732A99/TDDE01 Machine Learning

Lecture 3b Block 1: Support Vector Machines

Jose M. Peña
IDA, Linköping University, Sweden

Contents and Literature

- Content
  - Support Vector Regression
  - Support Vector Classification
  - Summary
- Literature
  - Lindholm, A., Wahlström, N., Lindsten, F. and Schön, T. B. *Machine Learning - A First Course for Engineers and Scientists*. Cambridge University Press, 2022. Chapter 8.3 and 8.5.

Support Vector Regression

- Kernel ridge regression = ridge regression in the feature space defined by the mapping $\phi(\cdot)$ rather than in the input space.

$$\widehat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \Big(\underbrace{\theta^{\mathsf{T}} \phi(\mathbf{x}_i)}_{\widehat{y}(\mathbf{x}_i)} - y_i \Big)^2 + \lambda \|\theta\|_2^2 = (\mathbf{\Phi}(\mathbf{X})^{\mathsf{T}} \mathbf{\Phi}(\mathbf{X}) + n\lambda \mathbf{I})^{-1} \mathbf{\Phi}(\mathbf{X})^{\mathsf{T}} \mathbf{y}, \tag{8.4a}$$

- Computing $\widehat{\theta}$ can be problematic for some feature spaces, since they can be high or even infinite dimensional. However, $\widehat{\theta}$ is just needed for prediction.

$$\widehat{y}(\mathbf{x}_\star) = \underbrace{\widehat{\theta}^{\mathsf{T}}}_{1\times d} \underbrace{\phi(\mathbf{x}_\star)}_{d\times 1} = \underbrace{\mathbf{y}^{\mathsf{T}}}_{1\times n} \underbrace{(\mathbf{\Phi}(\mathbf{X})\mathbf{\Phi}(\mathbf{X})^{\mathsf{T}} + n\lambda \mathbf{I})^{-1}}_{n\times n} \underbrace{\mathbf{\Phi}(\mathbf{X})\phi(\mathbf{x}_\star)}_{n\times 1}.$$

- The formulation above in terms of $\boldsymbol{\theta}$ is called the **primal formulation**. The formulation below in terms of $\boldsymbol{\alpha}$ is called the **dual formulation**. Note that $\boldsymbol{\alpha}$ is of dimension n (cf. $\boldsymbol{\theta}$).

$$\widehat{\boldsymbol{\alpha}} = \begin{bmatrix} \widehat{\alpha}_1 \\ \widehat{\alpha}_2 \\ \vdots \\ \widehat{\alpha}_n \end{bmatrix} = \mathbf{y}^{\mathsf{T}} (\boldsymbol{K}(\mathbf{X}, \mathbf{X}) + n\lambda \mathbf{I})^{-1}. \tag{8.14a}$$

$$\widehat{y}(\mathbf{x}_\star) = \widehat{\alpha}^{\mathsf{T}} \boldsymbol{K}(\mathbf{X}, \mathbf{x}_\star). \tag{8.14b}$$

- Clearly, $\widehat{\boldsymbol{\theta}} = \mathbf{\Phi}(\boldsymbol{X})^T \widehat{\boldsymbol{\alpha}}$. This result is known as the representer theorem.

Support Vector Regression

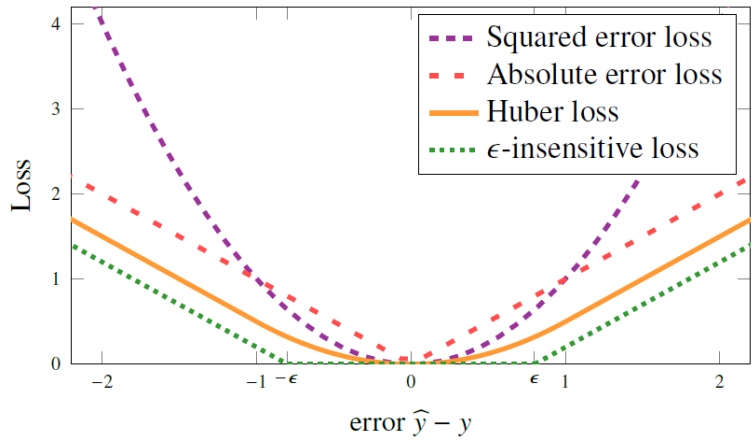

Figure 5.1: The loss functions for regression presented in the text, each as a function of the error $\widehat{y} - y$.

- Replace the squared error loss function in kernel ridge regression with the ϵ-insensitive loss function, where ϵ is a user-defined hyperparameter.

$$L(y, \widehat{y}) = \begin{cases} 0 & \text{if } |\widehat{y} - y| < \epsilon, \\ |\widehat{y} - y| - \epsilon & \text{otherwise,} \end{cases} \tag{5.9}$$

- In the primal formulation, we predict as

$$\widehat{y}(\mathbf{x}_\star) = \widehat{\boldsymbol{\theta}}^\mathsf{T} \boldsymbol{\phi}(\mathbf{x}_\star), \tag{8.18a}$$

where $\widehat{\boldsymbol{\theta}}$ is the solution to the optimization problem

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} \max\{0, |y_i - \underbrace{\boldsymbol{\theta}^\mathsf{T} \boldsymbol{\phi}(\mathbf{x}_i)}_{\widehat{y}(\mathbf{x}_i)}| - \epsilon\} + \lambda \|\boldsymbol{\theta}\|_2^2. \tag{8.18b}$$

which has to be solved numerically (i.e., no closed-form solution exists).

Support Vector Regression

- In the dual formulation, we predict as

$$\widehat{y}(\mathbf{x}_\star) = \widehat{\boldsymbol{\alpha}}^\mathsf{T} \boldsymbol{K}(\mathbf{X}, \mathbf{x}_\star), \tag{8.19a}$$

where $\widehat{\boldsymbol{\alpha}}$ is the solution to the optimization problem

$$\widehat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^\mathsf{T} \boldsymbol{K}(\mathbf{X}, \mathbf{X})\boldsymbol{\alpha} - \boldsymbol{\alpha}^\mathsf{T}\mathbf{y} + \epsilon\|\boldsymbol{\alpha}\|_1, \tag{8.19b}$$

$$\text{subject to } |\alpha_i| \leq \frac{1}{2n\lambda}. \tag{8.19c}$$

  which has to be solved numerically. Luckily, the objective function is concave up and, thus, **it can be minimized efficiently**. Proving that the primal and dual optimization problems are equivalent is non-trivial.

- It can be shown that $\widehat{\alpha}_i \neq 0$ only if $|\widehat{y}(\boldsymbol{x}_i) - y_i| \geq \epsilon$. The training data point $\{\boldsymbol{x}_i, y_i\}$ is called **support vector**. The prediction in Equation 8.19a depends only on the support vectors. This reduces the computation needed for prediction (cf. kernel ridge regression). This is the reason of using the ϵ-insensitive loss function.
- **The larger ϵ is, the fewer support vectors** and the fewer the computations needed for prediction. So, ϵ has a regularizing effect in the feature space.
- Note that all the training data points are used during training (Equation 8.19b).
- Since many kernel functions (e.g., Gaussian kernel) do not include a constant offset term, an intercept $\widehat{\alpha}_0$ is sometimes included in Equation 8.19a.

Support Vector Regression

Example 8.3: Support vector regression and kernel ridge regression

We consider yet again the car stopping distance problem from Example 2.2in Figure 8.4. With the combined squared exponential and polynomial kernel from Example 8.2, $\lambda = 0.01$ and $\epsilon = 15$, we apply support vector regression to the data (red line). As a reference we also show the corresponding kernel ridge regression (blue line).

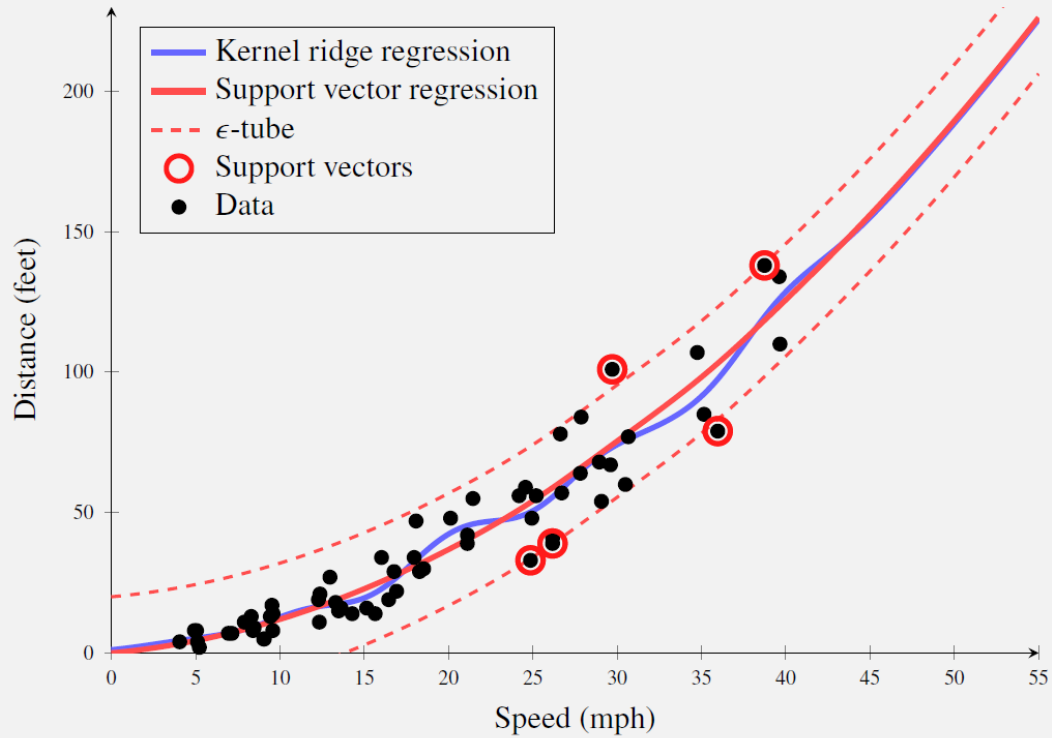

Fig. 8.4

In Figure 8.4 we have encircled (in red) all data points for which $\alpha_i \neq 0$, the so-called support vectors. We have also included the "ϵ-tube" ($\widehat{y}(\mathbf{x}) \pm \epsilon$; dotted lines), and we can confirm that all support vectors are located outside the "ϵ-tube". This is a direct effect of using the ϵ-insensitive loss, which explicitly encodes that the loss function for data points within ϵ from $\widehat{y}(\mathbf{x})$ is exactly zero. If choosing a smaller ϵ, we would have more support vectors, and vice versa. Another consequence of the sparsity of $\boldsymbol{\alpha}$ is that when computing a prediction (8.19a) with support vector regression, it is sufficient to do the computation using (in this case) only five data points. For kernel ridge regression, which does not have a sparse $\boldsymbol{\alpha}$, the prediction (8.14b) depends on all 62 data points.

Support Vector Classification

- Consider binary ($\{-1, +1\}$) classification with the classifier $\widehat{y}(\boldsymbol{x}) = sign\{f(\boldsymbol{x})\}$, where $f(\boldsymbol{x})$ is to be learned. The **margin** of the classifier for the point $\{\boldsymbol{x}, y\}$ is $y \cdot f(\boldsymbol{x})$. Consider the hinge loss function:

$$L(y \cdot f(\mathbf{x})) = \begin{cases} 1 - y \cdot f(\mathbf{x}) & \text{for } y \cdot f(\mathbf{x}) \leq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{5.17}$$

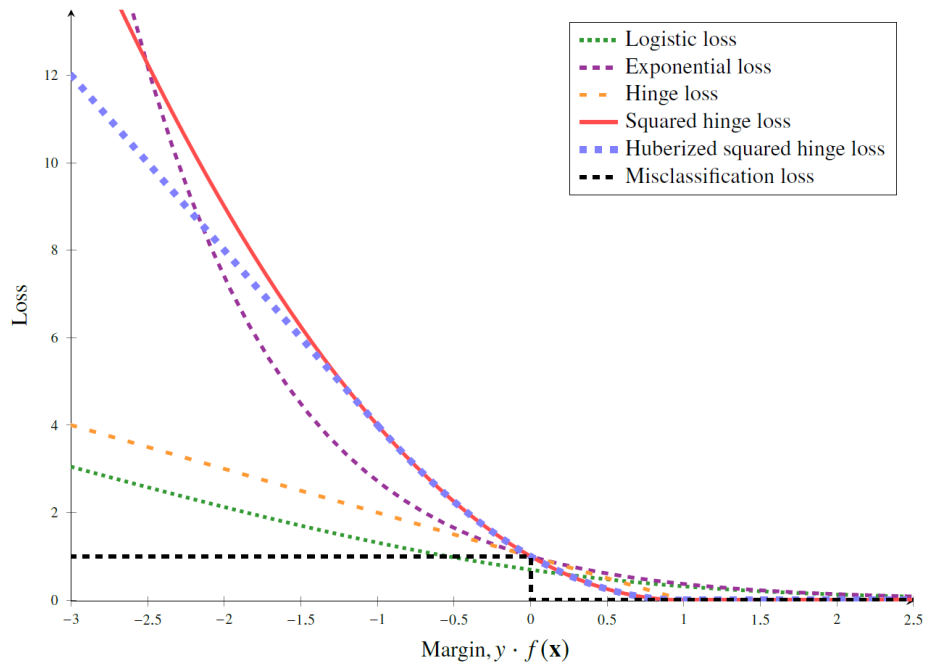

Figure 5.2: Comparison of some common loss functions for classification, plotted as a function of the margin.

Support Vector Classification

- In the primal formulation, we predict as

$$\widehat{y}(\mathbf{x}_\star) = \operatorname{sign}\left\{\boldsymbol{\theta}^\mathsf{T}\boldsymbol{\phi}(\mathbf{x}_\star)\right\}. \tag{8.32}$$

  where $\widehat{\boldsymbol{\theta}}$ is the solution to the optimization problem

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n} \max\left\{0, 1 - y_i\boldsymbol{\theta}^\mathsf{T}\boldsymbol{\phi}(\mathbf{x}_i)\right\} + \lambda\|\boldsymbol{\theta}\|_2^2. \tag{8.34}$$

  which has to be solved numerically (i.e., no closed-form solution exists).
- In the dual formulation, we predict as

$$\widehat{y}(\mathbf{x}_\star) = \operatorname{sign}\left(\widehat{\alpha}^\mathsf{T}\boldsymbol{K}(\mathbf{X}, \mathbf{x}_\star)\right) \tag{8.35c}$$

  where $\widehat{\boldsymbol{\alpha}}$ is the solution to the optimization problem

$$\widehat{\alpha} = \arg\min_{\alpha} \frac{1}{2}\alpha^\mathsf{T}\boldsymbol{K}(\mathbf{X}, \mathbf{X})\alpha - \alpha^\mathsf{T}\mathbf{y} \tag{8.35a}$$

$$\text{subject to } |\alpha_i| \leq \frac{1}{2n\lambda} \text{ and } 0 \leq \alpha_i y_i \tag{8.35b}$$

  which has to be solved numerically. Luckily, the objective function is concave up and, thus, **it can be minimized efficiently**. Proving that the primal and dual optimization problems are equivalent is non-trivial.
- It can be shown that $\widehat{\alpha}_i \neq 0$ only if the margin $y_i \cdot \widehat{\boldsymbol{\alpha}}^T\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x}_i) \leq 1$. The training data point $\{\boldsymbol{x}_i, y_i\}$ is called **support vector**. The prediction in Equation 8.35c depends only on the support vectors. This reduces the computation needed for prediction. This is the reason of using the hinge loss function.
- Note that all the training data points are used during training (Equation 8.35a).

Support Vector Classification

- It can be shown that **the support vectors are on the wrong side of the decision boundary or on the right side but too close to it**, exactly within $\frac{1}{||\widehat{\boldsymbol{\theta}}||_2} = \frac{1}{||\boldsymbol{\Phi}(\boldsymbol{X})^T \widehat{\boldsymbol{\alpha}}||_2}$ from it in the feature space.
- **The smaller** λ, the larger $\widehat{\boldsymbol{\theta}}$ and the smaller $\frac{1}{||\widehat{\boldsymbol{\theta}}||_2}$ and, thus, **the fewer support vectors** and the fewer the computations needed for prediction. So, $1/\lambda$ has a regularizing effect in the feature space (cf. λ has a regularizing effect in the input space, and the relation between both is given by the representer theorem).
- In the literature and software packages, it is common to use an equivalent formulation with $C = \frac{1}{2\lambda}$ or $C = \frac{1}{2n\lambda}$ as regularization hyperparameter.
- Since many kernel functions (e.g., Gaussian kernel) do not include a constant offset term, an intercept $\widehat{\alpha}_0$ is sometimes included in Equation 8.35c.

Support Vector Classification

Example 8.5: Support vector classification

We consider the binary classification problem with the data given in Figure 8.5 and apply support vector classification with the linear and the squared exponential kernel, respectively, in Figure 8.6. We mark the support vectors with yellow circles. For the linear kernel, the locations of the support vectors are either on the "wrong side" of the decision boundary or within $\frac{1}{\|\boldsymbol{\theta}\|_2}$ from it, marked with dashed white lines. As we decrease λ we allow for larger $\boldsymbol{\theta}$ and thereby a smaller band $\frac{1}{\|\boldsymbol{\theta}\|_2}$ and consequently fewer support vectors.

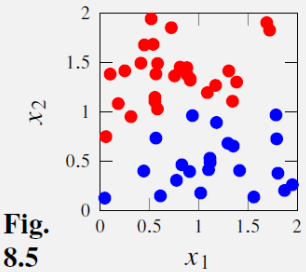

Fig. 8.5

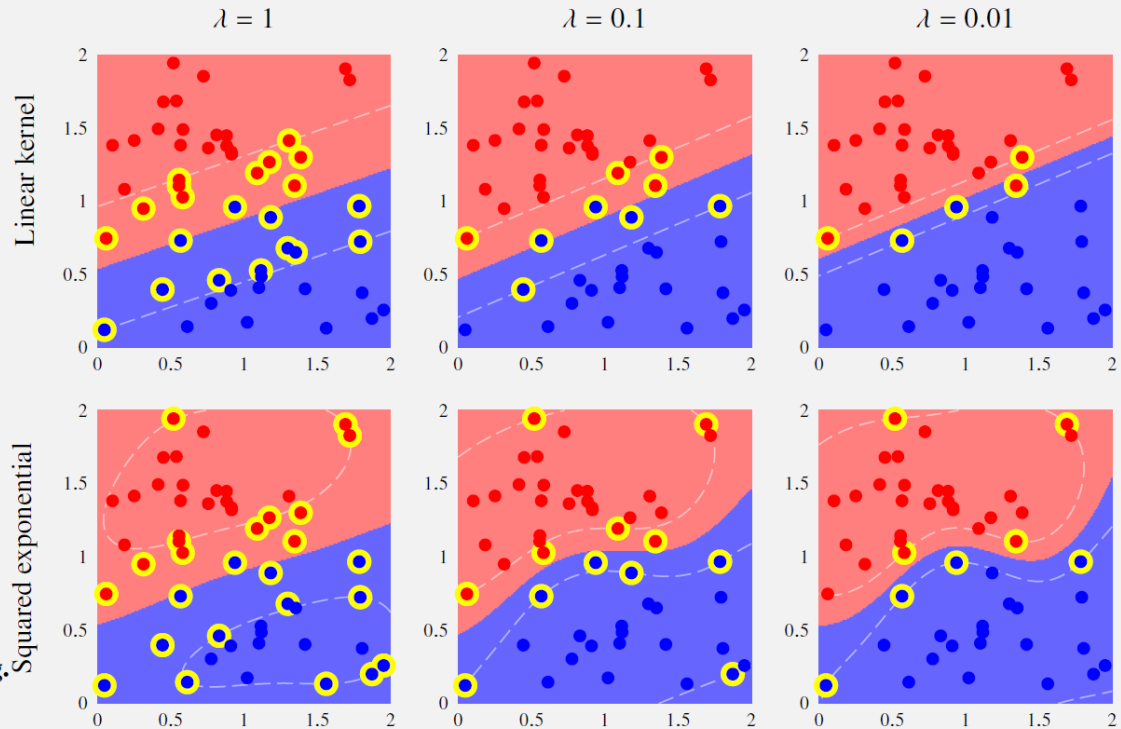

Fig. 8.6

When using the (indeed nonlinear) squared exponential kernel, the situation becomes somewhat harder to interpret. It still holds that the support vectors are either on the "wrong side" of the decision boundary or within $\frac{1}{\|\boldsymbol{\theta}\|_2}$ from it, but the distance is measured in the infinite dimensional $\boldsymbol{\phi}(\mathbf{x})$-space. Mapping this back to the original input space we observe this heavily nonlinear behavior. The meaning of the dashed white lines are the same as above. This also serves as a good illustration of the power in using kernels.

Summary

- Support Vector Regression
- Support Vector Classification