SORBONNE
UNIVERSITÉ

RDFIA

Pattern Recognition
for Image Analysis and Interpretation

Report Section 1:
SIFT, Bag Of Words, and SVM

Students:	*Student ID:*
Balkis Bouthaina DIRAHOUI	21113733
Lina MEZDOUR	21215990

October 2022

Contents

1 Summary

In this document, we aim to apply a Machine Learning algorithm on an image classification problem. However, as several pre-processing steps are needed in order to train any model, we are going to present each of these steps before displaying the classification results.

- ***SIFT (Scale-invariant Feature Transform) representation*** : In an image, some regions are more important than others (high contrast areas, corners...), these regions are called *Key Points.* It is important to detect and describe them in an efficient way in order to gain global understanding of the image. The goal of the SIFT algorithm is to capture these key points locally and encode them into a feature vector. The particularity of this algorithm (or descriptor) is its invariance to brightness and rotation. This means that if we take an image, and change its rotation or brightness, the local features vectors extracted will be the same.

- ***Visual dictionary*** : In this step, the goal is to capture patterns in the SIFTs, taken from a big dataset of images, and group them together to get a limited number of representations (that we call later: visual words). To do so, we create a visual dictionary with the help of the K-means Clustering algorithm. The dictionary contains the center of the clusters, which are the 'mean" SIFT representation of the SIFTs vectors belonging to the cluster.

- ***Bag Of Words (BOW)***: Now that we have the visual words ("mean" descriptors) and the SIFT representations, the goal is to create a global image representation. The image will be represented as a BoW (Bag of Words). A BoW is a sparse matrix. Its columns represent the SIFT descriptors (the visual words) and the rows represent the images, to draw an analogy with the NLP vocabulary, the words would be the SIFT descriptors and the documents would be the images. Therefore, each image is represented by a vector summarizing the frequencies of each visual word, present in the image.

- ***Classification***: Finally, the BoW representation can be used in solving different Machine Learning problems. In this document, we will do an image classification task with the help of the SVM algorithm (a supervised Machine Learning algorithm). The input is our BoW representation of the image and the output is whether the image belongs or not to a certain class. The problem here is a multi-class problem which leads us to choose the one-vs.-all strategy[1]. Moreover, the SVM algorithm has some key hyper-parameters like C or *kernel* that we will analyse further through a series of experiments.

[1] Meaning that we learn one SVM for each possible outcome

2 SIFT

2.1 Question 1

The two kernels are separable into $M_x = h_y.h_x$ and $M_y = h_x.h_y$, With:

- $h_x = 1/2 \times (-1, 0, 1)$
- $h_y = 1/2 \times (1, 2, 1)$

2.2 Question 2

It is useful because we **optimize** the time complexity by doing so. We go from a 3×3 matrix to 2 vectors of size 3 each.

- In **the first scenario** (using the 3x3 kernel), computing the convolution needs 8 multiplication operations and 8 additions to be performed for each pixel in the image of a dimension $n \times m$.

  Therefore, the complexity of computing the convolution using this matrix is $9mn$ multiplications and $8mn$ additions.

- However, in **the second scenario** (using the separable vectors): each vector needs 3 multiplications and 2 additions for each pixel (because the convolution keeps the dimensions of the input image). So, overall: Only $6mn$ multiplications and $4mn$ additions are needed.

Thus, as the kernel is separable into two vectors, using them will help the convolution operations run faster at least $9mn/6mn = 1.5$ times. (the acceleration in addition terms is $8mn/4mn = 2$ times faster).

2.3 Question 3

The Gaussian Mask is used to be centralized on the most **important** regions (the closest to the patch center). In fact, this mask's purpose is to give less importance to the gradients **farther** from the focus points in the center of the patch.

2.4 Question 4

The role of discretization is to make the algorithm more **robust** to **rotation**. In fact, if we take the same image, and rotate it, our SIFT descriptor would be invariant. It can also make it easier to build the histograms and thus, compare different images.

2.5 Question 5

The purpose of the first step, where we set to a null vector descriptors that have an L_2 norm below 0.5 is used to not consider descriptors that do not have enough interesting information or that are **low interest** points (as they don't have enough contrast). Furthermore, these points would have disturbed the clustering process as there are many points, and they should all belong to the same class. So, we remove them and add them directly to the dictionary to give more chance for the other points to be clustered correctly within the number of clusters given (1000).

Then, we normalize and set a threshold of 0.2 to make SIFT **invariant** to any brightness changes.

2.6 Question 6

SIFT is a good descriptor for image analysis, because it's simply **practical**. It is **robust** to change in rotation, light and other geometric transformations (zoom for instance). In fact, SIFT is invariant to brightness changes, as gradients are invariant to light intensity shift. It's also invariant to rotation, as histograms do not contain any geometric information and that can allow us to compare two images taken in **different contexts**.

2.7 Question 7

The patch represented in 2.1 is a part of the image treated by SIFT, we can interpret the results as follows;

- We have 8 orientations, as illustrated in the orientations image.
- Every patch is split into 4×4 regions. We can see that regions like **1,4,13** or **16** in the image are completely white, so their gradient, as we can see in the images, is **null**. The same would have happened if we had fully black regions.
- The rest is in between the black and white regions (borders) and therefore, they have a **non-null** gradient (**e.g, 15,2**), we can see their signals as it is represented in the spikes of the histogram.

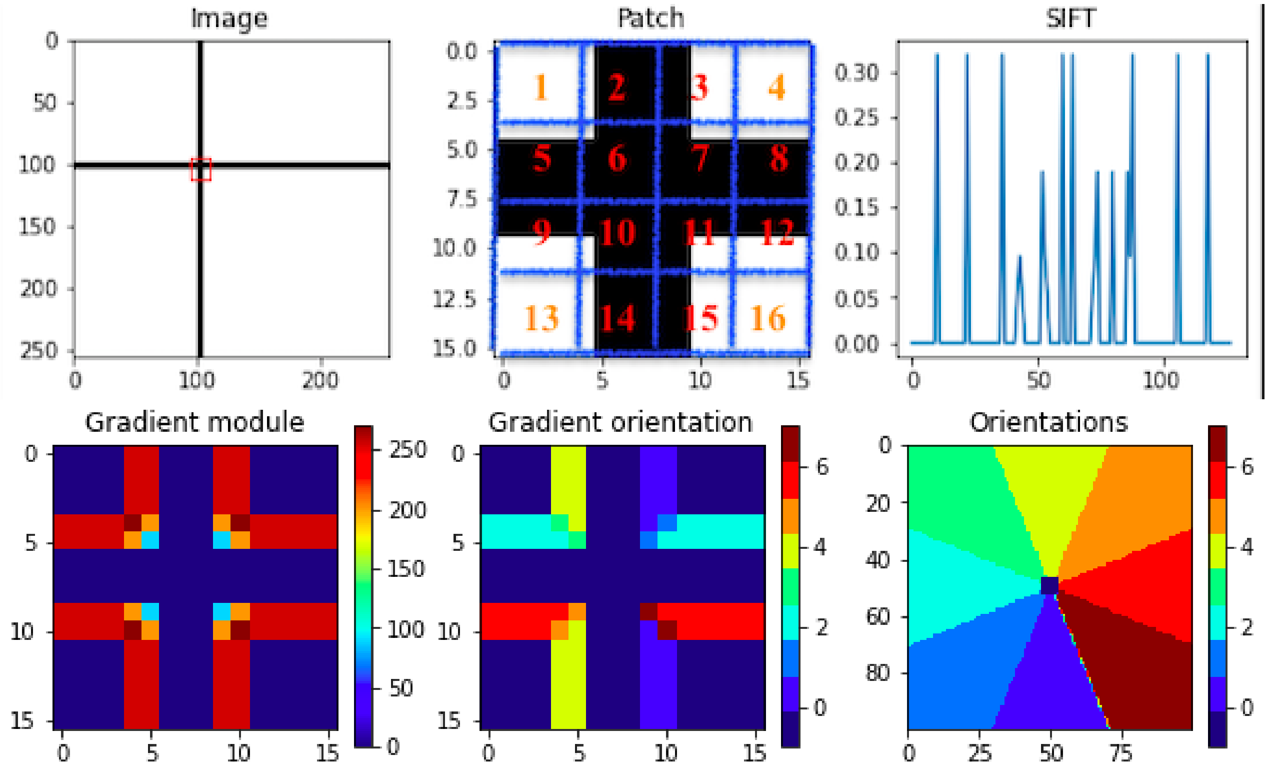


Figure 2.1: SIFT result example

3 Visual Dictionary

3.1 Question 8

In our classification task, the visual dictionary is needed in order to provide a more **compact** representation of our images. Instead of manipulating many SIFT vectors (whose number varies with the size of image), we quantize them in fixed well-defined feature space.

Furthermore, we can group similar features (patches with slight variations) together. Which reduces the **complexity** of the learning process, as we reduce noise, and as every image can be defined by the same set of features (visual words). This makes the training easier, as the model can access the different aspects of the input image without difficulty, and **compare** them easily.

3.2 Question 9

By definition, the center of the cluster is the point that is closest to all points of the cluster. This implies that the points inside a cluster are closer to each other, as they were pulled to the same cluster's center, and the points further away are pulled by another center. Therefore, the dispersion is small.

Furthermore, by definition, the **variance** in statistics is equal to the **mean of squares** of **the distances** between the **points** and the **center**. This is,(multipled by number of points N), is what we want the cluster's center to minimize. So, the center of the cluster that minimizes the dispersion is well the barycenter of the points of the cluster.

3.3 Question 10

Choosing the optimal number of clusters depends on the complexity of the recognition task, and the size of the data we have (to construct the visual dictionary). We do not want it to be too small, and thus the visual words won't provide an accurate representation of all patches, or too big, to prevent overfitting. Knowing this, we can empirically try multiple numbers and compare the results. If not possible, some techniques, like the elbow method and the Agglomerative Hierarchical clustering procedures, can help. Some other more sophisticated techniques exist, using feature selection and Optimum-Path Forest clustering .

3.4 Question 11

We use SIFTs instead of raw pixels to build the visual dictionary, because SIFTs (and local features) are better descriptors and more **informative** of the image than raw pixels. For instance, SIFTs are orientation-invariant, scale-invariant, and robust to small perturbations and transformations. This means that they produce similar outputs to similar patches, which is not possible for raw pixels, as they state statically the color of the pixel. Knowing this, the visual dictionary is much more general using SIFTs, as it inherits the properties of the SIFTs themselves.

3.5 Question 12

The regions that belong to the same cluster are alike globally (3.1). For instance, they have the same texture or the same pattern (lines, squares...). That's because their SIFT descriptors are **similar**, and closer. Thus, the k-means grouped them in one cluster.

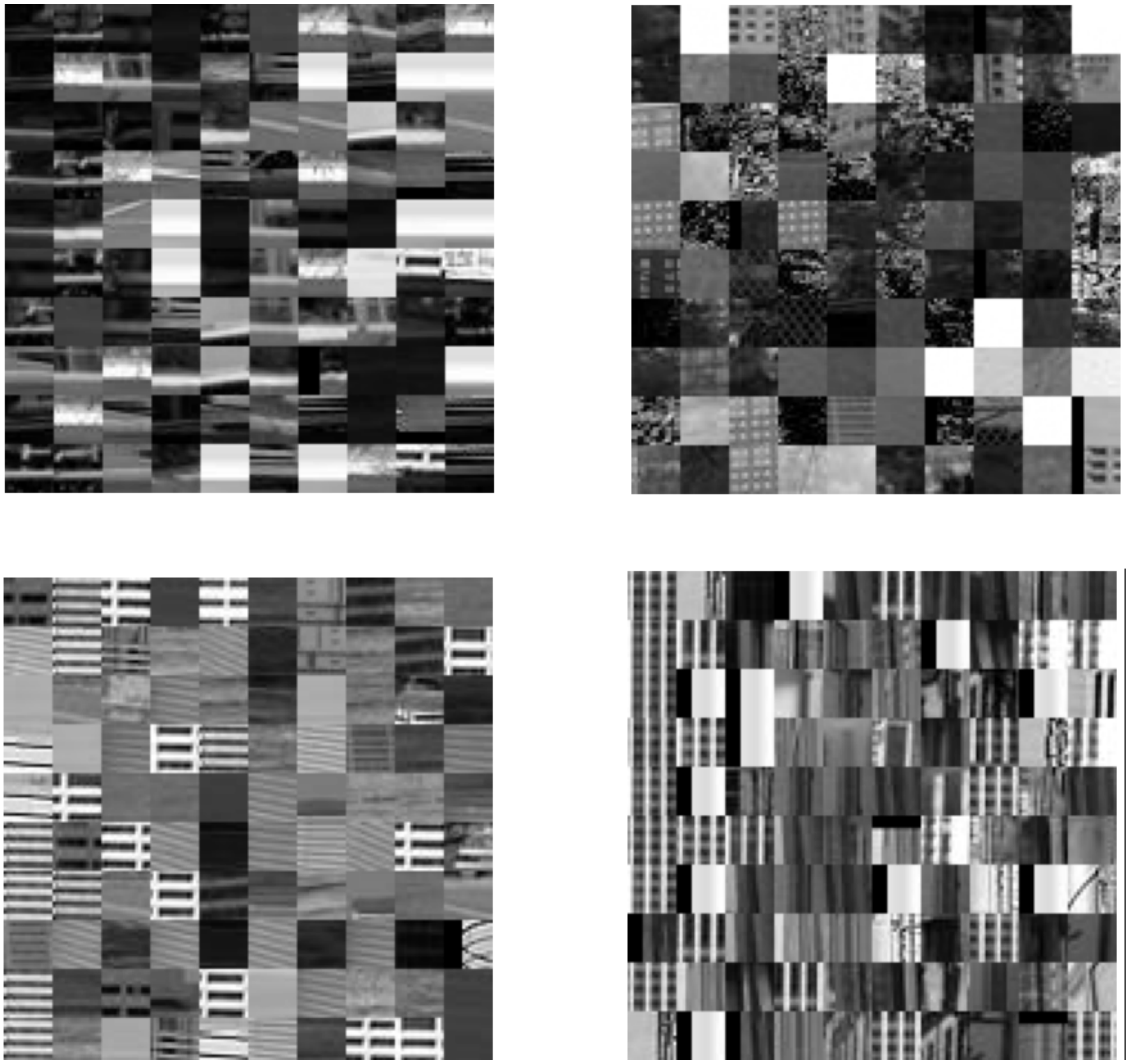
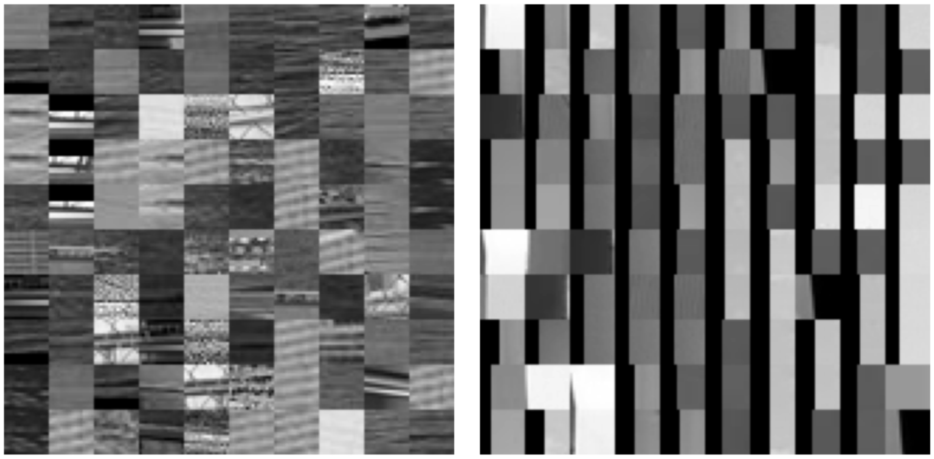


Figure 3.1: Examples of 4 clusters content

We can also see that, in one cluster, regions can vary in the intensity of **luminosity** and **darkness** (3.2a). However, they share the same orientation. And that's because, as SIFT works with gradients, the values of the pixels themselves are not as important as their **variation**. Some clusters contain regions/patches that look different from the others (3.2b), and they are the furthest generally from the cluster center. That's because we are using a heuristic clustering algorithm and the number of bins (clusters) was defined manually. Furthermore, a patch can be a mixture of multiple patches. Thus, putting it in the closest cluster does not imply it looks like it 100%.



(a) Best patches

(b) Worst patches

Figure 3.2: Examples of the best and worst patches of a cluster

4 Bag Of Words

4.1 Question 13

z represents a summary of the local features of the image, constructing its global representation. Concretely, it is the frequency (number of appearance) of each visual word of the dictionary in the image, independently of their location. It also a quantization of the SIFT features in a fixed space defined by the visual dictionary.

4.2 Question 14

The results can be analyzed as follows;

— For the histogram (4.1a), it presents the frequency of the visual words detected in the example image. We observe some spikes on some of the 1001 visual words: some words have a bigger frequency than others, like the fifth word with 34% frequency. Other words are not present in the image (the blanks). This is because the visual dictionary is aimed for features belonging over 30 scenes. This being just one of them, it may not contain features found on a pet image or a sea.

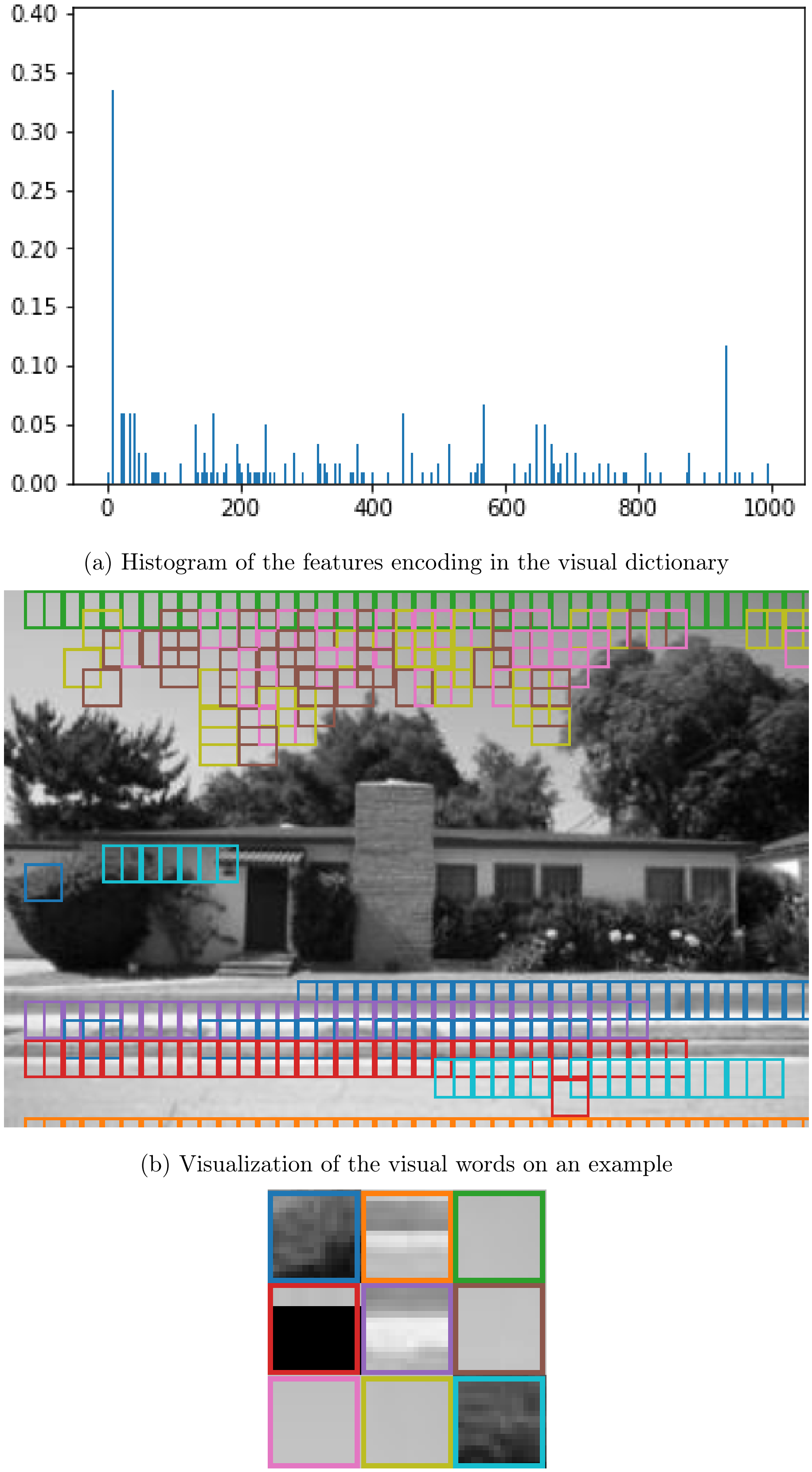
— As for the visualization of the BoW on the image (4.1b), it points out some occurrences of the 9 most common visual words (4.1c). We observe that they are concentrated in the uniform patches (where the same pattern is repeated multiple times), like the sky and the sidewalk. The top and lower edges of the photo represent important features, according to our descriptors, and follow the same pattern.
The yellow-, pink-, green- and brown-boxed visual words seem the same to the naked eye., and they all define the sky. However, the SIFTs and the visual dictionary consider them different regions. This may be in link with the number of words we have chosen for the dictionary: It is causing the same points to be divided into different clusters (overfiting). However, this may also be because the patches are, even though they look the same humanly, but they are not the same texture-like, and when changing lightning or other aspects, they would appear different.

4.3 Question 15

The interest of the nearest-neighbors encoding technique is the hard constraint that it provides (Hard Assignement). Every feature is represented by the single visual word, that is closest to it. This ensures that the feature and the visual word are similar, and have a constant representation. Furthermore, This is simple, **fast to compute**, and **invariant to spacial placement** of the feature. However, this may cause information loss.

Other encoding methods are:

- ***the Soft Assignment (SA) coding method:*** proposed to reduce information loss by assigning a local feature to different visual words according to its memberships to multiple visual words.



(a) Histogram of the features encoding in the visual dictionary

(b) Visualization of the visual words on an example

(c) Top 9 visual word

Figure 4.1: Example of the application of BoW technique using the visual dictionary and SIFTs

- ***Sparse Coding based SPM (ScSPM), Local Coordinate Coding (LCC) and Locality-constrained Linear Coding (LLC):*** aims at obtaining a nonlinear feature representation which works better with linear classifiers. They search for some weighted coefficients to linearly combine visual words of the dictionary to approximate the input low-level descriptor.
- ***Salient Coding (SaC) and its extension Group Salient Coding (GSC):*** is proposed to speed-up while reserving the classification accuracy.

4.4 Question 16

Sum pooling is interesting and is used because it gives more importance to the visual words that appear the most. Thus, we can give more importance to some clusters (features) than others.

One alternative to sum pooling is *max pooling*, which consists of only selecting the cluster (feature) that appears the most in our image.

Another alternative is the *Tf-Idf* representation, that works by giving more weight to less frequent visual words.

4.5 Question 17

We use the $L2$ normalization in order to compare the representations of different sized images (only the direction matters, as normalizing allows us to consider these vectors the same way regardless of the number of features). An alternative norm would be the $L1$ norm, but that would lead to the loss of this property.

5 Classification with *SVM*

5.1 Question 18

We trained several SVM models, while varying the hyper-parameters. In what follows, we are going to present the results while varying the value of C (soft vs hard SVM) and the kernel by the SVM, as shown in the figure 5.1. It is to mention that,in this experiment, we are using the default multi-classification strategy in *sklearn* for SVM, which is One-vs-Rest.

- ***C:*** We can see that, choosing high values for C (C>200), helps with the training accuracy. As seen in the first 3 figures on the right. We reach 100% of the accuracy in the training set because it does not have a lot of data, which means that the classification task is easier.
- ***Kernel:***
  - For the **sigmoid kernel**, the model is underfiting tremendously with no more than 20% of the accuracy. This kernel was incapable of capturing the prediction fuction. Which justifies the bad performances in the validation set. This kernel has been discarded.
  - We can see that the **linear kernel** provides the most stable and high values on both the training and the validation set, with accuracy of 74% on the validation set. The model may have overfited on both sets, as their sizes are small, but with accuracy of 73% on the test set, we can confirm its performances.
  - For the **poly kernel**, the training accuracy never reachs 100%, however is the value of C. This struggling in the training shows in the validation, as the accuracy is maximized by 62%.
  - Finally, the **RBF** would be the kernel to compete with the linear model, with 72% accuracy on the validation set. It provides, also, constant accuracy per values of C.

5.2 Question 19

The effect of each hyperparameter:

- ***C:*** The hyperparameter C adds a penalty to the classifier for each misclassified point, the penalty varies with the value of C.

  If C is small, we say that we have a *soft margin* and the penalty for the misclassified points is small. A decision boundary with a large margin is then chosen at the expense of more misclassifications. Furthermore, a *soft margin* leads in general to better generalization.

  If on the other hand C is big, we say that we have a *hard margin.* Meaning, we do not tolerate misclassifications (outliers) and we have a smaller margin. But, this can lead to divergence.
- ***Kernel:*** The kernel function is a kind of similarity measure. We give it as an input the original features. The output is a similarity measure in the new feature space.

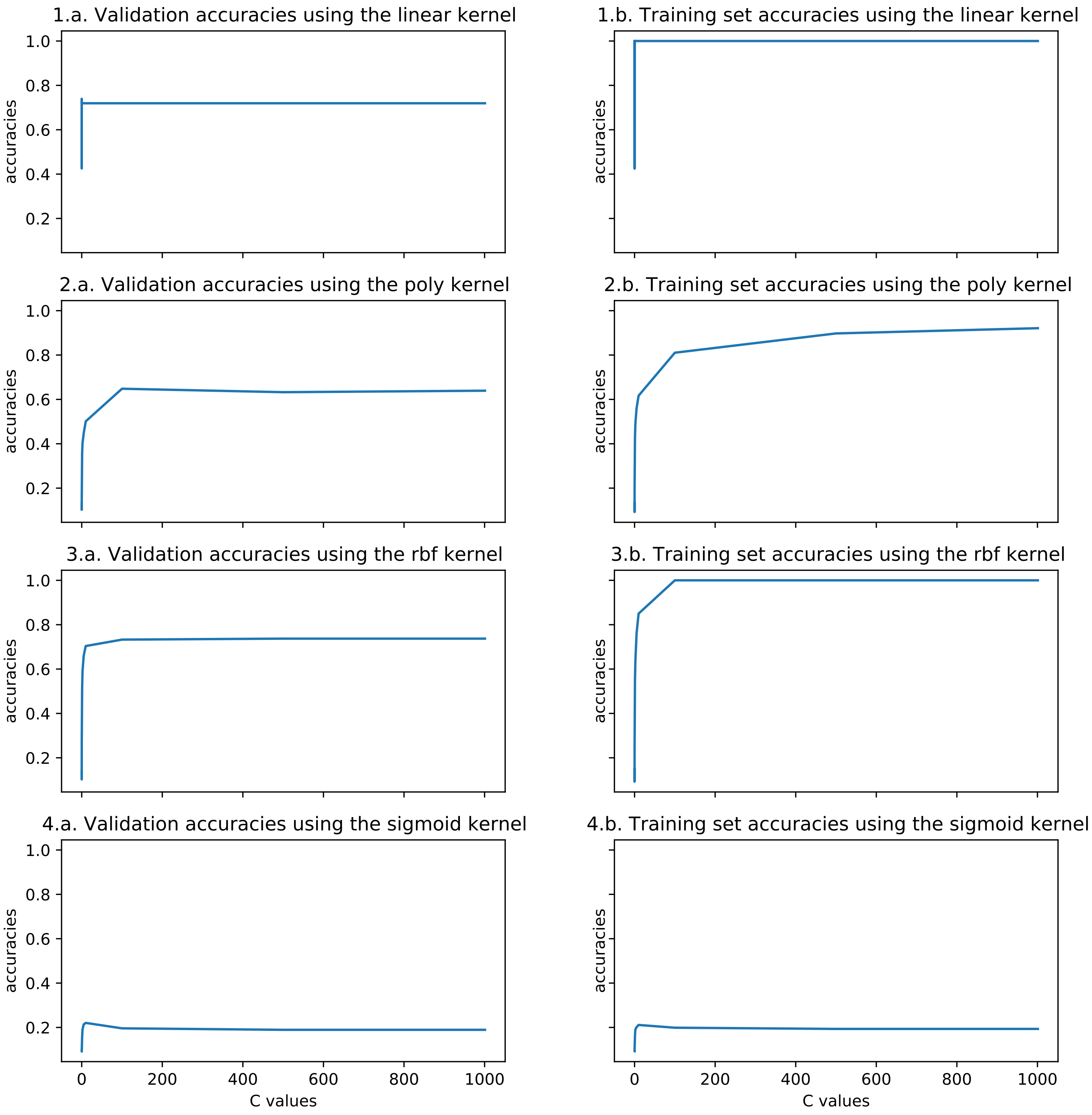


Figure 5.1: Graphs representing the variation of the accuracy in the validation and the training set using different kernels

5.3 Question 20

The validation set is needed in order to tune the hyper parameters, like C, the kernel, and the strategy of classification. We want to be able to know which values are best for our problem. So, it will contribute in the training process. Furthermore, We do not want to use the test set for this task. Because, we need it to test the performances of the final model on new data (non-seen during the training). Thus, we can accurately evaluate the model.