUM values completely known within shaded grey region.



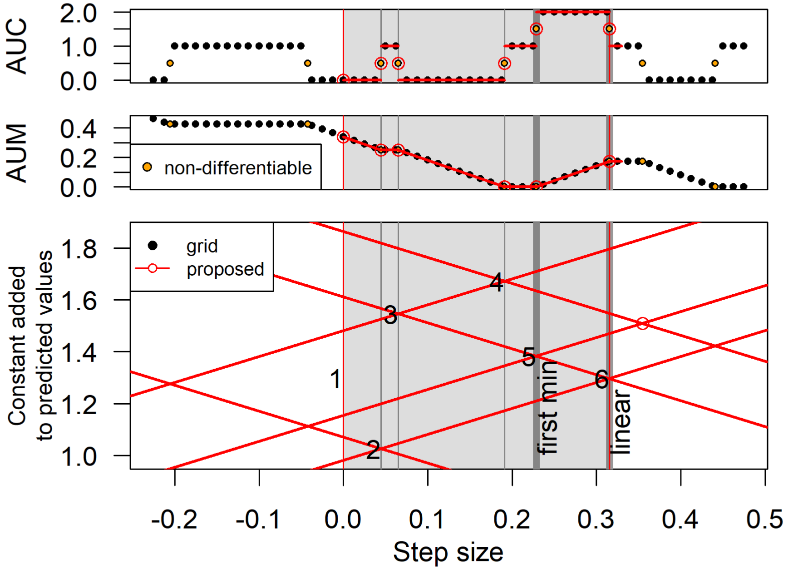
Proposed complete AUC/AUM line search, iteration 5

AUC/AUM values completely known within shaded grey region.



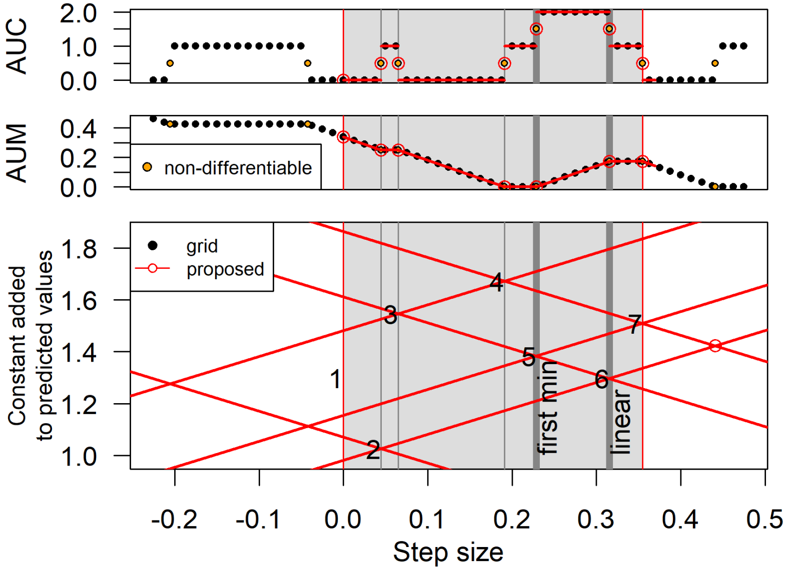
Proposed complete AUC/AUM line search, iteration 6

AUC/AUM values completely known within shaded grey region.



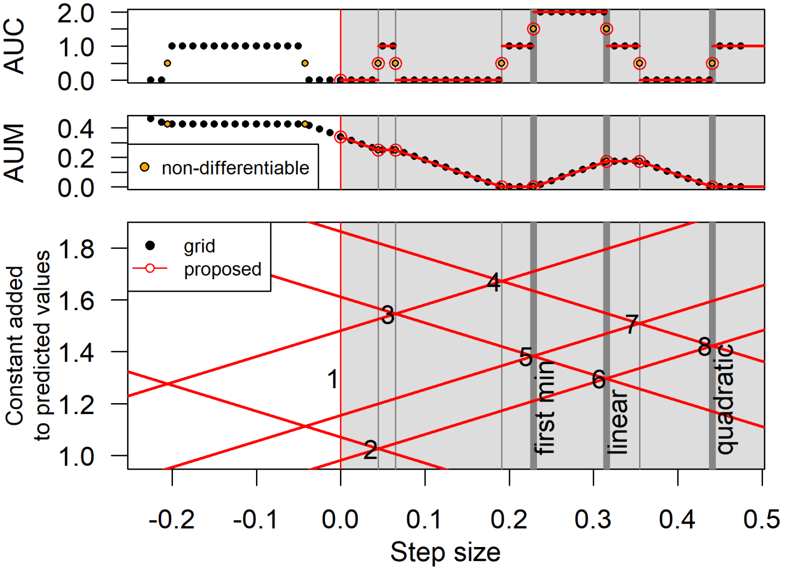
Proposed complete AUC/AUM line search, iteration 7

AUC/AUM values completely known within shaded grey region.



Proposed complete AUC/AUM line search, iteration 8

AUC/AUM values completely known within shaded grey region.



Complexity analysis of proposed algorithm

For N labeled observations, input N threshold line slope/intercept values. Possible next intersection points stored in a C++ STL map (red-black tree, sorted by step size), $O(\log N)$ time insertion, $O(1)$ lookup of next intersection. Worst case $O(N)$ space.

grid: standard grid search. $O(GN \log N)$ time per step, for G grid points.

linear(proposed): only first N intersections. $O(N \log N)$ time per step, relatively small step sizes chosen, relatively large number of steps overall in gradient descent.

quadratic(proposed): all $O(N^2)$ intersections. $O(N^2 \log N)$ time per step, large step sizes, small number of steps.

first min(proposed): keep iterating until first AUM increase. Same as quadratic in worst case, but may be faster on average (it was faster than both quadratic and linear for the example on the previous slide).

Problem Setting 1: ROC curves for evaluating supervised binary classification algorithms

Problem setting 2: ROC curves for evaluating supervised changepoint algorithms

ROC curve optimization using gradient descent

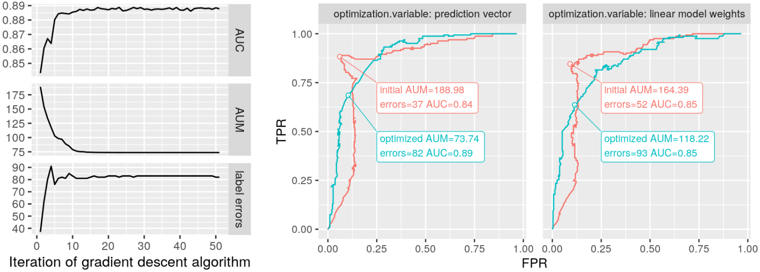
Proposed complete line search algorithm for surrogate loss: Area Under Min{FP,FN} (AUM)

Empirical results: increased speed and comparable accuracy using proposed complete line search

Discussion and Conclusions

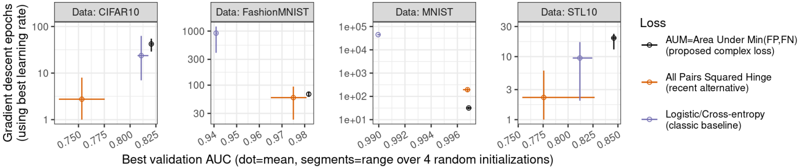
AUM gradient descent results in increased train AUC for a real changepoint problem

Hillman, Hocking, *Journal of Machine Learning Research* (2023).



- Left/middle: changepoint problem initialized to prediction vector with min label errors, gradient descent on prediction vector.
- Right: linear model initialized by minimizing regularized convex loss (surrogate for label error, Hocking *et al.* ICML 2013), gradient descent on weight vector.

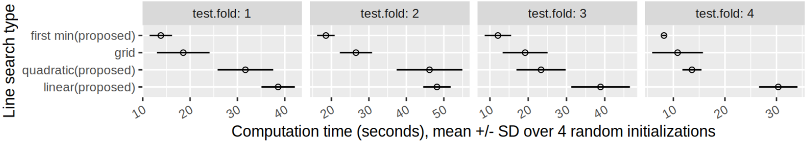
AUM gradient descent increases validation AUC, four image classification data sets



- Gradient descent with constant step size, best of 10^{-4} to 10^{5}.
- Max Validation AUC comparable or better than baselines.
- Computation time comparable or much faster.

Proposed search consistently faster than grid search

Analyzed supervised genomic change-point detection data set H3K4me3_TDH_immune ($N = 1073$ to 1248) from UCI Machine Learning Repository, `https://archive.ics.uci.edu/ml/datasets/chipseq`, Train/test splits defined via 4-fold CV, linear model initialized by minimizing regularized convex loss (surrogate for label error, Hocking *et al.* ICML 2013), keep doing AUM rate gradient descent steps (with line search) until subtrain loss stops decreasing.



first min(proposed): keep iterating until first AUM increase.

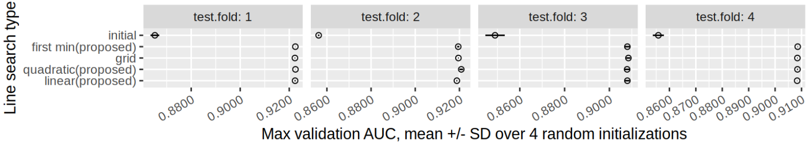
grid: search over step size$\in \{10^{-9}, 10^{-8}, \dots, 10^{1}, 10^{0}\}$.

quadratic(proposed): all line search iterations.

linear(proposed): only first N line search iterations.

Proposed search has similar accuracy as grid search

Analyzed supervised genomic change-point detection data set H3K4me3_TDH_immune ($N = 1073$ to 1248) from UCI Machine Learning Repository, `https://archive.ics.uci.edu/ml/datasets/chipseq`, Train/test splits defined via 4-fold CV, linear model initialized by minimizing regularized convex loss (surrogate for label error, Hocking *et al.* ICML 2013), keep doing AUM rate gradient descent steps (with line search) until subtrain loss stops decreasing.



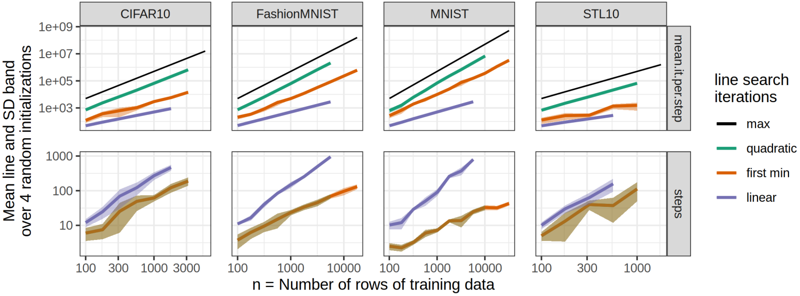
first min(proposed): keep iterating until first AUM increase.

grid: search over step size $\in \{10^{-9}, 10^{-8}, \ldots, 10^{1}, 10^{0}\}$.

quadratic(proposed): all line search iterations.

linear(proposed): only first N line search iterations.

Asymptotic time complexity analysis



max: theoretical worst case, $N(N-1)/2$ iterations.

quadratic(proposed): all line search iterations.

first min(proposed): keep iterating until first AUM increase (same number of steps/solution as quadratic, but asymptotically faster/smaller slope).

linear(proposed): only first N line search iterations.

Problem Setting 1: ROC curves for evaluating supervised binary classification algorithms

Problem setting 2: ROC curves for evaluating supervised changepoint algorithms

ROC curve optimization using gradient descent

Proposed complete line search algorithm for surrogate loss: Area Under Min{FP,FN} (AUM)

Empirical results: increased speed and comparable accuracy using proposed complete line search

Discussion and Conclusions

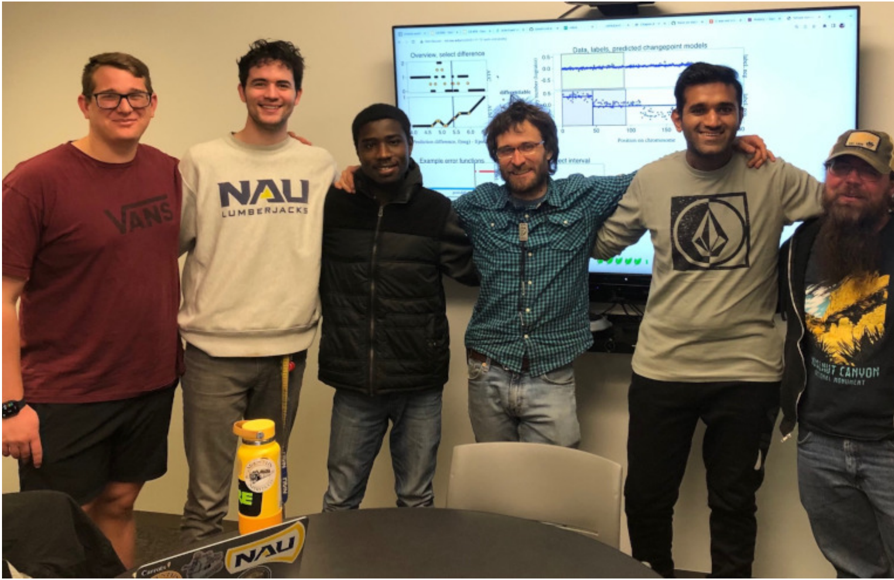
Discussion and Conclusions

- Area Under the ROC Curve (AUC) is used to evaluate binary classification and changepoint detection algorithms.
- Hocking, Hillman, *Journal of Machine Learning Research* (2023), proposed AUM=Area Under Min(FP,FN), a new differentiable surrogate loss for AUC optimization.
- In this talk we proposed new gradient descent algorithms with efficient complete line search, for optimizing AUM/AUC.
- Empirical results provide evidence that proposed complete line search is consistently faster than grid search, and has comparable accuracy (in terms of max validation AUC).
- Implementations available in R/C++ and python:

  `https://cloud.r-project.org/web/packages/aum/` (R/C++ line search)

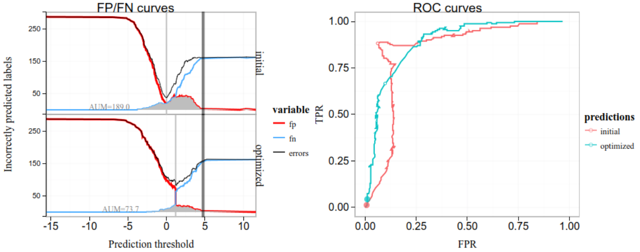
  `https://tdhock.github.io/blog/2022/aum-learning/` (pytorch AUM loss)
- Future work: non-linear learning algorithms that use AUM minimization as a surrogate for AUC maximization.

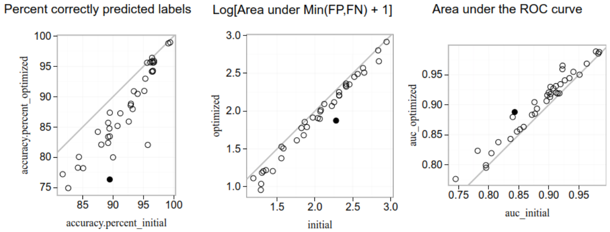
Thanks to co-author Jadon Fowler! (second from left)



Contact: toby.hocking@nau.edu

Initial/optimized AUC/AUM for change-point problems





https://tdhock.github.io/2023-11-21-auc-improved