CAP5415-Computer Vision
Lecture 13-Support Vector Machines for Computer Vision Applications

Guest Lecturer: Dr. Boqing Gong

Dr. Ulas Bagci
bagci@ucf.edu
Reminders

• October 14
  – Choose your mini-projects (both).
  – Send email with a short proposal/explanation.

• October 8
  – Due for Programming Assignment #3
Pattern Classification Problem

• Suppose we are given two classes of objects, we are then faced with a new object and we have to assign it to one of the two classes.
Motivation

denotes +1
denotes -1

How would you classify this data?
Motivation

How would you classify this data?

• denotes +1
• denotes -1
Motivation

How would you classify this data?

- denotes +1
- denotes -1
How would you classify this data?
Motivation

- denotes +1
- denotes -1

Any of these would be fine..

..but which is best?
Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.
The maximum margin linear classifier is the linear classifier with the, um, maximum margin. This is the simplest kind of SVM (Called an LSVM).
The maximum margin linear classifier is the linear classifier with the maximum margin. This is the simplest kind of SVM (Called an LSVM).
Lecture 13: Support Vector Machines for Computer Vision Applications

Maximum Margin

1. Intuitively this feels safest.
2. If we’ve made a small error in the location of the boundary (it’s been jolted in its perpendicular direction) this gives us least chance of causing a misclassification.
3. LOOCV is easy since the model is immune to removal of any non-support-vector datapoints.
4. There’s some theory (using VC dimension) that is related to (but not the same as) the proposition that this is a good thing.
5. Empirically it works very very well.
• A supervised approach for classification and regression
SVM

• A supervised approach for classification and regression
  – Developed in the computer science society at 1990s, has grown in popularity since then.
SVM

• A supervised approach for classification and regression
  – Developed in the computer science society at 1990s, has grown in popularity since then.
  – Shown to perform well in variety of settings, and often considered as one of the best “out-of-the box” classifiers.
SVM

• An approach for classification
  – Developed in the computer science society at 1990s, has grown in popularity since then.
  – Shown to perform well in variety of settings, and often considered as one of the best “out-of-the box” classifiers.

Max-margin Classifier  \rightarrow \text{Support Vector Classifier}  \rightarrow \text{Support Vector Machines}

(historical appearances in the literature)
Maximal-Margin Classifier

• In a $p$-dimensional space, a hyperplane is a flat affine subspace of dimension $p-1$. 
Maximal-Margin Classifier

• In a $p$-dimensional space, a hyperplane is a flat affine subspace of dimension $p-1$.
  – Ex. In 2D, a hyperplane is 1D line,
  – Ex. In 3D, a hyperplane is 2D plane.
Maximal-Margin Classifier

• In a $p$-dimensional space, a hyperplane is a flat affine subspace of dimension $p-1$.
  – Ex. In 2D, a hyperplane is 1D line,
  – Ex. In 3D, a hyperplane is 2D plane.

General hyperplane definition

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$$
Maximal-Margin Classifier

• In a $p$-dimensional space, a hyperplane is a flat affine subspace of dimension $p-1$.
  – Ex. In 2D, a hyperplane is 1D line,
  – Ex. In 3D, a hyperplane is 2D plane.

**General hyperplane definition**

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$$

**Hyperplane for 2D data.**

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$$
Maximal-Margin Classifier

- In a $p$-dimensional space, a hyperplane is a flat affine subspace of dimension $p-1$.
  - Ex. In 2D, a hyperplane is 1D line,
  - Ex. In 3D, a hyperplane is 2D plane.

General hyperplane definition

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_p X_p = 0$$

Hyperplane for 2D data.

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$$

$$1 + 2X_1 + 3X_2 > 0$$

$$1 + 2X_1 + 3X_2 < 0$$
Max-margin classifier

Left: There are two classes of observations: blue and in purple (each of which has measurements on two variables). Three separating hyperplanes, out of many possible, are shown in black. Right: A separating hyperplane is shown in black: a test observation that falls in the blue portion of the grid will be assigned to the blue class, and a test observation that falls into the purple portion of the grid will be assigned to the purple class.
Max-margin classifier

There are two classes of observations, shown in blue and in purple. The maximal margin hyperplane is shown as a solid line.

The margin is the distance from the solid line to either of the dashed lines. The two blue points and the purple point that lie on the dashed lines are the support vectors, and the distance from those points to the margin is indicated by arrows.

The purple and blue grid indicates the decision rule made by a classifier based on this separating hyperplane.
Construction of Max-Margin Classifier

• Consider $n$ training observations $x_1, \ldots, x_n \in \mathbb{R}^p$
Construction of Max-Margin Classifier

- Consider $n$ training observations $x_1, \ldots, x_n \in \mathbb{R}^p$
- and associated class labels $y_1, \ldots, y_n \in \{-1, +1\}$
Construction of Max-Margin Classifier

• Consider \( n \) training observations \( x_1, \ldots, x_n \in \mathbb{R}^p \) and associated class labels \( y_1, \ldots, y_n \in \{-1, +1\} \).

• Briefly, max-margin hyperplane is the solution to the optimization problem:

\[
\begin{align*}
\text{maximize} & \quad M \\
\text{subject to} & \quad \sum_{j=1}^{p} \beta_j^2 = 1, \\
& \quad y_i (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M \quad \forall \ i = 1, \ldots, n.
\end{align*}
\]
Construction of Max-Margin Classifier

• Consider n training observations \( x_1, \ldots, x_n \in \mathbb{R}^p \) and associated class labels \( y_1, \ldots, y_n \in \{-1, +1\} \).

• Briefly, max-margin hyperplane is the solution to the optimization problem:

\[
\begin{align*}
\text{maximize } & M \\
\text{subject to } & \sum_{j=1}^{p} \beta_j^2 = 1, \\
& y_i (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M \quad \forall i = 1, \ldots, n.
\end{align*}
\]

• This equation ensures that each observation is on the correct side of the hyperplane and at least a distance M from the hyperplane. Hence, M represents the margin of our hyperplane.
Non-Separable Case

The maximal margin classifier is a very natural way to perform classification, if a separating hyperplane exists.

In many cases there is no separating hyperplane exists!

In this case, we cannot exactly separate the two classes. (However, Notice soft margin in the following slides!)

The generalization of the maximal margin classifier to the non-separable case is known as the support vector classifier.
Support Vector Classifier - Separable

- Separable plane is shown.
- Decision boundary is the solid line.
- Broken lines bound the Shaded maximal margin of 2M.
Support Vector Classifier-Overlap

- Overlap case is shown.
- The points labeled $\xi_i$ are on the wrong side of the Margin.
- The margin is maximized subject to a total budget

$$\sum \xi_i \leq \text{constant}$$
Left: Two classes of observations are shown in blue and in purple, along with the maximal margin hyperplane.

Right: An additional blue observation has been added, leading to a dramatic shift in the maximal margin hyperplane shown as a solid line. The dashed line indicates the maximal Margin hyperplane that was obtained in the absence of this additional point. Max-margin classifier is sensitive to individual observations!
Soft-Margin ~ Support Vector Classifier

- Rather than seeking the largest possible margin so that every observation is not only on the correct side of the hyperplane but also on the correct side of the margin, we instead allow some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane.
Soft-Margin ~ Support Vector Classifier

- Rather than seeking the largest possible margin so that every observation is not only on the correct side of the hyperplane but also on the correct side of the margin, we instead allow some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane.

\[
\begin{align*}
\text{maximize} & \quad M \\
\text{subject to} & \quad \sum_{j=1}^{p} \beta_j^2 = 1, \\
y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) & \geq M(1 - \epsilon_i), \\
\epsilon_i & \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C,
\end{align*}
\]

- C: nonnegative tuning params, eps: slack variables allowing obs. to be on the wrong Side of the margin.
**Left:** A support vector classifier was fit to a small data set. The hyperplane is shown as a solid line and the margins are shown as dashed lines. Purple observations: Observations 3, 4, 5, and 6 are on the correct side of the margin, observation 2 is on the margin, and observation 1 is on the wrong side of the margin. Blue observations: Observations 7 and 10 are on the correct side of the margin, observation 9 is on the margin, and observation 8 is on the wrong side of the margin. No observations are on the wrong side of the hyperplane. **Right:** Two additional points, 11 and 12 are added.
Four different tuning parameters were used to fit SVM into a small number of data points.

The largest value of C was used in the top left panel, and smaller values were used in the top right, bottom left, and bottom right panels.

When C is large, there is a high tolerance for observations being on the wrong side of the margin, and so the margin will be large. As C decreases, the tolerance for observations being on the wrong side of the margin decreases, and the margin narrows.
In practice we are sometimes faced with non-linear class boundaries. Support vector classifier or any linear classifier will perform poorly here

**Left:** The observations fall into two classes, with a non-linear boundary between them.

**Right:** The support vector classifier seeks a linear boundary, and consequently performs very poorly.
Non-Linear Support Vector Machines

- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:

\[ \Phi: \mathbf{x} \rightarrow \varphi(\mathbf{x}) \]
The Kernel Trick

The linear classifier relies on dot product between vectors $K(x_i, x_j) = x_i^T x_j$

If every data point is mapped into high-dimensional space via some transformation $\Phi$: $x \rightarrow \phi(x)$, the dot product becomes:

$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$

A kernel function is some function that corresponds to an inner product in some expanded feature space.

Example:

2-dimensional vectors $x = [x_1, x_2]$; let $K(x_i, x_j) = (1 + x_i^T x_j)^2$.

Need to show that $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$:

$K(x_i, x_j) = (1 + x_i^T x_j)^2$

$= 1 + x_{i1}^2 x_{j1}^2 + 2 x_{i1} x_{j1} x_{i2} x_{j2} + x_{i2}^2 x_{j2}^2 + 2 x_{i1} x_{j1} + 2 x_{i2} x_{j2}$

$= [1 \ x_{i1}^2 \sqrt{2} \ x_{i1} x_{i2} \ x_{i2}^2 \sqrt{2} x_{i1} \sqrt{2} x_{i2}]^T [1 \ x_{j1}^2 \sqrt{2} \ x_{j1} x_{j2} \ x_{j2}^2 \sqrt{2} x_{j1} \sqrt{2} x_{j2}]$

$= \phi(x_i)^T \phi(x_j)$, where $\phi(x) = [1 \ x_1^2 \sqrt{2} \ x_1 x_2 \ x_2^2 \sqrt{2} x_1 \sqrt{2} x_2]$


**Example of Kernel Functions**

- **Linear:** $K(x_i, x_j) = x_i^T x_j$

- **Polynomial of power $p$:** $K(x_i, x_j) = (1 + x_i^T x_j)^p$

- **Gaussian (radial-basis function network):**

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

- **Sigmoid:** $K(x_i, x_j) = \tanh(\beta_0 x_i^T x_j + \beta_1)$
Non-linear SVM Mathematically

■ Dual problem formulation:

Find $\alpha_1...\alpha_N$ such that

$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j K(x_i, x_j)$ is maximized and

(1) $\sum \alpha_i y_i = 0$

(2) $\alpha_i \geq 0$ for all $\alpha_i$

■ The solution is:

$$f(x) = \sum \alpha_i y_i K(x_i, x_j) + b$$

■ Optimization techniques for finding $\alpha_i$’s remain the same!
Formulating the optimization problem

Constraint becomes:

\[ y_i (w \cdot x_i + b) \geq 1 - \xi_i, \quad \forall x_i \]
\[ \xi_i \geq 0 \]

Objective function penalizes for misclassified instances and those within the margin

\[ \min \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \]

\( C \) trades-off margin width and misclassifications
Non-Linear SVM Overview

- SVM locates a separating hyperplane in the feature space and classify points in that space.
- It does not need to represent the space explicitly, simply by defining a kernel function.
- The kernel function plays the role of the dot product in the feature space.
Disadvantages of Linear Decision Surfaces

\[ \text{Var}_1 \]

\[ \text{Var}_2 \]
Advantages of non-linear Decision Surfaces
Left: An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from previous slides, resulting in a far more appropriate decision rule.

Right: An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.
Multi-class SVM

• SVMs can only handle two-class outputs (i.e. a categorical output variable with arity 2).
• What can be done?
• Answer: with output arity N, learn N SVM’s
  – SVM 1 learns “Output==1” vs “Output != 1”
  – SVM 2 learns “Output==2” vs “Output != 2”
  – ...
  – SVM N learns “Output==N” vs “Output != N”
• Then to predict the output for a new input, just predict with each SVM and find out which one puts the prediction the furthest into the positive region.
Trade-off Between Flexibility and Interpretability

In general, as the flexibility of a method increases, its interpretability decreases.
Why do SVM generalize?

• Even though they map to a very high-dimensional space
  – They have a very strong bias in that space
  – The solution has to be a linear combination of the training instances

• Large theory on Structural Risk Minimization providing bounds on the error of an SVM
  – Typically the error bounds too loose to be of practical use
Practical Issues

• **Choice of kernel**
  - Gaussian or polynomial kernel is default
  - if ineffective, more elaborate kernels are needed
  - domain experts can give assistance in formulating appropriate similarity measures

• **Choice of kernel parameters**
  - e.g. \( \sigma \) in Gaussian kernel
  - \( \sigma \) is the distance between closest points with different classifications
  - In the absence of reliable criteria, applications rely on the use of a validation set or cross-validation to set such parameters.

• **Optimization criterion – Hard margin v.s. Soft margin**
  - a lengthy series of experiments in which various parameters are tested
CV Application of SVM: Human Detection

Navneet Dalal and Bill Triggs “Histograms of Oriented Gradients for Human Detection” CVPR05

- Each block consists of 2x2 cells with size 8x8
- Quantize the gradient orientation into 9 bins (0-180)
  - The vote is the gradient magnitude

Final Feature Vectors Go to SVM
CV Application of SVM: Pedestrian Detection

Feature vectors:
HOG: histogram of gradients
**CV Application of SVM: Pedestrian Detection**

### Training (Learning)

- Represent each example window by a HOG feature vector

\[
\begin{bmatrix}
\vdots \\
\vdots \\
\vdots \\
\end{bmatrix}
\]

\[x_i \in \mathbb{R}^d,\]

- Train a SVM classifier

### Testing (Detection)

- Sliding window classifier

\[f(x) = w^T x + b\]
References and Slice Credits

• An excellent tutorial on VC-dimension and Support Vector Machines:

• The VC/SRM/SVM Bible:

• Andrew W. Moore, CMU
  James, Witten, Hastie, Tibshirani: An Introduction to Statistical Learning