

General approaches for metrics optimization

Overview

- Loss vs metric
- Approaches to metrics optimization in general

Loss and metric

- **Target metric** is what *we* want to optimize
- **Optimization loss** is what *model* optimizes

Loss and metric

- **Target metric** is what we want to optimize
- **Optimization loss** is what *model* optimizes



Loss and metric

- **Target metric** is what we want to optimize
- **Optimization loss** is what *model* optimizes



Synonyms: loss, cost, objective

Approaches for target metric optimization

- **Just run the right model!**
 - MSE, Logloss

The most common metrics like MSE, Logloss are implemented as loss functions in almost every library.

Approaches for target metric optimization

- **Just run the right model!**
 - MSE, Logloss
- **Preprocess train and optimize another metric**
 - MSPE, MAPE, RMSLE, ...

Approaches for target metric optimization

- **Just run the right model!**
 - MSE, Logloss
- **Preprocess train and optimize another metric**
 - MSPE, MAPE, RMSLE, ...
- **Optimize another metric, postprocess predictions**
 - Accuracy, Kappa

Approaches for target metric optimization

- **Just run the right model!**
 - MSE, Logloss
- **Preprocess train and optimize another metric**
 - MSPE, MAPE, RMSLE, ...
- **Optimize another metric, postprocess predictions**
 - Accuracy, Kappa
- **Write custom loss function**
 - Any, if you can

Custom loss for XGBoost

It's actually quite easy to define a custom loss function for XGBoost. We only need to implement a single function that takes predictions and the target values and computes first and second-order derivatives of the loss function with respect to the model's predictions.

- **Define an 'objective':**
 - function that computes *first and second order derivatives* w.r.t. predictions.

```
def logregobj(preds, dtrain):  
    labels = dtrain.get_label()  
    preds = 1.0 / (1.0 + np.exp(-preds))  
    grad = preds - labels  
    hess = preds * (1.0 - preds)  
    return grad, hess
```

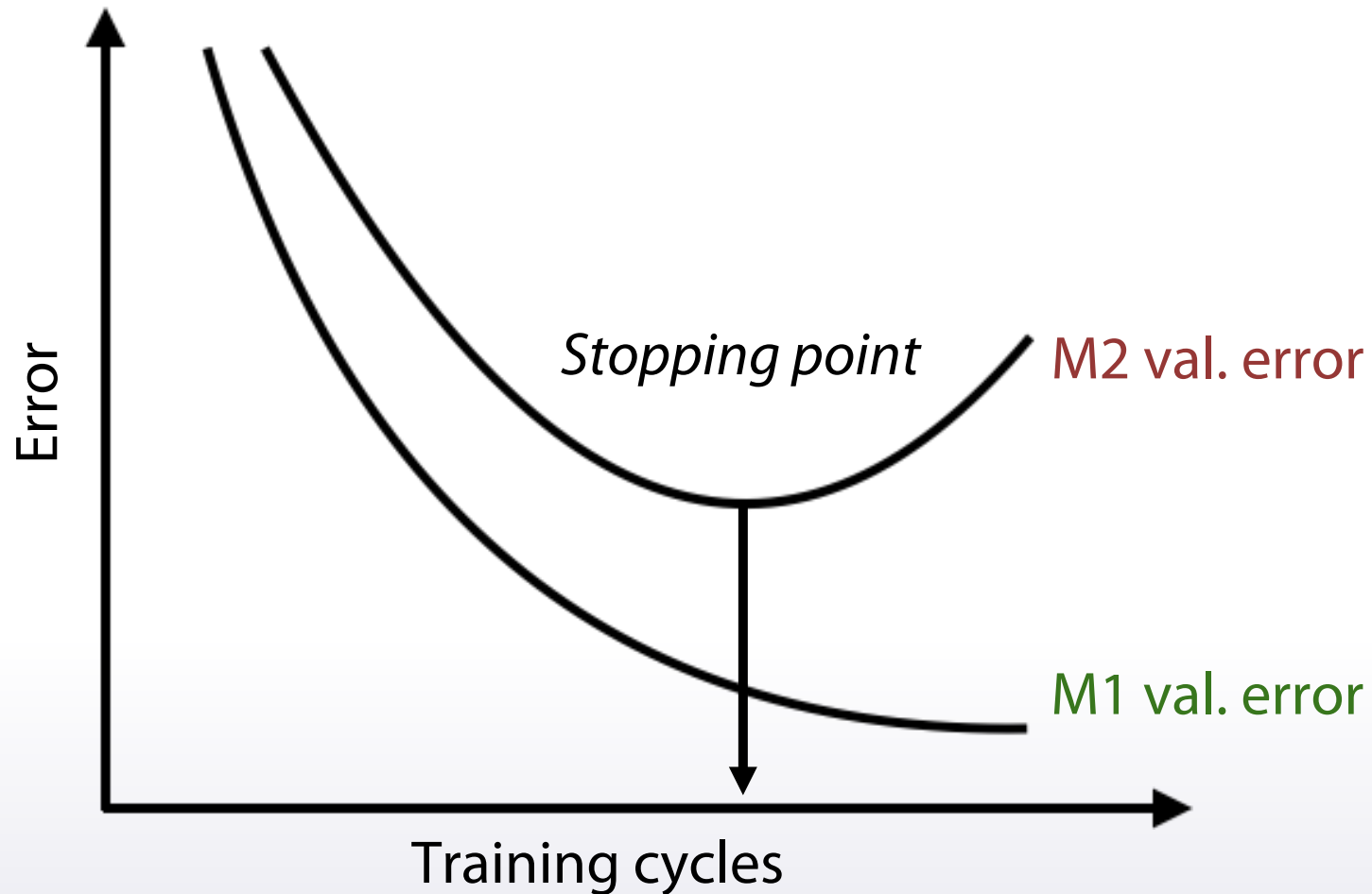
Approaches for target metric optimization

- **Just run the right model!**
 - MSE, Logloss
- **Preprocess train and optimize another metric**
 - MSPE, MAPE, RMSLE, ...
- **Optimize another metric, postprocess predictions**
 - Accuracy, Kappa
- **Write custom loss function**
 - Any, if you can
- **Optimize another metric, use early stopping**
 - Any

Early stopping

And you stop the training when the model starts to fit according to the desired metric and not according to the metric the model is truly optimizing

- Optimize metric **M1**, monitor metric **M2**
 - Stop when **M2 score** is the best



Conclusion

- **Loss vs metric**
- **Approaches in general:**
 - Just run the right model
 - Preprocess train and optimize another metric
 - Optimize another metric, postprocess predictions
 - Write a custom loss function
 - Optimize another metric, use early stopping