# Machine Learning Exercise 4

Florian Schneider, Phil Szalay, Amin Kasipour

03.12.12

# 1 Experiments on SVMs using libSVM[1]

## 1.1 Features

After some feature engineering I had the best scores with the following features.

- Mean of the image of all RGB channels
- Mean of the image of R channel
- Mean of the image of B channel
- Min of the image of all RGB channels
- Max of the image of all RGB channels
- Number of purple pixels in the RGB image. See 1.1.1

### 1.1.1 Number of purple pixels feature

This feature was computed by calculating the 'maximum' purple color $p_{max} = \begin{pmatrix} 157 & 80 & 139 \end{pmatrix}$ and the 'minimum' purple color $p_{min} = \begin{pmatrix} 116 & 20 & 130 \end{pmatrix}$. The number of purple pixel is the number of pixels which lay in between $p_{max}$ and $p_{min}$.

## 1.2 libSVM Experiments

In our experiments we used the tools provided by the libSVM package for python and Ubuntu.

The first step is to split the data in a train and test set $(0.8+0.2)$ and create a specially formed data files, what we accomplished with an own python script.

The second step is to scale the features so that they're all between $0 \ldots 1$ with the *svm-scale* tool.

[1] `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

Then we ran the parameter grid search for all kernels (c.f. Eq. 1, Eq. 2, Eq. 3, Eq. 4) on the training dataset for the parameters $C$ and  using the *svm-grid* tool as third step. The results are depicted in Table 1.2 and Figures 1.21.21.21.2

$$K_{linear}(\vec{x}, \vec{y}) = \vec{x} \cdot \vec{y} \tag{1}$$

$$K_{poly}(\vec{x}, \vec{y}) = (\gamma * \vec{x} \cdot \vec{y} + \text{coef})^d \tag{2}$$

$$K_{rbf}(\vec{x}, \vec{y}) = \exp\left(-\gamma * \|\vec{x} - \vec{y}\|^2\right) \tag{3}$$

$$K_{sigm}(\vec{x}, \vec{y}) = tanh(\gamma * \vec{x} \cdot \vec{y} + \text{coef}) \tag{4}$$

| **KERNEL** | $C$ | $\gamma$ | d | coef | CV |
|---|---|---|---|---|---|
| **linear** (Eq. 1) | 0.125 | 0.0078125 | not present | not present | 97.9167 |
| **polynomial** (Eq. 2) | 0.125 | 0.5 | 3 | 0 | 95.8333 |
| **RBF** (Eq. 3) | 2.0 | 0.5 | not present | not present | 100.0 |
| **sigmoid** (Eq. 4) | 32.0 | 0.0078125 | not present | 0 | 97.9167 |

Table 1: Results for the best parameters $C$ and $\gamma$ found by grid search using svm-grid on the training dataset with the the polynomial degree $d$ and/or the coefficient parameter coef. The resulting cross-validation rate (CV) is also shown.

After finding the best parameters we trained a model using *svm-train* for each kernel using the parameters from Table 1.2.

Those models were then used in *svm-predict* to classify the vectors in the test dataset . The results are shown in Table 2

| **Kernel** | **Accuracy** |
|---|---|
| **linear** | $\frac{12}{12} = 1.0$ |
| **polynomial** | $\frac{9}{12} = 0.75$ |
| **RBF** | $\frac{12}{12} = 1.0$ |
| **sigmoid** | $\frac{12}{12} = 1.0$ |

Table 2: Prediction results of the test set using svm-predict with models trained with parameters from Table

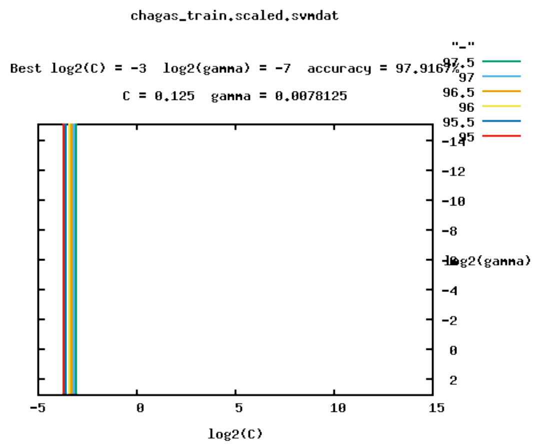


Figure 1: Visualization of grid search using a linear kernel.

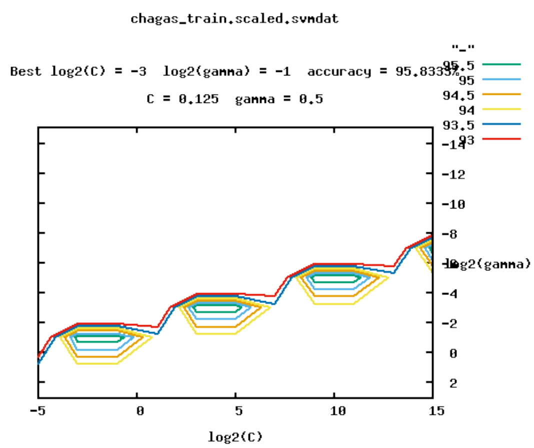


Figure 2: Visualization of grid search using a polynomial kernel with d = 3 and coef = 0

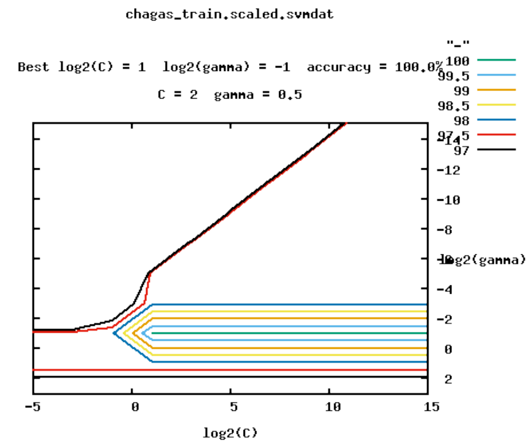


Figure 3: Visualization of grid search using a RBF kernel.

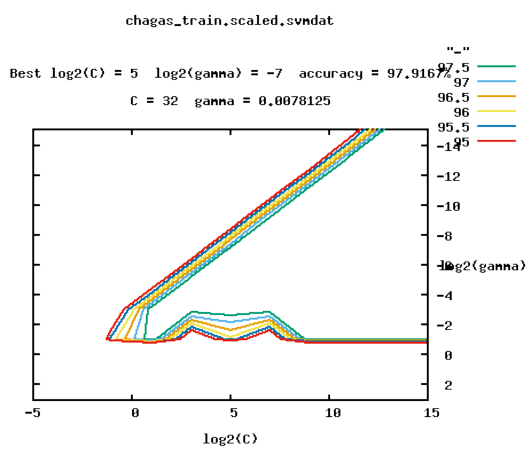


Figure 4: Visualization of grid search using a sigmoid kernel with coef = 0