PIXEL CLASSIFICATION USING MAXIMUM LIKELIHOOD ESTIMATION

1.0 Introduction

The report will explore Maximum Likelihood estimation as a method for classify pixels in an image as belonging to one of a set of classes. It also discusses Bayes decision rules and how they can be used to calculate the likelihood a pixel belongs to a class using a modified Gaussian distribution. Further experimentation is tried such as morphological operations to modify the classified pixels using expert knowledge, altering sample sizes and attributing weights to each class conditional pdf.

This implementation of Maximum Likelihood will use Python and OpenCV 4.0.

2.0 Methodology

2.1 Collecting data

Maximum likelihood estimation is a supervised training technique that requires labelled data. Fortunately, ground truth values have been provided as labelled data which makes sampling the data much easier. The author created a script which randomly selects X number of samples from the original image.

Each pixel (sample) is a D dimensional vector constructed of six channels, RGB (Red/Green/Blue) and three additional ones, Near-infrared, first echo and last echo (NIR/FE/LE).

The author found the best number of samples per class to be 30 due to having the highest average classification rate during testing (discussed in the results).

In a real-world scenario labelled data is hard to acquire, in which case, the programmer is to manually label the data. This would consist of inspecting a random selection of samples from the original image noting its features (R/G/B/NIR/FE/LE) and which class it belongs to.

Figure [1] – original image in RGB

2.2 Calculating the mean and covariance matrix

In order to establish a Maximum Likelihood model two import variable must be calculated, the mean of each dimension for each sample of classes creating a vector of means for each class. The other is the covariance matrix of each class, both will be explained in more detail with respect to Maximum Likelihood estimation further in the report. After collecting the samples, the author calculated both the means and covariances for each class.

2.3 Maximum Likelihood

Maximum Likelihood is the procedure of finding the value of one or more parameters for a given statistic which makes the known likelihood distribution a maximum.

In this implementation of Maximum Likelihood method it is possible to gather a small number of labelled training samples from an image and then classify the remaining pixels.

Therefore, its important to have a method for calculating the probability of an occurrence of a class given a pixel known as the posterior probability.

Bayes decision rule shows that the posterior probability can be calculated using the following rule:

$$\mathrm{P}(\omega_i \mid x) = \frac{P(\omega_i)\mathrm{P}(x \mid \omega_i)}{P(x)}$$

Prior probability:
P(x) is the probability of the D dimensional feature vector of the image that is being classified.
$P(\omega_i)$ is the probability of class ω_i
Class-conditional probability:
$P(x \mid \omega_i)$ is the probability of a feature vector x occurring given that it belongs to class ω_i
Posterior probability:
$P(\omega_i \mid x)$ is the probability of a class ω_i given that it is feature vector x.

The posterior probability can be estimated by assuming the priors ($p(\omega_i)$ and p(x)) are uniformly distributed as stated in central limit theorem. These priors are avoided by not making probability statements about the parameters, but only about their estimates, whose properties are fully defined by the observations and the statistical model. This also bypasses the difficulty of collecting/calculating prior probabilities. Due to the nature of random samples most data will fit a gaussian distribution given enough samples. Therefore, the class conditional probability density function (pdf) is expressed by a Gaussian function.

$$\mathrm{P}(\omega_i \mid x) \propto \mathrm{P}(x \mid \omega_i)$$

Therefore, it's possible to assume the posterior probability and class conditional pdf are proportional and can be estimated using just the class conditional pdf.

2.4 Gaussian Distribution

Gaussian's distribution is as follows:

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Given that x is normally distributed it's possible to calculate the mean and variance values for a set of samples and use the Gaussian distribution to calculate the posterior probability. To model the samples collected Gaussian's equation can be utilised by making a few small modifications to work in our domain.

$$p(x \mid \omega) = \frac{1}{2\pi^{\frac{n}{2}}\sqrt{|\Sigma_i|}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}$$

For example, changing the sample x into a vector x of size 1 x D (one for each dimension in the image) and representing the variance as a covariance matrix for each class ω_n which is of size D x D. This is because the determinant of the covariance matrix is equal to the variance of the samples collected. μ is now a vector of means which contains the means for each dimension and is of size 1 x D.

The exponent to Euler's number has been modified to support matrices. Due to there being no method of dividing by a matrix the equation must be multiplied by the inverse of the covariance matrix, this achieves the same result. The individual pixel variance (x - μ) can be squared by multiplying it by its transpose. However, because the order of operations is important with matrices the pixel variance is first multiplied by the inverse covariance matrix and then by it's transpose in order to reduce to a single number.

Because $\frac{1}{2\pi^{\frac{n}{2}}}$ is a constant it can be disregarded during the calculation of the class conditional pdf to save computational time.

2.5 Classifying the image

The class conditional pdf is calculated for each class on every pixel within the image. For every pixel the class with the highest pdf is selected as the most likely candidate.

Each pixel's most likely class is stored in a matrix with the same dimensions as the original image. To construct the final image each class is assigned a colour - Buildings (RED), Vegetation (GREEN), Cars (BLUE) and Ground (GREY). A new image is constructed mapping the classes to colours producing an RGB bitmap output.

3.0 Results

3.1 Classification vs ground truth image

Figure [2] – classification image vs ground truth

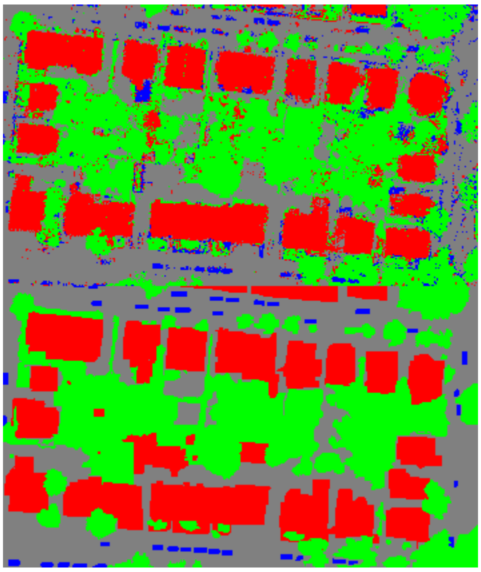


Figure [3] - Confusion matrix (sample size: 30)

	B	V	C	G
B	16234	1772	1039	2528
V	775	21025	644	1700
C	60	50	746	249
G	779	4293	1565	21657
User %	0.909	0.774	0.186	0.828
Producer %	0.752	0.870	0.675	0.765
Overall %	0.794			

The confusion matrix describes how classification errors are distributed between classes with their corresponding correct prediction percentage underneath. The trace of the confusion matrix represents the True Positive results with the others being False Positives.

The author found that the variance in prediction accuracy was high between random samples. Some samples heavily overclassify cars while others don't as much.

User and Producer accuracies

It is obvious that the model struggles to classify cars this can be seen with a very low user accuracy compared to the other classes. Buildings generally classify the best with some samples reaching 90% true positives. The author suspects this is due to the fact that buildings are very well defined within the LiDAR (FE/LE) features of the image.

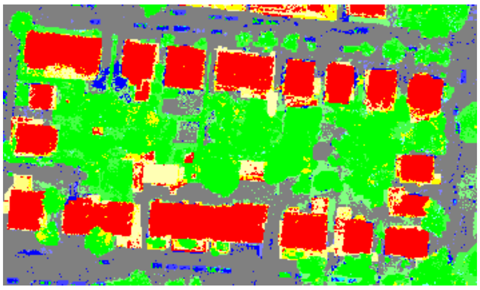
The producer accuracy of cars in much more in line with the other classes. This suggests that other classes are being miss-classified as cars and that generally when cars are (predicted) classified they are classified correctly.

Cohen's Kappa score

Cohen's Kappa score expresses the level of agreement between two annotators on a classification problem. A score of 0.699 suggests that the classification is much better than a random classification of pixels.

3.2 Visualising confusion matrix

Figure [4] – Confusion image



The Confusion image shows more clearly where in the image and what classes are being correctly and incorrectly classified. Twelve new colours represent the miss-classification of pixels (see key in appendix). Viewing the image like this distinguishes errors such as car-ground from ground-car miss-classifications.

From the Confusion image it is clear to see that many ground pixels are being classified as cars. This would explain the very poor prediction rate for cars.

Another observation made from the confusion image is that some of the smaller buildings and the edges of almost all of them are being miss classified as vegetation. This can be seen on the buildings surrounded by light yellow pixels.

4.0 Results of further research

The Maximum Likelihood is a blind technique that has no understanding of the classes or the problem itself. Therefore, it's possible to use Expert Knowledge about the domain to remove some obviously incorrect data using just a few assumptions.

4.1 Finding a suitable sample size

It's very important to ensure a balanced sample size for testing data to ensure it does not under or overfit the data. To find the optimal region the Maximum likelihood was calculated for ten different randomly selected samples, for each sample size increasing by ten each time. The average of each sample size is given below:

Figure [5] – average prediction rate given number of samples per class

Sample size	10	20	30	40	50
Prediction Avg of 10 samples	0.52	0.73	0.77	0.74	0.66

A sample size between 20 – 40 appears to be sufficient. The author investigated much larger samples sizes of 60 and 70 but they wildly over fitted the data and become unreliable predictions.

4.2 Removing dimensions from the source image

The author removed dimensions from the samples one by one and discovered that the LiDAR dimensions have the largest impact on the classification of a class. When removed accuracy drops to around 40%. The RGB dimensions appear to have the least effect on the classification.

4.3 Weighting the class conditional pdf

In an attempt to get better classification of cars the author decided to add unique weights between 0 and 1 for each class. This weight is used to multiply the corresponding class conditional pdf before selecting the best candidate. The reasoning behind this is based on the assumption that not all classes will appear equally within the image. For instance, cars make up a very small percentage of the image so are weighted with a value around 0.5. This helped remove much of the clustering of cars making it easier for morphological operations.

4.4 Morphological operation

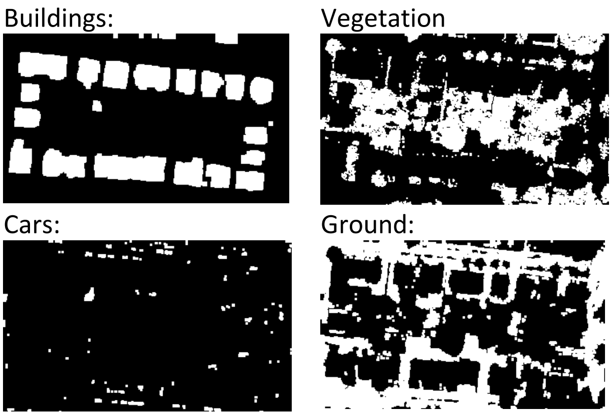
In this scenario the author can make the assumption that no cars can be one pixel in size. During classification if a feature vector is classified as a car but is not connected to any other car feature vectors then it can be replaced with the next most likely class. It's also reasonable to assume no cars will be on top of building so any blue pixels on top of buildings can be also replaced. These

To preform morphological operations on this image the first step is to separate each of the individual classes from one another by assigning any pixels classified as that class to 255 (white) and all other pixels to 0 (black). The four images produced are similar to a threshold image. The reasoning behind this idea is that cars seem to either form to small (one pixel big) or too large (large clusters bigger than 20 pixels). Morphological operates have the ability to reshape these cars while maintaining their position within the image.

4.4.1 Kernels

The threshold images are then morphologically operated on to close gaps, smooth edges and remove single pixels. This is done using the base operations of Erosion and Dilation which together can form Opening and Closing operations. The kernels used are a 2 x 2 and 3 x 3 matrix of 1's. Closing operation were useful for closing up small holes within the building, ground and vegetation layers. Erosion was used to make the large clusters of cars smaller making them a more appropriate size.

Figure [6] – Threshold images of classes

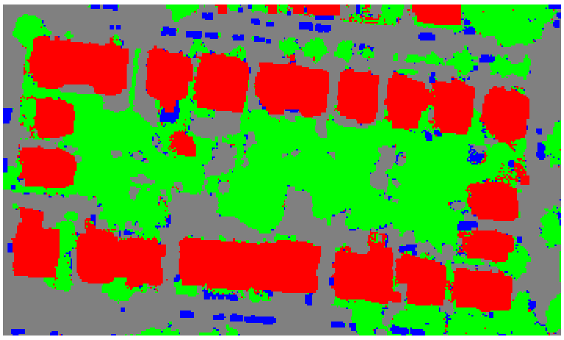


4.4.2 Modified predictions

The reconstruction of the image also uses expert knowledge. The image reconstruction is based on the assumption that the ground will always been on the bottom, cars and vegetation next with buildings almost always being in the foreground.

Each layer is then recombined in this order, ground, vegetation, cars, buildings. If pixels are missing values, then they are replaced with the most likely class. Once compiled it forms the final image.

Figure [7] – Final output image



The final recombined image has a classification percentage of 82.43% a 2.49% increase after morphological operations.

However, the best order of morphological operations seems to vary between samples. It appears to be hard to develop a consistent set of operations that works well for all classified samples. The author once again suspects this is mainly due to the inconsistency of classifying cars. Due to this uncertainty this method may be somewhat impractical.

5.0 Summary

5.1 Conclusion

This report introduced the author to Maximum Likelihood as a method for classifying images. The Maximum Likelihood produces a decent estimation given such a small sample and no method of learning. Accuracies generally ranged between 75% and 79% given a sample of 30.

Further improvement is made by utilising morphological operations. These operations remove single pixels, dilate the building edges which are classified as vegetation and close small holes made by miss-classification. With these modifications an 82% accuracy is achieved. However, the author feels that the small increase in accuracy is negligible and not worth additional effort. The morphological operations do clean up the image and make it far more understandable and presentable.

This implementation of Maximum Likelihood estimation is fast enough to be used in real time systems and achieves accurate results given how small the sample size and time to compute is.

5.2 Reflections

The author wonders if this technique could be used as a pre-processing step for a more advanced object detection system.

The author would also have liked to have found a simple way to remove pixel clusters larger than X size using morphological operations. This would have improved the accuracy by setting a minimum size for buildings or maximum size for cars.

6.0 Acknowledgement

[1] - Comparison of pixel-based and object-based classifications of high resolution satellite data in urban fringe areas – Noritoshi Kamagata

[2] – MLE Gaussian equation http://sar.kangwon.ac.kr/etc/rs_note/rsnote/cp11/cp11-7.htm

[3] - CS3VI18-CSMVI16 Visual Intelligence Dr. Hong Wei

Appendix

Confusion image colour matrix:

User error are seen in the light colours while producer error is seen in the darker colours.

	B	V	C	G
B	RED	YELLOW	MAGENTA	PINK
V	LIGHT YELLOW	GREEN	CYAN	LIGHT GREEN
C	LIGHT MAGENTA	LIGHT CYAN	BLUE	LAVENDER
G	LIGHT PINK	VERY-LIGHT GREEN	LIGHT LAVENDER	GREY

Code:

Source code found in folder