Remote Sensing Lab
Spring Semester 2022

Lab 2 Week 2 (L2W2)
Bayes Theorem and SVM for Classification

Rodrigo Daudt
rcayedaudt@ethz.ch
HIL D43.3

Introduction

- Based on the example regions that were labeled in L2W1, our objective is to classify every pixel in the image into one of the considered classes.

- For this, it is useful to think of pixels not as colors, but as either a point in a high dimensional space or an object with several feature values.

- We will be exploring two pixel-wise classifiers: Bayes rule classifier and support vector machine (SVM).

Bayes Rule Classifier

- The per-class probability density of an object (pixel) is given by:

$$f_{\mathbf{x}}(x_1, \ldots, x_k) = \frac{1}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right),$$

Where **x** is a vector containing all features (containing all **k** channels) of a given pixel.

μ is the mean vector of the class (containing one value for each of the **k** channels)

k is the number of channels (k=10 in your case)

|Σ| is the determinant of Σ

Σ is the covariance matrix

Covariance Matrix

- The covariance matrix has to be computed for each class.

$$\Sigma = (\mathbf{x}_{tr} - \mu_{\mathbf{x}})^{\mathrm{T}}(\mathbf{x}_{tr} - \mu_{\mathbf{x}})/\mathrm{N}$$

Where $\mathbf{x}_{\mathbf{tr}}$ are k-dimensional **training** pixels for one class (number of pixels *times* k = **N**)

k is the number of channels of the image

N is the number of pixels in $\mathbf{x}_{\mathbf{tr}}$

$\boldsymbol{\mu}_{\mathbf{x}}$ is the mean vector of $\mathbf{x}_{\mathbf{tr}}$ (the training pixels)

Uncertainty Map

- After normalizing the likelihoods to obtain the per-pixel class probabilities, we can calculate the classification uncertainty at each pixel to obtain an uncertainty map.

$$\text{ClassificationUncertainty} = 1 - \frac{\text{max} - \frac{\text{sum}}{\text{n}}}{1 - \frac{1}{\text{n}}}$$

where

max	=	the maximum set membership value for that pixel
sum	=	the sum of the set membership values for that pixel
n	=	the number of classes (signatures) considered

Hardening the Results

- Taking the index of the highest probability density value per pixel is in fact the **Maximum Likelihood** result, which consists of finding the highest a-posteriori likelihood to classify each pixel in the image.

Your Task

Implement the **Maximum Likelihood** classifier

- You need to copy the labels.tif file that you have generated in L2W1.

- You must:
  - Calculate the mean vector and covariance matrix for each class
  - Evaluate the probability density function (pdf) to calculate the likelihood of each pixel of belonging to each class
  - Normalize the likelihoods to obtain class probabilities (the probabilities in each pixel must sum to 1)

- You must visualize the **probability maps** of each class and the **uncertainty map**

- Harden the Bayes result

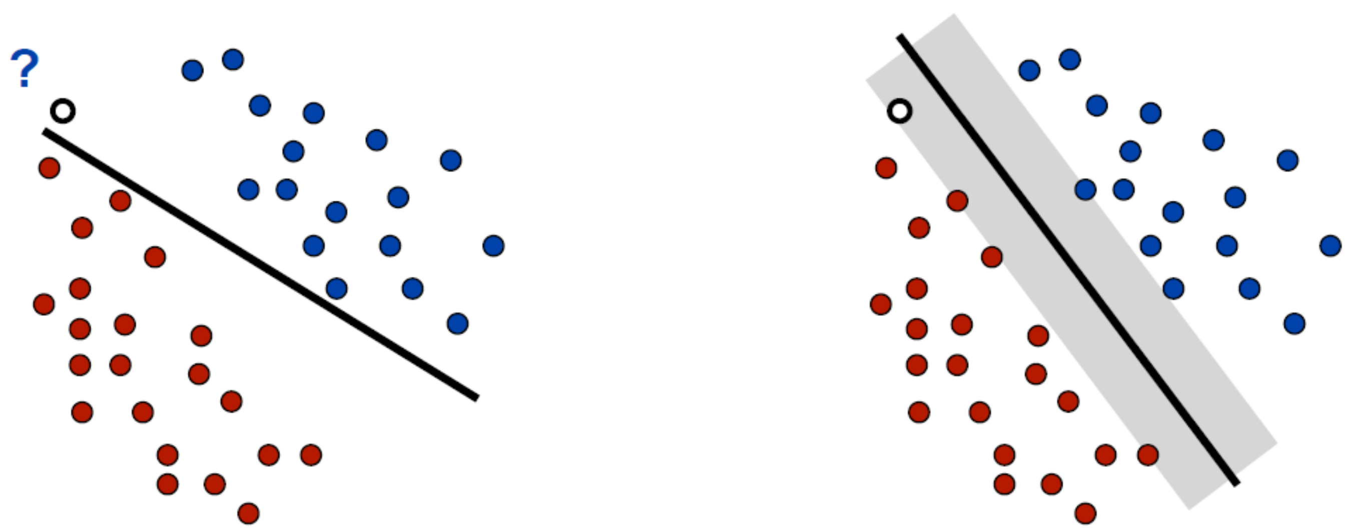
Support Vector Machines

- SVMs look for decision boundaries between classes.
- **Principle**: Training points should not only be separated, but should be far away from the decision boundary.
- A *hyperplane* separates the space into two half spaces when we have two target classes.

See Image interpretation slides for more information.

SVM Theory

- Points that are considered in finding a hyperplane are those nearest to the hyperplane.
- The “best” hyperplane would be equidistant to the bordering pixels on each side.
- Large distance between the classes -> the decision rule is expected to perform well with unseen data.



Your Task

- Train an SVM classifier.
- Use sklearn.svm.LinearSVC to create an SVM object.
- You can use the *fit()* method to train the SVM on a dataset, and *predict()* to classify unseen data.
- It may be useful to look at the documentation for LinearSVC to understand how the inputs should be organised:
https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

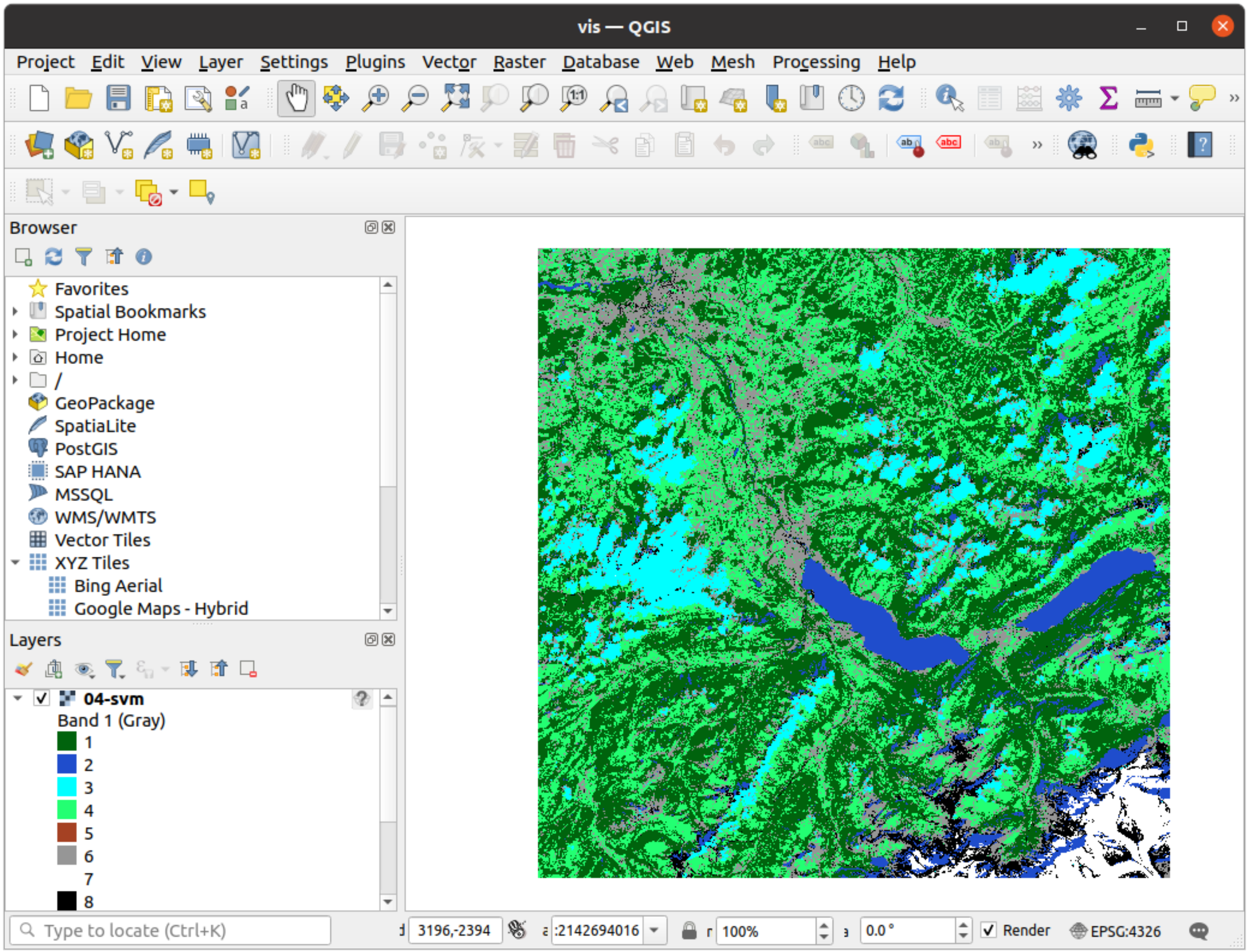
Visualizing the Results

- For interpreting the results, you can use QGIS to open:
    - The (automatically generated) RGB composite
    - The Bayes classifier predictions
    - The Bayes classifier uncertainty map
    - The SVM predictions

- You can follow the same process as in L2W1 to colorize the classification layers. You can change the colors individually to highlight classes one at a time.

- Tip: you can copy and paste a layer's style by right clicking it, then going to the "Styles" menu.

Visualizing the Results



Deliverables

- Upload your code and all the generated images (except for the RGB composite) on Moodle.
- Answer the quiz on Moodle.
- Deadline for submission and quiz: May 3rd 15:45

For Next Week

- Next we will study deep learning and convolutional neural networks.

- Please make sure you install the following libraries:
  - PyTorch
  - torchvision

- Installation instructions: https://pytorch.org/