Geospatial Deep Learning

Losses and Assessment Metrics

Image from NASA: https://commons.wikimedia.org/wiki/File:Earth_Western_Hemisphere_transparent_background.png#filelinks

WV VIEW

AmericaView
Empowering Earth Observation Education
AmericaView.org - Est. 2003

The loss metric is a key component of the learning process, as it guides the parameter updates. Also, obtaining an unbiased assessment of model performance generally requires predicting to withheld testing or validation data and using appropriate assessment metrics. Thus, it is important to understand assessment and loss metrics if you want to implement deep learning.

This section will begin with a discussion of assessment metrics for both regression and classification problems followed by a discussion of loss metrics.

Module Topics

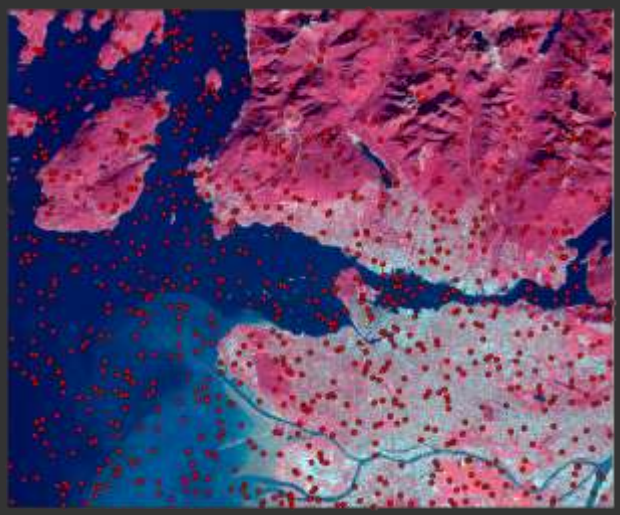
1. Training and validation data
2. Regression assessment metrics: RMSE, R-squared, AIC, and BIC
3. Multiclass classification assessment metrics: confusion matrix, overall accuracy, user's accuracy, and producer's accuracy
4. Kappa statistic and alternatives
5. Binary classification assessment metrics: precision, recall, F1-scoare, specificity, and negative predictive value
6. Aggregation of class-level metrics: micro vs. macro averaging
7. Multi-threshold assessment metrics: ROC curve, PR curve, and AUC
8. IoU and mAP
9. Regression loss functions: MSE, MAE, Huber, MSLE, and log-cosh
10. Classification loss metrics: BCE, CE, weighted CE, focal, Dice, Tversky, focal Tversky, and combo

2

These are the topics covered in this module.

Developing Testing or Validation Data

- Unbiased
- Randomized
- Non-overlapping with your training data
- Correctly proportioned
- Accurate



3

Before we begin, let's review the concepts of validation and test data. In deep learning, the validation data are used to assess the model at the end of each training epoch while the test data are used to assess the final model.

In order to produce an unbiased estimate of performance, the testing and validation data must be randomized in some way.

Also, the training and validation data should not overlap with each other or the test samples, or the same samples should not be used for both training and assessment. This is because algorithms tend to do a better job of predicting the training samples as opposed to new locations, a phenomenon known as overfitting. So, including training samples as test or validation samples could inflate the reported performance.

It is also recommended that test and validation samples be correctly proportioned relative to the map. For example, if you are mapping land cover and 70% of the mapped area is forest, then 70% of the testing and validation samples should also be forest.

Lastly, test and validation data should be accurate. The goal here is to compare the map product to reference data of high quality. It is generally assumed that no data are perfect. So, even the test and validation data will have some error.

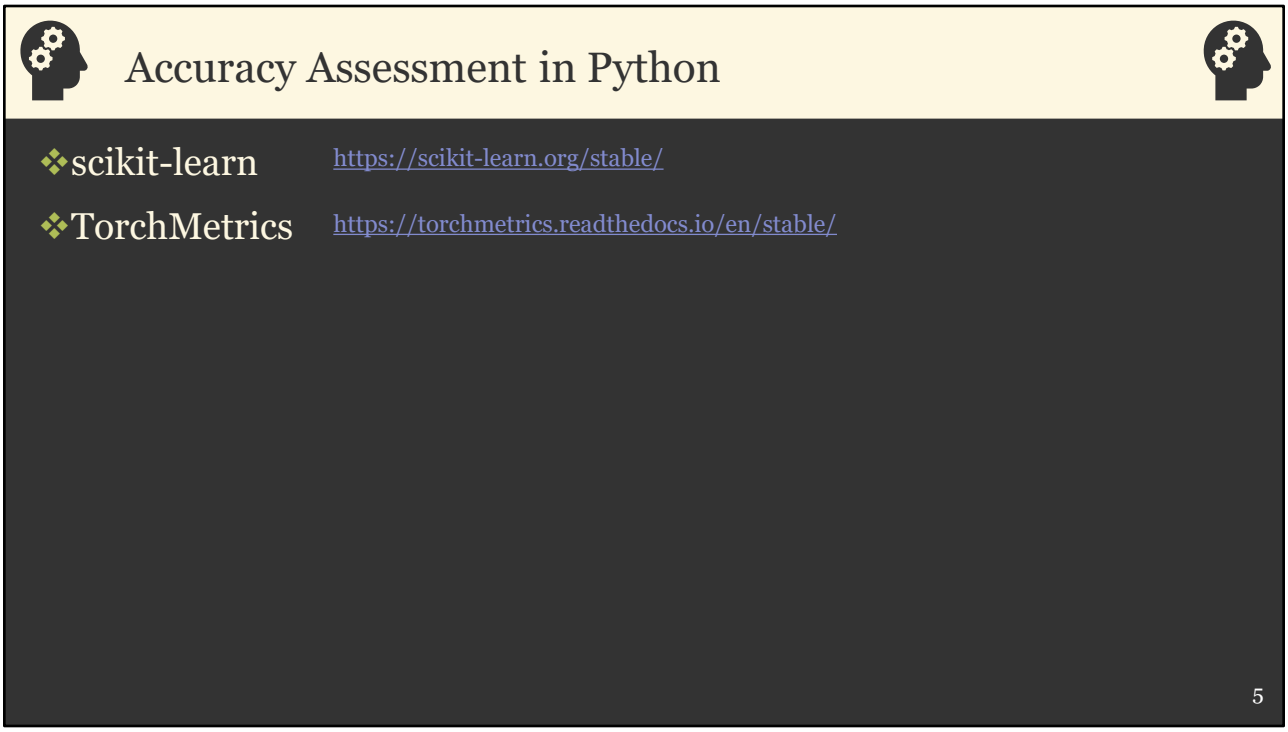
We try to avoid using the terms “ground truth” or “ground truthing” for this reason.

Accuracy Assessment in R

- yardstick https://yardstick.tidymodels.org/
- caret https://topepo.github.io/caret/
- rfUtilities https://cran.r-project.org/web/packages/rfUtilities/index.html
- diffeR https://cran.r-project.org/web/packages/diffeR/index.html
- pROC https://www.rdocumentation.org/packages/pROC/versions/1.18.0
- multiROC https://cran.r-project.org/web/packages/multiROC/index.html
- micer https://github.com/maxwell-geospatial/micer

4

I have found the R language and environment to be very useful for assessing models since several packages provide implementations of key assessment metrics. This slide provides links to some R packages that are especially useful for performing model assessment.



When using PyTorch, I have found the TorchMetrics package to be especially useful for implementing assessment metrics within training and validation loops as it allows for metrics to be accumulated or aggregated across multiple mini-batches. This package will be demonstrated throughout the PyTorch modules.

Scikit-learn also provides access to a variety of assessment metrics and tools.

Accuracy Measures: Regression

RMSE

- RMSE = $\sqrt{\Sigma(y - \hat{y})^2/n}$
- Root Mean Square Error (RMSE) will be in the units of y
- Mean Square Error (MSE) = $\Sigma(y - \hat{y})^2/n$

7

Continuous or numeric predictions, as opposed to classifications, can be assessed using root mean square error, or RMSE. MSE, or Mean Square Error, is simply RMSE without the square root applied. RMSE is reported in the units of the variable being predicted while MSE is reported in the square of the units. For example, if you are predicting chemical concentration in parts per million, then RMSE will be reported in parts per million and MSE will be reported in parts per million squared.

RMSE is calculated by comparing the predicted values to the reference values provided in the validation or test data. The values are subtracted to obtain a residual or error. The residuals are then summed, squared, then divided by the number of samples. This will provide MSE. The square root is taken to obtain RMSE. Lower RMSE and MSE suggest better performance.

R^2

- $R^2 = \frac{Total\ Sum\ of\ Squares - Residual\ Sum\ of\ Squares}{Total\ Sum\ of\ Squares}$

- Total Sum of Squares = $\Sigma(y - \bar{y})^2$
- Residual Sum of Squares = $\Sigma(y - \hat{y})^2$
- Proportion of variance in y explained by the model

8

Another means by which to assess continuous predictions is R^2. This value represents the proportion of variance in the quantity being predicted explained by the model. It is scaled from 0 to 1. 0 indicates that no variance is explained, or this is a poor model, while 1 indicates that all variance is explained. So, higher values are better. Again, this measure would not be appropriate for assessing a classification.

R^2 is not strictly a measure of model accuracy or performance. Instead, it is a measure of variance explained. However, it is still useful for assessing models.

Adjusted R^2

- When you have more than one x (multiple regression) you have to use adjusted R^2
- Adjusted $R^2 = 1 - \frac{(1-R^2)(N-1)}{(N-p-1)}$
- N = sample size, p = number of variables

9

When using the training data and when more than one predictor variable is used, R^2 must be modified to obtain adjusted R^2 to deal with inflation. The equation requires altering R^2 based on the sample size and number of predictor variables.

Note that it is not necessary to perform this adjustment when the withheld test or validation data are used to obtain R^2 as opposed to the training data.

Other Metrics

Akaike Information Criterion (AIC)
- Used for model comparisons
- Based in information theory
- Considers goodness of fit
- Penalizes for increased number of predictor variables/model complexity
- Larger value is better

Bayesian Information Criterion (BIC)
- Based on Bayes' Theorem
- Considers goodness of fit
- Penalizes for increased number of predictor variables/model complexity
- Lower value is better

10

Another option is the Akaike Information Criterion, or AIC. This is a measure of the relative quality of statistical models based on information theory. It takes into account both the goodness of fit (i.e., accuracy) and the complexity of the model. Specifically, models are penalized for having a larger number of included predictor variables. AIC is generally used to compare between models and select the best performing models while also taking into account the number of predictor variables. Larger values are preferred.

An alternative to AIC is the Bayesian Information Criterion (BIC). Similar to AIC, this method takes into account model performance while also penalizing for a large number of predictor variables. In contrast to AIC, it is based on Bayes' Theorem, and lower values indicate a better model as opposed to larger values, which is the case for AIC.

AIC and BIC are not commonly used in deep learning applications.

Accuracy Measures: Multiclass Classification

Confusion Matrix or Error Matrix

- Explores confusion between classes
- How were the classes confused?
- Should be created from randomized validation data

		Reference Data							
		Forested	*Pasture/ Grass*	*Barren*	*Cropland*	*Developed*	*Water*	Total	User's Accuracy
Classified Data	*Forested*	91	8	1	11	0	2	113	81%
	Pasture/ Grass	4	81	0	15	0	0	100	81%
	Barren	0	0	87	2	10	2	101	86%
	Cropland	3	11	0	69	0	0	83	83%
	Developed	0	0	10	0	73	0	83	88%
	Water	2	0	2	3	17	96	120	80%
	Total	100	100	100	100	100	100		
	Producer's Accuracy	91%	81%	87%	69%	73%	96%		

12

Many assessment metrics for classification problems are based on the confusion matrix, which is also commonly referred to as an error matrix.

Using validation or test samples, a confusion matrix compares the correct or reference classification at a location to the predicted class. Diagonal cells, shaded gray in the example, represent correct classifications. For example, 91 samples in the example were of the forested class and were correctly labelled as forested. All off-diagonals cells represent errors or misclassifications. For example, 4 samples were forested but were incorrectly labelled as pasture/grass.

The confusion matrix is very informative since it doesn't just summarize the total amount of error, but also differentiates sources of error. An analysis of the confusion matrix allows for an understanding of what classes are well mapped or poorly mapped and what classes are most confused with one another. For example, 15 cropland locations were misclassified as pasture/grass while 11 pasture/grass samples were misclassified as cropland. This indicates that these two classes are commonly confused. In contrast, only 4 of the water samples were misclassified to another class, suggesting that water is well mapped or differentiated.

It is important to note that the quality of the assessment will depend greatly on

the quality of the validation or test data. In remote sensing, we tend to avoid using the term “ground truth” for validation or test samples since it is assumed that even ground samples will have some errors or miss-classifications. Instead, we tend to use the term validation data, test data, or reference data. Even if there are errors, it is assumed that the validation or test data are more accurate than the classification that is being assessed. So, it is important to compare your classification to a dataset this is more accurate. Also, it is generally best that validation samples are collected using randomized sampling methods so as not to bias the assessment. If an analyst hand-picks validation or test samples, these samples are likely to not represent the true landscape conditions.

Confusion Matrix or Error Matrix

- Explores confusion between classes
- How were the classes confused?
- Should be created from randomized validation data

		Reference Data			
		Forested	*Not Forest*	Total	User's Accuracy
Classified Data	*Forested*	94	15	109	86%
	Not Forest	6	85	91	93%%
	Total	100	100		
	Producer's Accuracy	94%	85%		

13

This slide provides another example of a confusion matrix. In this example, only two classes differentiated.

Metrics

- What can be calculated from a confusion matrix?
- Overall Accuracy = $\frac{\text{Number of Features Correctly Classified}}{\text{Total Number of Features}} \times 100$
- Class User's Accuracy = $\frac{\text{Number of Features of Specific Class Correctly Classified}}{\text{Row Total}} \times 100$
- Class Producer's Accuracy = $\frac{\text{Number of Features of Specific Class Correctly Classified}}{\text{Column Total}} \times 100$

14

From the confusion matrix, it is possible to derive measures of overall classification accuracy and class-level accuracies.

Overall accuracy is simply the number of correctly classified samples divided by the total number of samples. In other words, it is the sum of the diagonal divided by the sum of the entire table. This metric is generally reported either as a proportion (0 to 1) or as a percentage (0% to 100%).

We will discuss the class-level accuracies on the next slide.

Metrics

User's Accuracy

- 1 - Commission error
- Probability that a feature classified on the map actually represents that category

$$\frac{\text{Number of Features of Specific Class Correctly Classified}}{\text{Row Total}} \times 100$$

Producer's Accuracy

- 1 - Omission error
- Probability of a reference feature being correctly classified

$$\frac{\text{Number of Features of Specific Class Correctly Classified}}{\text{Column Total}} \times 100$$

15

For each mapped class, two different measures of accuracy can be calculated. User's accuracy is 1 minus the commission error (error associated with samples being included in the wrong class). In contrast, producer's accuracy is 1 minus omission error (error associated with samples not being included in the correct class). In other words, user's accuracy represents the probability that a feature classified on the map actually represents that category on the ground whereas producer's accuracy relates to the probability of a reference feature being correctly classified. Similar to overall accuracy, these metrics are reported as proportions (0 to 1) or percentages (0% to 100%).

The standard way to generate a confusion matrix, at least in remote sensing, is to define the columns using the reference data classifications and the rows using the predicted classes. When the rows and columns are defined using this standard, user's accuracy for each class is calculated as the number correct for the class divided by the row total, which represents the number of samples mapped to that class. In contrast, producer's accuracy for a class is the number correct for the class divided by the column total, which represents the number of samples from that class in the validation data.

I actually find these terms to be very confusing. I prefer to simply report commission and omission error, as I feel these terms are more easily understood. However, reporting user's and producer's accuracy is the standard

in remote sensing.

Kappa

- Kappa, Kappa Statistic, Cohen's Kappa, K-Hat ($\widehat{K}$)
- Sometimes you are right by random chance
- Kappa offers an adjustment of overall accuracy that takes into account random or change agreement
- It is considered to be more robust than overall agreement

$$\frac{[\text{Total Number of Features} * \text{Number of Correct Features}] - [\text{Sum of All Row Totals} * \text{Column Totals}]}{[\text{Total Number of Features squared}] - [\text{Sum of All Row Totals} * \text{Column Totals}]}$$

16

Sometimes features are correctly classified simply by random chance. The Kappa statistic offers a correction of overall accuracy that takes into account chance agreement. It is a commonly reported metric in remote sensing.

The numerator of Kappa is calculated as the total number of features multiplied by the number of correct features. From this, you subtract the sum of all the row and associated column totals. The denominator is calculated as the square of the number of samples with the sum of all the row and associated column totals subtracted.

Other terms used for Kappa include the Kappa statistic, Cohen's Kappa, and K-Hat ($\widehat{K}$). $\widehat{K}$ indicates that a population statistic is being estimated from a sample.

Kappa is generally reported as a proportion (0 to 1) as opposed to a percentage. It is also possible to calculate confidence intervals and statistically compare Kappa values for different classifications. We will not discuss these additional statistical considerations in this course.

Calculation Example 1

		Reference Data							
		Forested	*Pasture/ Grass*	*Barren*	*Cropland*	*Developed*	*Water*	Total	User's Accuracy
Classified Data	*Forested*	91	8	1	11	0	2	113	81%
	Pasture/ Grass	4	81	0	15	0	0	100	81%
	Barren	0	0	87	2	10	2	101	86%
	Cropland	3	11	0	69	0	0	83	83%
	Developed	0	0	10	0	73	0	83	88%
	Water	2	0	2	3	17	96	120	80%
	Total	100	100	100	100	100	100		
	Producer's Accuracy	91%	81%	87%	69%	73%	96%		

Overall Accuracy

$$= \frac{91+81+87+69+73+96}{600}$$

$$= 0.828 \text{ or } 82.8\%$$

Kappa

$$= \frac{(((91+81+87+69+73+96)*600) - (100*113+100*100+100*101+100*83+100*83+100*120))}{(600^2 - (100*113+100*100+100*101+100*83+100*83+100*120))} = 0.794$$

17

Class user's and producer's accuracies have already been calculated on the margins of this example confusion matrix. Make sure you understand how these measures were derived.

I have now provided the calculations for overall accuracy and Kappa. Make sure you understand how these calculations were derived. For this example, overall accuracy was 0.828 while Kappa was 0.794.

Calculation Example 2

		Reference Data			
		Forested	*Not Forest*	Total	User's Accuracy
Classified Data	*Forested*	94	15	109	86%
	Not Forest	6	85	91	93%%
	Total	100	100		
	Producer's Accuracy	94%	85%		

Overall Accuracy

$$= \frac{94+85}{200}$$

$= 0.895$ or 89.5%

Kappa

$$= \frac{(((94+95)*200) - (100*109+100*100*91))}{(200^2 - (100*109+100*100*91))}$$

$= 0.890$

18

This slide provides the metric calculations for the second example.

Should we use Kappa?

International Journal of Remote Sensing
Vol. 32, No. 15, 10 August 2011, 4407–4429

Taylor & Francis

Death to Kappa: birth of quantity disagreement and allocation disagreement for accuracy assessment

ROBERT GILMORE PONTIUS JR* and MARCO MILLONES
School of Geography, Clark University, Worcester, MA 01610, USA

(Received 27 August 2010; in final form 20 December 2010)

The family of Kappa indices of agreement claim to compare a map's observed classification accuracy relative to the expected accuracy of baseline maps that can have two types of randomness: (1) random distribution of the quantity of each category and (2) random spatial allocation of the categories. Use of the Kappa indices has become part of the culture in remote sensing and other fields. This article examines five different Kappa indices, some of which were derived by the first author in 2000. We expose the indices' properties mathematically and illustrate their limitations graphically, with emphasis on Kappa's use of randomness as a baseline, and the often-ignored conversion from an observed sample matrix to the estimated population matrix. This article concludes that these Kappa indices are useless, misleading and/or flawed for the practical applications in remote sensing that we have seen. After more than a decade of working with these indices, we recommend that the profession abandon the use of Kappa indices for purposes of accuracy assessment and map comparison, and instead summarize the cross-tabulation matrix with two much simpler summary parameters: quantity disagreement and allocation disagreement. This article shows how to compute these two parameters using examples taken from peer-reviewed literature.

Foody, G.M., 2020. Explaining the unsuitability of the kappa coefficient in the assessment and comparison of the accuracy of thematic maps obtained by image classification. *Remote Sensing of Environment*, *239*, p.111630.

Pontius Jr, R.G. and Millones, M., 2011. Death to Kappa: birth of quantity disagreement and allocation disagreement for accuracy assessment. *International Journal of Remote Sensing*, *32*(15), pp.4407-4429.

Remote Sensing of Environment

Explaining the unsuitability of the kappa coefficient in the assessment and comparison of the accuracy of thematic maps obtained by image classification

Giles M. Foody

19

Some researchers have argued that the Kappa statistic should no longer be used. Specifically, they argue that the adjustment for chance agreement is not necessary and misleading. Despite these suggestions, Kappa is still a very commonly reported metric. So, I have included an explanation of it in this course. However, you should be aware that its use is currently somewhat controversial.

If you are interested in this debate, I would recommend the papers referenced on this slide.

Alternatives to Kapha

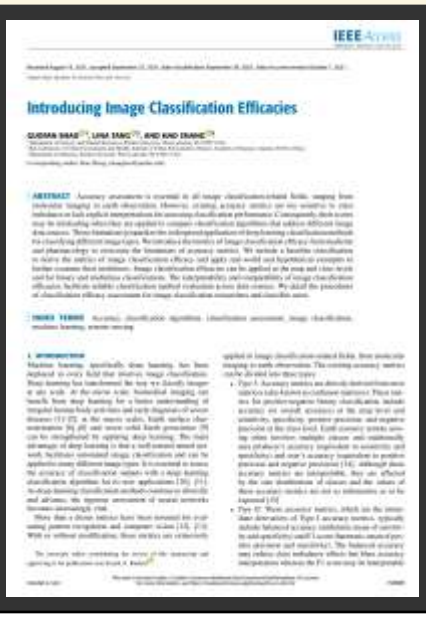
- Quantity disagreement = Difference in proportion of classes
- Allocation disagreement = Difference in spatial allocation of classes (further divided into exchange and shift)
- Quantity Disagreement + Allocation Disagreement = Overall Disagreement

20

Pontius and Millones (2011) suggest some alternative metrics to replace Kappa. Specifically, they have argued for the use of quantity disagreement and allocation disagreement. Quantity disagreement relates to the differences in proportions of classes between the reference data and classification result whereas allocation disagreement, which is further divided into exchange and shift disagreement, relates to differences in the location or spatial allocation of the refence data and classification result.

I feel that these metrics are informative and worth including in accuracy assessment workflows and reports. However, they have not gained wide use and acceptance in the field.

MICE



- Map-Level Image Classification Efficacy
- Alternative to Kappa

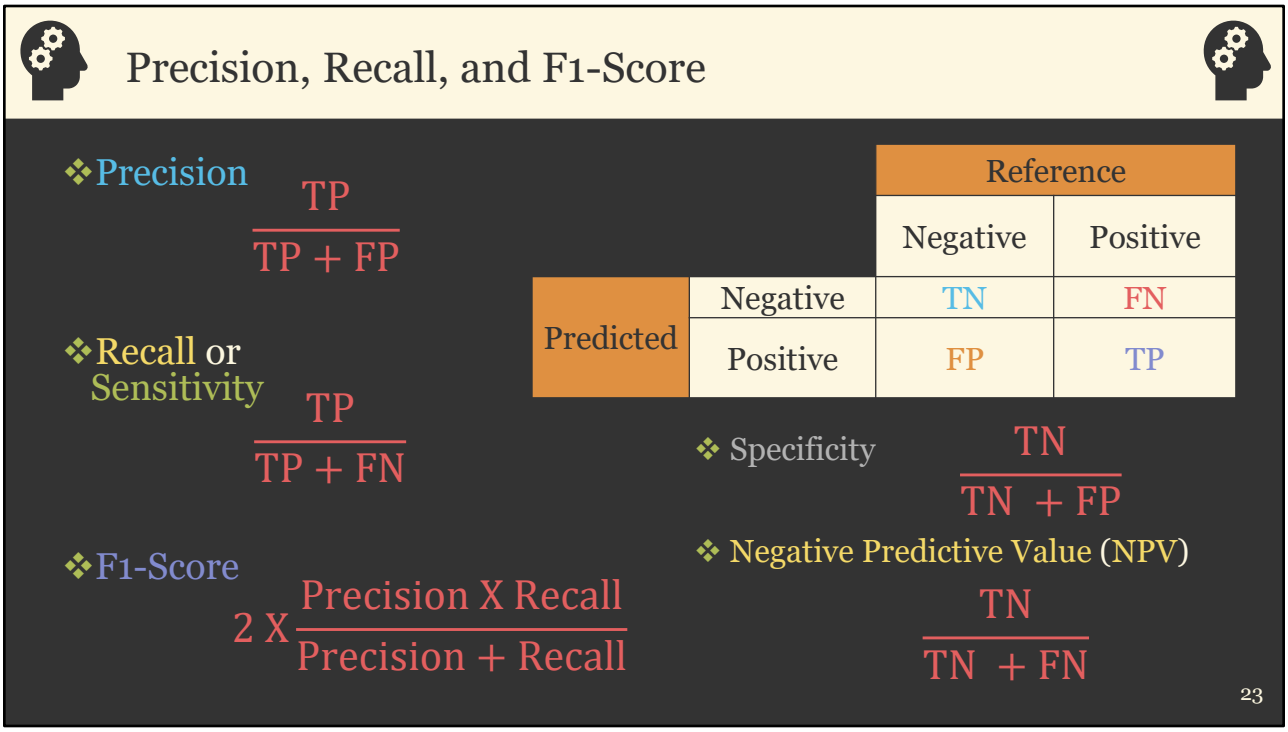
Shao, G., Tang, L. and Zhang, H., 2021. Introducing image classification efficacies. *IEEE Access*, *9*, pp.134809-134816.

21

Another alternative to Kappa that has been recently suggested is the map-level image classification efficacy (MICE) metric. We will not discuss this metric in detail in this course. However, if you are interested in learning more, please have a look at the publication referenced here.

In contrast to Kappa, MICE only uses the reference class margin totals as opposed to both the reference and classification margin totals to adjust overall accuracy for chance agreement. This method has been suggested to be robust to accuracy inflation due to chance agreement and meaningful when the number of samples per class is imbalanced while not suffering from the logical flaws of Kappa.

Accuracy Measures: Binary Classification



When only a single class is differentiated from the image background, as is common in feature extraction tasks, or when only two classes are mapped to generate a binary output, an alternative terminology is commonly used in comparison to multiclass classification.

Specifically, if one category represents a positive case while the other represents a background or negative case, the terms true positive (TP), false positive (FP), true negative (TN), and false negative (FN) are commonly used. TP samples are those that are in the positive class and are correctly mapped as positive while FPs are not in the positive class but are incorrectly mapped as a positive case. TNs are in the negative class and are correctly mapped as negative while FNs are mapped as negative when they are actually positive. From this cross tabulation, it is possible to derive a variety of metrics.

Precision represents the proportion of the samples that is correctly classified within the samples predicted to be positive and is equivalent to user's accuracy (1 – commission error) for the positive class. Recall or sensitivity represents the proportion of the reference data for the positive class that is correctly classified and is equivalent to producer's accuracy for that class.

The F1-score, or Dice coefficient, is the harmonic mean of precision and recall, while specificity represents the proportion of negative reference samples that is

correctly predicted and is thus equivalent to the producer's accuracy for the negative class. The user's accuracy for the negative class is termed the negative predictive value (NPV).

Binary and Multiclass Assessment Crosswalk

Metric	Equation	Relation to Multi-Class Metrics
Overall Accuracy (OA)	$\frac{TP + TN}{TP + TN + FP + FN}$	Overall Accuracy
Recall (Sensitivity) True Positive Rate (TPR)	$\frac{TP}{TP + FN}$	Producer's Accuracy for positive class
Precision Positive Predictive Value (PPV)	$\frac{TP}{TP + FP}$	User's Accuracy for positive case
F1-Score (Dice Coefficient)	$2 \text{ X} \frac{\text{Precision X Recall}}{\text{Precision} + \text{Recall}}$	Harmonic mean of Producer's Accuracy and User's Accuracy for the positive case
Specificity True Negative Rate (TNR)	$\frac{TN}{TN + FP}$	Producer's Accuracy for negative case
Negative Predictive Value (NPV)	$\frac{TN}{TN + FN}$	User's Accuracy for negative case

24

The goal of this slide is to provide a crosswalk between binary and multiclass assessment metric terminology.

Overall accuracy is calculated the same for binary and multiclass classification problems as the total correct divided by the total number of samples.

Recall (also sometimes referred to as sensitivity or true positive rate (TPR)) is equivalent to the producer's accuracy for the positive case. All these metrics quantify 1 – omission error.

Precision (also sometimes referred to as positive predictive value (PPV)) is equivalent to the user's accuracy for the positive case. These metrics quantify 1 – omission error.

F1-score, which is also referred to as the Dice coefficient, is equivalent to the harmonic mean of the user's and producer's accuracies for the positive case.

For the negative case, specificity represents producer's accuracy (i.e., 1 – omission error) while negative predictive value (NPV) represents user's accuracy (i.e., 1 – commission error).

Aggregated Multiclass Metrics

- Macro-Averaging
- Micro-Averaging

$$\text{Precision} = \frac{TP}{TP+FP} \qquad \text{Recall} = \frac{TP}{TP+FN}$$

$$\text{Dice} = 2\ X \frac{\text{Precision X Recall}}{\text{Precision+Recall}}$$

$$\text{Dice} = 2\ X \frac{2\ X\ TP}{(TP+FP)+(TP+FN)}$$

Class	TP	FP	FN	Precision	Recall	F1-Score/ Dice
A	120	17	41	0.88	0.75	0.81
B	96	12	17	0.89	0.85	0.87
C	111	32	21	0.78	0.84	0.81

$$\text{Dice (Macro)} = \frac{0.81+0.87+0.81}{3l} = 0.83$$

$$\text{Dice (Micro)} = \frac{2*(120+96+111)}{(120+96+111+17+12+32)+(120+96+111+41+17+21)} = 0.83$$

https://stephenallwright.com/micro-vs-macro-f1-score/

25

It is possible to aggregate class-level metrics to a single value. Examples include class aggregated recall (i.e., producer's accuracy), precision (i.e., user's accuracy), and F1-score (i.e., the Dice coefficient). This slide provides an example for the Dice coefficient or F1-score. When more than two classes are differentiated, or there is not a positive and background case, background or negative case metrics, such as specificity and negative predictive value, do not have meaning.

There are generally two means to aggregate class-level metrics: macro-averaging and micro-averaging. Macro-averaging consists of calculating the metric separately for each class then averaging the results. When macro-averaging is used, all classes are treated equally in the average, regardless of the relative number of samples assigned to each class. This can be useful when minority classes are of importance, or you are attempting to balance performance across all classes.

Micro-averaging consists of aggregating all samples before calculating the metric. Effectively, the confusion matrix is aggregated to TP, TN, FP, and FN counts prior to calculating metrics. When this averaging method is used, classes with more samples have a larger weight in the final calculation.

Since commission errors are omission errors when referenced relative to

another class, it turns out that micro-averaged precision, recall, and F1-score are all equivalent, Further, they are actually equivalent to overall accuracy; as a result, if class aggregated metrics are reported, we recommend reporting the macro-averaged versions alongside overall accuracy.

Summary of Classification Metrics

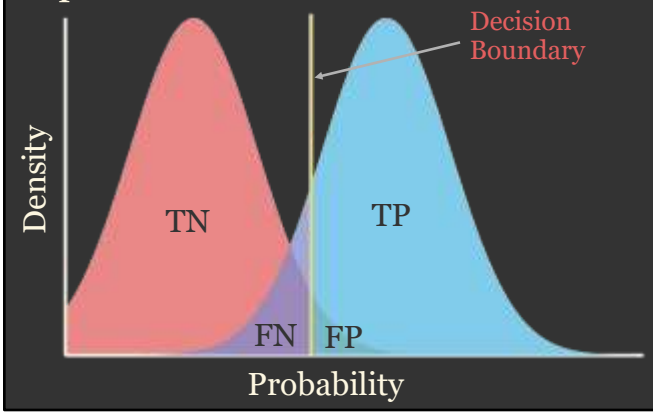
- Confusion matrix provides a detailed summary of the sources of error and confusion in the calculation
- 1 – Commission Error = User's Accuracy = Precision
- 1 – Omission Error = Producer's Accuracy = Recall
- F1-Score = harmonic mean of Precision and Recall = harmonic mean of User's Accuracy and Producer's Accuracy
- Micro-Averaged Class Aggregated Recall = Micro-Averaged Class Aggregated Precision = Micro-Averaged Class-Aggregated F1-Score = Overall Accuracy
- Suggest reporting confusion matrix, class-level metrics, macro-averaged class aggregated metrics, and overall accuracy

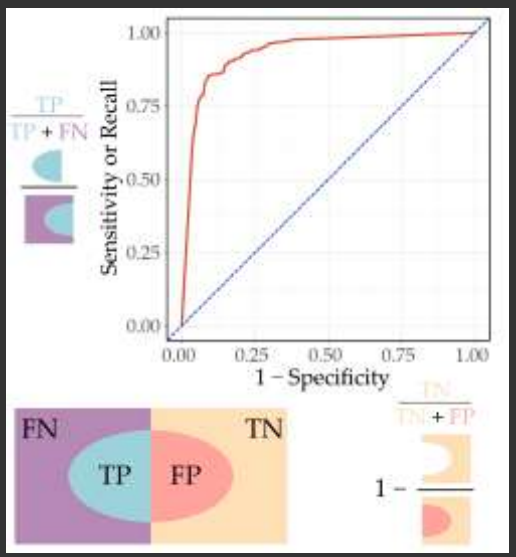
26

This slide summarizes the classification metrics and our reporting recommendations.

ROC Curves

- True Positive Rate = fraction of cases predicted as true that were true
- False Positive Rate = fraction of cases predicted as true that were false





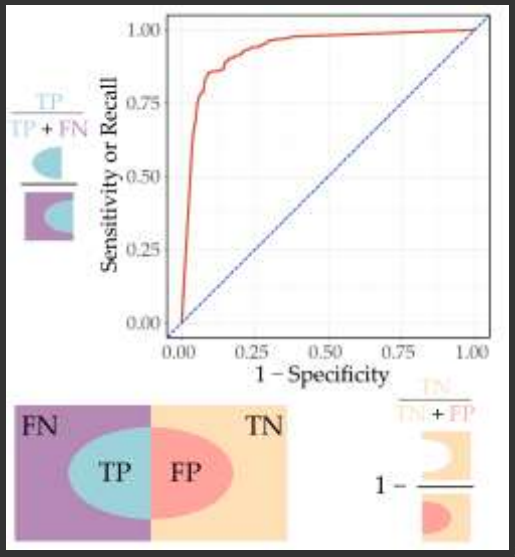
27

Probabilistic predictions are commonly assessed using Receiver Operating Characteristic curves, or ROC curves. These curves make use of the true positive and false positive rates. True positives represent cases predicted as true that were actually true whereas false positives are cases predicted as true that were actually false.

For example, if you were trying to predict which subjects had a certain disease, the true positive rate would be the fraction of the total number of subjects that were predicted as having the disease that actually have it. In contrast, the false positive rate would be the fraction of the total number of subjects that were predicted to have the disease but did not.

ROC Curves

- Receiver Operating Characteristic (ROC) Curve
- Plots False Positive Rate and True Positive Rate at all decision/probability thresholds
- Sensitivity = True Positive Rate (Recall)
- Specificity = 1 – False Positive Rate



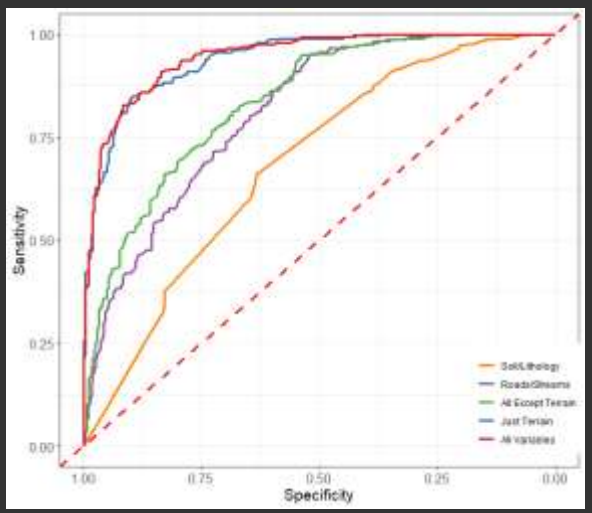
28

The ROC curve specifically plots sensitivity, which is equivalent to recall, against specificity. Sensitivity is the true positive rate (or recall) while specificity is 1 minus the false-positive rate. The rates depend on what probability threshold is used. So, instead of performing the assessment at one threshold, assessment is performed at all thresholds to plot a continuous curve.

ROC Curves

- Area Under Curve (AUC) = area under ROC Curve (0-1)
- Larger suggest better models
- Can be used to compare models

0.90-1 = excellent (A)
0.80-0.90 = good (B)
0.70-0.80 = fair (C)
0.60-0.70 = poor (D)
0.50-0.60 = fail (F)



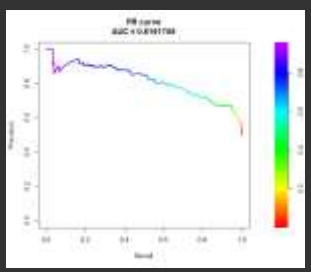
29

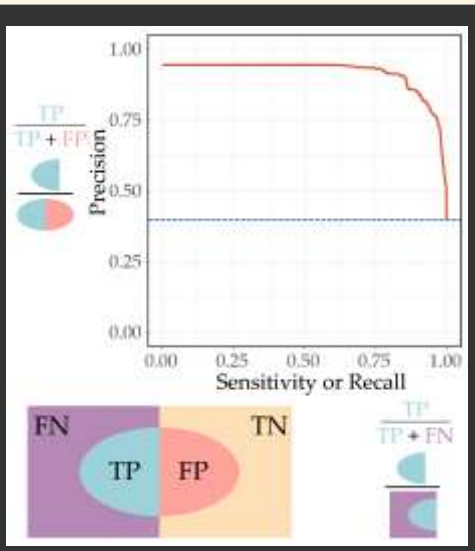
Using the ROC curve, the AUC, or area under the ROC curve, can be calculated. As the name implies, this is simply the area under the ROC curve. This measure is scaled from 0-1, where 1 is the entire area of the ROC graph. This is generally interpreted like a grade scale, as described on the slide. So, larger values are better. An AUC of 0.5 indicates that your model is not doing better than simply taking a random guess.

Since many algorithms generate estimated class probabilities as a means to obtain the hard classification, ROC curves and the AUC measure are sometimes used to assess classification models. This is especially true for binary classifications. However, a multiclass version of the ROC curve can be generated, and a multiclass AUC can be calculated.

P-R Curves

- Precision-Recall Curve
- Graph precision vs. recall at different decision/probability thresholds
- Can calculate area under the curve, similar to ROC



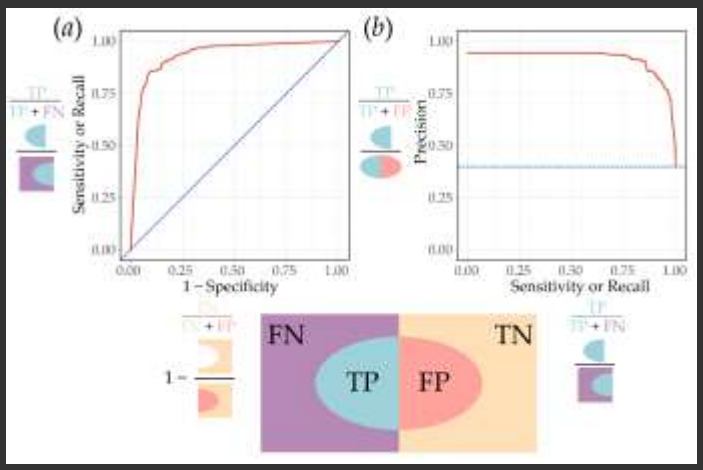


30

Assessments based on ROC curves do not take into account data imbalance, or the impact of a different number of samples in the two classes. The precision-recall (P-R) curve offers an alternative that takes this imbalance into account. Instead of using specificity, precision is graphed against recall or sensitivity. Similar to the ROC curve, the area under the P-R curve can be calculated as an assessment metric.

ROC Curve vs. P-R Curve

- Both are used for binary classification problems
- ROC Curve can be flawed when highly imbalanced data are used
- ROC Curve only considers omission error
- P-R Curve considers omission and commission error for positive case
- P-R Curve is better if data set contains moderate to large class imbalance



31

Since ROC curves and the associated AUC ROC metric rely on recall and specificity, which are both insensitive to data imbalance, they can be misleading in cases where data imbalance should be taken into account, such as when the mapped classes take up very different proportions on the landscape. Specifically, reported metrics can be overly optimistic in cases of severe class imbalance and/or when the class of interest makes up a small percentage of the landscape.

For example, if the goal is to map the locations of wetlands in a landscape where this class makes up less than 1% of the landscape, recall provides a quantification of the percentage of wetland samples that were correctly mapped as wetlands while specificity represents the percentage of not wetland samples that were correctly mapped. These two metrics and the associated ROC curve do not incorporate precision, which quantifies the percentage of the samples that were predicted to be wetlands that were actually wetlands. If a large percentage of the landscape is not wetlands, then the ROC curve and AUC ROC measure will not capture issues of FP cases, which may be an important criteria in determining the quality of the classification and the amount of manual labor necessary to improve the results (i.e., manually re-labelling the FP cases to not wetland).

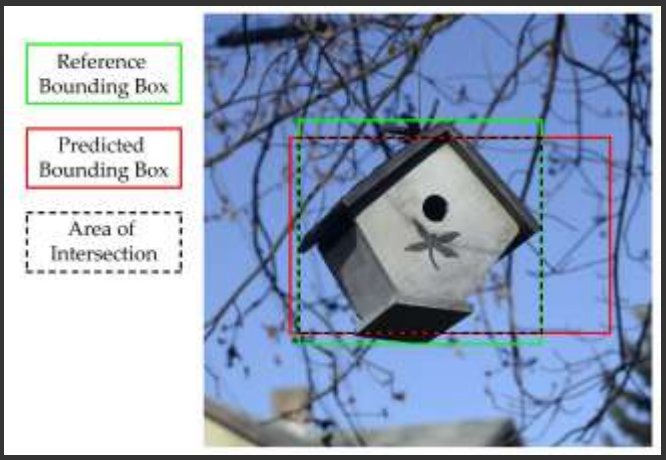
In such cases, a precision-recall (PR) curve may be more informative since it

does incorporate precision, which is sensitive to class imbalance and quantifies the percentage of samples predicted to the positive class that were TPs. This curve plots sensitivity or recall to the x-axis and precision to the y-axis. Similar to ROC, it is possible to generate an area under the curve (AUC PR) metric to obtain a single summary statistic.

Intersection over Union (IoU)

$$\frac{Area\ of\ Intersection}{Area\ of\ Union}$$

- Use bounding box or pixels
- Compare predicted extent or bounding box to reference extent or bounding box
- Used for object detection
- Used in computer vision
- Average for multiple class = mean IoU (mIoU)



32

Expanding upon binary metrics, the computer vision, feature recognition, and deep learning research communities often calculate additional measures by comparing each validation feature to its predicted extent in the image based on overlapping bounding boxes or feature masks.

This can be accomplished by comparing bounding boxes or classifications at the pixel-level.

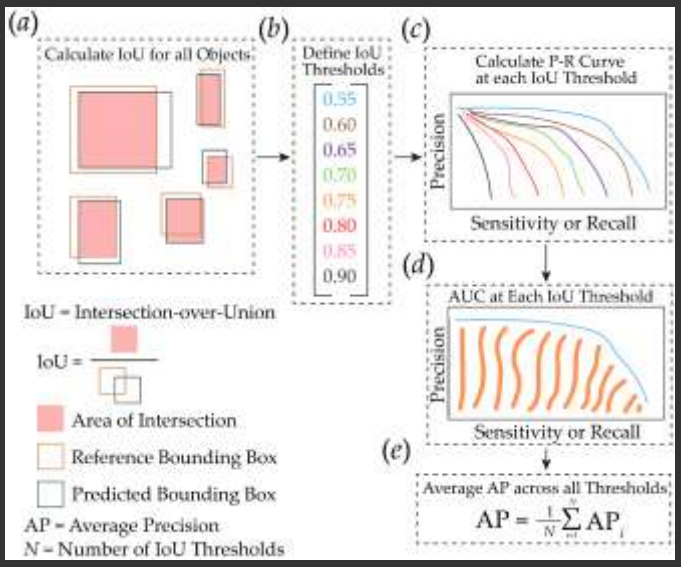
Intersection-over-union (IoU), also known as the Jaccard Index, represents the proportion or area that is shared of the combined area of the reference bounding box and predicted bounding box, or the area of intersection divided by the area of union of the two objects. A larger proportion of shared area or overlap suggests better performance.

In the provided example, the green box represents the correct or reference bounding box for the bird house object. The red rectangle represents a prediction, and the dashed, black line represents the area of overlap.

Mean IoU (mIoU) is the mean of IoU for all mapped classes. It is also possible to calculate a weighted mIOU that takes into account differences in class proportions or abundance.

Mean Average Precision (mAP)

- Define IoU thresholds
- Calculate PR Curves
- AP = area under curve averaged at different thresholds
- mAP = average AP for multiple classes



33

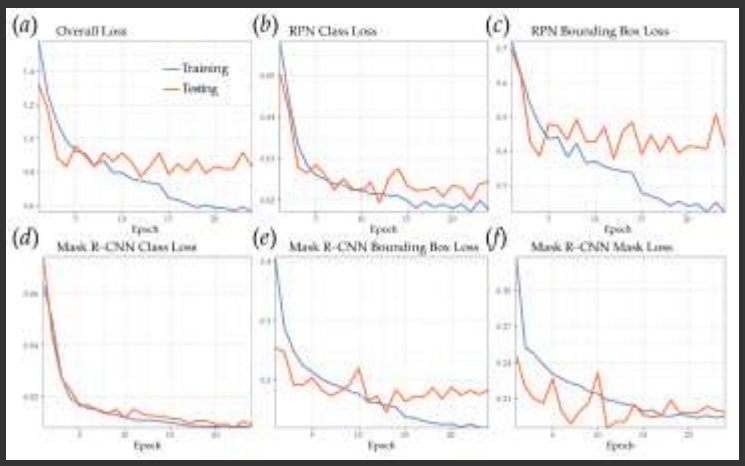
The figure on this slide conceptualizes the average precision (AP) or mean average precision (mAP) metrics. For each individual feature or detected object, correct or incorrect predictions can be defined based on a minimum IoU measure that is deemed acceptable to generate a classification and associated metrics. For example, features correctly classified and with an areal intersection of more than 60% with the associated reference feature could be labelled as TPs while features with less than 60% overlap, or those incorrectly classified, could be labelled as FNs. PR curves can then be generated based on multiple IoU thresholds by comparing the predicted probabilities of samples labelled as TP, FN, and FP. The area under each of these PR curves can then be calculated and averaged to obtain a combined average precision (AP). It is also possible to calculate AP for multiple classes and average them to a single AP value, which is commonly termed mean average precision (mAP); however, the distinction between AP and mAP is often confused in the literature.

Traditionally, IoU, mIoU, and AP, have been calculated using bounding boxes; however, it is possible to calculate them using pixel-level reference and predicted masks.

Loss Functions: Regression

Loss Functions

- Measure of error
- Used to assess learning during training
- Goal is to update weights/parameters to minimize the loss
- Use different measures for different situations
- Can created custom measures or use multiple measures



35

As the DL algorithm learns from the training data and the optimization algorithm updates the weights/parameters, it must monitor some metric of performance in order to measure whether the predictions of the training data are improving and learning is occurring.

This is accomplished by monitoring a loss metric. Loss metrics are measures of error as opposed to measures of accuracy, and the goal is to reduce or minimize this metric via the learning processes.

A variety of loss metrics have been proposed, and you can even define a custom metric for a specific problem.

The appropriate loss metric partially depends on whether classification or regression is being performed.

Mean Square Error (MSE)

- Sum of squared residuals divided by the number of samples
- RMSE without square root
- Perfect value is 0
- More weight is placed on samples with a larger relative prediction error

$$\Sigma(y - \hat{y})^2/n$$

36

Mean square error (MSE) is commonly used to evaluate regression problems. MSE as a loss metric is equivalent to MSE as an assessment metric.

Note that more weight is placed on samples with a larger residual or prediction errors since squaring a larger residual results in a larger relative increase in magnitude in comparison to squaring a smaller residual.

Mean Absolute Error (MAE)

- Sum of absolute value of residuals divided by the number of samples
- Perfect is 0
- Does not place increased weight on samples with a larger relative prediction error

$$\Sigma|y - \hat{y}|/n$$

37

Mean absolute error (MAE) is more robust to outliers than MSE and is an alternative loss metric for regression problems. It is the average of the absolute values of the residuals.

In contrast to MSE, increased weight is not placed on samples with larger residuals since the residuals are not squared.

Huber Loss

- Combine positive characteristics of MSE and MAE
- residual <= δ: $\frac{1}{2}(y_i - \hat{y}_i)^2$
- Residual > δ: $\delta|y_i - \hat{y}_i| - \frac{1}{2}\delta^2$

$$\text{Huber loss} = \frac{1}{n}\sum_{i=1}^{n} \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2 \ for |y_i - \hat{y}_i| \leq \delta, \\ \delta|y_i - \hat{y}_i| - \frac{1}{2}\delta^2 \ otherwise \end{cases}$$

38

The Huber loss attempts to combine the positive characteristics of the MSE and MAE losses.

When the absolute value of the residual for a sample is less than or equal to a user-defined threshold, defined with the delta parameter, the loss for that sample is calculated as the square of the residual divided by two. If the residual is larger than the user-defined delta, the loss is calculated as delta squared divide by two subtracted from delta multiplied by the absolute value of the residual.

This configuration reduces the disproportionate impact of samples with a larger residual magnitude. Below delta, larger residual values will have larger relative weightings, similar to MSE loss. Above delta, the loss behaves more similarly to MAE where all samples have equal magnitude or contribution in the loss calculation. The loss is also designed to have a smooth transition between the two means to calculate the loss to make the optimization process easier (i.e., the function is differentiable at all locations along the curve).

Mean Square Logarithmic Error (MSLE)

- Valuable if predicting large range of values
- Less punishment for large differences in large predicted values

$$\sqrt{\Sigma(\log(y + 1) - \log(\hat{y} + 1))^2/n}$$

39

The mean square logarithmic error can be used as a loss metric for regression, as opposed to MSE or MAE, and is particularly useful if the values of y, or the variable being predicted, vary greatly, such as over orders of magnitude.

It decreases the impact or weight of errors when predicting large values.

Log-cosh loss

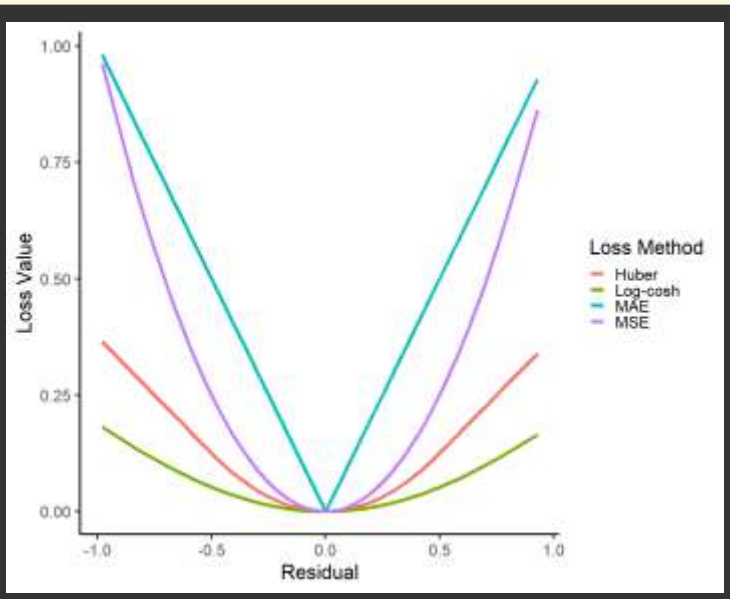
- Assigns a larger relative impact to observations with larger residuals
- Impact is reduced in comparison to MSE loss

$$\text{Log-cosh loss} = \frac{1}{n}\sum_{i=1}^{n} \log(\cosh(y_i - \hat{y}_i))$$

40

The log-cosh loss is a modified version of MSE loss. Similar to MSE loss, it assigns a larger relative impact to observations with larger residual magnitudes. However, this impact is reduced in comparison to MSE loss.

Regression Loss Comparisons



41

This graph provides a comparison of the MSE, MAE, Huber, and log-cosh losses.

There is not necessarily a preferred loss. The best loss is case specific, and all losses have pros and cons.

Loss Functions: Classification

Binary Cross Entropy (BCE)

- Log Loss
- Most common for binary classification problems
- One class coded as 1 (Positive), other coded as 0 (Negative/Background)
- Based on probability of samples occurring in each class

$$= -\frac{1}{n}\sum_{i=1}^{n}[y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)]$$

y_i = Truth value (0 or 1)
$\hat{y}_i$ = Softmax probability for positive class
n = Number of samples

43

Binary cross entropy (BCE) is commonly used as a loss metric for binary classification problems. It specifically makes use of the predicted class probabilities for the positive case. Again, there is some argument as to whether probability is the correct term here. I prefer to use the term rescaled logits.

When using binary cross entropy as a loss metric, the final layer should output one prediction, which represents the logit associated with the positive case. This logit can then be passed through a sigmoid activation function to obtain an estimated probability or rescaled logit. If the binary classification architecture generates two logits, one for the positive and one for the negative or background class, then you generally will not use binary cross entropy as a loss metric. Instead, you will use cross entropy (CE) loss, which will be discussed next.

In the equation, y_i is the class label (1 or 0) and p_i is the estimated probability of the sample belonging to Class 1 (i.e., the positive class). If the reference point belongs to Class 1, then the second component of the equation drops out (1-1 = 0). If it belongs to Class 0 (i.e., the negative or background class), then the first part of the equation drops out.

Since a negative is applied, this measure is minimized when features in Class 1 have a high predicted probability of being in Class 1 while features in Class 0

have a low predicted probability of being in Class 1.

To obtain a single metric from all of the predictions and associated labels, results are summed then divided by the number of samples.

Cross Entropy (CE)

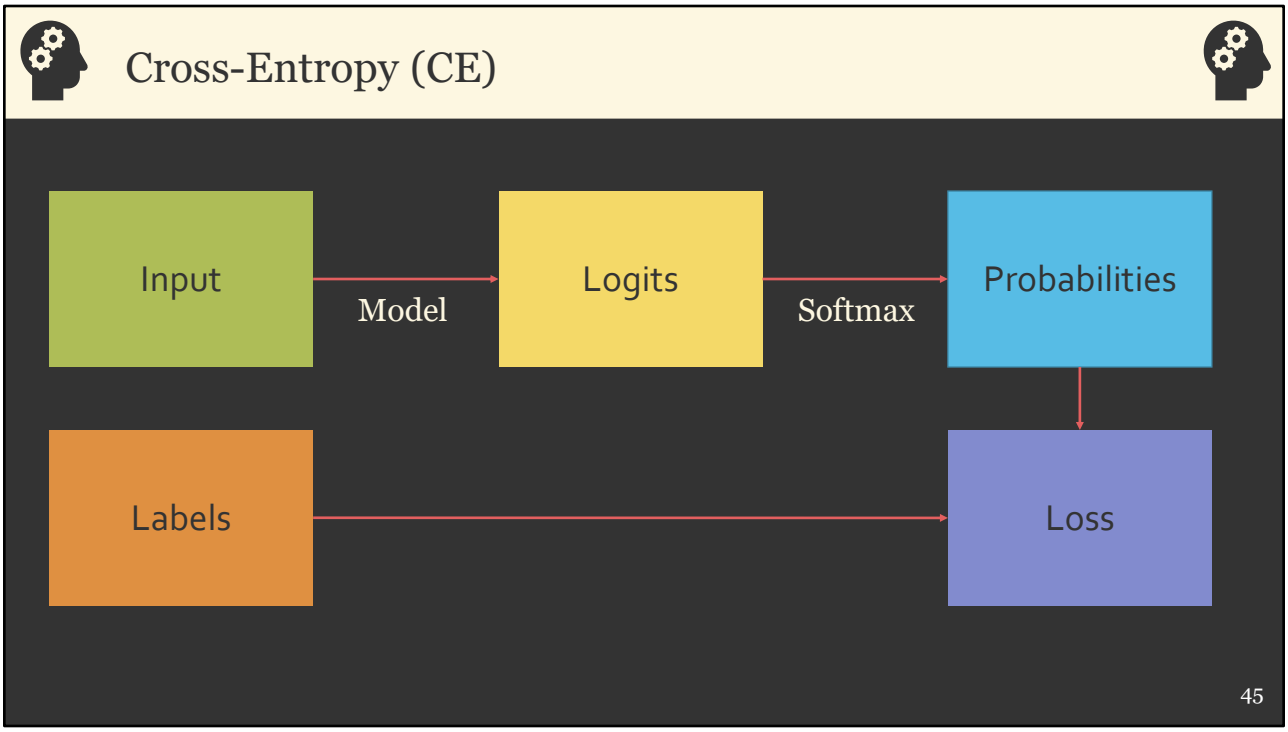
- Expands cross entropy for use with more than two classes
- Generally, default option for multiclass problems

$$= -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{C} y_{ij} \cdot \log\left(\widehat{y_{ij}}\right)$$

y_{ij} = current sample for current class
$\widehat{y_{ij}}$ = softmax probability for current class
n = number of samples
C = number of classes

44

Cross entropy (CE) loss is a multiclass version of binary cross entropy loss. The process of calculating this loss metric is discussed in the following slides.



In order to calculate cross entropy loss, the model must output logits for all differentiated classes. These logits can then be converted to estimated probabilities, or rescaled logits, that sum to 1 using a softmax activation. The loss metric requires the predicted class probabilities and associated reference labels as input.

Cross-Entropy (CE)

Label	*Prob*
0	.12
0	.05
0	.21
1	.45
0	.17

$$= -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{C} y_{ij} \cdot \log(\widehat{y_{ij}})$$

-(1)*log(0.45)

- Sum for each sample (i) for all samples (n)
- Sum for each class (j) for all classes (C)
- Divide by number of samples (n)

46

On this slide, I have provided an example output for a single sample. Note that a softmax activation has already been applied to rescale the raw logits. The estimated class probabilities, or rescaled logits, for the sample now sum to 1.

In this case, there are four classes being differentiated. The correct class is one-hot encoded. In other words, the correct class is assigned a value of 1 and all other classes are assigned a value of 0. For each sample, the log of the rescaled logit is calculated then multiplied by -1. Results are then summed for each sample (i) for all samples (N) for each class (j) for all classes (C). The sum is then divided by the number of samples to obtain a single loss metric from the available samples.

Weighted CE

- Augmentation CE loss that incorporates different weights for each class
- w_j = class weightings
- Weightings inversely proportional to class abundance
- Can be useful in cases of class imbalance

$$= -\frac{1}{\sum_{i=1}^{n} w_j} \sum_{i=1}^{n} \sum_{j=1}^{C} w_j \cdot y_{ij} \cdot \log\left(\widehat{y_{ij}}\right)$$

y_{ij} = current sample for current class
$\widehat{y_{ij}}$ = softmax probability for current class
n = number of samples
C = number of classes
w_j = class weightings

47

Cross entropy loss can be negatively impacted by class imbalance. Since the more abundant classes have more samples, they will have a larger impact on the aggregated loss calculation. This can cause the model to focus more on the abundant classes and less on the less abundant classes. This can result in poorer prediction accuracies for the less abundant class, overpredicting the abundance of the more abundant class, and/or underpredicting the abundance of the less abundant classes.

One means to potentially combat this issue is to apply class weightings. It is common to weight the relative influence of each class based on the inverse of its abundance in the dataset. This can result in a more balanced influence between the classes and, potentially, improved model performance, especially for rarer classes.

Focal Loss

- Focal loss allows for the prioritization of or focus on difficult-to-classify samples over multiple iterations over the training data
- Add γ (larger = more weight on difficult-to-classify samples)

$$= -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{C}\left[\left(1-\widehat{y_{ij}}\right)^{\gamma}\cdot y_{ij}\cdot\log\left(\widehat{y_{ij}}\right)\right]$$

y_{ij} = current sample for current class
$\widehat{y_{ij}}$ = softmax probability for current class
n = number of samples
C = number of classes
γ = focal weightings

48

Focal losses allow for the prioritization of difficult-to-classify samples. Over multiple iterations over the training data, the weight updates will be made such that the classification of difficult samples, or those with lower probabilities of membership to the correct class, are prioritized. There are focal versions of both cross entropy and Dice loss available. Dice will be discussed next.

Larger values of γ will result in putting more weight on difficult training samples. A γ of 0 equates to not implementing a focal adjustment (i.e., the result will be equivalent to CE loss).

Dice Loss

- Dice is equivalent to F1 Score

$$\frac{2 * \mathrm{TP}}{(TP + FP) + (TP + FN)}$$

- Dice can be augmented to a loss metric by:
  - Using estimated probabilities as opposed to counts
  - Subtracting from one to obtain a measure of error as opposed to accuracy

49

As noted above, the Dice coefficient is equivalent to the F1-score.

This measure can be modified to serve as a loss function by (1) using class probabilities in the calculation as opposed to counts and (2) subtracting the metric from 1. Probabilities are used to make the loss differentiable, which is required for gradient calculation and to perform backpropagation. Subtracting the metric from 1 is necessary to obtain a measure of error as opposed to accuracy. Remember that loss metrics must be minimized as the model's predictions improve.

Dice Loss

$$\text{Binary Dice Loss} = 1 - \left(\frac{(2 \times \Sigma \hat{p}_{TP}) + \varepsilon}{(2 \times \Sigma \hat{p}_{TP}) + \Sigma \hat{p}_{FN} + \Sigma \hat{p}_{FP} + \varepsilon}\right)$$

$$\text{Multiclass Macro-Averaged Dice Loss} = \frac{1}{C} \sum_{j=1}^{C} \left(1 - \left(\frac{(2 \times \Sigma \hat{p}_{TP}) + \varepsilon}{(2 \times \Sigma \hat{p}_{TP}) + \Sigma \hat{p}_{FN} + \Sigma \hat{p}_{FP} + \varepsilon}\right)\right)$$

$$\text{Weighted Macro-Averaged Dice Loss} = \frac{1}{\sum_{j=1}^{C} w_j} \sum_{j=1}^{C} w_j \left(1 - \left(\frac{(2 \times \Sigma \hat{p}_{TP}) + \varepsilon}{(2 \times \Sigma \hat{p}_{TP}) + \Sigma \hat{p}_{FN} + \Sigma \hat{p}_{FP} + \varepsilon}\right)\right)$$

$$\text{Log-cosh Dice Loss} = \log(\cosh(\text{Dice loss}))$$

$\hat{p}_{xx}$ = softmax probability for current class
C = number of classes
ε = smoothing factor
w_j = class weightings

50

This slide provides a few different versions of the Dice loss. The first equation is used for a binary classification and considers estimated probabilities for TP, FN, and FP samples relative to the positive case. This metric is appropriate when only predictions for the positive case are provided.

The second equation is for a multiclass macro-averaged Dice loss where all classes are given equal weight in the combined loss.

It is also possible to calculated a weighted multiclass macro-averaged Dice loss where the user specifies the relative weight of each class in the aggregated metric.

Lastly, the log-cosh version of the Dice loss has been suggested since the Dice loss can have complex gradients and can be difficult to optimize against. Using the log-cosh transformation can allow for smoother gradients and easier optimization. However, I have not seen this augmentation applied often in practice.

Tversky Loss

- Similar to Dice
- Provide different weightings for FNs (α) and FPs (β)
- Provide different weightings for recall and precision

$$\text{Binary Tversky loss} = 1 - \left(\frac{\Sigma \hat{p}_{TP} + \varepsilon}{\Sigma \hat{p}_{TP} + \alpha \Sigma \hat{p}_{FN} + \beta \Sigma \hat{p}_{FP} + \varepsilon}\right)$$

$$\text{Multiclass Tversky loss} = \frac{1}{C} \sum_{j=1}^{C} (1 - TI_j)$$

$\hat{p}_{xx}$ = softmax probability for current class
C = number of classes
ε = smoothing factor
α = coefficient for FNs
β = coefficient for FPs

51

The Tversky loss is an augmentation of the Dice loss and allows for applying different weightings to false negative (FN) and false positive (FP) errors relative to each class. In other words, different weightings can be applied to recall and precision. The key here is that the user can prioritize different types of error during model training: omission and commission errors do not need to be equally weighted.

The relative weightings of FN and FP outcomes are controlled by user-defined α and β terms. α is the weighting for FNs, and β is the weighting for FPs. It has been suggested that applying these weightings can be of particular importance when classes are heavily imbalanced. Specifically, for a binary classification where the positive case is not abundant, it may be useful to use a higher α relative to β to improve recall.

The slide provides Tversky loss equations for both binary and multiclass classification.

Focal Tversky Loss

- Provide different weightings for FNs (α) and FPs (β)
- Focal loss allows for the prioritization of or focus on difficult-to-classify samples over multiple iterations over the training data (γ)
- Applied per class as opposed to per sample

$$\text{Focal multiclass Tversky loss} = \frac{1}{C}\sum_{j=1}^{C}(1 - TI_j)^{1/\gamma}$$

C = number of classes
γ = focal weighting

52

Focal Tverksy loss adds a γ parameter, which can be used to control the relative impact of difficult-to-classify samples in the aggregated loss, similar to focal CE loss.

When γ is larger than 1, difficult samples take on a larger relative weight in the loss calculation.

Top *K* Loss

- Another way to focus on difficult-to-classify samples
- Only considers top *K* number of samples contributing to the loss
- Top *K* BCE/CE
- Top *K* Dice

53

Another means to weight difficult-to-classify samples is to only use the difficult-to-classify samples in the calculation of the loss. In other words, the *K* number of samples that contribute most to the loss will be used to guide the weight/parameter updates. All other samples will be ignored. Similar to focal loss, there are Top K versions of cross entropy and Dice loss.

Combining Metrics

- Loss = CE + *w*Dice
- Loss = Focal CE + *w*Dice
- Loss = Top K + *w*Dice
- Etc.

54

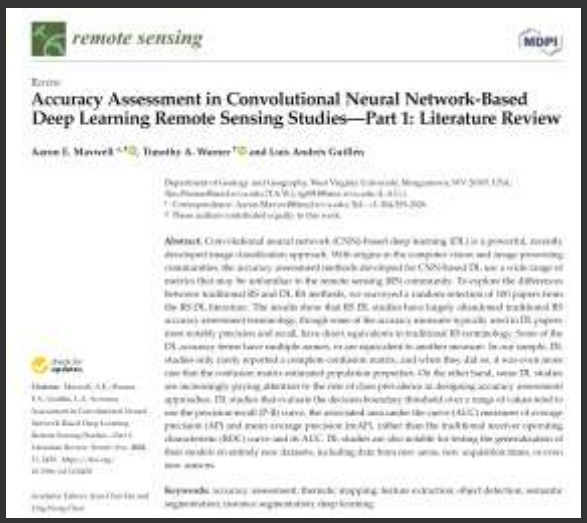
It is also possible to combine loss metrics into a single measure to be minimized during training.

For example, for binary classification problems, the total loss to be minimized could be the sum of the binary cross entropy and Dice losses.

It is also possible to apply weightings to each metric to adjust the relative influences.

This slide provides some examples of combined loss options.

Maxwell, Warner, and Guillen

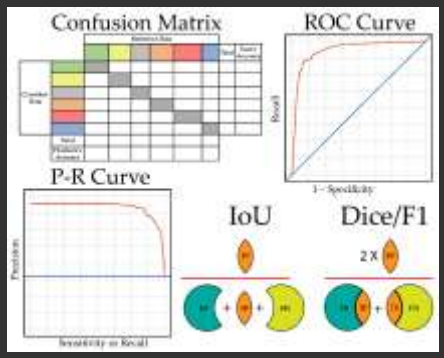

remote sensing

MDPI

Review

Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—Part 1: Literature Review

Aaron E. Maxwell *, Timothy A. Warner * and Luis Andrés Guillén



Maxwell, A.E., T.A. Warner, and L.A. Guillen, 2021. Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies – Part 1: Literature Review, *Remote Sensing*, 13(13): 1-27.

55

If you are interested in accuracy assessment methods for CNN-based DL in remote sensing, we recommend this two-part paper series. The citation for the second component is provided on the next slide. Both are freely available and open-source.

Maxwell, Warner, and Guillen

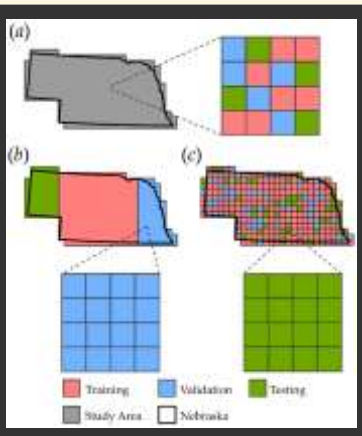

remote sensing

MDPI

Article

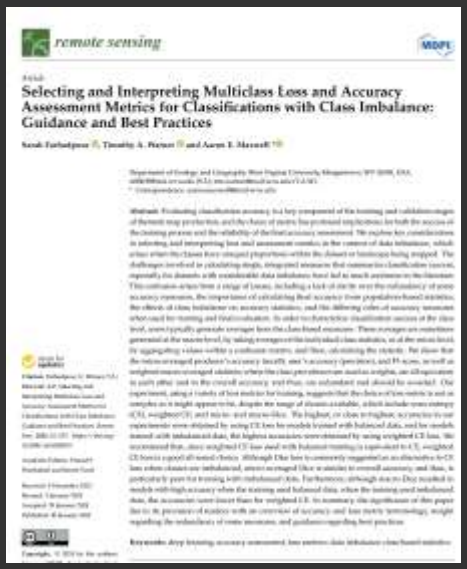
Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—Part 2: Recommendations and Best Practices

Aaron E. Maxwell *, Timothy A. Warner and Luis Andres Guillen



Maxwell, A.E., T.A. Warner, and L.A. Guillen, 2021. Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies – Part 2: Recommendations and Best Practices, *Remote Sensing*, 13(13): 1-22. https://doi.org/10.3390/rs13132591

Farhadpour et al., 2024


remote sensing

MDPI

Article

Selecting and Interpreting Multiclass Loss and Accuracy Assessment Metrics for Classifications with Class Imbalance: Guidance and Best Practices

Sarah Farhadpour, Timothy A. Warner and Aaron E. Maxwell

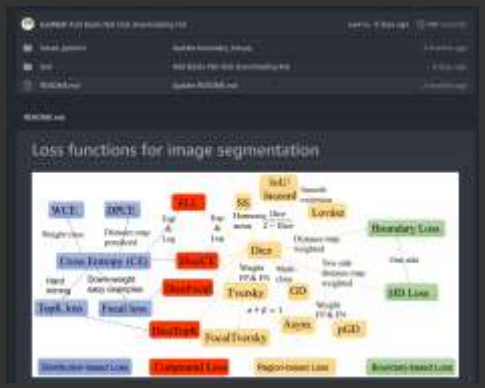
Farhadpour, S., T.A. Warner, and A.E. Maxwell, 2024. Selecting and interpreting Multiclass loss and accuracy assessment metrics for classifications with class imbalance: guidance and best practices, *Remote Sensing*, 16(3): 533. https://doi.org/10.3390/rs16030533.

57

This paper explores the impact of class imbalance on choosing an appropriate loss metric.

Ma et al. (2021)

https://github.com/JunMa11/SegLoss



Ma J, Chen J, Ng M, Huang R, Li Y, Li C, Yang X, Martel AL. Loss odyssey in medical image segmentation. *Med Image Anal.* 2021 Jul;71:102035. doi: 10.1016/j.media.2021.102035. Epub 2021 Mar 19. PMID: 33813286.

58

For more information on loss metrics associated with classification and image segmentation problems, we recommend Ma et al. (2021). Even though this paper focuses on medical image analysis, its findings are applicable to geospatial science.

This is the end of this lecture module.

Please return to the West Virginia View Webpage for additional content.

Thanks! Hope you found this useful.