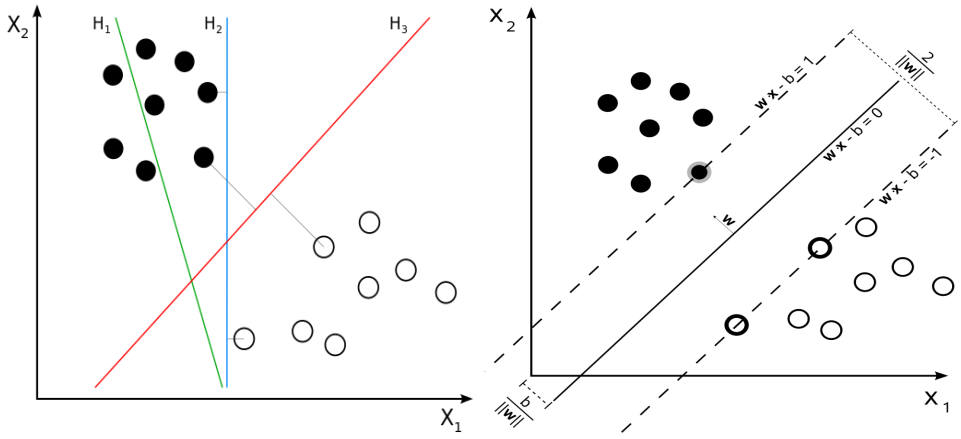
Classifier 2: Support Vector Machine

Introduction

A support vector machine(SVM) is a classifier that uses a separating hyperplane to analyze and classify data. The following images visualize a 2 dimensional situation of SVM. The separating hyperlane represents as the lines. And two classes (black circles, white circles) can be seen as well in the images. The question is which one is the optimal hyperplane. The answer is to find a hyperplane that is as far as possible from all the points. To put it more formal, training SVM classifier is the problem of finding the optimal hyperplane that gives the largest minimum distance to the training data points. It turns SVM to an optimization problem.



Images source: https://en.wikipedia.org/wiki/Support_vector_machine

The formula to find the optimal hyperplane would be

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$s.t. \sum_{i=1}^{n} y_i \alpha_i = 0$$

$$0 \leq \alpha_i \leq C$$

Originally, hyperplane can only be a linear classifier. By applying the kernel trick, we are able to create nonlinear classifiers. There are several kernels can be used in practice. For our case, we use Gaussian radial basis function (RBF SVM), which gives us $k(\vec{x_i}, \vec{x_j}) = \exp(-\gamma \|\vec{x_i} - \vec{x_j}\|^2)$. As a result, we will have two parameters, C and gamma, to decide for our SVM classifier. The C parameter represents the trades off between misclassification of training examples and simplicity of the decision surface, while the gamma parameter specifies how far the influence of a single training example reaches.

Feature Extraction(SVM)

For the feature extractions, we use several feature extractors provided by the scikit image library. These extractors enable us to build the feature vectors that are later used as the input for SVM classifier. We test different features provided by scikit image features and choose those who provide the best results. The short explanation of each extractors are introduced below.

- ORB: Oriented FAST and rotated BRIEF feature detector and binary descriptor extractor.
- BRIEF: Binary Robust Independent Elementary Features is an efficient feature point descriptor. For each keypoint, intensity comparisons are carried out for a specifically distributed number N of pixel-pairs resulting in a binary descriptor of length N. For binary descriptors the Hamming distance can be used for feature matching, which leads to lower computational cost in comparison to the L2 norm (scikit)
- CENSURE: Center Surround Extremas for Realtime Feature Detection and Matching
- Histogram of the grey scale image
- Corner peaks: Find corners in corner measure response image.
- Histogram of Oriented Gradients
  An image gradient is a concept to describe the change of intensity or color in an image. In every pixel of an image, a gradient contains two values: a magnitude and a direction. A magnitude is a description about how strong is the change, while the value of a direction is used to describe the angle of the changes. There are several steps for building a HOG. First, preprocessing the image to set a certain size ratio for every images. Then, computing the horizontal and vertical gradients. Next, dividing the image pixels into cells and calculating the Histogram of Gradients in each designated cells. For example, an 128*64 pixels images can be divided into 128 8*8 cells. The x-axis of a HOG is divided by the angles of gradients and y-axis represents the magnitudes. In order to less sensitive to the overall lighting changes, we need to normalize the HOG. Finally, to compute the feature vectors, the normalized HOG is turned into one vector.

Feature Selection Optimization

Since the different fruits that are used for classification have many different characteristics and forms, it was noticed that extracting certain features from certain fruits resulted worse or not optimal accuracy rates. For this reason, an algorithm was developed where a configuration list is created that details which feature extractors should be utilized for a certain fruits. This is accomplished by creating a list of every possible status (included/not included represented as booleans) for each extractor, thus resulting in 16 configuration lists. For example, when comparing apples to bananas extracting only the corner peaks the and the edge histogram results in a better result (90,5%; [False, False, True, True]) than if the ORB, BRIEF and edge histogram are extracted without the corner peaks (81%; [True, True, True, False]). A possible explanation for this is that some fruits lack certain characteristics or possess them in insufficient quality and quantity. To correctly estimate which features worsen model accuracy, a sample of 100 images is extracted from the each fruits training images and then fed into the SVM model. The accuracy for

each configuration list is returned and finally the configuration list chosen which bears the highest one. This list is later used for determining which features need to be extracted during the main process.

It was noticed that the HOG and grayscale histogram extractors always returned more positive results if they were included which is why they are automatically used to extract their respective features and feed them into the model.

Grid Search

As described during the theoretical introduction of the SVM classifier, two parameters need to be chosen carefully to deliver the best possible results, C and gamma. Their values can greatly influence the resulting accuracy of the model and thus a careful fine-tuning. The SkLearn machine learning library offers the Method *GridSearchCV* which calculates the best possible parameter combination for a certain parameter range. To calculate C, the values from 1.0e-02 to 1.0e10 were used. To calculate gamma, the values from 1.0e-09 to 1.0e3 were used. This process resulted in increased accuracy of up to 3% compared to the default values of gamma = 1.0e-9 and C = 10.0 which have been calculated earlier for specific differences in C and gamma, e.g. with C being 1.0 or 100.0. However, when using grid search it is important to consider that the model is at risk of being overfitted towards the used training data.

Result of SVM Testing

By changing the parameters in SVM, we tried to find the best combination of features among every possible combination of four features: ORB, Corner peaks, Edge and CENSURE. Since the remaining two features, histogram of gray scale image and HOG, always increased prediction accuracy we decided to always include them. The test accuracy of the best combinations of features for each pair of fruits is depicted below. The best combination with optimized grid parameters for n = 400 is depicted as well as the mean accuracy of all possible combinations and the best possible combination for n = 100.

Fruit Pair	Histogram of gray scale image	HOG	ORB	Corner peaks	Edge	CENSURE	Accuracy n=100 (mean)	Accuracy n=100 (best combination)	Accuracy n=400 (best combination)
	✓	✓	x	✓	✓	x	84.3%	90.5%	92%
	✓	✓	✓	x	x	✓	78.8%	88.5%	86%
	✓	✓	✓	x	✓	x	97.6%	98.4%	97%
	✓	✓	✓	✓	x	x	86.1%	92.0%	88%
	✓	✓	x	x	x	✓	76%	84.1%	82%
	✓	✓	✓	x	✓	✓	74.9%	84.9%	92%

The results show that there is no feature from the combination that was included in every best possible combination. One can also notice that the accuracy for n=100 samples varies greatly depending on if the best possible feature combination is extracted. One can also see that the best possible combination for n=100 and n=400 are not clearly superior or superior to each other. This high variance could also be caused by the relatively low sample number. However, the mean accuracy of all possible combinations is always lower than for the best possible combination for n = 400.

Reference

1. http://scikit-image.org/docs/dev/api/skimage.feature.html
2. http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html
3. https://en.wikipedia.org/wiki/Support_vector_machine
4. http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html