ROB 101 - Fall 2021

Soft Margin Classifier, Gaussian Support Vector Machine

December 1, 2021

UNIVERSITY OF MICHIGAN

Learning Objectives

- Introduce material that is assumed in UofM Computer Science courses that have Math 214 as a prerequisite.

- Provide a resource for use after you leave ROB 101.

Outcomes

- Soft Margin Classifier.
- Gaussian Support Vector Machine.

Signed Distance to a Hyperplane

- For any point that does not lie on the hyperplane, we have $\langle a, x - x_c \rangle \neq 0$ or $a^\mathsf{T} x \neq a^\mathsf{T} x_c$.

- We define the signed distance of a point to the hyperplane by the amount of deviation from the hyperplane equation.

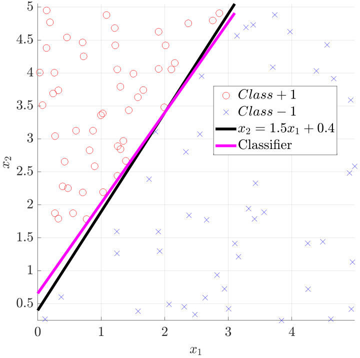
$$y(x) = \frac{\langle a, x - x_c \rangle}{\|a\|}$$

- We normalize by $\|a\|$ to avoid scaling the space.

Maximum Margin Classifier

- We wish to find a classifier (here a hyperplane) that separates × and ∘ categories.

- Furthermore, we want to predict the label for a new input (called query or test point).



Maximum Margin Classifier

- We are given a data set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, where the inputs are $x_i \in \mathbb{R}^2$ and targets are $y_i \in \{+1, -1\}$.

- Our model is a hyperplane $a^{\mathsf{T}} x + a_0 = 0$. a_0 is called the bias term.

- Define $w := \begin{bmatrix} a \\ a_0 \end{bmatrix}$ and $\bar{x} := \begin{bmatrix} x \\ 1 \end{bmatrix}$. Then $w^{\mathsf{T}} \bar{x} = 0$.

Maximum Margin Classifier

We define the following *hard margins*.

- $w^\mathsf{T}\bar{x} = 1$, anything on or above this boundary belongs to class $+1$.
- $w^\mathsf{T}\bar{x} = -1$, anything on or below this boundary belongs to class -1.
- We get the following constraints:

$$w^\mathsf{T}\bar{x}_i \geq 1, \quad \text{if } y_i = 1,$$
$$w^\mathsf{T}\bar{x}_i \leq -1, \quad \text{if } y_i = -1.$$

Maximum Margin Classifier

- We get the following constraints:

$$\begin{aligned} w^{\mathsf{T}}\bar{x}_i &\geq 1, \quad \text{if } y_i = 1, \\ w^{\mathsf{T}}\bar{x}_i &\leq -1, \quad \text{if } y_i = -1. \end{aligned}$$

- We can combine both constraints into one as

$$y_i \cdot w^{\mathsf{T}}\bar{x}_i \geq 1, \quad \text{for } i = 1, \ldots, n.$$

Maximum Margin Classifier

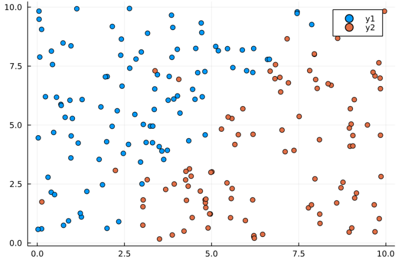
We now formulate the following *constrained* optimization problem.

$$\begin{aligned} \min_{w \in \mathbb{R}^3} \quad & \frac{1}{2} w^\mathsf{T} w \\ \text{subject to} \quad & y_i \cdot w^\mathsf{T} \bar{x}_i \geq 1, \quad \text{for } i = 1, \ldots, n. \end{aligned}$$

Soft Margin Classifier

- If two categories have overlaps, the hard margin approach does not work.

- **Proposed fix:** Introduce soft margins via some slack variables.



Soft Margin Classifier

The new problem is still a QP.

$$\begin{aligned} \min_{w\in\mathbb{R}^3,\xi\in\mathbb{R}^n} \quad & \frac{\lambda}{2}w^\mathsf{T}w + \frac{1}{2}\xi^\mathsf{T}\xi \\ \text{subject to} \quad & y_i \cdot w^\mathsf{T}\bar{x}_i \geq 1 - \xi_i, \quad \text{for } i = 1,\dots,n \\ & \xi_i \geq 0, \quad \text{for } i = 1,\dots,n. \end{aligned}$$

Remark

The new variables $\xi = \begin{bmatrix}\xi_1 \\ \vdots \\ \xi_n\end{bmatrix}$ *are called slack variables. We now solve for both* $w \in \mathbb{R}^3$ *and* $\xi \in \mathbb{R}^n$. λ *is a constant tunable parameter.*

Soft Margin Classifier

We can use a vectorized notation.

$$\begin{aligned} \min_{w \in \mathbb{R}^3, \xi \in \mathbb{R}^n} \quad & \frac{\lambda}{2} w^\mathsf{T} w + \frac{1}{2} \xi^\mathsf{T} \xi \\ \text{subject to} \quad & \Phi w \preceq -1 + \xi \\ & 0 \preceq \xi. \end{aligned}$$

We define $\Phi := \begin{bmatrix} -y_1 \bar{x}_1^\mathsf{T} \\ \vdots \\ -y_n \bar{x}_n^\mathsf{T} \end{bmatrix}_{n \times 3}$.

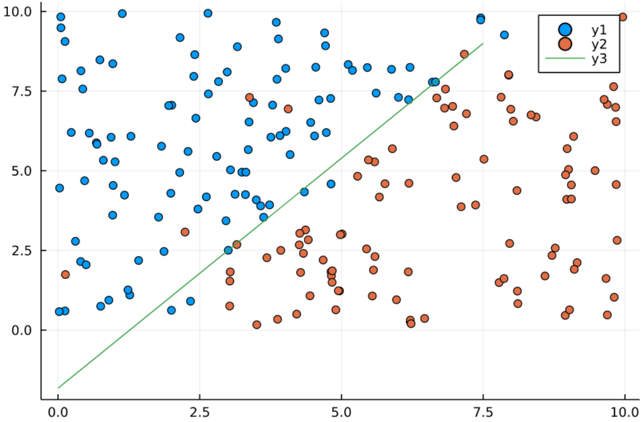
Soft Margin Classifier

This final form is more suitable for implementation using OSQP solver.

$$\begin{aligned}
\min_{w\in\mathbb{R}^3,\xi\in\mathbb{R}^n} \quad & \frac{1}{2}\begin{bmatrix} w \\ \xi \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \lambda I_3 & 0_{3\times n} \\ 0_{n\times 3} & I_n \end{bmatrix} \begin{bmatrix} w \\ \xi \end{bmatrix} \\
\text{subject to} \quad & \begin{bmatrix} -\infty_n \\ 0_n \end{bmatrix} \preceq \begin{bmatrix} \Phi_{n\times 3} & -I_n \\ 0_{n\times 3} & I_n \end{bmatrix} \begin{bmatrix} w \\ \xi \end{bmatrix} \preceq \begin{bmatrix} -1_n \\ \infty_n \end{bmatrix}.
\end{aligned}$$

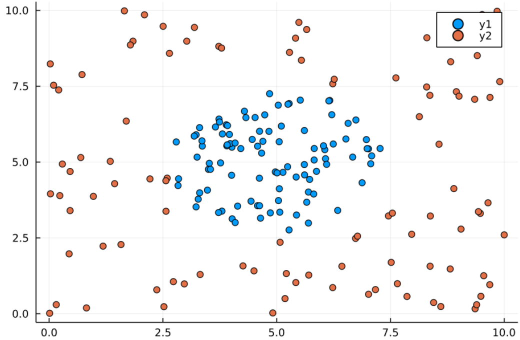
We defined $1_n := \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1}$ and $\infty_n := \begin{bmatrix} \infty \\ \vdots \\ \infty \end{bmatrix}_{n\times 1}$.

Soft Margin Classifier



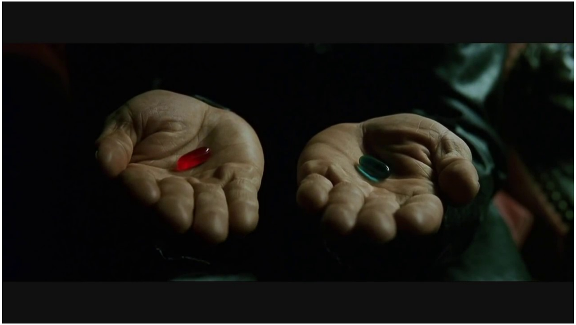
Gaussian (RBF) Support Vector Machine (SVM)

- What if data is not linearly separable even using soft margins?



Gaussian SVM

From Matrix: Choose between potentially unsettling or life-changing truth or remaining in contented ignorance.



Gaussian SVM

- **Key idea:** Lift the inputs using a feature map to a higher-dimensional space where data is separable.

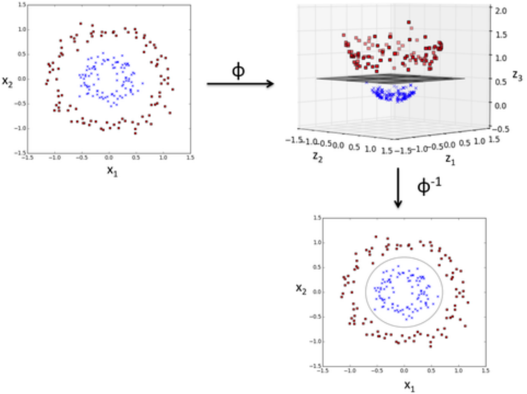


Figure: Image credit: `https://sebastianraschka.com/faq/docs/select_svm_kernels.html`

Gaussian SVM

Our outputs are of this form $y(x) = \langle w, \bar{x} \rangle = w^\mathsf{T} \bar{x}$. We use the following trick to implicitly lift the problem to the feature space.

- Define a feature map as $\bar{x} \mapsto \phi(x)$.
- Define a new output function as $y(x) = \sum_{j=1}^{n+1} \alpha_j \phi_j(x) = \alpha^\mathsf{T} \phi(x)$.
- Notice now $\alpha \in \mathbb{R}^{n+1}$ as opposed to $w \in \mathbb{R}^3$. n is the number of data points in the data set.
- We choose ϕ to be an RBF, i.e., $\phi_j(x) = \exp(-\frac{\|x-x_j\|^2}{2s^2})$

Gaussian SVM

$$\min_{\alpha \in \mathbb{R}^m, \xi \in \mathbb{R}^n} \quad \frac{1}{2} \begin{bmatrix} \alpha \\ \xi \end{bmatrix}^\top \begin{bmatrix} \lambda I_m & 0_{m \times n} \\ 0_{n \times m} & I_n \end{bmatrix} \begin{bmatrix} \alpha \\ \xi \end{bmatrix}$$

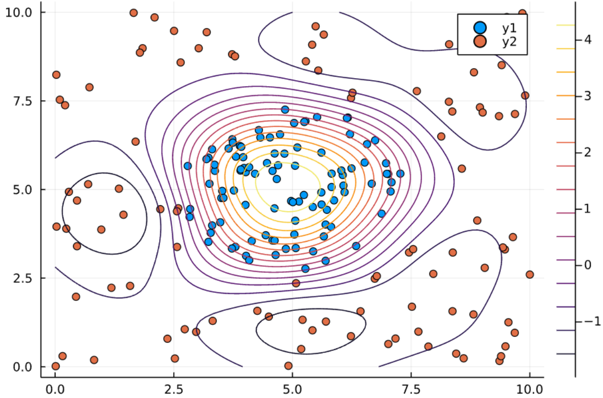
$$\text{subject to} \begin{bmatrix} -\infty_n \\ 0_n \end{bmatrix} \preceq \begin{bmatrix} \Phi_{n \times m} & -I_n \\ 0_{n \times m} & I_n \end{bmatrix} \begin{bmatrix} \alpha \\ \xi \end{bmatrix} \preceq \begin{bmatrix} -1_n \\ \infty_n \end{bmatrix}.$$

Where now $m = n + 1$,

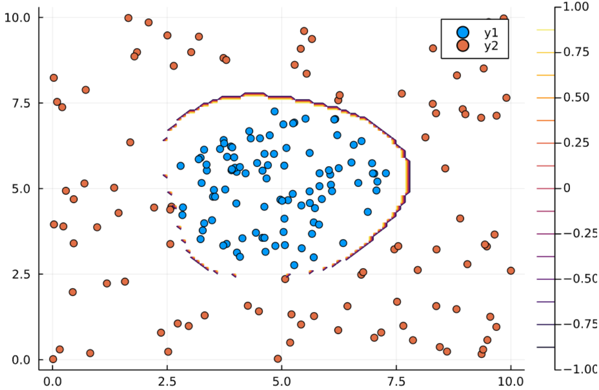
$$\Phi_{n \times m} = \begin{bmatrix} -y_1 \phi_1^\top \\ \vdots \\ -y_n \phi_n^\top \end{bmatrix}.$$

Gaussian (RBF) Support Vector Machine (SVM)

The contour plot shows the *decision boundaries*.



Gaussian (RBF) Support Vector Machine (SVM)



Gaussian (RBF) Support Vector Machine (SVM)

Remark

In kernel machines such as SVM, the feature map ϕ is implicitly defined using a kernel, e.g., Radial Basis Functions (RBFs) that you used in Project 2 for regression.

Remark

In neural networks, we explicitly search for a finite-dimensional feature vector ϕ. This process is called training the network where we automatically "learn" the features ϕ from data. We then use the learned features for regression and classification tasks. This is currently the mainstream approach in machine learning because of its better scalability using parallel computing technology.

Example

Let's switch to the Julia notebook.

Next Time

- Convolution
- Final Lecture.