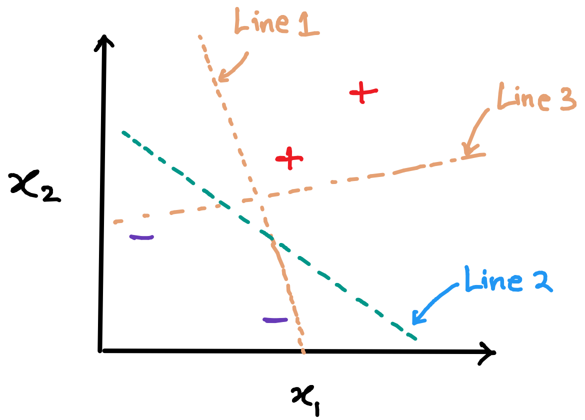
Maximum Margin Classifier and SVM

- In kernel methods, we have introduced two tools for regression → KRR, SVR

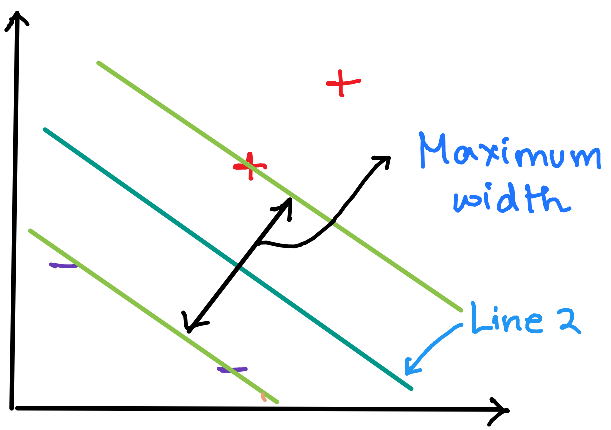
- What about use of kernels in classification?

- Reconsider binary classification with $y \in \{-1, 1\}$ and a linearly separable dataset

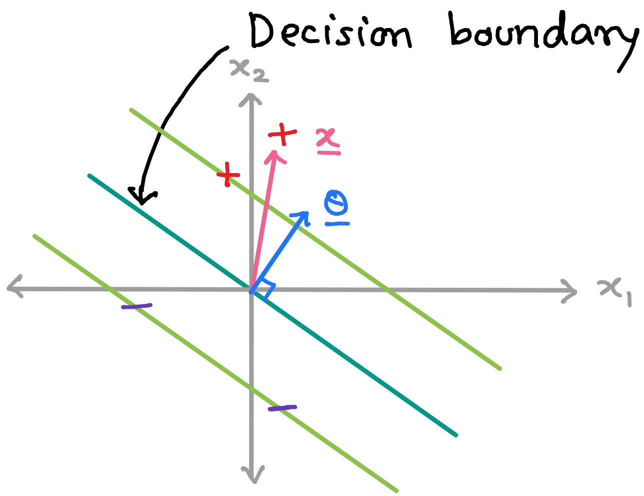


- For this dataset shown, how do you draw a line to separate the positive '+' data points from the negative '−' data points?

  - Several possible choices



- The best line (in 2D) that separates the two classes lies midway of the widest street that separates the '+' samples from '−' samples



- We know decision boundary of a classifier is where the prediction switches from one class to another and therefore at the decision boundary, we have

$$f_{\underline{\theta}}(\underline{x}) = 0$$

$$\boxed{\underline{\theta}^T \underline{x} = 0}$$ (for linear classifier)

Decision rule

- The parameter vector $\underline{\theta}$ must be mutually orthogonal to the decision boundary (i.e. the line in 2D)

- All '+' samples should have $\underline{\theta}^T \underline{x} > 0$ and '−' samples $\underline{\theta}^T \underline{x} < 0$

- However, to create the widest street, we will constrain that

$$\underline{\theta}^T \underline{x} \geq 1 \quad \text{for '+' samples}$$
$$\underline{\theta}^T \underline{x} \leq -1 \quad \text{for '−' samples}$$

- We can use the concept of margin ($y \cdot f_{\underline{\theta}}(\underline{x})$) to compactly represent

$$\underline{\theta}^T \underline{x} \geq 1 \quad \text{for '+' samples}$$

$$\underline{\theta}^T \underline{x} \leq -1 \quad \text{for '−' samples}$$

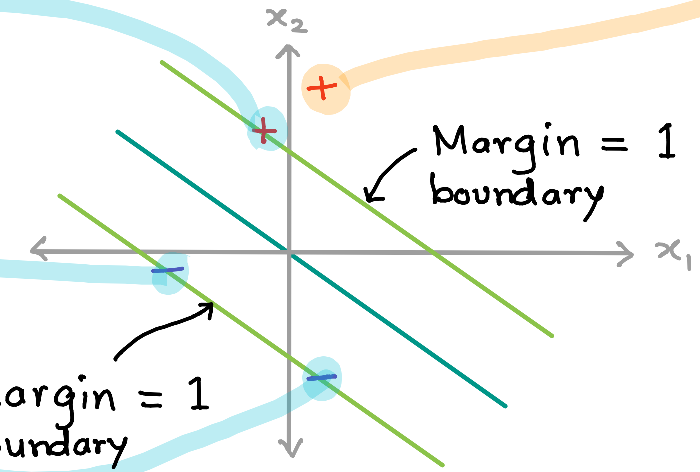
as

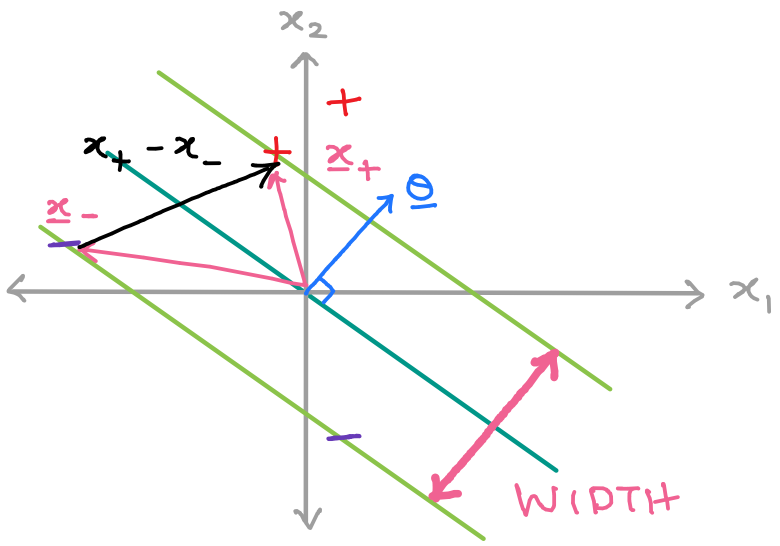
$$\overbrace{y \cdot \underline{\theta}^T \underline{x}}^{\text{Margin}} \geq 1 \quad \text{for all samples}$$

$$\Rightarrow \quad y \, \underline{\theta}^T \underline{x} - 1 \geq 0$$

- All samples that lie on the margin = 1 will satisfy the equality

$$y \, \underline{\theta}^T \underline{x} - 1 = 0$$





- We now want to calculate the width of the street and then maximize it

- Distance between '+' and '–' samples lying on the margin PROJECTED along the direction of $\underline{\theta}$

dot product

Normal direction

$$\text{WIDTH} = (\underline{x}_+ - \underline{x}_-) \cdot \frac{\underline{\theta}}{\|\underline{\theta}\|}$$

These datapoints lie on the margin

$$= (1 - (-1)) / \|\underline{\theta}\|$$

- Using the equality for datapoints lying on margin, we can use

$$y \cdot \underline{\theta}^T \underline{x} = 1$$

For '+' sample on margin $\Rightarrow$ $\underline{\theta}^T \underline{x}_+ = 1$

For '–' sample on margin $\Rightarrow$ $-\underline{\theta}^T \underline{x}_- = 1$

$\Rightarrow$ $\underline{\theta}^T \underline{x}_- = -1$

$$\boxed{\text{WIDTH} = \frac{2}{\|\underline{\theta}\|}}$$

- We want to maximize the width of the street

mathematically convenient

$$\max \frac{2}{\|\underline{\theta}\|} \equiv \max \frac{1}{\|\underline{\theta}\|} \equiv \min \|\underline{\theta}\| \equiv \min \|\underline{\theta}\|^2$$

subject to constraints that

$$y\, \underline{\theta}^T \underline{x} - 1 = 0$$

- Now, constrained optimization can be converted into unconstrained optimization using Lagrange multipliers

primal parameter

$$L(\underline{\theta}, \underline{\alpha}) = \|\underline{\theta}\|^2 - \sum_{i=1}^{N} \alpha_i \left(y^{(i)}\, \underline{\theta}^T \underline{x}^{(i)} - 1 \right)$$

dual parameter

$$= \underline{\theta}^T \underline{\theta} - \sum_{i=1}^{N} \alpha_i \left(y^{(i)}\, \underline{\theta}^T \underline{x}^{(i)} - 1 \right)$$

We minimize $L(\underline{\theta}, \underline{\alpha})$ w.r.t $\underline{\theta}$ and maximize it w.r.t $\underline{\alpha}$ to find the optimum

$$L(\underline{\theta}, \underline{\alpha}) = \underline{\theta}^T \underline{\theta} - \sum_{i=1}^{N} \alpha_i \left(y^{(i)} \underline{\theta}^T \underline{x}^{(i)} - 1 \right)$$

- $\frac{\partial L}{\partial \underline{\theta}} = 0 \Rightarrow 2\underline{\theta} - \sum_{i=1}^{N} \alpha_i \, y^{(i)} \underline{x}^{(i)} = 0$

$$\Rightarrow \boxed{\underline{\theta} = \frac{1}{2} \sum_{i=1}^{N} \alpha_i \, y^{(i)} \underline{x}^{(i)}}$$

[The primal parameter vector $\underline{\theta}$ turns out to be a linear combination of the data points weighted by the dual parameters]

- Plug $\underline{\theta}$ into the Lagrangian L, we get

$$L(\underline{\alpha}) = \frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i \, y^{(i)} \underline{x}^{(i)}\right) \cdot \frac{1}{2}\left(\sum_{j=1}^{N} \alpha_j \, y^{(j)} \underline{x}^{(j)}\right) - \sum_{i=1}^{N} \alpha_i \, y^{(i)} \underline{x}^{(i)} \cdot \frac{1}{2}\left(\sum_{j=1}^{N} \alpha_j \, y^{(j)} \underline{x}^{(j)}\right) - \sum_{i=1}^{N} \alpha_i$$

$$= \sum_{i=1} \alpha_i - \frac{1}{4} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \, y^{(i)} y^{(j)} \underbrace{\underline{x}^{(i)} \cdot \underline{x}^{(j)}}_{\text{dot product}}$$

Many times the dataset in the original input space may not be LINEARLY SEPARABLE

- The dataset might be linearly separable in the transformed feature space!

- Using transformed features $\underline{\phi}(\underline{x})$ instead of $\underline{x}$, the decision rule becomes

$$\underline{\theta}^T \underline{\phi}(\underline{x}) = 0 \qquad [\text{Decision boundary}]$$

- The margins are given by $y\,\underline{\theta}^T \underline{\phi}(\underline{x}) - 1 = 0$

- $$L(\underline{\theta}, \underline{\alpha}) = \|\underline{\theta}\|^2 - \sum_{i=1}^{N} \alpha_i \left(y^{(i)} \underline{\theta}^T \underline{\phi}(\underline{x}^{(i)}) - 1 \right)$$

- $$\underline{\theta} = \frac{1}{2} \sum_{i=1}^{N} \alpha_i\, y^{(i)} \underline{\phi}(\underline{x}^{(i)})$$

- $$L(\underline{\alpha}) = \sum_{i=1} \alpha_i - \frac{1}{4} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j\, y^{(i)} y^{(j)} \underbrace{\underline{\phi}(\underline{x}^{(i)})^T \underline{\phi}(\underline{x}^{(j)})}_{\substack{K(\underline{x}^{(i)},\, \underline{x}^{(j)}) \\ \text{PSD kernel}}}$$

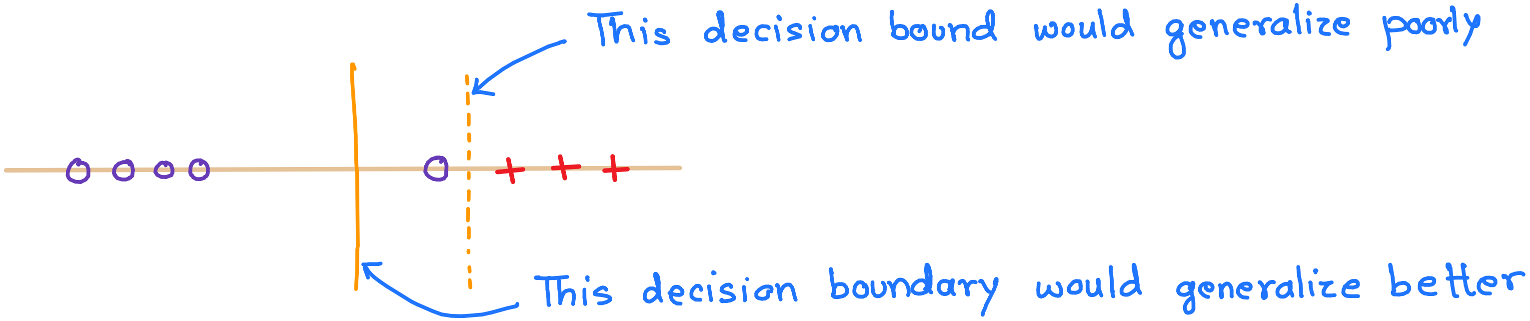
$\boxed{\text{SVM}}$

By maximizing $L(\underline{\alpha})$ one would get an expression for the dual parameter $\underline{\alpha}$ in terms of the kernel K

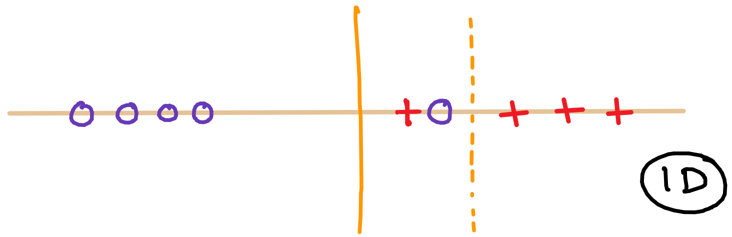
- In order to classify test input $\underline{x}^*$, one needs to evaluate the sign of $y(\underline{x}^*)$

$$y(\underline{x}^*) = \text{sign}\left\{ \underline{\theta}^T \underline{\phi}(\underline{x}^*) \right\} = \text{sign}\left\{ \sum_{i=1}^{N} \alpha_i\, y^{(i)} K(\underline{x}^{(i)}, \underline{x}^*) \right\}$$

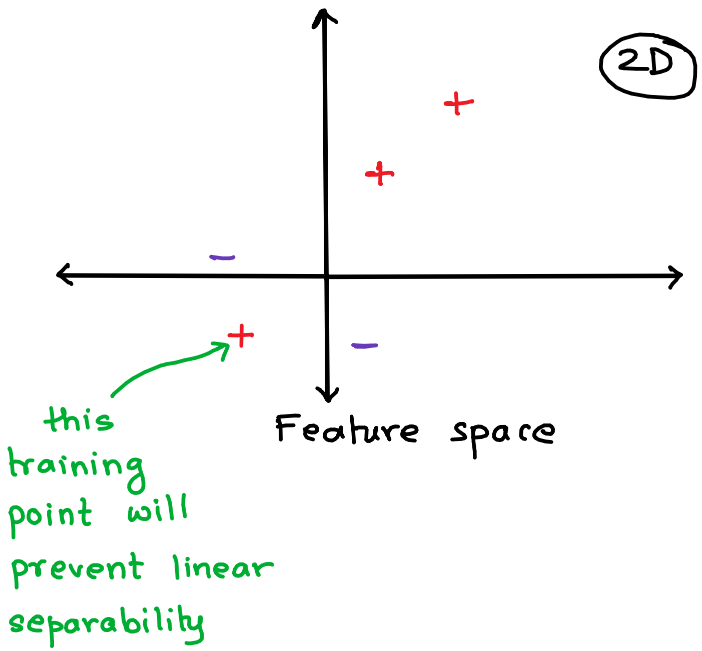
- In practice, exact separation of training data may lead to poor generalization



- Even worse, there might be overlap of classes



  - We therefore need a way to modify the SVM so as to allow some of the training points to be misclassified

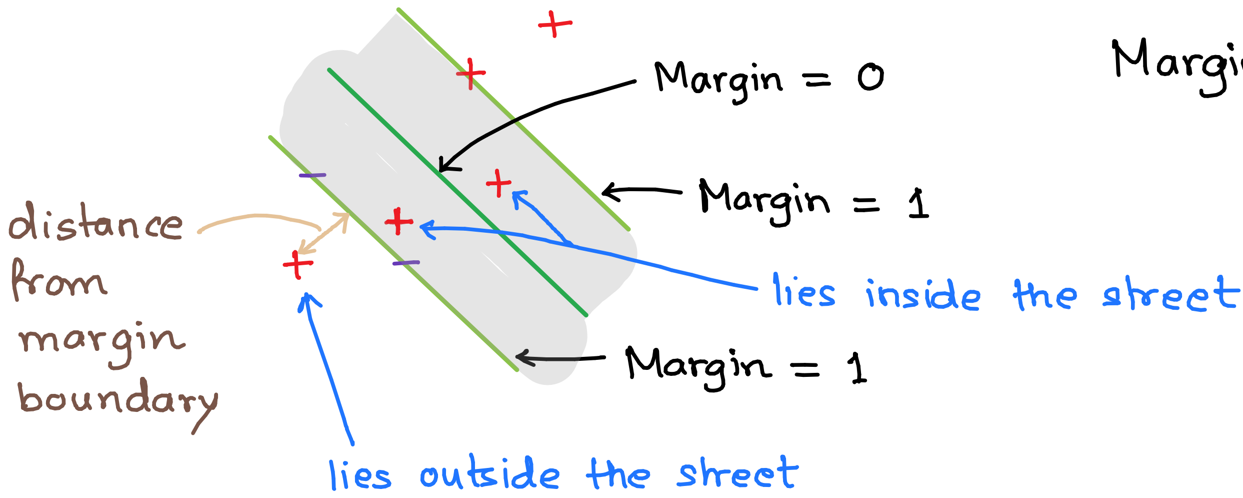


  - The current loss function can be expressed in the following equivalent form:

$$\sum_{i=1}^{N} E_\infty\left(y^{(i)}\underline{\theta}^T\underline{\phi}(\underline{x}^{(i)})-1\right)+\lambda\|\underline{\theta}\|_2^2$$

$$E_\infty(u)=\begin{cases}0 & \text{if } u \geq 0\\ \infty & \text{otherwise}\end{cases}$$

- We now modify the loss function such that some data points are allowed to be on the 'wrong side' of the street
  - Introduce a penalty that increases with distance from the margin boundary

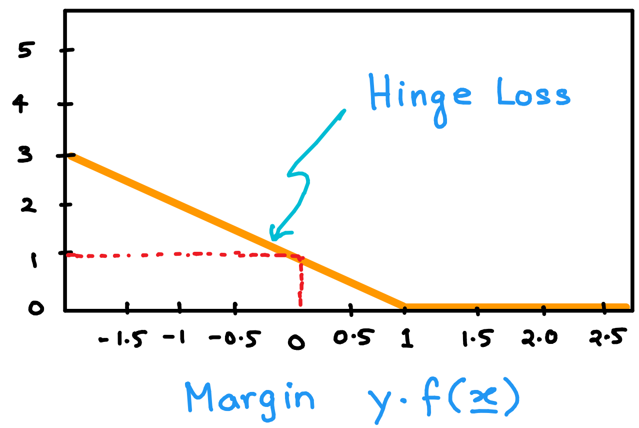


Margin $\triangleq y\, \underline{\theta}^T \underline{\phi}(\underline{x})$

- It is convenient to make the penalty a linear function of the distance

$$L(\underline{\theta}) = \begin{cases} 1 - y \cdot \underline{\theta}^T \underline{\phi}(\underline{x}) & \text{for } y\, \underline{\theta}^T \underline{\phi}(\underline{x}) < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$= \max \{ 0,\ 1 - y\, \underline{\theta}^T \underline{\phi}(\underline{x}) \}$$



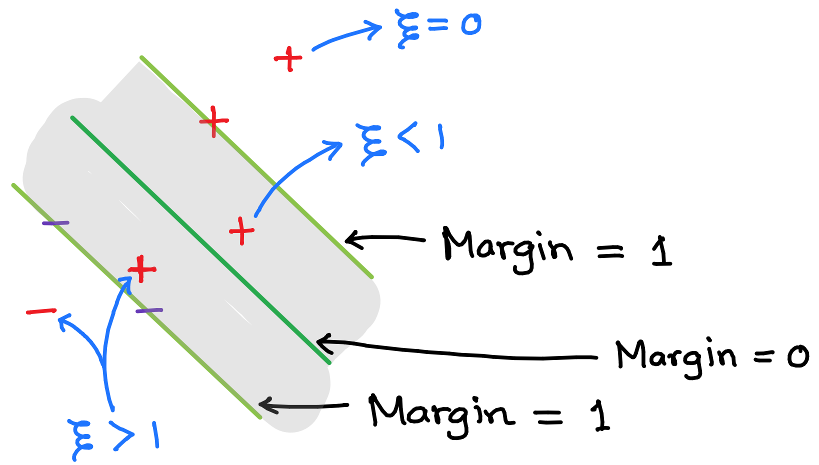
- Primal formulation with $\underline{\theta}$

non-differentiable due to max fun.

$$\hat{\underline{\theta}} = \underset{\underline{\theta}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{n} \max\left\{0, 1 - y^{(i)} \underline{\theta}^T \underline{\phi}(\underline{x}^{(i)})\right\} + \lambda \|\underline{\theta}\|_2^2$$

- An easier way to tackle this optimization is by introducing slack variables
  - We introduce a slack variable ξ_i for each datapoint $(\underline{x}^{(i)}, y^{(i)})$
  - By definition, slack variables $\xi_i \geq 0$
  - To replace the max function in the hinge loss with slack variables, constraints are shifted on to the slack variables

$$\xi_i \geq 1 - y^{(i)} \underline{\theta}^T \underline{\phi}(\underline{x}^{(i)})$$



- Equivalent optimization

$$\hat{\underline{\theta}} = \underset{\underline{\theta}}{\operatorname{argmin}} \ \frac{1}{N}\sum_{i=1}^{n} \max\left\{0,\ 1 - y^{(i)}\,\underline{\theta}^T \underline{\phi}(\underline{x}^{(i)})\right\} + \lambda \|\underline{\theta}\|_2^2$$

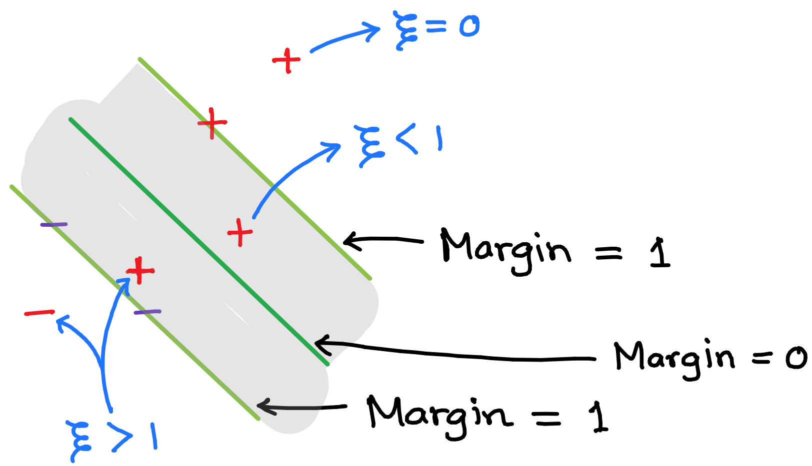
$$\underset{\underline{\theta},\,\underline{\xi}}{\text{minimize}} \ \frac{1}{N}\sum_{i=1}^{N} \xi_i + \lambda\|\underline{\theta}\|_2^2$$

$$\text{subject to} \quad \left.\begin{aligned} &\xi_i \geq 0 \\ &\xi_i \geq 1 - y^{(i)}\,\underline{\theta}^T\underline{\phi}(\underline{x}^{(i)}) \end{aligned}\right\} \ \forall\ i = 1, 2, \ldots, N$$

- Datapoints lie on the margin and on the correct side of margin → $\xi = 0$ (Correctly classified)

- Points that lie inside the street (margin) and on correct side of decision boundary → $0 < \xi \leq 1$ (Correctly classified)

- Points on wrong side of decision boundary → $\xi > 1$ (Incorrectly classified)



- The goal is to now maximize the width of the street while softly penalizing points that lie on the wrong side of the margin boundary

$$\underset{\underline{\Theta},\, \underline{\xi}}{\text{minimize}} \quad \frac{1}{N}\sum_{i=1}^{N} \xi_i + \lambda \|\Theta\|_2^2$$

$$\text{subject to} \quad \left.\begin{aligned} &\xi_i \geq 0 \\ &\xi_i \geq 1 - y^{(i)} \underline{\Theta}^T \underline{\phi}(\underline{x}^{(i)}) \end{aligned}\right\} \quad \forall\, i = 1, 2, \cdots, N$$

- Regularization parameter λ controls the trade-of between the slack variable penalty $\frac{1}{N}\sum_{i=1}^{N}\xi_i$ and the margin width given by $\frac{1}{\|\Theta\|_2}$

  - $\lambda > 0$
  - $\lambda \to \infty$ will get us back the SVM for linearly separable case

- Lets now minimize the new equivalent objective function using Lagrange multipliers (i.e. constrained optimization to unconstrained optimization)

- The Lagrangian associated with soft margin SVM

$$L(\underline{\theta}, \underline{\xi}, \underline{\beta}, \underline{\gamma}) = \frac{1}{N}\sum_{i=1}^{N} \xi_i + \lambda \|\underline{\theta}\|_2^2 - \sum_{i=1}^{N} \beta_i \left(\xi_i + y^{(i)} \underline{\theta}^T \underline{\phi}(\underline{x}^{(i)}) - 1\right) - \sum_{i=1}^{N} \gamma_i \xi_i$$

Lagrange multipliers

$\underline{\beta} \geqslant 0$

$\underline{\gamma} \geqslant 0$

- We minimize L w.r.t $\underline{\theta}$ and $\underline{\xi}$, and maximize w.r.t. $\underline{\beta}$ and $\underline{\gamma}$

- $$\frac{\partial L}{\partial \underline{\theta}} = 0 \quad \Rightarrow \quad \underline{\theta} = \frac{1}{2\lambda}\sum_{i=1}^{N} y^{(i)} \beta_i \, \underline{\phi}(\underline{x}^{(i)}) \quad \text{—— (1)}$$

- $$\frac{\partial L}{\partial \xi_i} = 0 \quad \Rightarrow \quad \gamma_i = \frac{1}{N} - \beta_i \quad \text{—— (2)}$$

- Inserting (1) and (2) in the Lagrangian and eliminating $\underline{\theta}$, ξ_i, we get

$$\tilde{L}(\underline{\beta}) = \sum_{i=1}^{N} \frac{\beta_i}{2\lambda} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y^{(i)} y^{(j)} \frac{\beta_i \beta_j}{4\lambda^2} \underline{\phi}^T(\underline{x}^{(i)}) \, \underline{\phi}(\underline{x}^{(j)})$$

- We need to maximize $\tilde{L}(\underline{\beta})$ w.r.t $\underline{\beta}$ to get the solution variable $\underline{\beta}$

$$\tilde{L}(\underline{\beta}) = \sum_{i=1}^{N} \frac{\beta_i}{2\lambda} - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y^{(i)} y^{(j)} \frac{\beta_i \beta_j}{4\lambda^2} \underline{\phi}^T(\underline{x}^{(i)}) \underline{\phi}(\underline{x}^{(j)})$$

- However, we also have constraints here:
  - We note that $\beta_i \geq 0$, $\gamma_i \geq 0$ (since they are Lagrange multipliers)
  - However, $\gamma_i = \frac{1}{N} - \beta_i$ implies $\beta_i \leq 1/N$. Thus, $0 \leq \beta_i \leq \frac{1}{N}$

- Now setting $\alpha_i = \dfrac{y^{(i)} \beta_i}{2\lambda}$, we see that the equivalent minimization:

$$\underset{\underline{\alpha}}{\text{minimize}} \quad \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \underbrace{\underline{\phi}^T(\underline{x}^{(i)}) \underline{\phi}(\underline{x}^{(j)})}_{K(\underline{x}^{(i)}, \underline{x}^{(j)})} - \sum_{i=1}^{N} y^{(i)} \alpha_i$$

subject to $\alpha_i y^{(i)} \geq 0$ and $|\alpha_i| \leq \dfrac{1}{2N\lambda}$

$$\underset{\underline{\alpha}}{\text{minimize}} \quad \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \underbrace{\underline{\phi}^T(\underline{x}^{(i)}) \, \underline{\phi}(\underline{x}^{(j)})}_{K(\underline{x}^{(i)}, \underline{x}^{(j)})} - \sum_{i=1}^{N} y^{(i)} \alpha_i$$

$$\text{subject to} \quad \alpha_i y^{(i)} \geq 0 \quad \text{and} \quad |\alpha_i| \leq \frac{1}{2\lambda N}$$

- Using kernels, we can write in matrix notation the minimization problem:

$\boxed{\text{Dual formulation}}$

$$\underset{\underline{\alpha}}{\text{minimize}} \quad \frac{1}{2} \underline{\alpha}^T \underline{\underline{K}}(\underline{\underline{X}}, \underline{\underline{X}}) \underline{\alpha} - \underline{\alpha}^T \underline{y}$$

Dual parameter $\rightarrow \underline{\alpha}$

$$\text{subject to} \quad \alpha_i y^{(i)} \geq 0$$
$$|\alpha_i| \leq \frac{1}{2\lambda N}$$

No closed-form solution; you need an optimizer to find solution numerically

- Prediction : $\hat{y}(\underline{x}^*) = \text{sign}\left(\hat{\underline{\alpha}}^T \underline{K}(\underline{\underline{X}}, \underline{x}^*) \right)$

- Dual formulation with $\underline{\alpha}$

$$\underset{\underline{\alpha}}{\text{minimize}} \quad \frac{1}{2} \underline{\alpha}^T \underline{\underline{K}}(\underline{\underline{X}}, \underline{\underline{X}}) \underline{\alpha} - \underline{\alpha}^T \underline{y}$$

$$\text{subject to} \quad \alpha_i y^{(i)} \geq 0$$

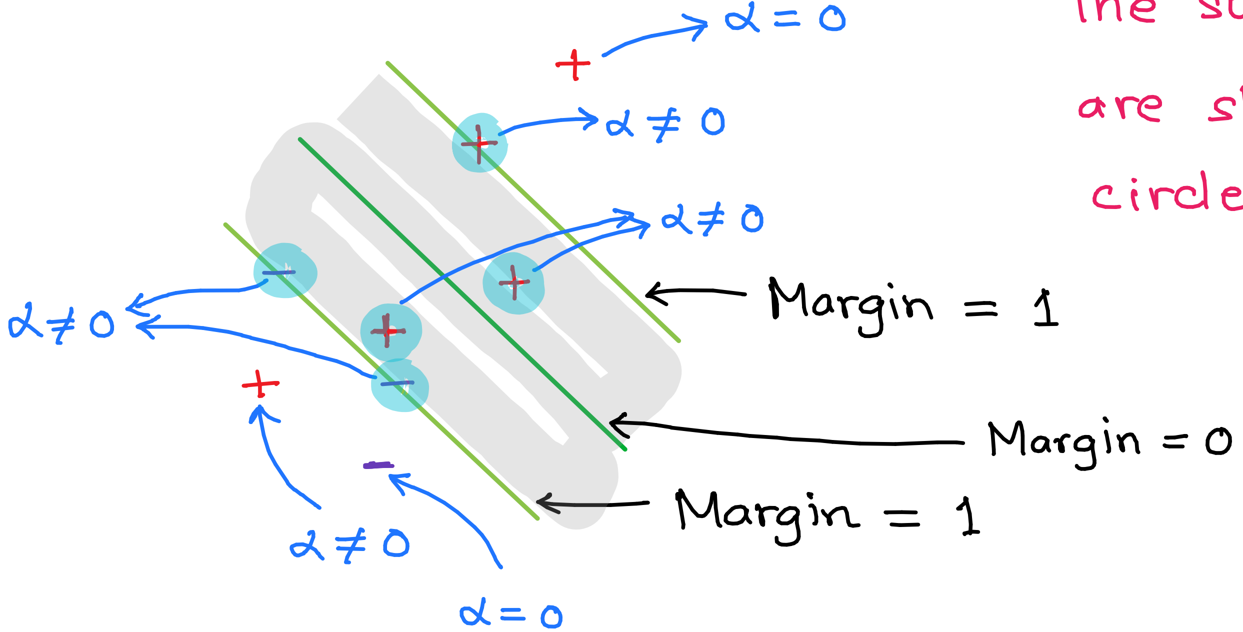
$$|\alpha_i| \leq \frac{1}{2\lambda N}$$

- The interesting point in SVM is that the dual parameter $\underline{\alpha}$ turns out to be sparse

- Similar to SVR, prediction $\hat{y}(\underline{x}^*)$ depends only on a subset of training points. Note, however, all training points are needed during training

- The support vector property is due to the fact that the hinge loss function is exactly zero when the margin $y\,\underline{\theta}^T \underline{\phi}(\underline{x}) \geq 1$

- The dual parameter $\alpha_i \neq 0$ only if the margin for $\underline{x}^{(i)}$ is ≤ 1



The support vectors are shown in filled circles

Support Vector Classification

Training

Data : Training data $\mathcal{T} = \{ \underline{x}^{(i)}, y^{(i)} \}_{i=1}^{N}$, choice of kernel

Result : Learned dual parameters $\hat{\underline{\alpha}}$

Procedure : Compute $\hat{\underline{\alpha}}$ by numerically minimizing

$$\underset{\underline{\alpha}}{\text{minimize}} \quad \frac{1}{2} \underline{\alpha}^T \underline{\underline{K}}(\underline{\underline{X}}, \underline{\underline{X}}) \underline{\alpha} - \underline{\alpha}^T \underline{y}$$

$$\text{subject to} \quad \alpha_i y^{(i)} \geq 0$$

$$|\alpha_i| \leq \frac{1}{2\lambda N}$$

Prediction

Data : Learned parameters $\hat{\underline{\alpha}}$ and test input $\underline{x}^*$

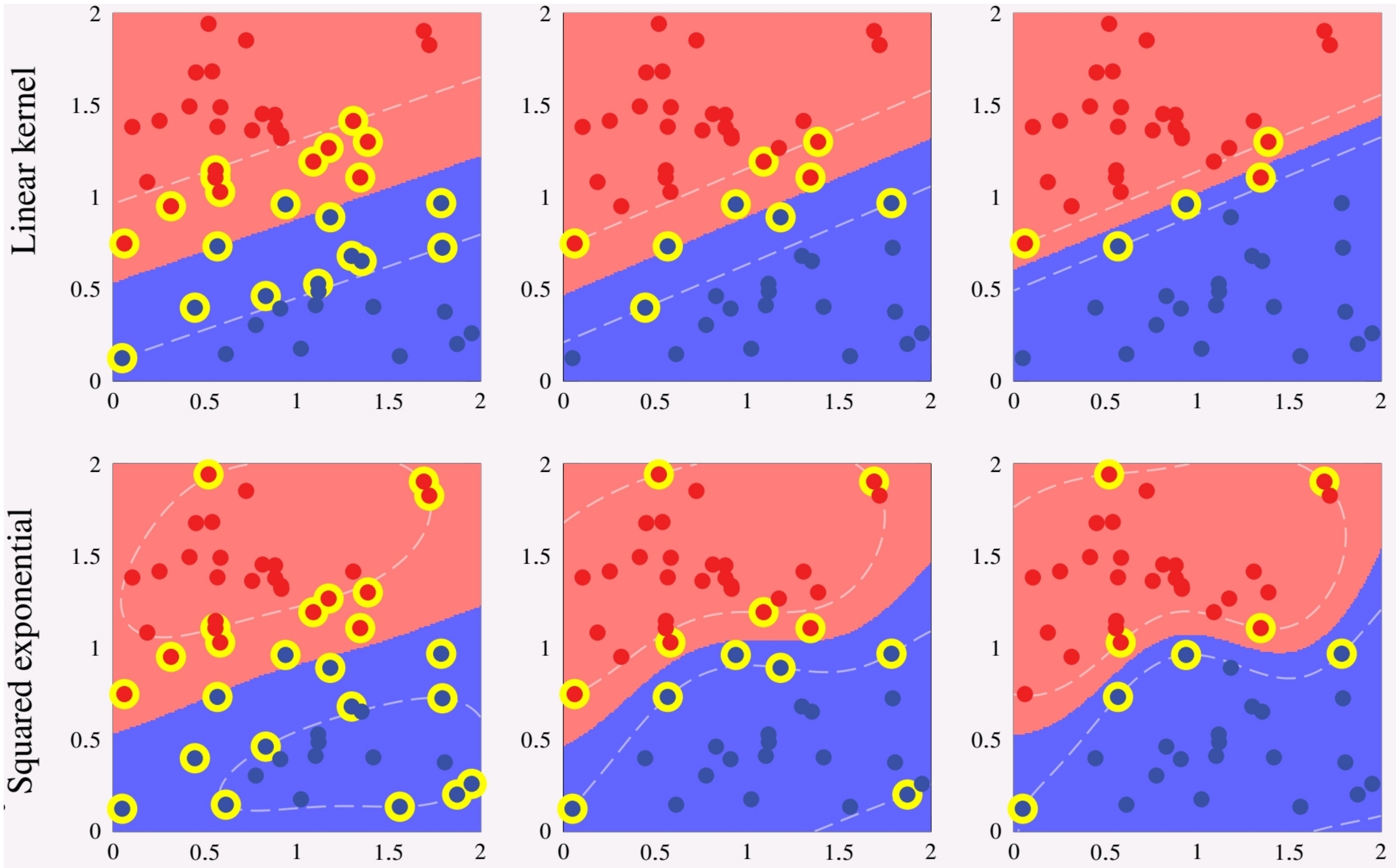
Result : Prediction $y(\underline{x}^*) = \text{sign}\left(\hat{\underline{\alpha}}^T \underline{K}(\underline{X}, \underline{x}^*) \right)$

Example of binary classification with SVC

- Linear kernel
- Squared exponential kernel

} Used kernels

$\lambda = 1$ $\quad\quad$ $\lambda = 0.1$ $\quad\quad$ $\lambda = 0.01$



As you decrease λ, we allow for larger $\underline{\theta}$, which means a narrower street, and fewer support vectors