Optimizing Looping ROC Curves

Jadon Fowler

Area Under the ROC Curve (AUC)

- Setup: model outputs a prediction vector given some examples, comparing the values to a threshold of zero to get classifications.

- Different points on the ROC curve are obtained by adding a constant to the predicted values (TPR vs FPR).

- AUC is an evaluation metric which accounts for every possible threshold

- AUC can be interpreted as the probability of ranking a positive example higher than a negative example

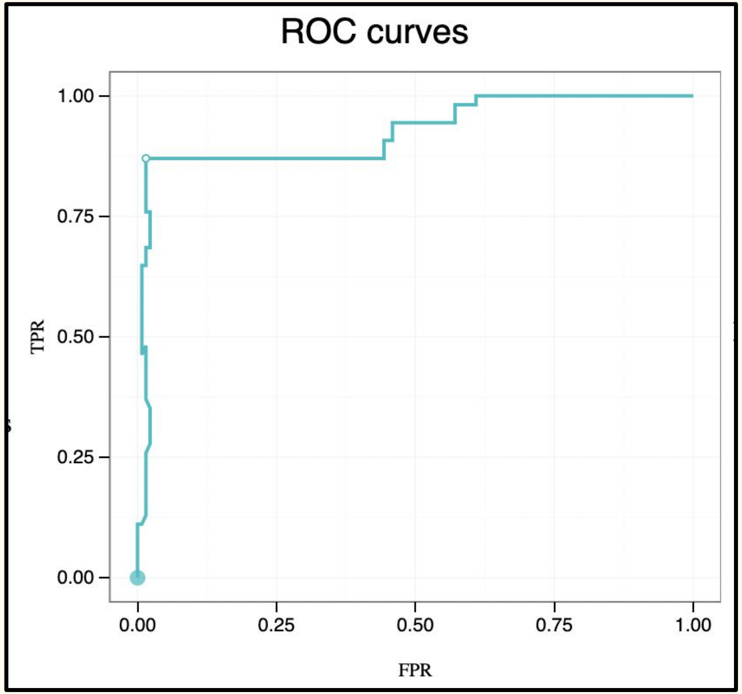
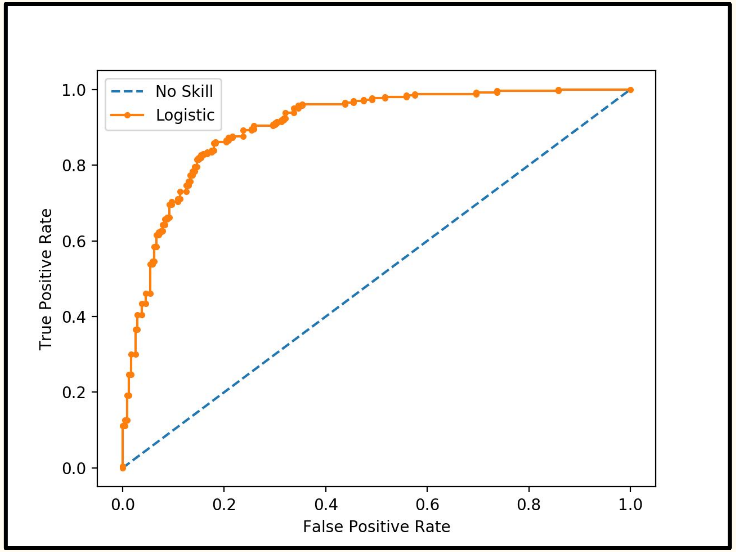


Figure from Dr. Hocking

Where do non-convex ROC curves come from?

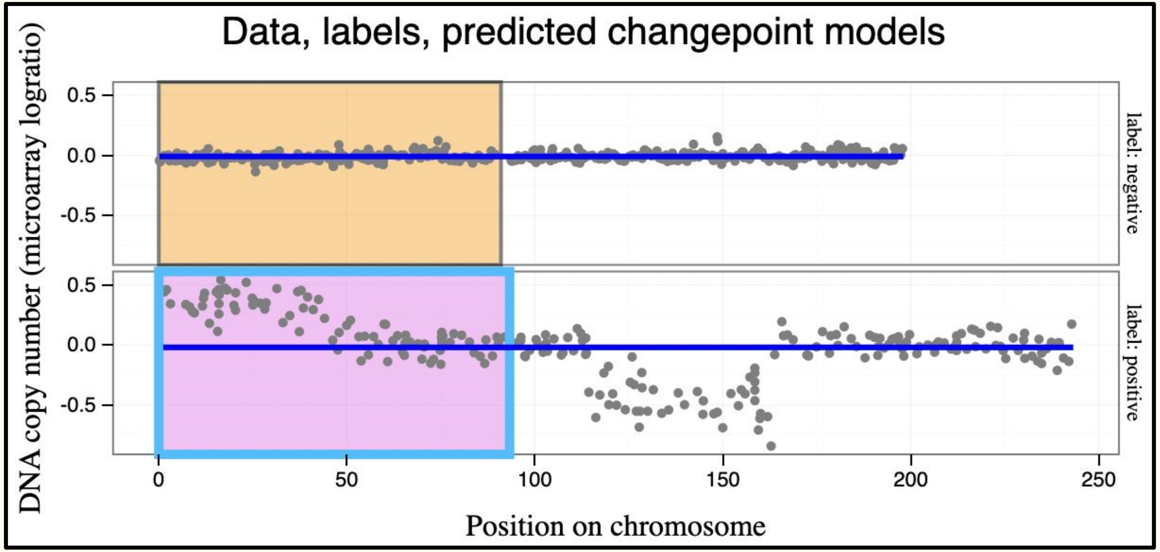


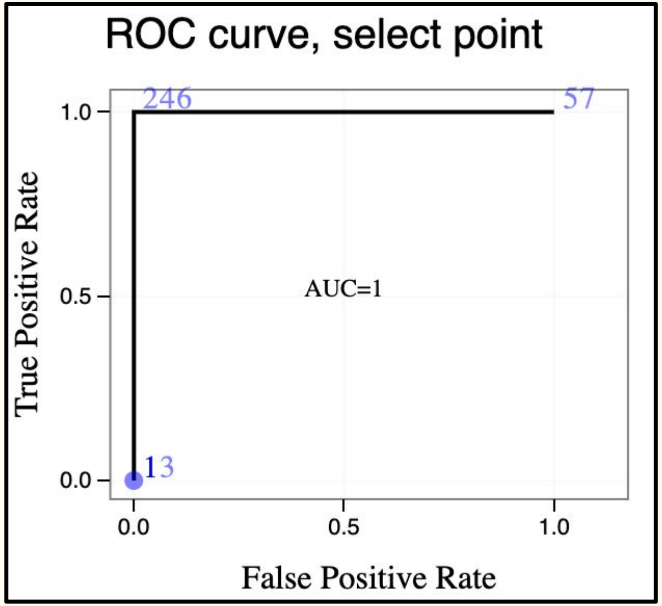
Concave ROC curve from [ML Mastery]

- In binary classification problems, there's a relationship between the increasing the classification thresholding and the TPR/FPR.

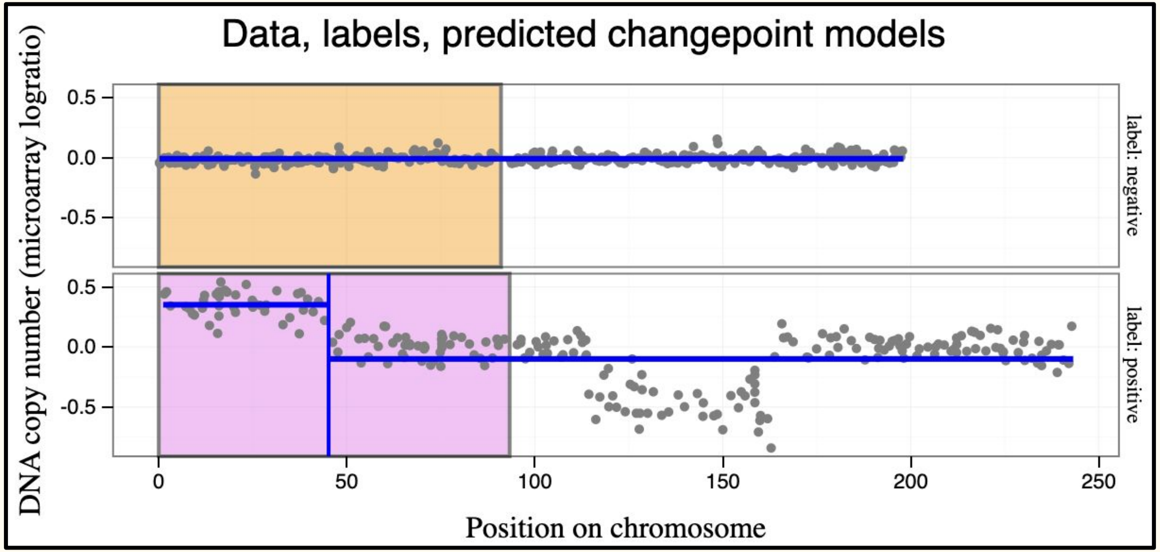
- In change point detection, gradual increases to the constant added to this prediction vector don't imply gradual increases to ROC points.

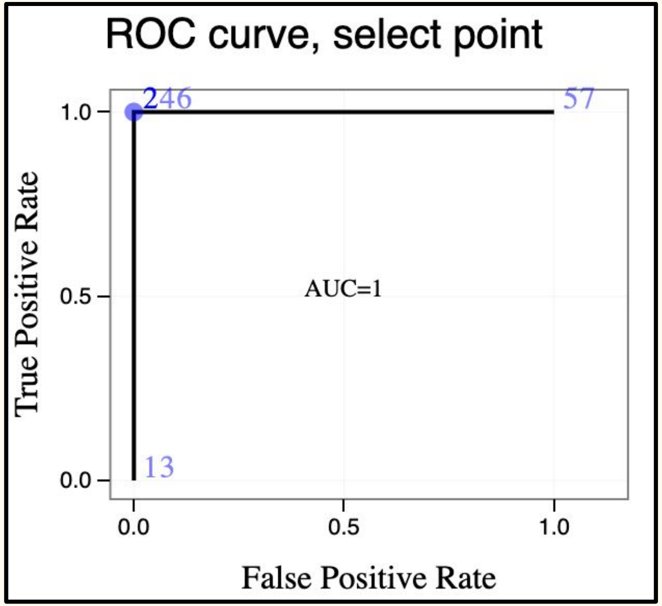
Expected changepoints in orange box: 0
Expected changepoints in pink box: 1



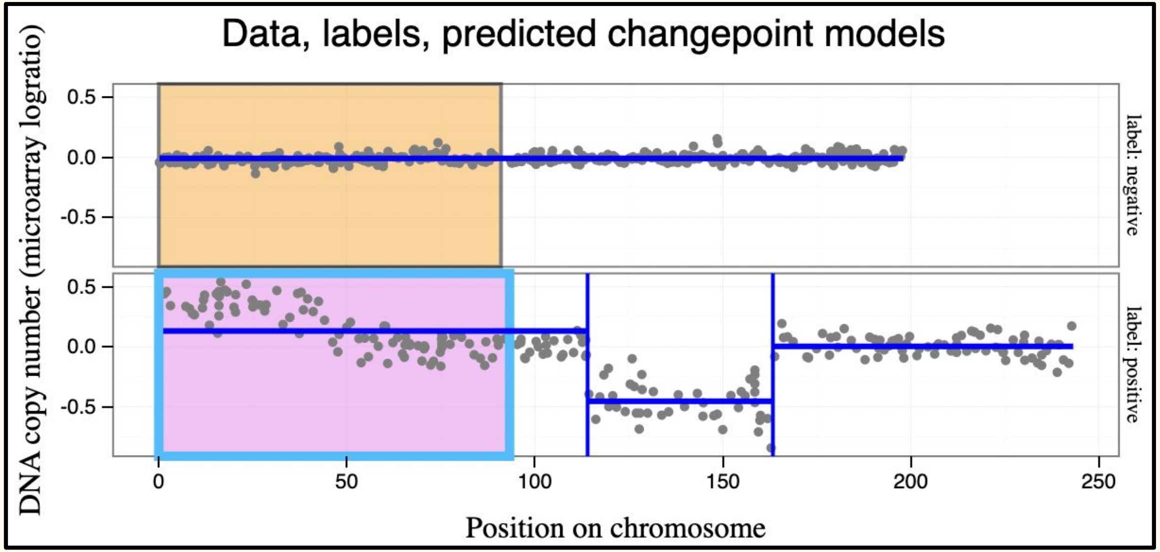


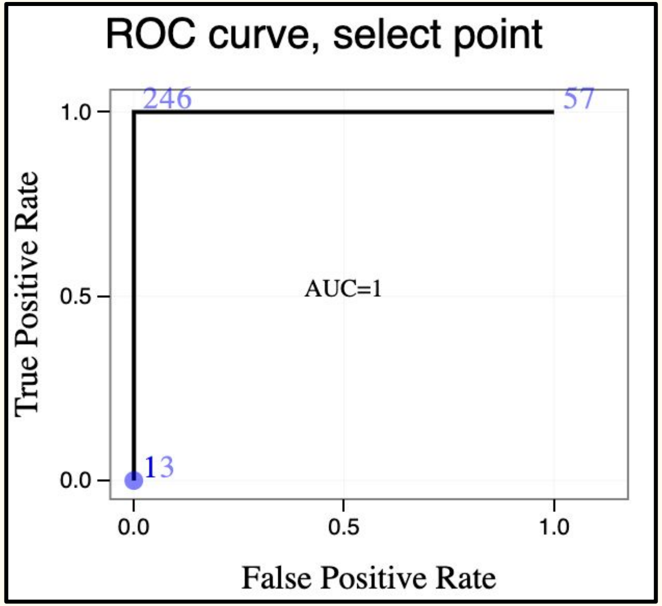
ROC Point: 1


Data, labels, predicted changepoint models
DNA copy number (microarray logratio)
0.5
0.0
-0.5
label: negative
label: positive
0
50
100
150
200
250
Position on chromosome

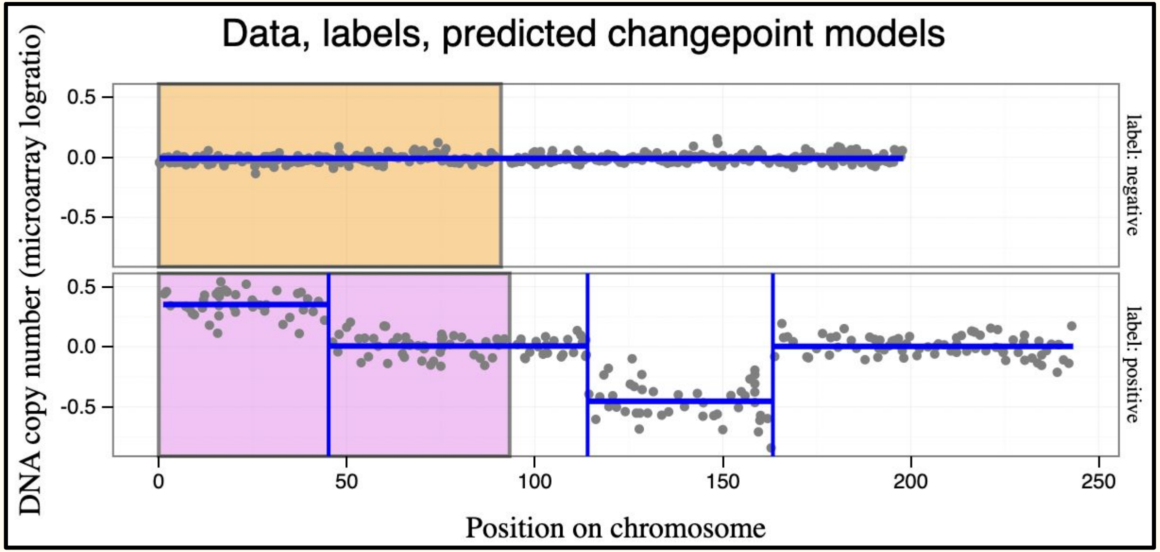

ROC curve, select point
True Positive Rate
False Positive Rate
1.0
0.5
0.0
2
46
57
13
AUC=1

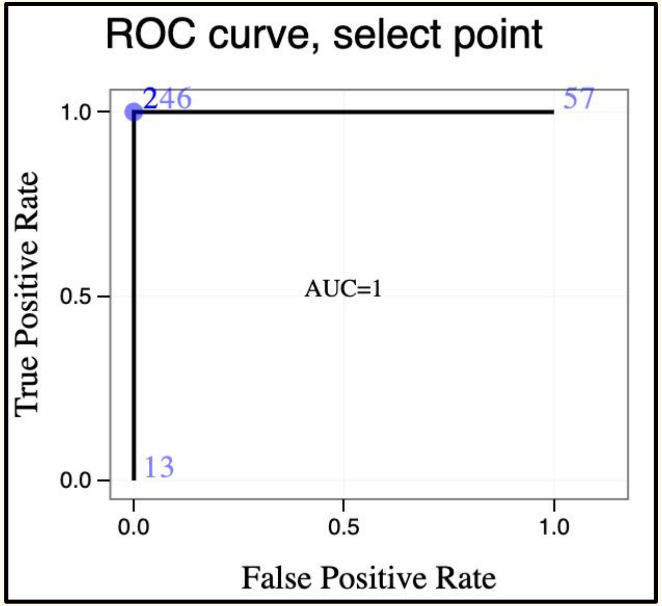
ROC Point: 2


Data, labels, predicted changepoint models
DNA copy number (microarray logratio)
0.5
0.0
-0.5
label: negative
label: positive
0
50
100
150
200
250
Position on chromosome

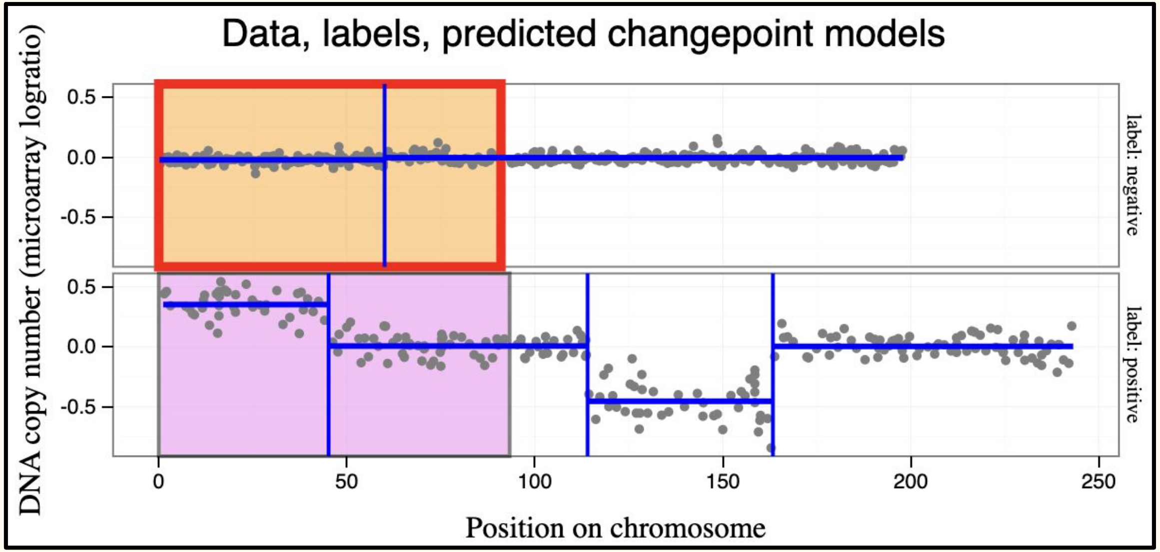

ROC curve, select point
246
57
13
AUC=1
1.0
0.5
0.0
True Positive Rate
False Positive Rate

ROC Point: 3

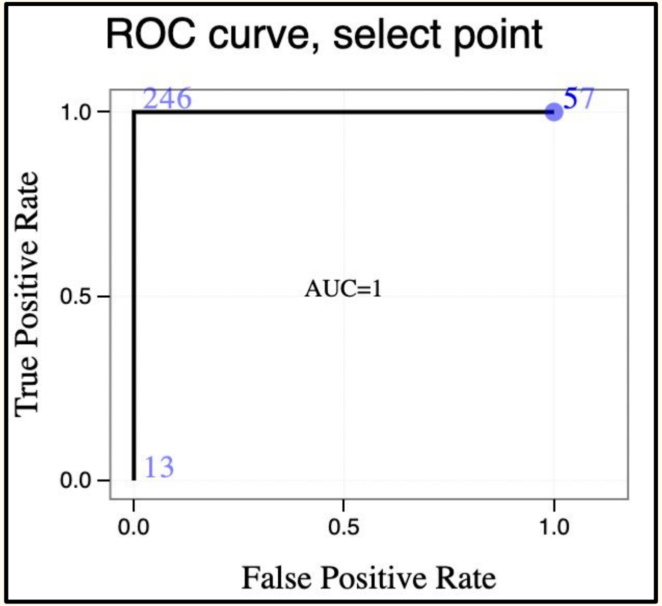

Data, labels, predicted changepoint models
DNA copy number (microarray logratio)
0.5
0.0
-0.5
0.5
0.0
-0.5
label: negative
label: positive
0
50
100
150
200
250
Position on chromosome


ROC curve, select point
True Positive Rate
1.0
0.5
0.0
246
57
13
AUC=1
0.0
0.5
1.0
False Positive Rate

ROC Point: 4

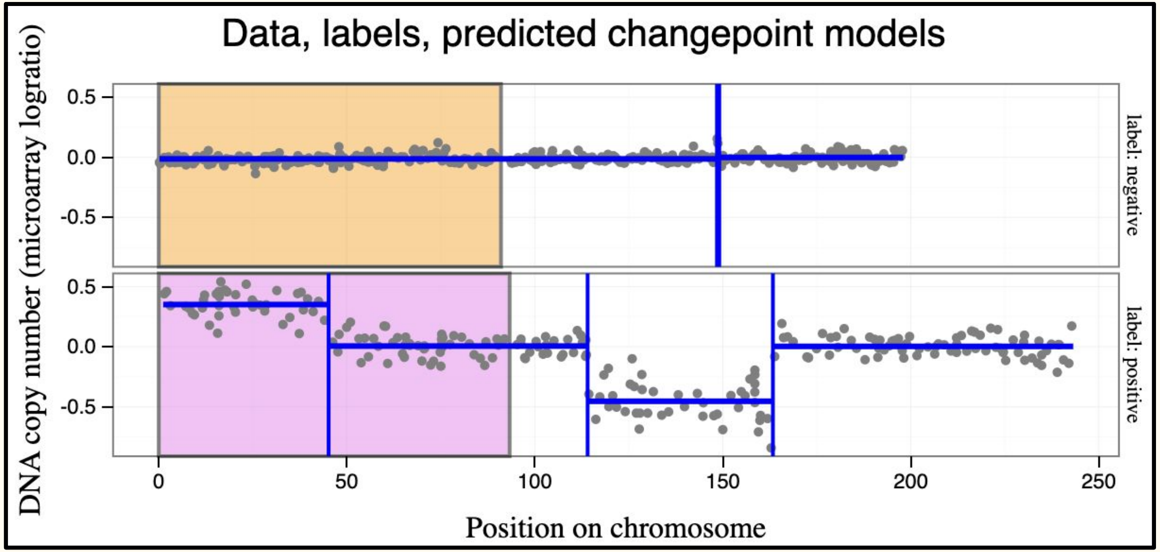


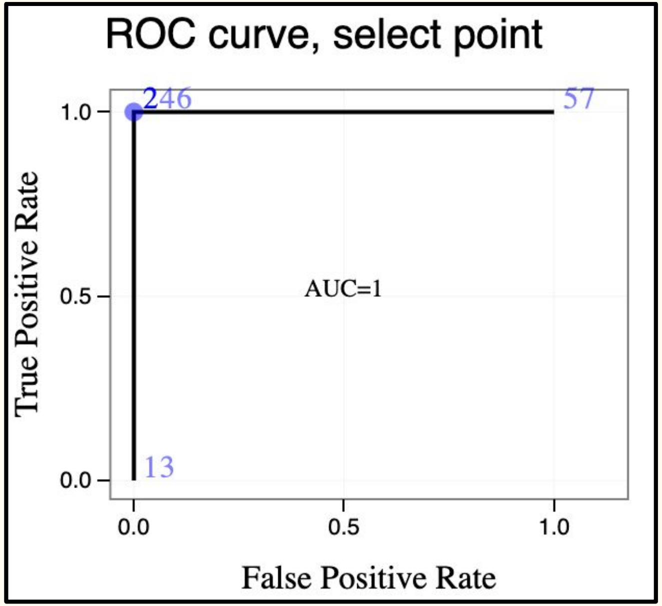
Data, labels, predicted changepoint models
DNA copy number (microarray logratio)
0.5
0.0
-0.5
label: negative
label: positive
0
50
100
150
200
250
Position on chromosome



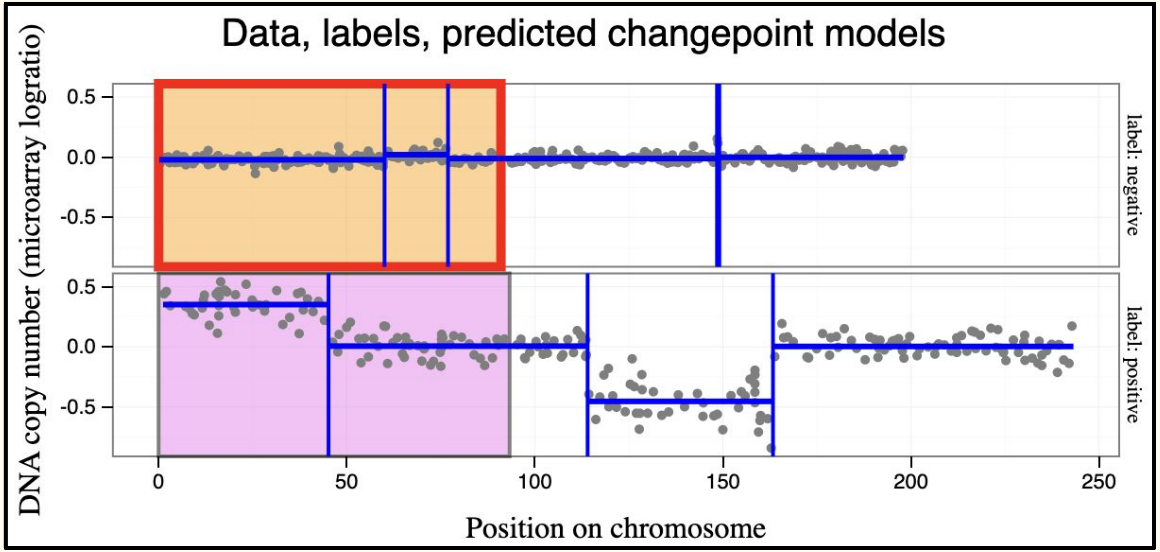
ROC curve, select point
246
57
13
AUC=1
True Positive Rate
False Positive Rate
1.0
0.5
0.0

ROC Point: 5

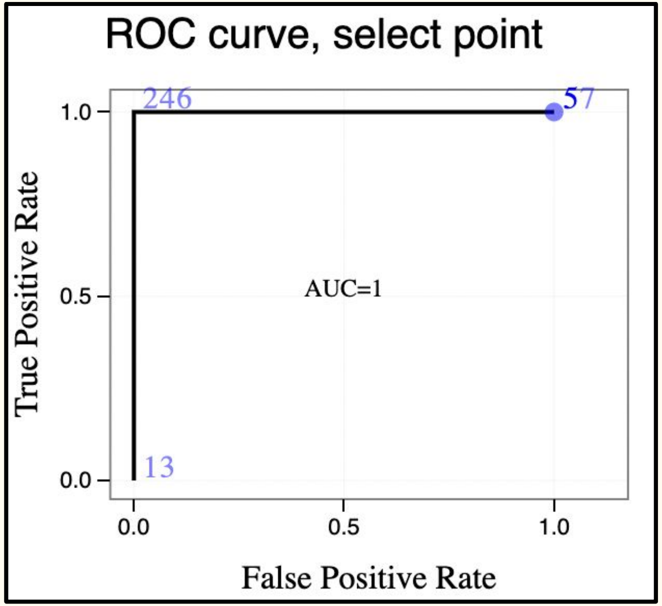

Data, labels, predicted changepoint models
DNA copy number (microarray logratio)
0.5
0.0
-0.5
label: negative
label: positive
0
50
100
150
200
250
Position on chromosome


ROC curve, select point
True Positive Rate
False Positive Rate
1.0
0.5
0.0
246
57
13
AUC=1

ROC Point: 6

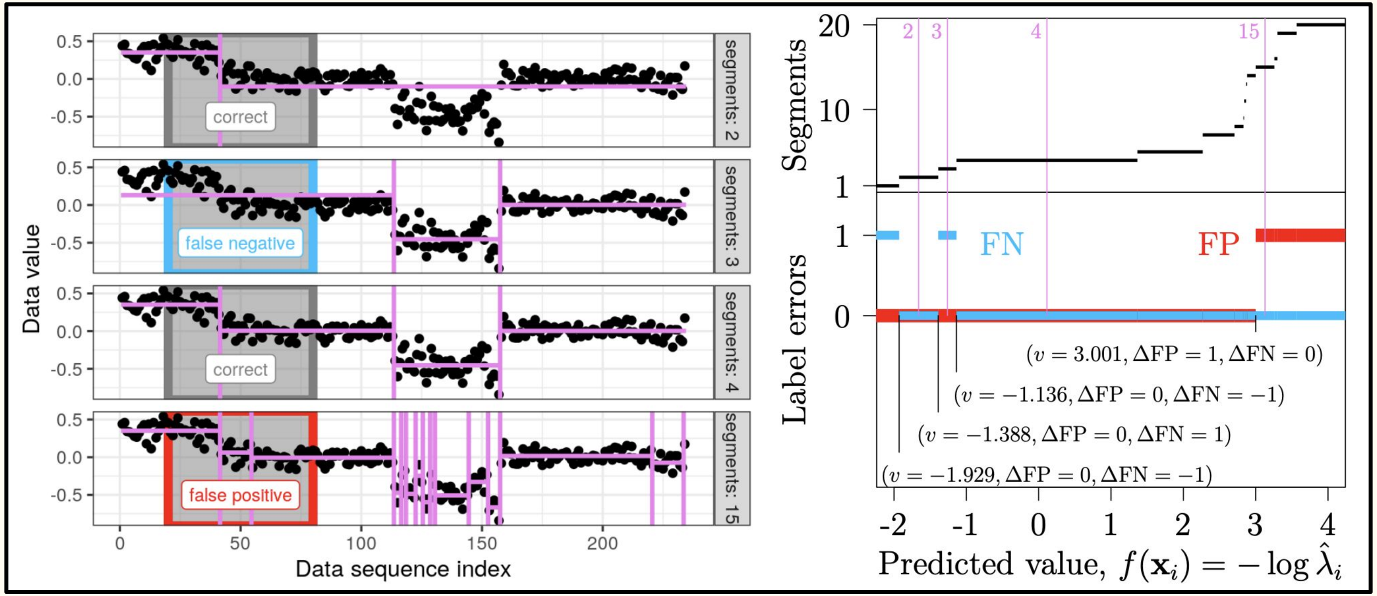


Data, labels, predicted changepoint models
DNA copy number (microarray logratio)
0.5
0.0
-0.5
0.5
0.0
-0.5
label: negative
label: positive
0
50
100
150
200
250
Position on chromosome



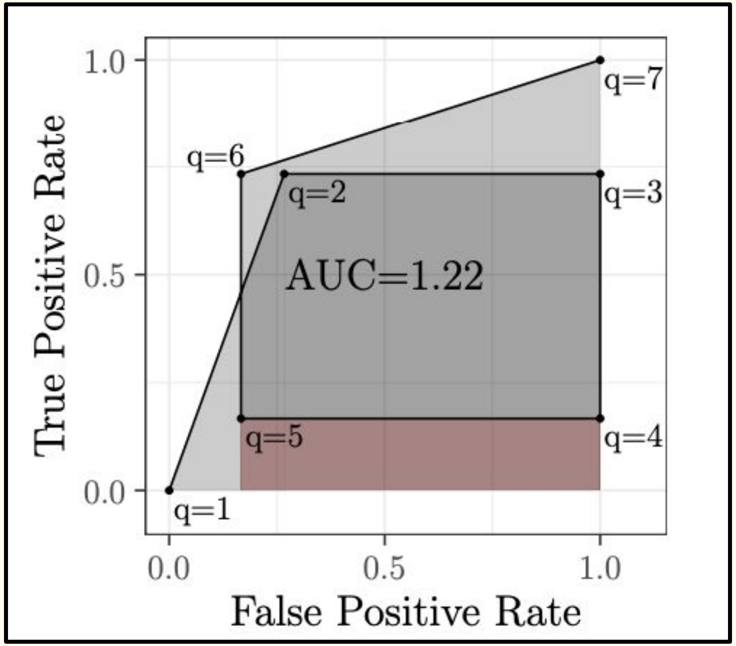
ROC curve, select point
True Positive Rate
False Positive Rate
246
57
13
AUC=1
1.0
0.5
0.0
0.0
0.5
1.0

ROC Point: 7



FN and FP are piecewise functions

ROC Curves can Loop?!



Example ROC curve with AUC $\geq$ 1 due to the loop which results in double counting the dark grey area (but single counting the red area which is positive counted twice and negative counted once). *[Hillman & Hocking 2021]*

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

Jonathan Hillman and
Toby Dylan Hocking — toby.hocking@nau.edu

July 6, 2021

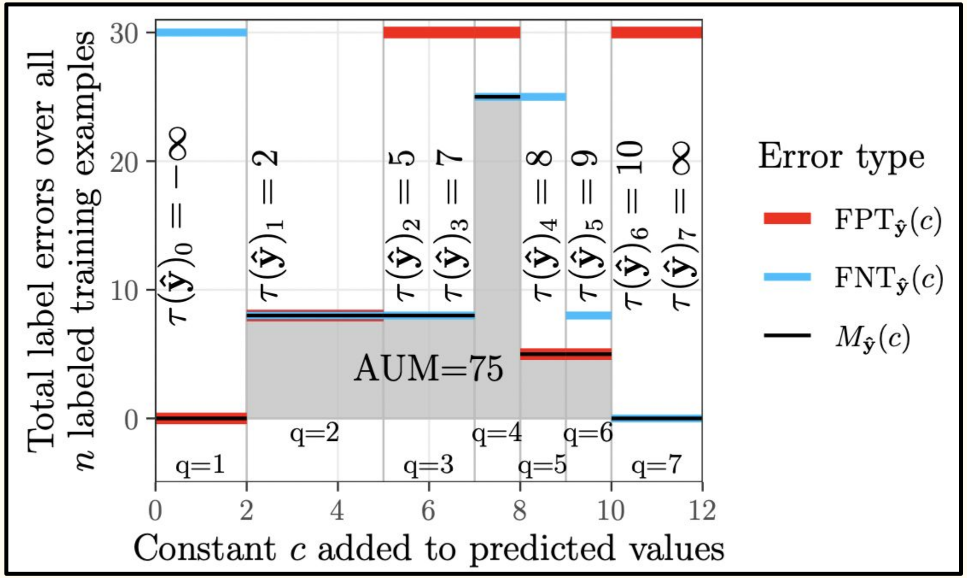
arxiv link

Area Under Min(FP, FN)

$$M_{\hat{\mathbf{y}}}(c) = \min\{\mathrm{FPT}_{\hat{\mathbf{y}}}(c), \mathrm{FNT}_{\hat{\mathbf{y}}}(c)\}.$$

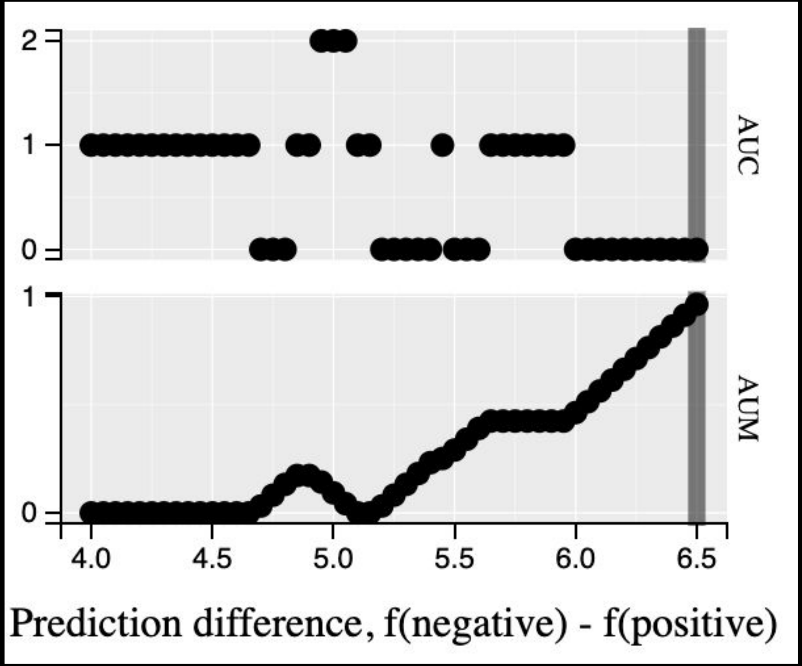
$$\mathrm{AUM}(\hat{\mathbf{y}}) = \int_{-\infty}^{\infty} M_{\hat{\mathbf{y}}}(c)\, dc.$$

- Motivation: certain problems non-convex AUC
- M(c) = Minimum of the False Positive Total and the False Negative Total for some constant added to our prediction vector
- FPT / FNT change at breakpoints
- Shown than minimizing AUM corresponds to maximizing AUC
- How do we minimize this?



How to minimize?

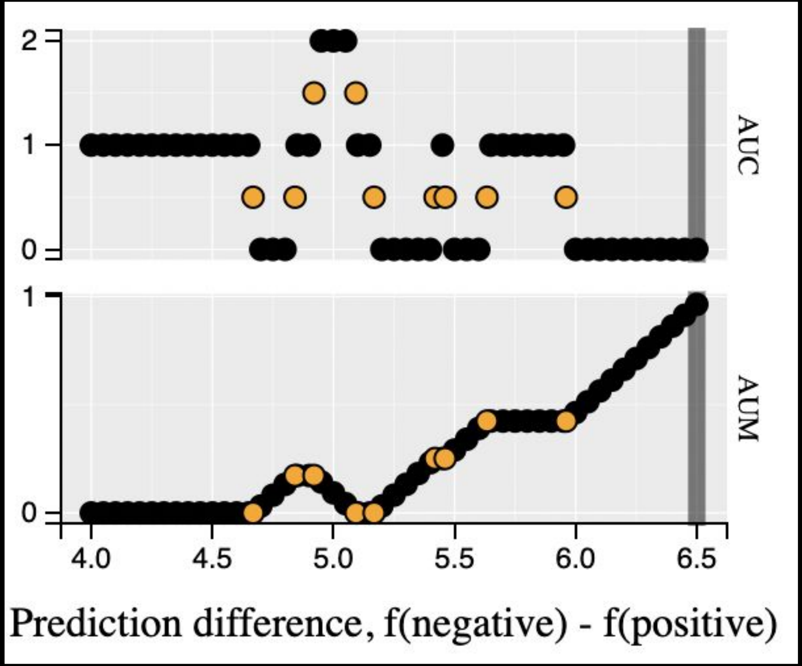
- Walk down AUM function
- Calculate gradient and move in the descent direction
- Line search to find the step size that decreases AUM the most



AUC vs. AUM

How to minimize?

- Non-differentiable points occur at breakpoints
- The gradient is not defined at these points!
- We use directional derivatives as a replacement



AUC vs. AUM with
non-differentiable points?!

Minimizing AUM for a Linear Model

Given a Linear Model,

$$\boxed{f(\mathbf{x}) = \mathbf{w}^\intercal \mathbf{x} + \beta}$$

we can define a weight update

$$\boxed{\mathbf{w}^{(j+1)} = \mathbf{w}^{(j)} - \alpha^{(j)} \mathbf{X}^\intercal \bar{\mathbf{D}}_{\text{AUM}}(\mathbf{X}\mathbf{w}^{(j)} + \beta^{(j)})}$$

using a matrix of directional derivatives for a given prediction vector

$$\boxed{\mathbf{D}_f(\mathbf{x}) = \begin{bmatrix} \nabla_{\mathbf{v}(-1,1)} f(\mathbf{x}) & \cdots & \nabla_{\mathbf{v}(-1,n)} f(\mathbf{x}) \\ \nabla_{\mathbf{v}(1,1)} f(\mathbf{x}) & \cdots & \nabla_{\mathbf{v}(1,n)} f(\mathbf{x}) \end{bmatrix}^\intercal}$$

Minimizing AUM for a Linear Model

Given a Linear Model,

$$\boxed{f(\mathbf{x}) = \mathbf{w}^\intercal \mathbf{x} + \beta}$$

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

Directional derivatives of AUM

we can define a weight update

$$\boxed{\mathbf{w}^{(j+1)} = \mathbf{w}^{(j)} - \alpha^{(j)} \mathbf{X}^\intercal \bar{\mathbf{D}}_{\text{AUM}}(\mathbf{X}\mathbf{w}^{(j)} + \beta^{(j)})}$$

using a matrix of directional derivatives for a given prediction vector

$$\boxed{\mathbf{D}_f(\mathbf{x}) = \begin{bmatrix} \nabla_{\mathbf{v}(-1,1)} f(\mathbf{x}) & \cdots & \nabla_{\mathbf{v}(-1,n)} f(\mathbf{x}) \\ \nabla_{\mathbf{v}(1,1)} f(\mathbf{x}) & \cdots & \nabla_{\mathbf{v}(1,n)} f(\mathbf{x}) \end{bmatrix}^\intercal}$$

Condensed running title of 50 characters or less: Sort-Based Surrogate for AUC Optimization

Five keywords: AUC, ROC, loss functions, gradient descent, optimization, line search.

Efficient line search for a piecewise linear relaxation of ROC curve optimization

Jonathan Wheeler JLW697@NAU.EDU
College of the Environment, Forestry, and Natural Sciences
Northern Arizona University
Flagstaff, AZ, USA

Toby Dylan Hocking TOBY.HOCKING@NAU.EDU
School of Informatics, Computing, and Cyber Systems
Northern Arizona University
Flagstaff, AZ, USA

Efficient AUM Line Search

- Exact line search is $O(n^2)$ in the worst case
- New approximate line search is O(n*log(n)) in the worse case

Condensed running title of 50 characters or less: Sort-Based Surrogate for AUC Optimization

Five keywords: AUC, ROC, loss functions, gradient descent, optimization, line search.

Efficient line search for a piecewise linear relaxation of ROC curve optimization

Jonathan Wheeler JLW697@NAU.EDU
College of the Environment, Forestry, and Natural Sciences
Northern Arizona University
Flagstaff, AZ, USA

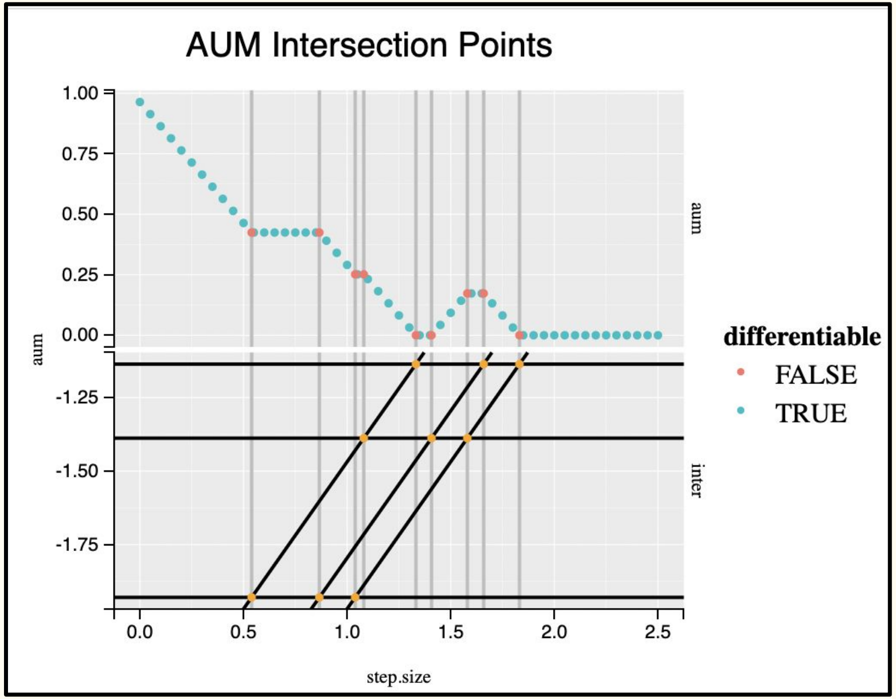
Toby Dylan Hocking TOBY.HOCKING@NAU.EDU
School of Informatics, Computing, and Cyber Systems
Northern Arizona University
Flagstaff, AZ, USA

Reconstructing the AUM line

- Breakpoints are tuples: (v, ΔFP, ΔFN). They tell us how the FP/FN functions change over time.

- Non-differentiable points line up with these intersections of the bottom lines.

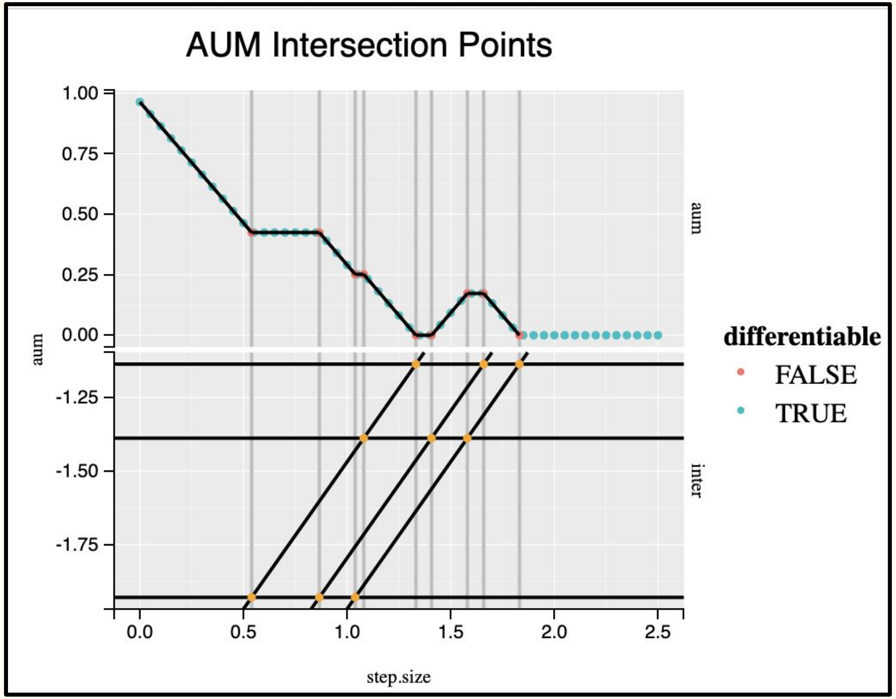
- Bottom lines can be constructed from breakpoint data



Graph by Jadon

Reconstructing the AUM line

- Goal: reconstruct the AUM line using these breakpoints.
- We wouldn't have to calculate the AUM at every point
- Gradient descent for AUM would be much faster



Graph by Jadon