3 Receiver Operating Characteristic (ROC) Curves and Shortest-Path Examples

The key idea of this paper is to show that paths graphs correspond to areas certain shape (or can be computed by counting lattice points). In this section we illustrate this idea using a few simple examples related to a Cayley graph generated from a vector and restricted to the orbit of the multiset $0^{n-k}1^k$ — i.e., the multipermutahedron graph.

In classification models True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN) are computed by comparing model prediction and ground truth values. A well-know metrics for a classification models are

$$\begin{aligned}\text{TPR (True Positive Rate)} &= \text{Sensitivity} = \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{FPR (False Positive Rate)} &= \frac{\text{FP}}{\text{FP} + \text{TN}}\end{aligned}$$

Definition 9 (ROC AUC)**.** *Let $P = [p_0, ...p_{n-1}]$ vector of distinct reals in $[0,1]$ listed in ascending order. Let v be a binary vector of length n with $(n-k)$ zeros and k ones. Intuitively, P is a vector of n propensity values computed by a binary classifier. v is a ground truth for the classifier.*

Let $T = \mathrm{P} + \epsilon$ be all possible threshold values we will consider. Assume ϵ is sufficiently small so that $\forall j, T_j + \epsilon < T_{j+1}$.

For each T_j from T we define a vector of "prediction" PRED_j as follows. Entries with $P < T_j$ are assigned value 0, and entries with $P >= T_j$ are assigned value 1. From PRED_j and v we can calculate the corresponding TP, FP, TN, FN, TPR, FPR, and hence (TPR_j, FPR_j) pair corresponding to T_j. Given such pairs for all j, we can plot ROC curve: TPR vs FPR. AUC is the area under this curve.

Proposition 1 (Path length and Area relation)**.** *Let $S(v)$ be a graph generated from a binary vector v with $(n-k)$ zeros and k ones by adjacent transpositions. Let L be the length of the shortest path from v to $\mathrm{sorted}(v) = [0, \ldots, 0, 1 \ldots, 1]$. Then $L = (1 - AUC)(n-k)k$.*

Or, equivalently, if A is the AREA ABOVE the ROC curve measured in the unit squares, then $A = L$.

In examples shown below we will assume $n = 10$ $k = 5$,

$$\begin{aligned}P &= [0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95] \\ T &= P + 0.01 \\ \mathrm{PRED}_j &= \mathrm{int}(P > T_j), \quad \forall j = 0, \ldots, (n-1)\end{aligned}$$

3.1 Example 1

$$\begin{aligned}v &= [0, 0, 0, 0, 0, 1, 1, 1, 1, 1] \\ FPR &= [1.0, 0.8, 0.6, 0.4, 0.2, 0.0, 0.0, 0.0, 0.0, 0.0, 0] \\ TPR &= [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.8, 0.6, 0.4, 0.2, 0]\end{aligned}$$

In this example area above the curve is 0, i.e. $A = 0$. Since $\text{sorted}(v) = v$, $L = 0 = A$.

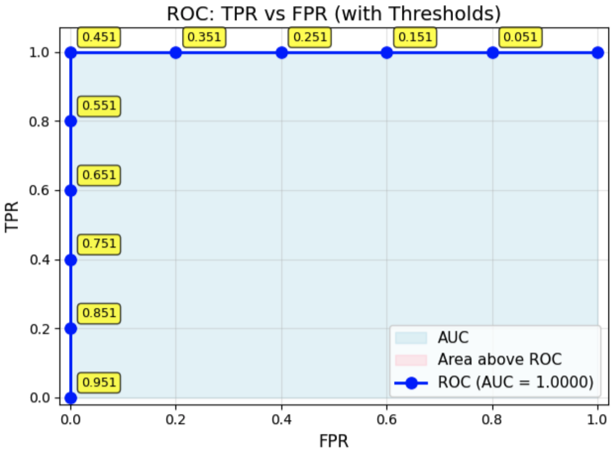

Figure 3: ROC: TPR vs FPR - perfect predictions case.

Suppose now $v = [0, 0, 0, 0, 1, 0, 1, 1, 1, 1]$, i.e., we modified the previous v by swapping adjacent elements only once. That is to say the path from new v to $\text{sorted}(v)$ is 1. It is easy to see that two adjacent predictions in the middle will be wrong. Hence we will reduce the area under ROC curve by one unit. So $A = L$.

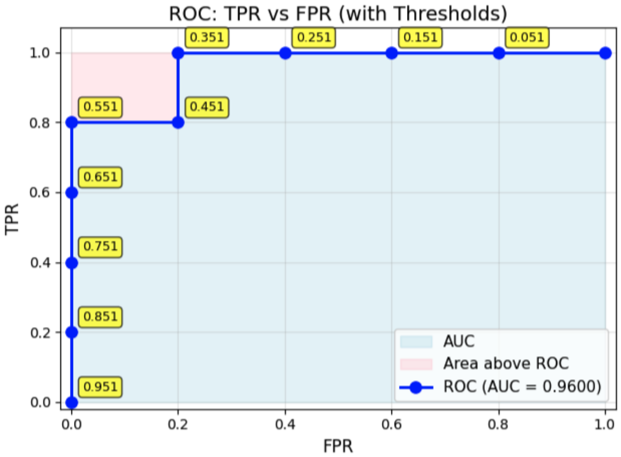

Figure 4: ROC: TPR vs FPR. One swap.

3.2 Example 2

$$\begin{aligned}
v &= [0,0,0,1,0,1,1,0,1,1] \\
FPR &= [1,0.8,0.6,0.4,0.4,0.2,0.2,0.2,0.0,0.0,0.0] \\
TPR &= [1,1.0,1.0,1.0,0.8,0.8,0.6,0.4,0.4,0.2,0.0]
\end{aligned}$$

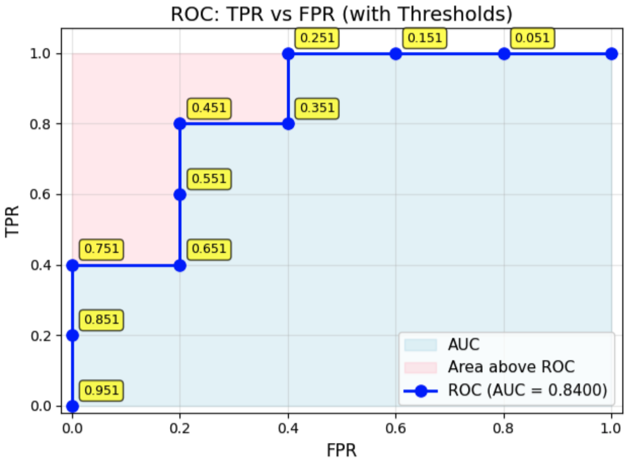

Figure 5: ROC Curve showing TPR vs FPR. Four swaps.

Sorting v using adjacent transpositions requires four swaps. Hence the shortest path from $v = [0,0,0,1,0,1,1,0,1,1]$ to sorted$(v) = [0,0,0,0,0,1,1,1,1,1]$ is 4. On the other hand ROC AUC is 0.84. The maximum area is 25 (5 x 5). The area above the curve is $(1 - 0.84) \cdot 25 = 4$.

One can compute the ROC curve (and the shortest path length) by looking at the original v. Start at the lower left corner of the TPR vs FPR plot, i.e., at point (0,0). Look at the right most value of v and keep moving to the left. For each element of v do the following: if it is 1, move up; if it is 0 move to the right.

Swap	**Transposition**	**Node in $S(v)$**
0	–	$v_0 = v = [0,0,0,1,0,1,1,0,1,1]$
1	$(3,4)$	$v_1 = [0,0,0,0,1,1,1,0,1,1]$
2	$(6,7)$	$v_2 = [0,0,0,0,1,1,0,1,1,1]$
3	$(5,6)$	$v_3 = [0,0,0,0,1,0,1,1,1,1]$
4	$(4,5)$	$v_4 = \text{sorted}(v) = [0,0,0,0,0,1,1,1,1,1]$