Slide 1 – Title Slide

Hello and welcome to Week 2, Part 3 of EGM703: Hyperspectral Image Analysis 2, 2 Hype 2 Image. In this lesson, we'll look at a few more methods that we can use to analyze hyperspectral images, including a few that we've seen before.

Slide 2 – Band Indices

We can create normalized difference indices or perform band arithmetic with hyperspectral images, just like we do with multispectral images – the main difference is we have a much, much larger number of band combinations that we can try to exploit. We've seen the normalized difference vegetation index, or NDVI, before – this is just the difference between the reflectance in the near infrared minus the reflectance in the red, divided by the sum of these reflectances. With hyperspectral data, we normally have multiple bands in each of these different wavelength ranges. This figure shows how well the NDVI correlates to wet biomass as a function of band combinations for two crops: soybean and crop. From this, we can see that the optimal choice of band combination differs between the two crops – for cotton, the highest correlation value comes with a NIR wavelength of around 920 nm and a red wavelength of 680 nm, while for soybeans the ideal range is between 800-900 nm in the NIR and about 710 nm for the red wavelength.

Slide 3 – Recall: Decision trees

Continuing the theme of "methods that we've seen before," I'm going to use the next few slides to refresh your memory on different machine learning techniques, starting with decision trees. Recall that a decision tree is similar to a flow chart: we have input data, and a series of steps to follow to determine what to classify that input data as. We have nodes in our decision tree, where we test different attributes. For example, here we have a node where we're looking at the value in band 1, which might be, for example, the red band, and we're classifying, or dividing, our data into two different groups based on whether the pixel values are bright or dark in the red band. After that, we move on to our next set of nodes, in order to continue the process until we get to our final classification layer, where each pixel value has been assigned to a class. The branches of our decision tree are the particular test outcomes that we have. For the band 1 example here, one branch is where the pixel value is "bright" (above 82), and one is where the value is "dark" (less than or equal to 82). Leaf nodes are the final nodes, where we have our class labels – here, we have wetlands, dead vegetation, bare soil, and water. Another way to think about this is that we are recursively partitioning our dataset into more and more homogenous subsets of that data – we're starting with objects or pixel values as an input, and returning classes as the output.

Slide 4 – Recall: Machine Learning

When we talked about machine learning, we mentioned that there are 3 main approaches to machine learning. First, we have unsupervised machine learning – this is where we provide the machine with no

information about the input data, and essentially ask it to sort, or categorize, or find the structure that is present in the data. In supervised machine learning, we provide the machine with labeled input data that it then uses to figure out what the function, or pattern, is that determines how to classify that data. Then, when we provide the machine with new, unlabelled, data, it uses that pattern it has "learned" to label the new data. Finally, we have reinforcement learning, where depending on the outcomes or actions that the machine learning algorithm takes, we reward it or punish it based on whether the action it takes helps it to achieve the goal that we have put in place. In remote sensing, the most prevalent approach is supervised machine learning, which is what most of our examples will fall under.

Slide 5 – Recall: Artificial Neural Networks (ANNs)

An Artificial Neural Network, or ANN, is designed to mimic how a human or animal brain works. It's a network of different connected nodes that can "communicate" with each other, or transmit signals to each other. Each of the different nodes is known as a neuron – again, this is all based on how we understand human brains to actually function, where we have different neurons that are communicating with each other in order to process all of the different things that our brains process. Each of these connections has a weight that is adjusted as the network learns – we can have more communication between particular neurons, we can have less communication – these are the things that are tweaked as we train up our neural network. The neuron is something that processes a particular input signal, or value, that then outputs a value to the next neuron in the network using a non-linear function that is known as an activation or transfer function. This is just a way of determining whether the neuron will transmit a signal ("activate") to the next neuron, or not. Once, or if, the neuron transmits the signal to other neurons, they can move on to the next layers in the network. Eventually, from our initial inputs, we have outputs that are, at least in our case, thematic map classes – at the end of this, we have a classified map. Each of these neurons are aggregated into different layers, which are also sometimes called hidden layers – we can have any number of different layers that build up our network.

Slide 6 – ANN classification

This figure comes from a 2012 study by Zhang and Xie that combined hyperspectral data, object-based properties such as texture, and different image classification algorithms, including spectral angle mapping, support vector machine, and an artificial neural network. In this study, the authors used AVIRIS data in the Everglades, Florida, USA, to try to classify different vegetation types. They tested whether the minimum noise fraction helped to improve the classification results, finding that it did – the classifications using the reduced dataset with the top 15 MNF layers had a kappa value of over 0.84, compared to a value of 0.52 with the full dataset. The ANN showed better performance than the other classification methods tested – the highest kappa value for the other methods was the support vector machine, at 0.86; with the ANN, kappa values ranged between 0.89 and 0.94, depending on the size of the training data used. Including object-based texture properties was extremely useful, with an accuracy of 100% for mangrove swamps, an important landcover type. The authors also found high accuracy for identification of invasive plant species, again an important consideration for critical wetland habitat such as the Everglades. This study combines a number of different approaches we've introduced before: object-based analysis, machine-learning methods, and hyperspectral analyses; a

2011 study by Licciardi and Del Frate combined another approach we've introduced, spectral unmixing, with artificial neural networks – as always, you check check this paper out in the module library.

Slide 7 – Recall: Support Vector Machine (SVM)

Another type of machine learning algorithm that we've introduced previously is the support vector machine, or SVM. The aim behind using a support vector machine is to find the location of a decision boundary in order to separate different classes. So, if we're given two different classes, here represented by the red and yellow dots and denoted by class a and b, respectively, then our goal is to find the line, or hyperplane that leaves the greatest margin, or space, between the two classes. The margin is the distance between the hyperplane and the closest points in each of the different classes, and these closest points are known as support vectors – hence the name. If our classes are not linearly separable – that is, there's no simple way to differentiate between them, maybe we have some overlap between the two classes, then the goal is to find the hyperplane that maximizes the margin, while also minimzing the misclassification. Rather than using a single line here, we might have a piecewise hyperplane separating our classes. Originally, SVM was developed for binary classification problems, but it can easily be adapted to multi-class applications.

Slide 8 – SVM classification

Like other machine learning methods, Support Vector Machines are an ideal method for analyzing hyperspectral data. They are highly generalizable, meaning that they are flexible and adapt to many applications; the cost function for SVM is convex, which means that we can identify an optimal solution – this is something that we want, because it means we can actually solve the problem. SVM are also well-suited to ill-posed problems, which most hyperspectral classification applications are. A "well-posed" problem is one where: a solution exists; the solution is unique; and the solution is stable, meaning that changing the initial conditions will change the solution in a "nice" way. Ill-posed problems, on the other hand, violate at least one of these conditions: the solution may not exist; there may be many solutions; or the solutions may be unstable. In addition to all of these nice reasons, SVM also don't take much effort for us to set up, relative to other methods. In this study by Dalponte and others from 2009, they tested three different classification methods for classifying trees in a hyperspectral image: SVM, gaussian maximum likelihood with a leave-one-out-covariance estimator, and linear discriminant analysis. They also varied the spectral resolution of the dataset by averaging the values in adjacent bands – starting with the full-resolution dataset at 4.6 nm, averaging every two bands to get 9.2, every 3 to get 13.8, and so on. From the graphs that you can see here, SVM outperformed the other two methods in both study areas, and especially with higher spectral-resolution datasets. The study area on the left, Bosco della Fontana, is a much more complicated forest for classification, as it is a natural forest reserve with 19 different species to classify, while the Val di Sella study area has only 6 tree species – you can see how the SVM classifier does better in both cases, but especially so in the more complicated application.

Slide 9 – Summary

In this lesson, we've seen how many of the approaches that we've introduced in other modules, such as machine learning or spectral indices, can still be applied to hyperspectral data.

In particular, machine learning algorithms can be especially useful, as they are often suited to applications where we have large amounts of information.

As always, though, the choice of algorithm we use is going to depend on the data we have and the application we are interested in – there aren't really any "one size fits all" algorithms.

Slide 10 – Additional resources

You can read more about the topics we've discussed here in the textbooks – Jensen, Chapters 10 and 11. For an additional refresher on some of the topics we've covered here, I've linked to the EGM702 lessons on Image Segmentation and Object-Based Image Analysis, Machine Learning, and Machine Learning classification – if you don't still have access to the slides for those lessons, let me know and I will make them available. I've also included links to the different papers referenced in this lesson, and of course there are more examples available in the Zotero Group Library for the module. 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!