Multimedia Retrieval Exercise Course

7 Image Classification: Support Vector Machine

Kimiaki Shirahama, D.E.

Research Group for Pattern Recognition
Institute for Vision and Graphics
University of Siegen, Germany

Overview of Today’s Lesson

- Three Machine Learning Approaches
  - Similarity Search
  - Unsupervised Learning (Clustering)
  - Supervised learning (Classification)

- Support Vector Machine (SVM)

- LIBSVM (famous SVM software)
  - How to use LIBSVM

Three Machine Learning Approaches

Machine learning is the discipline to classify examples

NOTE: Each example is represented as a feature. In our case,

- *One image is an example*
- *A feature is, for example, the colour histogram extracted from the image*

1. Similarity search: Given an example, search the most similar example

2. Unsupervised learning (Clustering): Find groups (clusters) of similar examples

3. Supervised learning (Classification): Given examples with labels, build a model which can distinguish examples with different labels

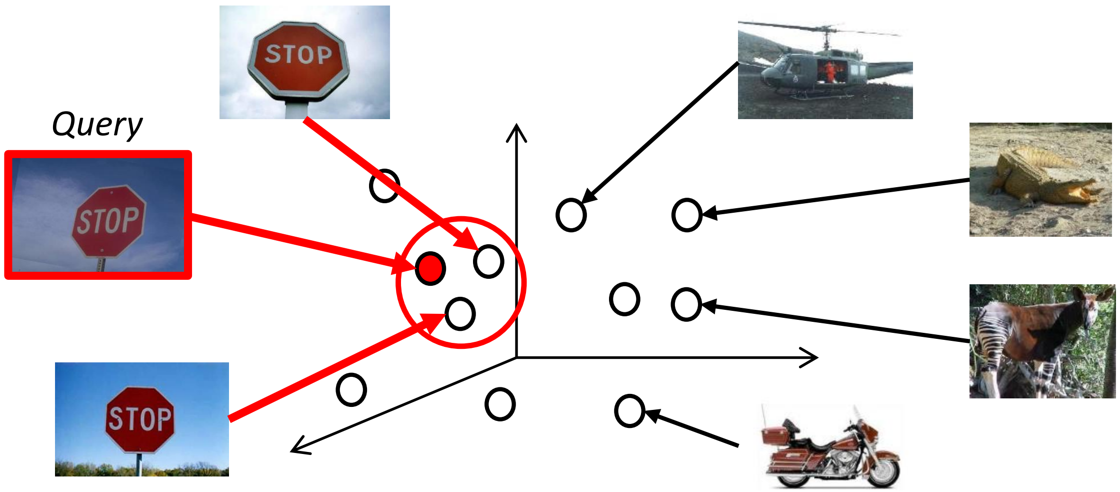
Three Machine Learning Approaches

Machine learning is the discipline to classify <u>examples</u>

NOTE: Each example is represented as a feature. In our case,
- *One image is an example*
- *A feature is, for example, the colour histogram extracted from the image*

1. Similarity search: Given an example, search the most similar examples

➡ *Query by example* that we have implemented in the last lessons



2. Unsupervised learning (Clustering): …

3. Supervised learning (Classification): …

Three Machine Learning Approaches

Machine learning is the discipline to classify <u>examples</u>

NOTE: Each example is represented as a feature. In our case,

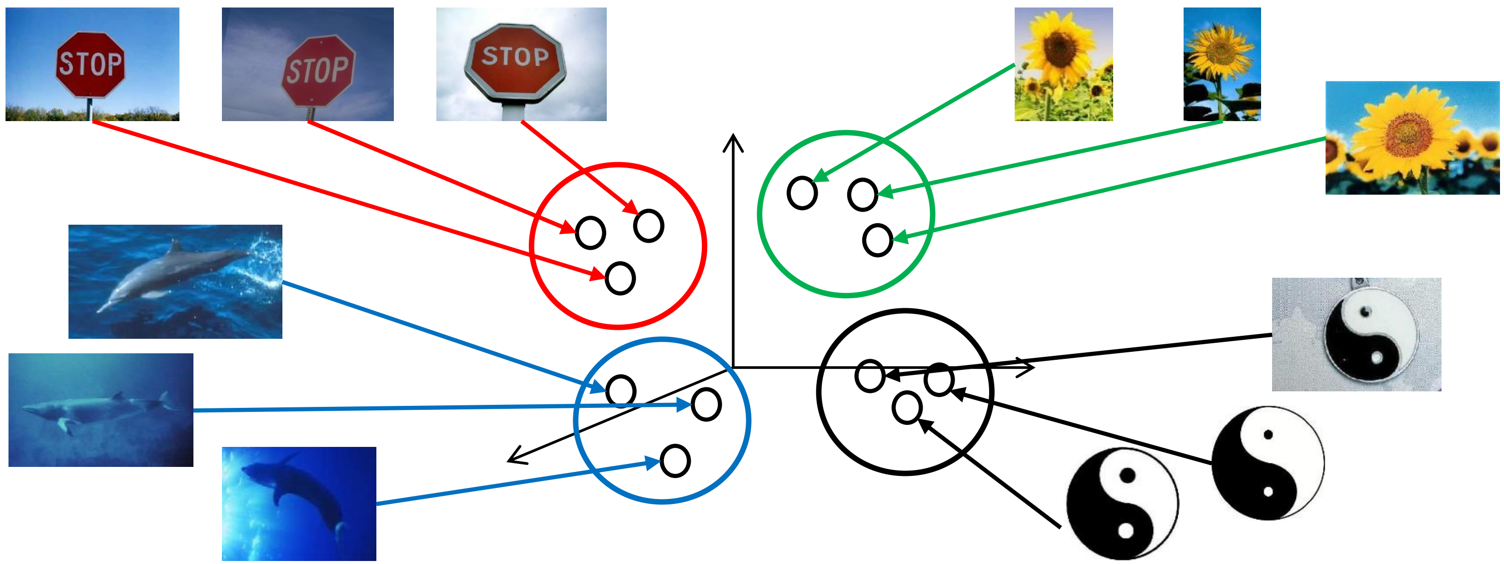
- *One image is an example*
- *A feature is, for example, the colour histogram extracted from the image*

1. Similarity search: …

2. Unsupervised learning (Clustering): Find groups (clusters) of similar examples

(Group of examples having many red and blue pixels)

(Group of examples having many yellow and blue pixels)



(Group of examples having many blue pixels)

(Group of examples having many black and white pixels)

3. Supervised learning (Classification): …

Three Machine Learning Approaches

Machine learning is the discipline to classify examples

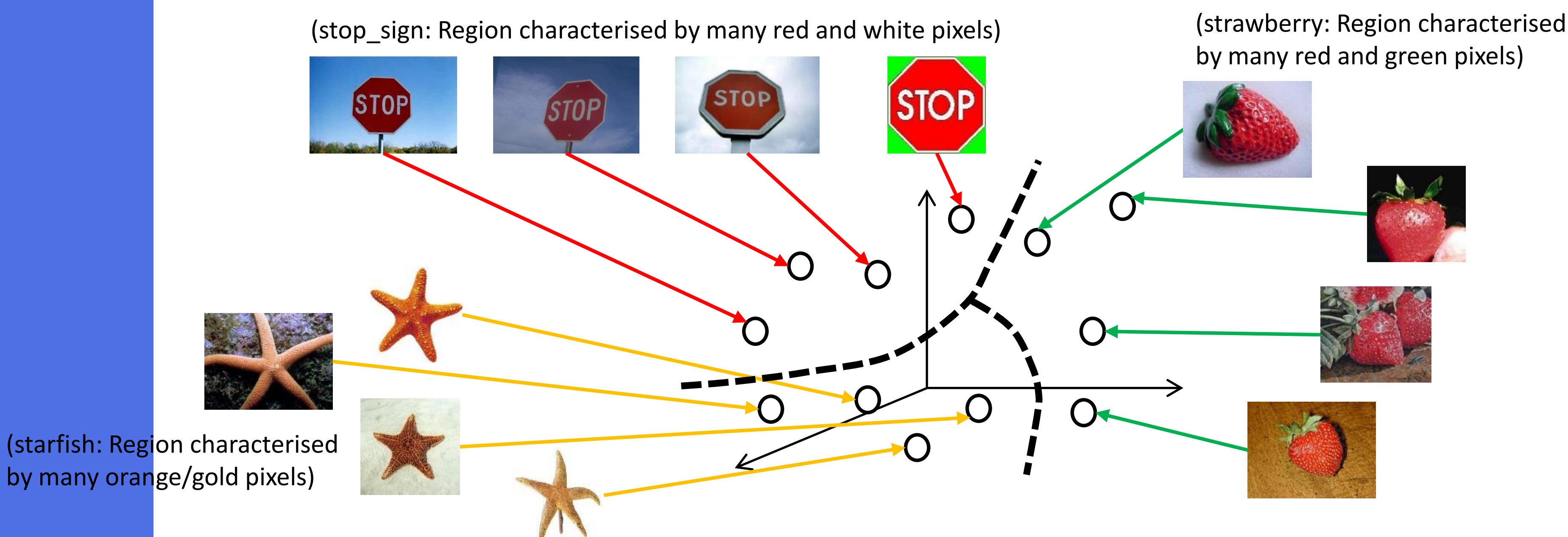
NOTE: Each example is represented as a feature. In our case,

- *One image is an example*
- *A feature is, for example, the colour histogram extracted from the image*

1. Similarity search: …

2. Unsupervised learning (Clustering): …

3. Supervised learning (Classification): Given examples with labels, build a model which can distinguish examples with different labels



In the feature space, find regions each of which is occupied by examples with the same label

Formal Definition of (Binary) Classification

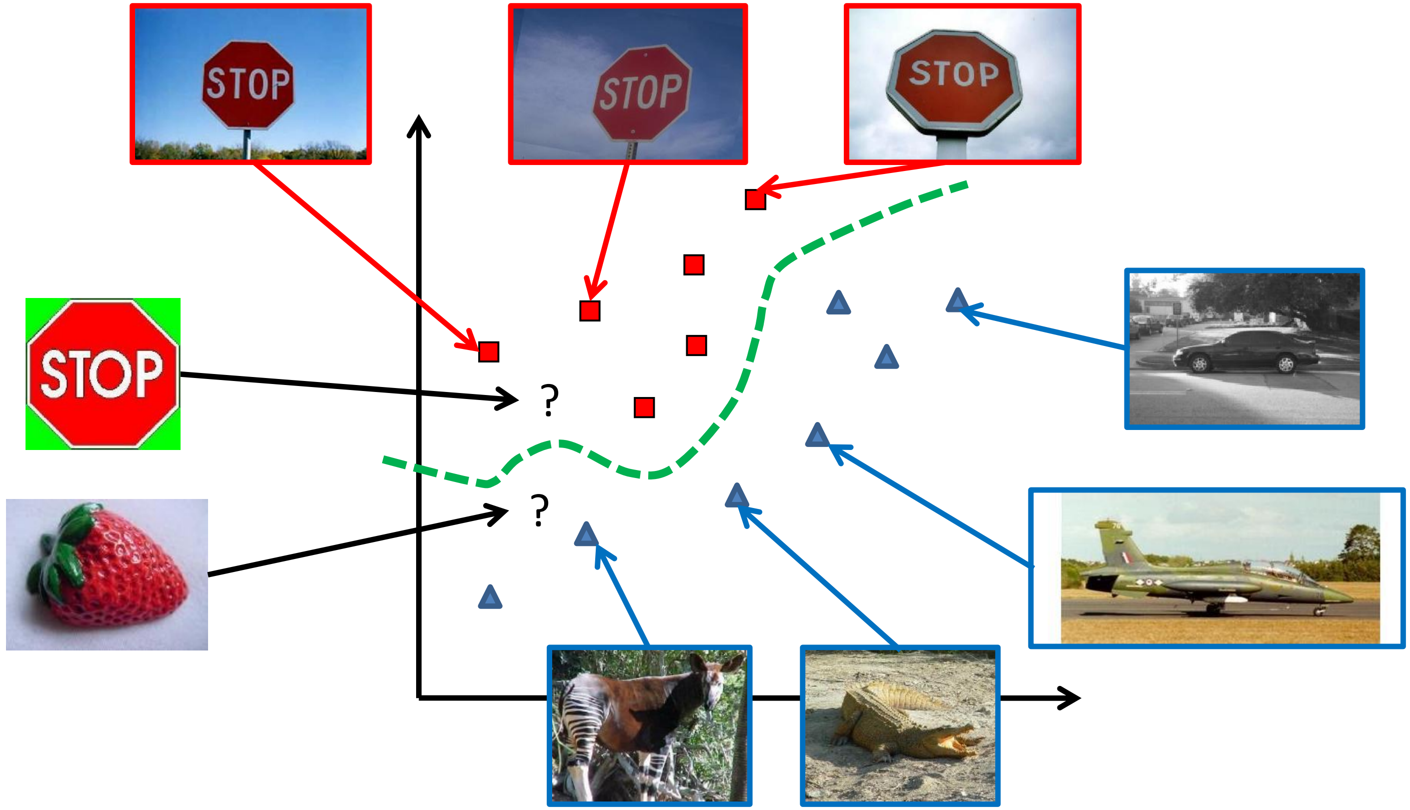
Problem: Given training examples, predict the label of a test example

We adopt binary classification where training examples consist of

- **Positive examples** where a target object (e.g., stop_sign) is shown
- **Negative examples** where the other objects (e.g., airplane, car, etc.) are shown

For test examples, we don't know if the target object is shown or not

NOTE: Examples mean images represented by colour histograms



Support Vector Machine (SVM)

SVM places a classification boundary in the middle between positive and negative examples
⇨ SVM extracts a useful similarity measure which combines weighted similarities between training examples and a test example

(Decision function of SVM)

$$h(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) - b$$

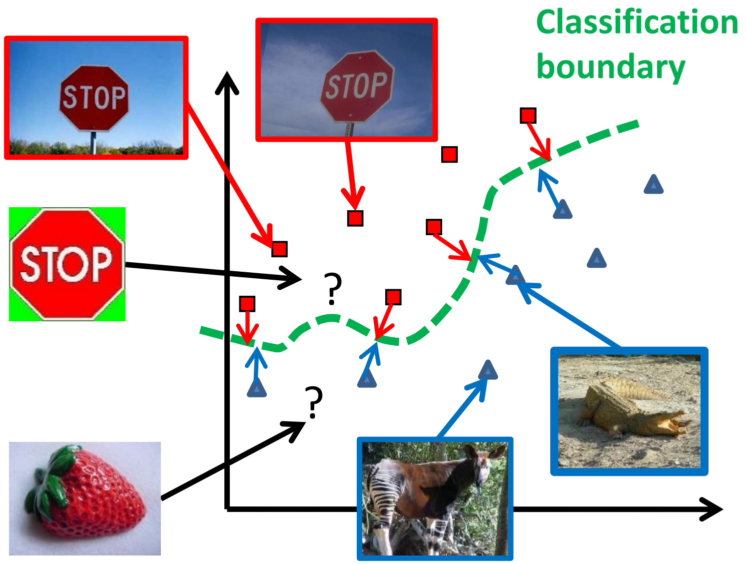
Weight

The optimal set of weight is computed using training examples

Kernel function (similarity between x_i and x)

$$K(x, x_i) = \exp\left(-\frac{\| x - x_i \|^2}{\gamma}\right)$$

Euclidian distance



- ■ If x is similar to positive examples, $+\alpha_i K(x_i, x)$
- □ If x is similar to negative examples, $-\alpha_i K(x_i, x)$

⇨ If the combined similarity is larger than 0, the object (Car) is regarded to be shown in x, otherwise, it is not shown in x

For more details, please refer to:
https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf
https://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf

Characteristics of SVM

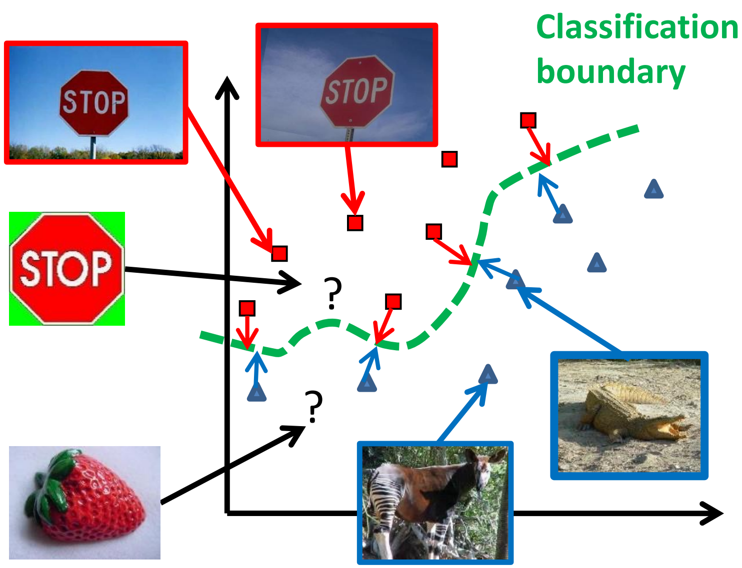
Advantage

1. Effectiveness for high-dimensional data: Placing a classification boundary in the middle between positive and negative examples (max margin principle), is theoretically proven to be independent of the number of dimensions.

2. Effectiveness for a small number of training data: A classification boundary is not probabilistically determined, but determined based on the geometry of positive and negative examples.

3. Convergence to the optimal solution: The objective function for SVM training is convex.

4. Reduction of the computational cost for SVM test: Only training examples selected as **support vectors** are needed to test the SVM.



Disadvantage

1. Quadratic increase of the computation time and the memory space depending on the number of training examples: In the case of extracting a non-linear classification boundary (non-linear SVM), it is needed to compute the similarity for each pair of training examples.

2. Difficulty of understanding results: Because of the high-dimensionality, it is difficult to know why good or bad classification is obtained. Especially, this is much difficult for non-linear SVMs.

LIBSVM

- One of the most famous SVM software (http://www.csie.ntu.edu.tw/~cjlin/libsvm/)
- A lot of SVM variations and options are supported
- Fast SVM training technique is implemented

- Another most famous software is SVM^{light}

How to Use LIBSVM

1. Download LIBSVM from http://www.csie.ntu.edu.tw/~cjlin/libsvm/
2. Run LIBSVM

Windows: Run executable files in the folder “windows”
Linux and Mac: Compile the source codes with “make” to create executable files

Open a command prompt or terminal

:SVM training svm-train (or ./svm-train) [options] training_set_file [model_file]
training_set_file: A file where training examples are written
model_file: A file where a trained SVM is output (if this is not specified, “training_set_file.model” is generated)

Example: svm-train ..\heart_scale

SVM test: svm-predict (or ./svm-predict) [options] test_file model_file output_file
test_file: A file where test examples are written
model_file: A file generated by SVM training
output_file: A file where classification results of test examples are written

Example: svm-predict ..\heart_scale.t ..\heart_scale.model ..\heart_scale.t.res.txt

NOTE: I created two files, the one named “heart_scale” contains the first 180 examples in the original “heart_scale”, the remaining 90 examples are contained in “heart_scale.t”.