Classification metrics optimization: AUC and Kappa

In this video

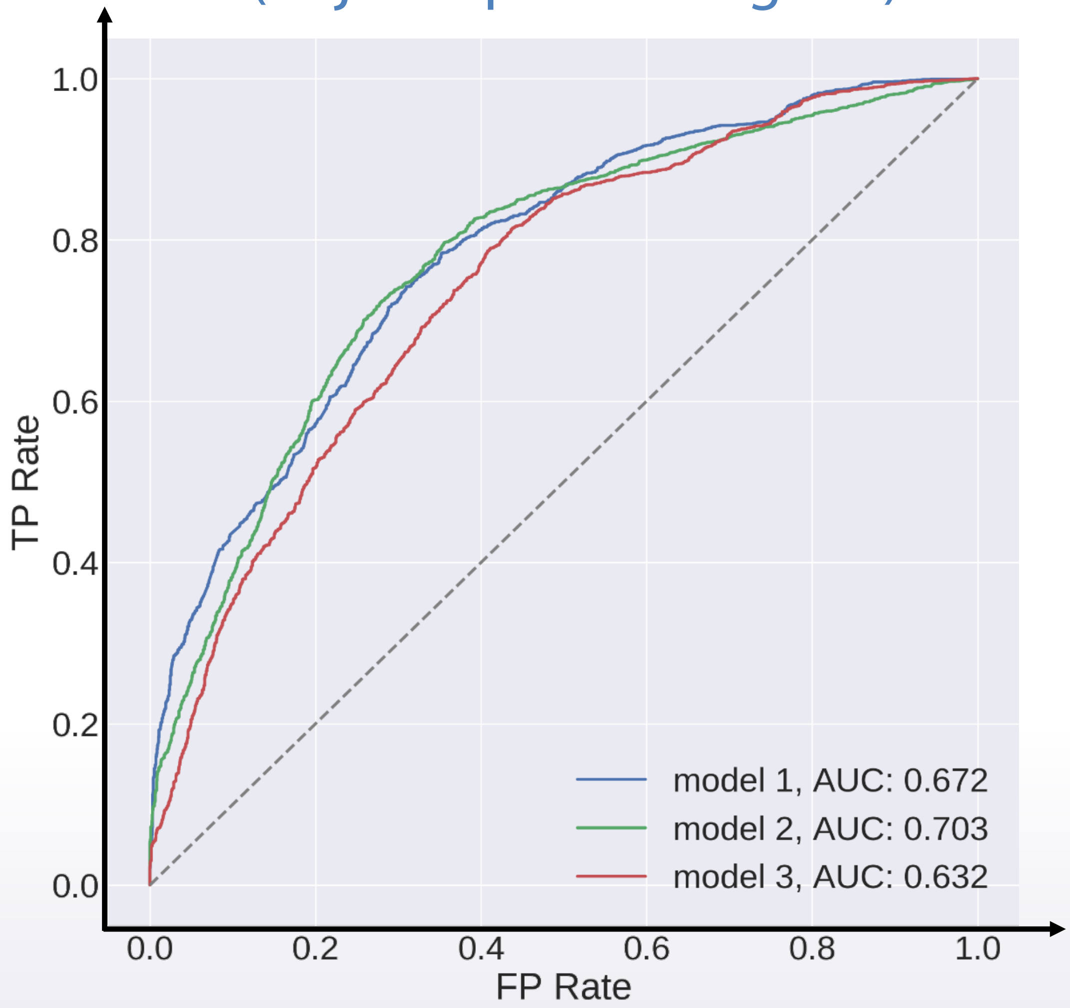
- Logloss
- Accuracy
- AUC
- (Quadratic weighted) Kappa

AUC (ROC)

How do you optimize it?

Run the right model

(or just optimize logloss)

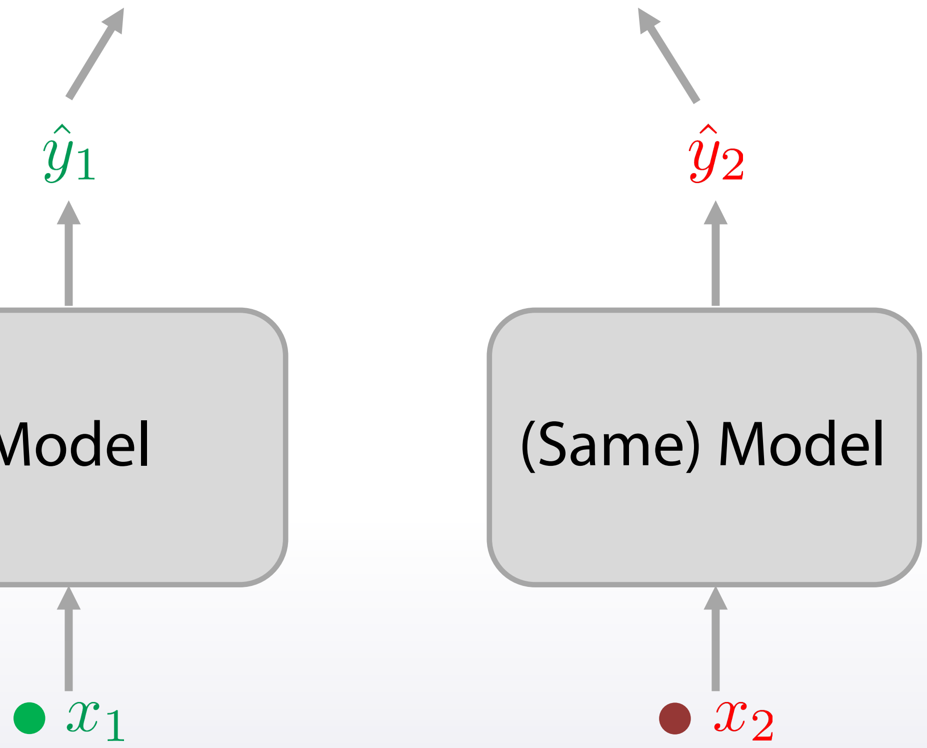


Pointwise loss

$$\min \sum_{i}^{N} l_{point}(\hat{y}_i; y_i)$$

We want:

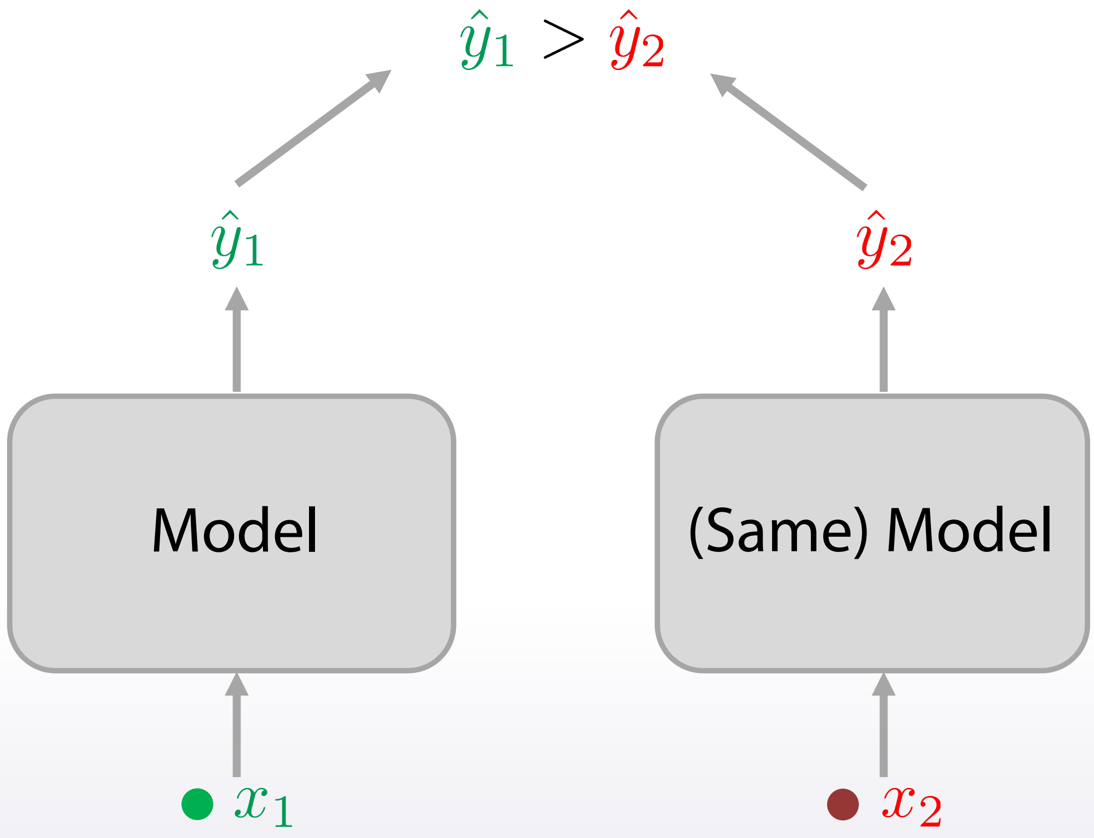
$$\hat{y}_1 = 1 \qquad \hat{y}_2 = 0$$



Pairwise loss

$$\min \sum_{i}^{N} \sum_{j}^{N} l_{pair}(\hat{y}_i, \hat{y}_j; y_i, y_j)$$

We want:



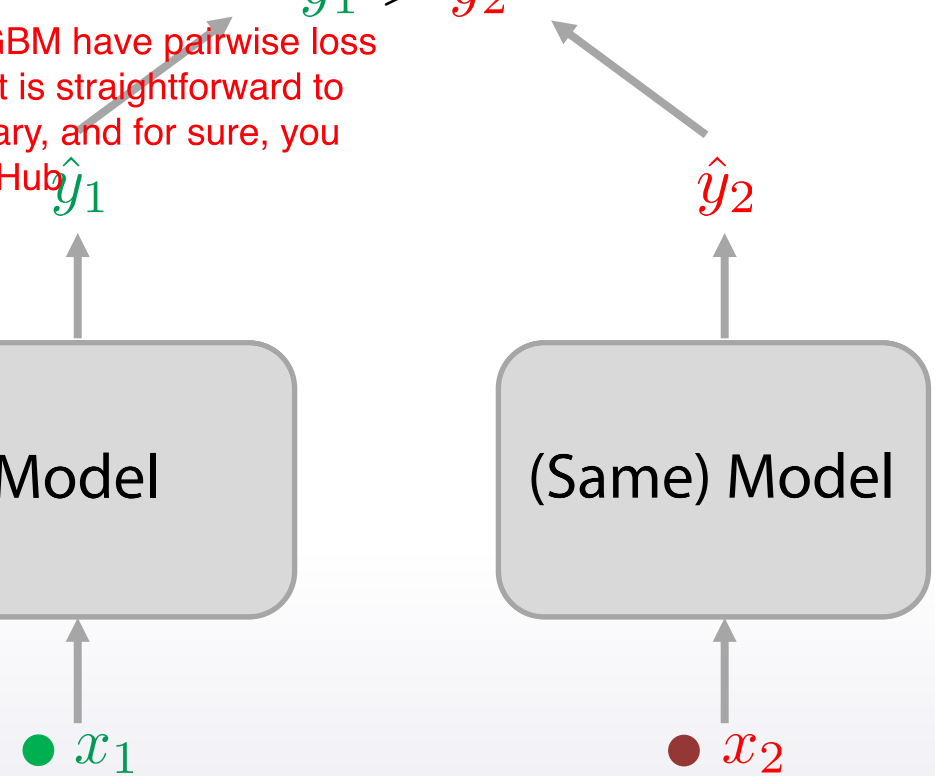
Pairwise loss

$$\text{Loss} = -\frac{1}{N_0 N_2} \sum_{j:y_j=1}^{N_1} \sum_{i:y_i=0}^{N_0} \log(\text{prob}(\hat{y}_j - \hat{y}_i))$$

We want:

$$\hat{y}_1 > \hat{y}_2$$

Well, basically, XGBoost, LightGBM have pairwise loss we've discussed implemented. It is straightforward to implement in any neural net library, and for sure, you can find implementations on GitHub



AUC

in practice, most people still use logloss as an optimization loss
without any more post processing.
I personally observed XGBoost learned with loglosst to give comparable AUC
score to the one learned with pairwise loss.

- **Tree-based**

  `XGBoost, LightGBM`

  ~~`sklearn.RandomForestClassifier`~~

- **Linear models**

  ~~`sklearn. LogisticRegression`~~

  ~~`sklearn.SGDRegressor`~~

  ~~`Vowpal Wabbit`~~

- **Neural nets**

  `PyTorch, Keras, TF –` not out of the box

Read the docs!

Quadratic weighted Kappa

How do you optimize it?

- **Optimize *MSE* and find right *thresholds***
  - *Simple*
- **Custom smooth loss for GBDT or neural nets**
  - Harder

MSE + tresholds

1. **Optimize MSE**

$$\text{Kappa}(y, \hat{y}) \approx 1 - \frac{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\text{hard to deal with part}} =$$

$$= 1 - \frac{\text{MSE}(y, \hat{y})}{\text{hard to deal with part}}$$

And it looks like everyone's logic is, well, there is MSE in the denominator, we can optimize it, and let's don't care about denominator. Well, of course it's not correct way to do it, but it turns out to be useful in practice.

MSE + tresholds

1. **Optimize MSE**

$$\text{Kappa}(y, \hat{y}) \approx 1 - \frac{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\text{hard to deal with part}} =$$
$$= 1 - \frac{\text{MSE}(y, \hat{y})}{\text{hard to deal with part}}$$

2. **Find right *thresholds***
   - Bad: `np.round(predictions)`
   - Better: optimize thresholds (by
grid search)

Smooth loss

Computer Science > Learning

On The Direct Maximization of Quadratic Weighted Kappa

David Vaughn, Derek Justice

(Submitted on 23 Sep 2015 (v1), last revised 6 Dec 2015 (this version, v3))

In recent years, quadratic weighted kappa has been growing in popularity in the machine learning community as an evaluation metric in domains where the target labels to be predicted are drawn from integer ratings, usually obtained from human experts. For example, it was the metric of choice in several recent, high profile machine learning contests hosted on Kaggle : this https URL , this https URL , this https URL . Yet, little is understood about the nature of this metric, its underlying mathematical properties, where it fits among other common evaluation metrics such as mean squared error (MSE) and correlation, or if it can be optimized analytically, and if so, how. Much of this is due to the cumbersome way that this metric is commonly defined. In this paper we first derive an equivalent but much simpler, and more useful, definition for quadratic weighted kappa, and then employ this alternate form to address the above issues.

Comments: realized some inaccuracies, and some sloppy reasoning. Need some time to fix
Subjects: **Learning (cs.LG)**
Cite as: arXiv:1509.07107 [cs.LG]
(or **arXiv:1509.07107v3** [cs.LG] for this version)

Submission history

From: David Vaughn [view email]
[v1] Wed, 23 Sep 2015 19:39:39 GMT (209kb,D)
[v2] Tue, 29 Sep 2015 21:30:43 GMT (199kb,D)
[v3] Sun, 6 Dec 2015 15:16:19 GMT (0kb,I)

Lesson conclusion

- **Target metric is how competitors are scored**
- **Target metric VS optimization loss**
- **Regression metrics**
  - MSE, RMSE, R-squared
  - MAE
  - MSPE, MAPE
  - RMSLE
- **Classification metrics**
  - Accuracy
  - Logloss
  - AUC
  - (Quadratic weighted) Kappa