

Slide 1 – Title Slide

Hello and welcome to Week 11, part 5 of EGM310: Image classification. In this lesson, we will learn about different methods of automated classification – how we can use the computer to help us identify the objects in an image.

Slide 2 – Image Classification

As we have seen, remotely sensed images can be difficult to interpret – they're usually taken from angles that we don't normally see, and they often have scales that we aren't familiar with. Our goal with remote sensing is to derive information from the raw image data – we want to classify, or identify, the objects in an image. We might be interested in x identifying the burn scars from wildfires, mapping water bodies, or mapping different vegetation types. As you might guess, identifying each and every pixel by hand is an extremely time-consuming task. It can also be physically exhausting, so we normally want to avoid this as much as possible.

Slide 3 – Classification

The process of categorizing raw image data into information that can be used by non-specialists, or that can be used as input for further study, is what we refer to as classification. The output of a classification routine is a thematic map of the image – you can see here an example of an image on the left, and a thematic map identifying each of the different landcovers present in the image on the right. We're going to be talking about automated classification routines – how we can make the computer do a lot of the work for us.

Slide 4 – Characterizing Classification Algorithms

There are a number of different ways that we can characterize different algorithms. We've already seen one of these ways in the second practical – we can do a manual classification, whereby we manually identify every pixel in an image. Or, we can do a supervised classification as we did in the second practical: we identify some of the features in an image, and the computer uses that information to determine which of the identified classes each pixel belongs to. In an unsupervised classification, on the other hand, the computer automatically decided how to separate groups of pixels with no input from the user.

Classifications can be spatial, where we use spatial information to identify the objects in an image; or we can use the spectral characteristics of the image to identify objects. We can also first group pixels into objects, then determine how to classify them.

Parametric classification methods use an assumption about the statistical distribution of the input data to parameterize the input data and somehow interpolate the pixel values to do the classification. Non-parametric algorithms use some other way to classify the image based on the input data.

Physical models involve modelling physical processes – for example, modelling the interaction of electromagnetic radiation and a surface in order to help identify that surface in an image; empirical models use the properties of the image data to assign real-world properties, or categories.

We might also think about the fact that many pixels actually cover multiple different surface types. In that case, we assume that the spectral value of each pixel is some combination of defined pure materials, also called “endmembers” that provide a proportion of the pixel value. This is also what is known as a “mixed pixel.” We can then use this to estimate the fractional area of the pixel is covered by a given class type. Alternatively, we can assign the class based on the characteristics of the pixel, and ignore the potential problem posed by these mixed pixels.

Finally, “hard” classifications make a definitive definition about what class each pixel, or object, belongs to, while “fuzzy” classifications allow for some level of uncertainty and overlap by estimating the degree of similarity that a given pixel has to each class.

Note that these are not mutually exclusive categories. The classification that we did in Practical 2 is a supervised Maximum likelihood classification – meaning that it is both a supervised classification, but also a parametric classification, as the algorithm uses a parametrization of the training data to assign class values to each pixel in the image. In this lesson, we’re going to focus on some of the most common distinctions: supervised and unsupervised classification, spectral vs object-oriented, and parametric vs non-parametric classifications.

Slide 5 – Unsupervised Classification

As we have discussed in the practical, “unsupervised” classification means we have little to no user input in the classification routine. Instead, the algorithm determines how best to group pixels, based on their statistical properties. Common examples of different types of unsupervised classification routines include k-means clustering, or ISODATA clustering, which produced the image shown here. The output classes have no meaning – they’re just grouped based on the statistics of the image data. The user has to identify what each of the spectral classes represents – you have to give the classes meaning in order for them to be useful. Once you’ve run an unsupervised classification, you can also combine different classes – in the example shown here, you can see the at least 3 different classes present over the water, depending in part on the colors we can see in the true-color image.

Slide 6 – Supervised Classification

In supervised classification, as we’ve done in the second practical, we train the algorithm using pre-identified areas. These training areas should have uniform characteristics – the the larger the spread of pixel values that we include in a training class, the harder it is to have clean separation between our different classes. We also want them to be spatially-distributed throughout the scene as best as possible, to help counteract any potential differences in illumination across the image. The computer then uses the data contained within the training samples to determine which class each pixel belongs to, depending on the type of algorithm used. Some different examples of algorithm include maximum likelihood, as we used in the second practical; minimum distance, which calculates the geometric

“center” of each of the classes based on the training data. For each pixel, the “distance” to each class is calculated, and the pixel is assigned to the closest class. In a similar vein, k-nearest neighbors takes the k closest training pixels to each pixel, and assigns the pixel to the majority class. Finally, parallelepiped essentially draws a box around the training classes, and assigns all of the pixels that fall within each box to that class. For most of the software packages that do supervised classification, such as ArcMap or ERDAS Imagine, you can normally look in the help menu for more information about the different available algorithms.

Slide 7 – Maximum Likelihood

Next, we’ll show how one of these examples, maximum likelihood classification, works. In this example, we have three different training classes shown here, with the rest of the pixels in the image displayed as red x’s. Using the training data, the algorithm calculates a probability distribution function, or pdf, for each class. The pdf is a measure of how likely a given pixel value, or combination of pixel values, is to fall within the given class. So, a pixel located here would be more likely to fall within the dark green class – it’s pretty close to the peak of the probability distribution. A pixel here would be more likely to fall within the light green class for the same reason. Note that this approach usually assumes that your training data are normally distributed – that is, they follow a normal, or Gaussian, distribution. This is of course not always the case, but it’s another thing to keep in mind as we are selecting training samples.

Using this example, we can determine which class a pixel located at the location of this red square should belong to. You can see that it looks like it could belong to either the dark green or the light green class – you can see that the color level of the probability distribution is the same for each of these two classes. It’s slightly closer to the dark green training pixels. And, if we extend a line from the red square, the probability of a pixel at this location being part of the dark green class is just slightly higher. But it’s quite close – this might be an example where we need more training data to calculate a better probability distribution, or it might be better to employ a fuzzy classification scheme.

Slide 8 – Object-based image analysis

Object-based classification does a classification based on image objects, not individual pixels. This is analogous to how we see the world – we see the world as being made up of discrete objects, rather than pixels. The different steps for this are first, we segment the image, to estimate discrete objects. Quite often, we do this based on the pixels values, grouping nearby pixels together based on their values in different bands. You can see what this looks like here – the image objects are delineated by white lines, so we can see the islands in the river outlined, the river itself is also outlined, and we see different delineations at different vegetation boundaries, and so on. After this, we do a classification – this is where we take the objects and give them some meaning. With this approach, we can incorporate spectral properties – for example, the dark green objects are a certain kind of tree, the dark black objects are water, and so on. But, we can also take into account the shape of the objects – for example, we could differentiate between rivers and lakes by taking into account that a river will most likely be long and thin relative to its area, while a lake might normally be more compact. We can also use texture

– for example, we can see the trees in this image have a very rough texture, while the river, even though it has a similar dark color, is relatively smooth. Area is another category we can use to classify image objects – larger buildings might be industrial or commercial, while smaller buildings are more likely to be residential. We can also include context – we wouldn't expect a white sand beach to be surrounded by agriculture, for example – we can use the relationship between our different classes to help identify objects. Finally, we can use other layers, or other datasets, such as elevation, to help us classify objects – we don't just have to rely on the image data.

Slide 9 – Summary

In this lesson, we have discussed how our goal in remote sensing is often to extract meaningful information from raw image data.

We want to avoid classifying images by hand – it's normally extremely time-consuming, and it can be difficult to reproduce.

We discussed how automated classification schemes can be categorized in a number of different ways that are not mutually exclusive – a given algorithm can actually fit into several of these different categories.

The choice of algorithm we use is going to depend on the input data we have, and the particular application we have in mind.

Slide 10 – Additional resources

Once again, you can read more about the concepts we've covered in this lesson in the textbooks, Chapter 7 of Lillesand, Kiefer & Chipman; and Chapter 12 of Campbell & Wynne. I've included a link to a video that discusses k-means and image segmentation, as well as an article on image classification in ArcMap. I've also uploaded a copy of an article that provides an overview of different classification algorithms – it's a long one, but you can read it to find out more about a number of these different types of classification. That's all for this lesson – I hope you found it interesting, and if you have any questions, please don't hesitate to e-mail me or post in the discussion forum on blackboard. Bye!