Optimization
Machine learning

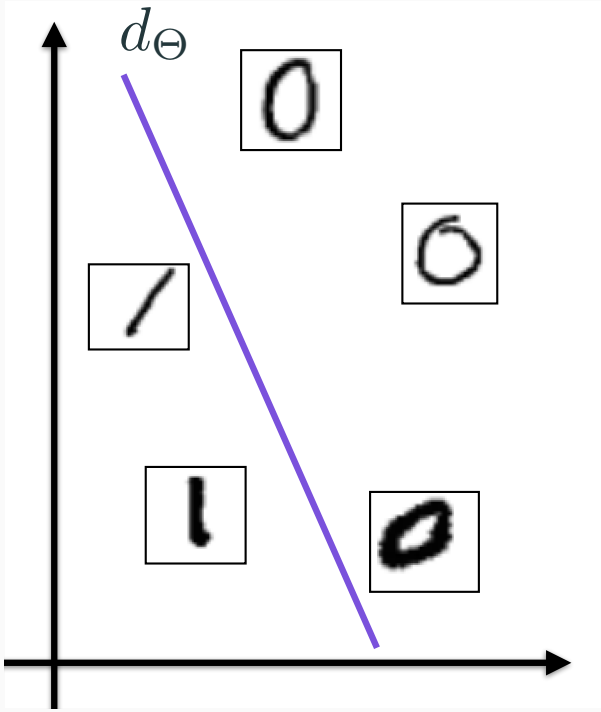
Nelly Pustelnik

CNRS, Laboratoire de Physique de l'ENS de Lyon, France

Example: Supervised learning

- **Database**: $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{H} \times \mathcal{G} \;\middle|\; \ell \in \{1, \ldots, L\}\right\}$
  e.g. $\mathrm{u}_\ell \in \underbrace{\mathbb{R}^N}_{\mathcal{H}}$ image and $\mathrm{c}_\ell \in \underbrace{\{-1, +1\}}_{\mathcal{G}}$ classe

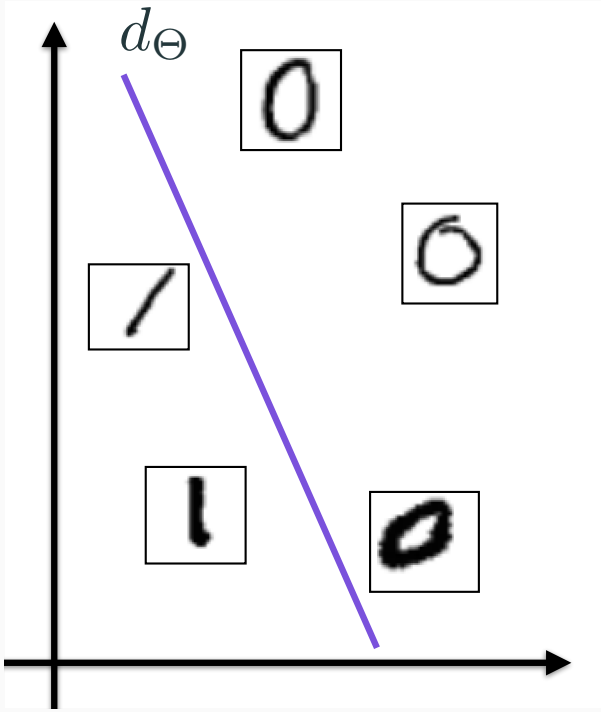
- **Goal**: Learn a prediction function $d_\Theta : \mathcal{H} \to \mathcal{G}$



Example: Supervised learning

- **Database**: $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{H} \times \mathcal{G} \;\middle|\; \ell \in \{1, \ldots, L\}\right\}$
  e.g. $\mathrm{u}_\ell \in \underbrace{\mathbb{R}^N}_{\mathcal{H}}$ image and $\mathrm{c}_\ell \in \underbrace{\{-1, +1\}}_{\mathcal{G}}$ classe

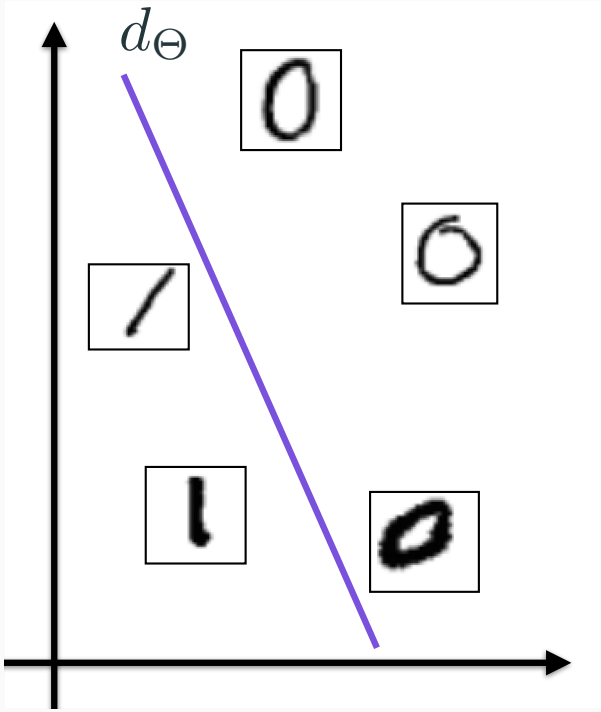
- **Goal**: Learn a prediction function $d_\Theta : \mathcal{H} \to \mathcal{G}$



Example: Supervised learning

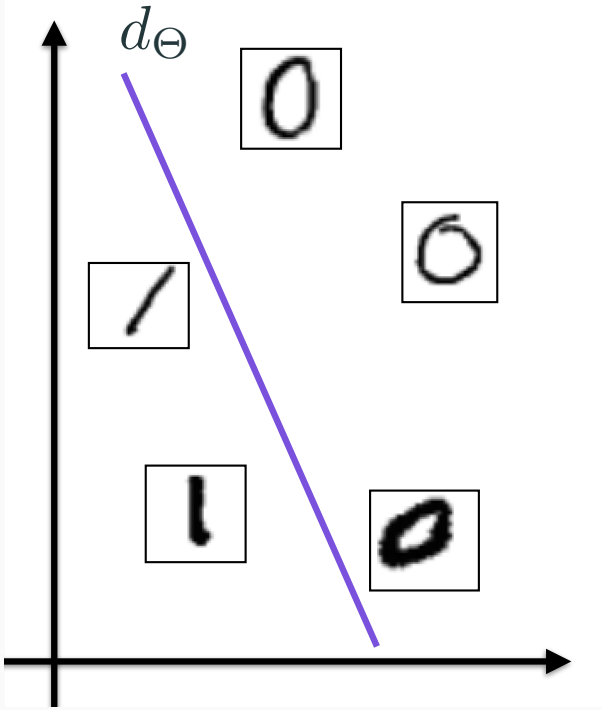
- **Database**: $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{H} \times \mathcal{G} \;\middle|\; \ell \in \{1, \ldots, L\}\right\}$
  e.g. $\mathrm{u}_\ell \in \underbrace{\mathbb{R}^N}_{\mathcal{H}}$ image and $\mathrm{c}_\ell \in \underbrace{\{-1, +1\}}_{\mathcal{G}}$ classe

- **Goal**: Learn a prediction function $d_\Theta : \mathcal{H} \to \mathcal{G}$



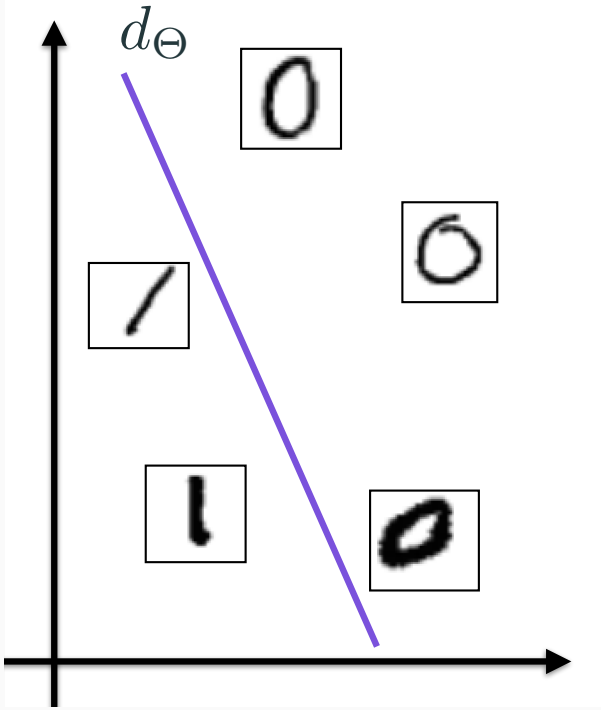
Example: Supervised learning

- **Database**: $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{H} \times \mathcal{G} \;\middle|\; \ell \in \{1, \ldots, L\}\right\}$
- **Goal**: Learn a prediction function $d_\Theta : \mathcal{H} \to \mathcal{G}$
- **Typical choices for $\mathcal{H}$**:
  - $\mathcal{H} = \mathbb{R}^N$ for image of size $N = N_1 \times N_2$;
  - $\mathcal{H} = \mathbb{R}^{N \times M}$ for multivariate images with N samples and M components;



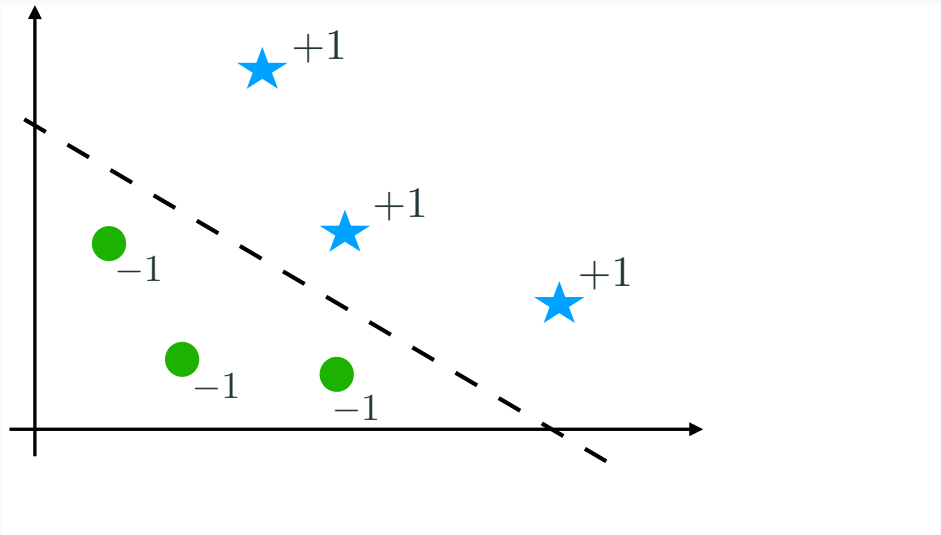
Example: Supervised learning

- **Database**: $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{H} \times \mathcal{G} \;\middle|\; \ell \in \{1, \ldots, L\}\right\}$
- **Goal**: Learn a prediction function $d_\Theta : \mathcal{H} \to \mathcal{G}$
- **Typical choices for $\mathcal{H}$**:
  - $\mathcal{H} = \mathbb{R}^N$ for image of size $N = N_1 \times N_2$;
  - $\mathcal{H} = \mathbb{R}^{N \times M}$ for multivariate images with N samples and M components;
- **Typical choices for $\mathcal{G}$**:
  - $\mathcal{G} = \{-1, +1\}$ for binary classification;
  - $\mathcal{G} = \{1, \ldots, K\}$ for multiclass classification;
  - $\mathcal{G} = \mathbb{R}$ for regression;
  - $\mathcal{G} = \mathbb{R}^K$ for multivariate regression;



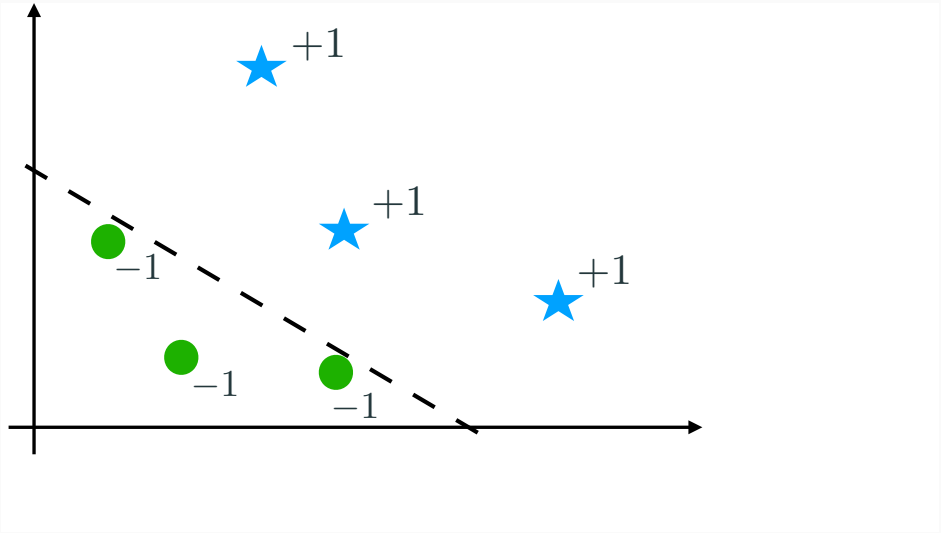
Support-vector machine (SVM)

- SVM – **Vladimir Vapnik** (early 90's) – Supervised classification.
- Example : dataset $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathbb{R}^2 \times \{-1; +1\} \;\middle|\; \ell \in \{1, \ldots, 6\}\right\}$.



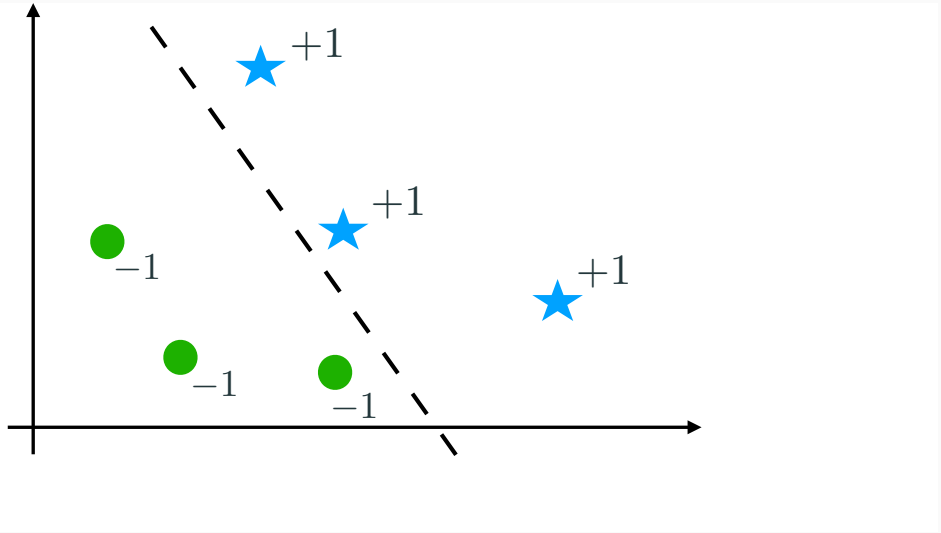
Support-vector machine (SVM)

- SVM – **Vladimir Vapnik** (early 90's) – Supervised classification.
- Example : dataset $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathbb{R}^2 \times \{-1; +1\} \;\middle|\; \ell \in \{1, \ldots, 6\}\right\}$.
- Many hyperplanes that might classify the data.



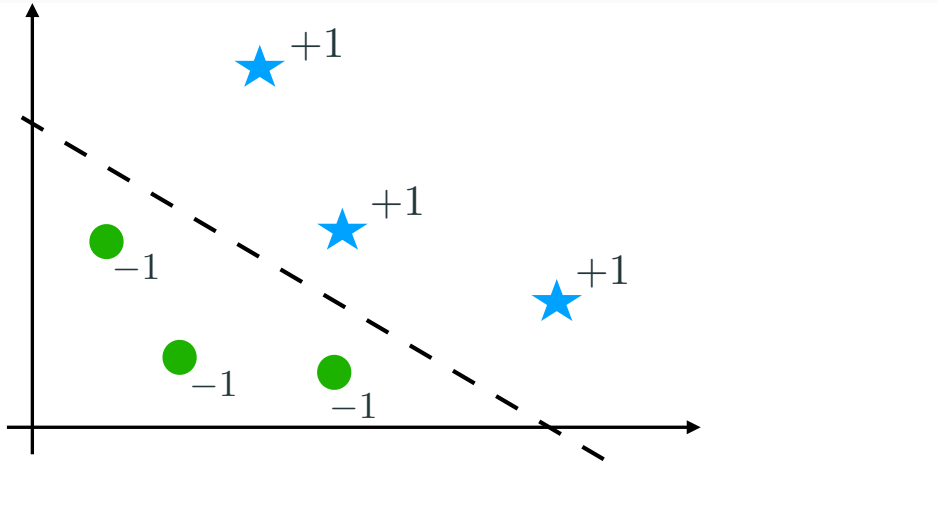
Support-vector machine (SVM)

- SVM – **Vladimir Vapnik** (early 90's) – Supervised classification.
- Example : dataset $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathbb{R}^2 \times \{-1; +1\} \;\middle|\; \ell \in \{1, \ldots, 6\}\right\}$.
- Many hyperplanes that might classify the data.



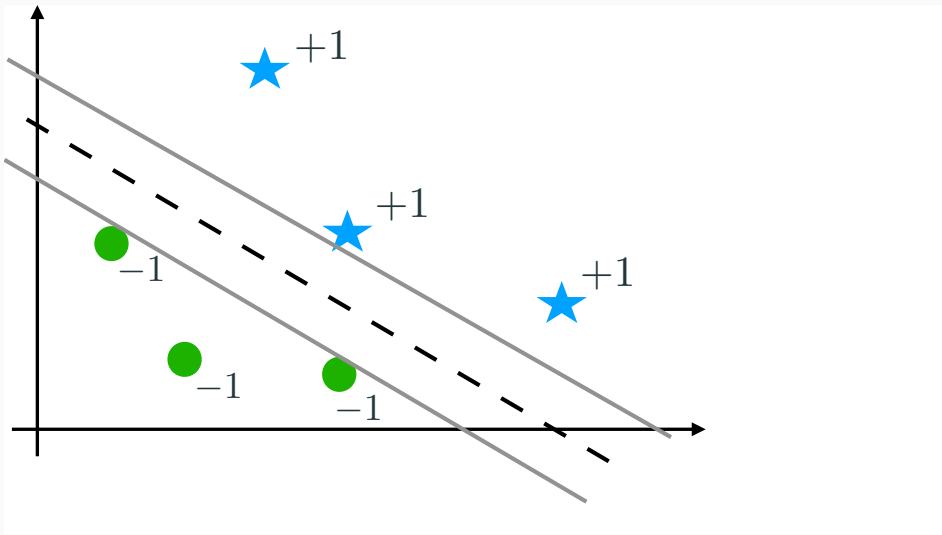
Support-vector machine (SVM)

- SVM – **Vladimir Vapnik** (early 90's) – Supervised classification.
- Example : dataset $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathbb{R}^2 \times \{-1; +1\} \;\middle|\; \ell \in \{1, \ldots, 6\}\right\}$.
- Many hyperplanes that might classify the data.



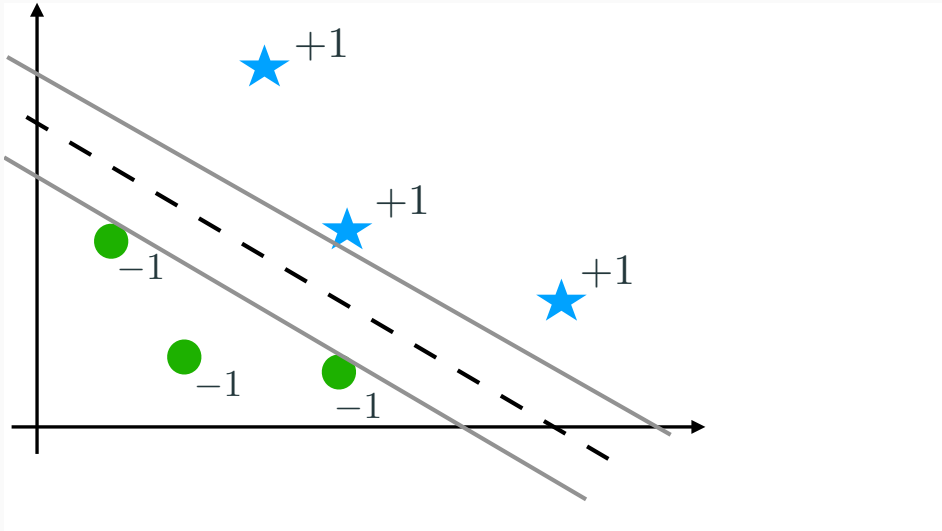
Support-vector machine (SVM)

- SVM – **Vladimir Vapnik** (early 90's) – Supervised classification.
- Example : dataset $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathbb{R}^2 \times \{-1; +1\} \;\middle|\; \ell \in \{1, \ldots, 6\}\right\}$.
- Many hyperplanes that might classify the data.
- **Best hyperplane = largest margin between the two classes**.



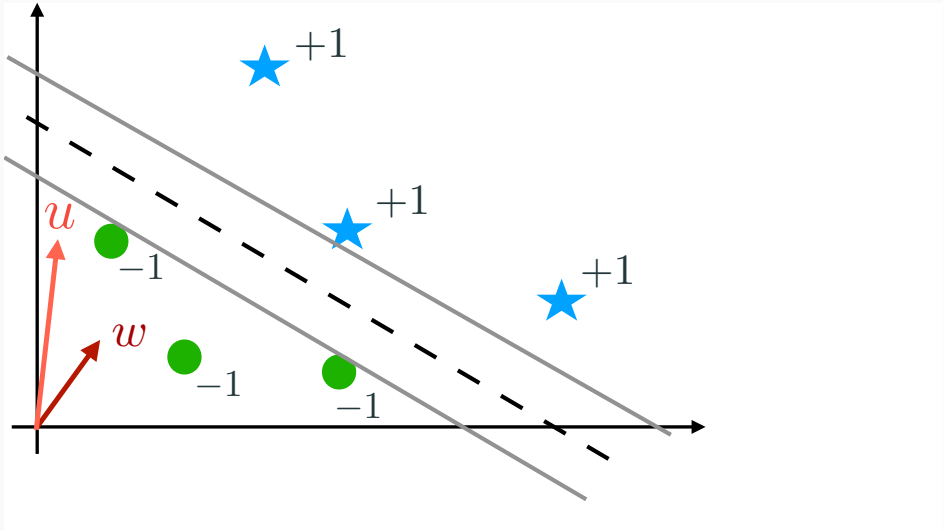
Support-vector machine (SVM)

- **Best hyperplane = largest margin between the two classes.**



Support-vector machine (SVM)

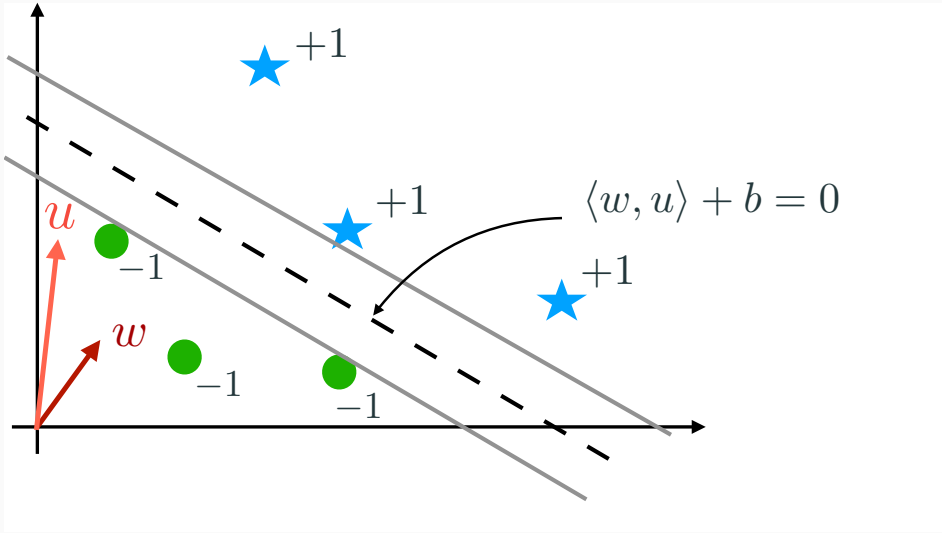
- **Best hyperplane = largest margin between the two classes**.



- **Decision rule**: $\langle \mathrm{w}, \mathrm{u} \rangle \geq \mathrm{cte}$

Support-vector machine (SVM)

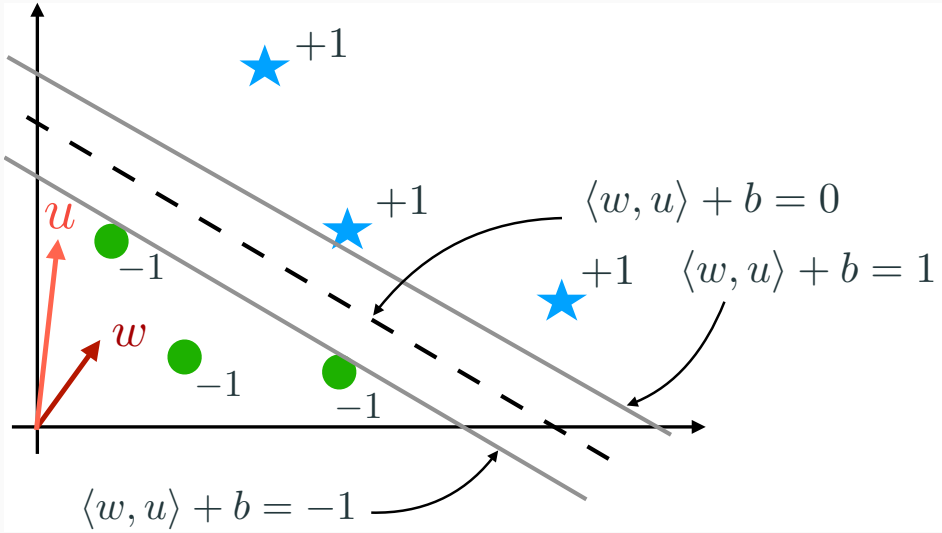
- **Best hyperplane = largest margin between the two classes**.



- **Decision rule**: $\langle \mathrm{w}, \mathrm{u} \rangle \geq \mathrm{cte} \quad \Leftrightarrow \quad \boxed{\langle \mathrm{w}, \mathrm{u} \rangle + \mathrm{b} \geq 0}$

Support-vector machine (SVM)

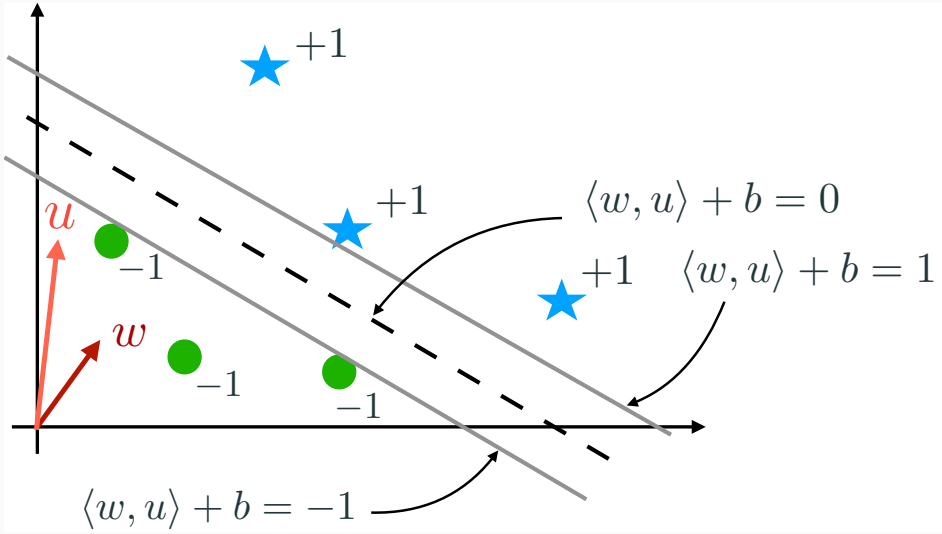
- **Best hyperplane = largest margin between the two classes**.



- **Decision rule**: $\langle \mathrm{w}, \mathrm{u} \rangle \geq \mathrm{cte} \quad \Leftrightarrow \quad \boxed{\langle \mathrm{w}, \mathrm{u} \rangle + \mathrm{b} \geq 0}$
- Deal with margin: $\mathrm{c}\Big(\langle \mathrm{w}, \mathrm{u} \rangle + \mathrm{b}\Big) \geq 1 \quad \Rightarrow \begin{cases} \langle \mathrm{w}, \mathrm{u}_+ \rangle + \mathrm{b} \geq 1 \\ \langle \mathrm{w}, \mathrm{u}_- \rangle + \mathrm{b} \leq -1 \end{cases}$

Support-vector machine (SVM)

- **Best hyperplane = largest margin between the two classes**.

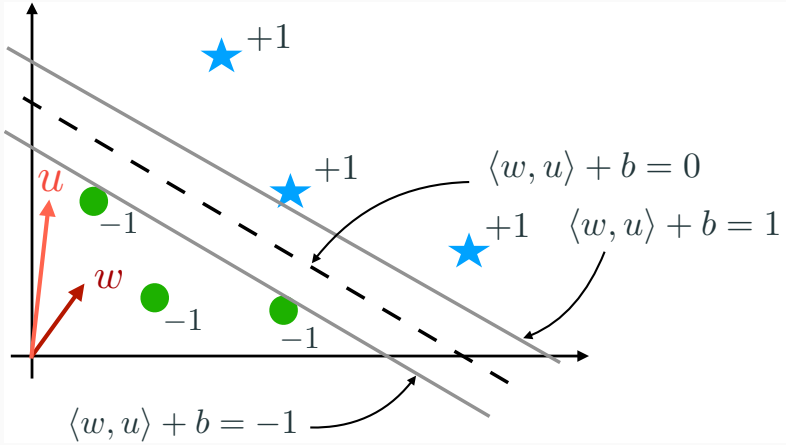


- **Decision rule**: $\langle \mathrm{w}, \mathrm{u} \rangle \geq \mathrm{cte} \quad \Leftrightarrow \quad \boxed{\langle \mathrm{w}, \mathrm{u} \rangle + \mathrm{b} \geq 0}$
- Deal with margin: $\mathrm{c}\Big(\langle \mathrm{w}, \mathrm{u} \rangle + \mathrm{b}\Big) \geq 1 \quad \Rightarrow \begin{cases} \langle \mathrm{w}, \mathrm{u}_+ \rangle + \mathrm{b} \geq 1 \\ \langle \mathrm{w}, \mathrm{u}_- \rangle + \mathrm{b} \leq -1 \end{cases}$

  $\rightarrow$ **Margin delimitation**: $\boxed{\mathrm{c}\Big(\langle \mathrm{w}, \mathrm{u} \rangle + \mathrm{b}\Big) - 1 = 0}$

Support-vector machine (SVM)

- **Margin delimitation**: $\mathrm{c}\Big(\langle \mathrm{w}, \mathrm{u}\rangle + \mathrm{b}\Big) - 1 = 0$

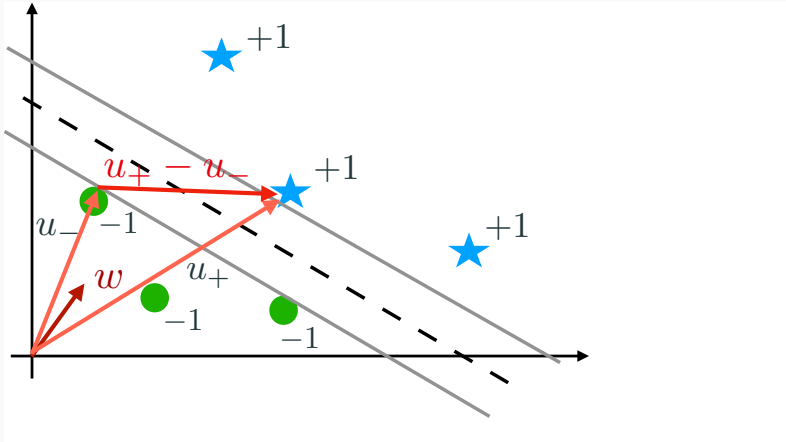


- **Objective**: Maximize the width of the margin

$$\begin{aligned}\text{Width} &= \frac{1}{\|\mathrm{w}\|}\langle \mathrm{u}_+ - \mathrm{u}_-, \mathrm{w}\rangle \\ &= \frac{1}{\|\mathrm{w}\|}(1 - b + 1 + b) \\ &= \frac{2}{\|\mathrm{w}\|}\end{aligned}$$

Support-vector machine (SVM)

- **Margin delimitation**: $\mathrm{c}\Big(\langle \mathrm{w}, \mathrm{u}\rangle + \mathrm{b}\Big) - 1 = 0$

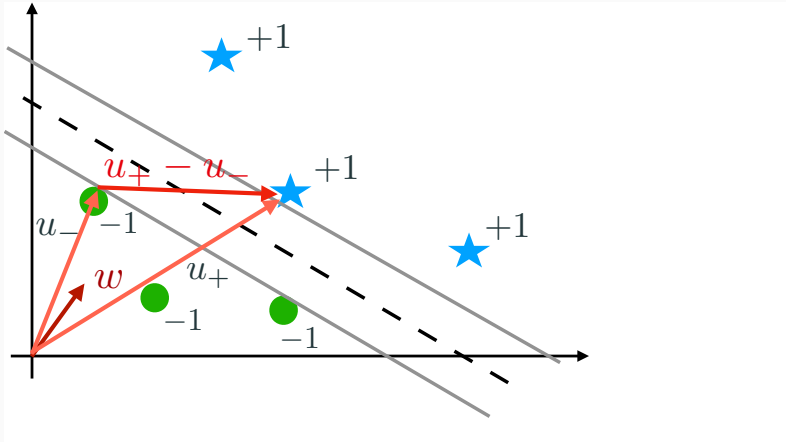


- **Objective**: Maximize the width of the margin

$$\begin{aligned}\mathrm{Width} &= \frac{1}{\|\mathrm{w}\|}\langle \mathrm{u}_+ - \mathrm{u}_-, \mathrm{w}\rangle \\ &= \frac{1}{\|\mathrm{w}\|}(1 - b + 1 + b) \\ &= \frac{2}{\|\mathrm{w}\|}\end{aligned}$$

Support-vector machine (SVM)

- **Margin delimitation**: $\mathrm{c}\Big(\langle \mathrm{w}, \mathrm{u}\rangle + \mathrm{b}\Big) - 1 = 0$

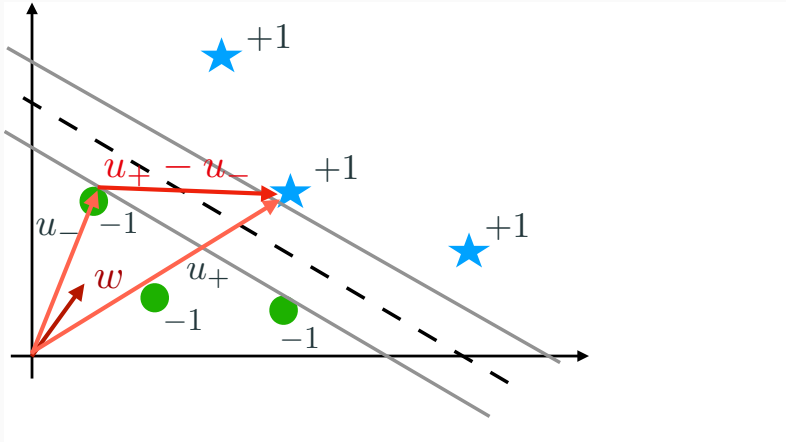


- **Objective**: Maximize the width of the margin

$$\begin{aligned}\text{Width} &= \frac{1}{\|\mathrm{w}\|}\langle \mathrm{u}_+ - \mathrm{u}_-, \mathrm{w}\rangle \\ &= \frac{1}{\|\mathrm{w}\|}(1 - b + 1 + b) \\ &= \frac{2}{\|\mathrm{w}\|}\end{aligned} \quad \Rightarrow \quad \boxed{\min_{\mathrm{w}} \|\mathrm{w}\| \quad \text{s.t.} \quad (\forall \ell)\ \mathrm{c}_\ell\Big(\langle \mathrm{w}, \mathrm{u}_\ell\rangle + \mathrm{b}\Big) \geq 1}$$

Support-vector machine (SVM)

- **Margin delimitation**: $\mathrm{c}\Big(\langle \mathrm{w}, \mathrm{u}\rangle + \mathrm{b}\Big) - 1 = 0$



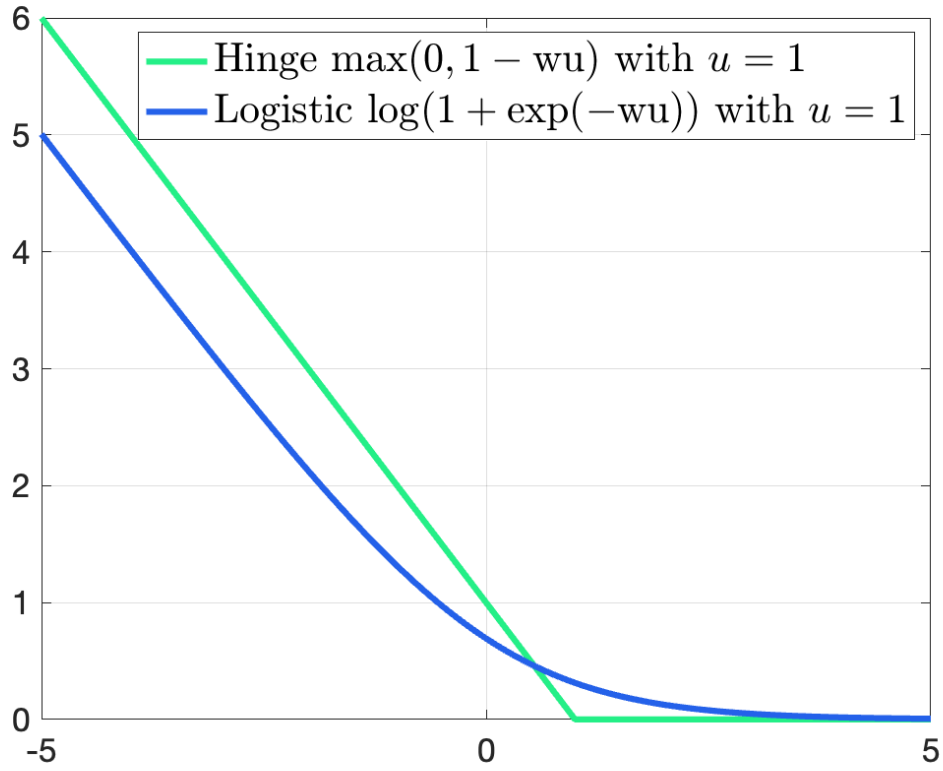
- **Objective**: Maximize the width of the margin

$$\begin{aligned}\text{Width} &= \frac{1}{\|\mathrm{w}\|}\langle \mathrm{u}_+ - \mathrm{u}_-, \mathrm{w}\rangle \\ &= \frac{1}{\|\mathrm{w}\|}(1 - b + 1 + b) \\ &= \frac{2}{\|\mathrm{w}\|}\end{aligned} \quad \Rightarrow \quad \boxed{\min_{\mathrm{w}} \ \frac{1}{2}\|\mathrm{w}\|^2 \ \text{ s.t. } (\forall \ell)\ \mathrm{c}_\ell\Big(\langle \mathrm{w}, \mathrm{u}_\ell\rangle + \mathrm{b}\Big) \geq 1}$$

Losses

- **SVM** : $\min_{\mathrm{w}} \ \frac{1}{2}\|\mathrm{w}\|^2 \ \text{s.t.} \ (\forall \ell) \ \mathrm{c}_\ell \Big(\langle \mathrm{w}, \mathrm{u}_\ell \rangle + \mathrm{b} \Big) \geq 1$
- **Hinge** + L2 : $\min_{\mathrm{w}} \ \frac{1}{2}\|\mathrm{w}\|^2 + \frac{\lambda}{L} \sum_\ell \max \Big(0, 1 - \mathrm{c}_\ell \big(\langle \mathrm{w}, \mathrm{u}_\ell \rangle + \mathrm{b} \big) \Big)$
- **Logistic** + L2 : $\min_{\mathrm{w}} \ \frac{1}{2}\|\mathrm{w}\|^2 + \frac{\lambda}{L} \sum_\ell \log \Big(1 + \exp(-\mathrm{c}_\ell (\langle \mathrm{w}, \mathrm{u}_\ell \rangle + \mathrm{b}) \Big)$

where $\lambda > 0$.



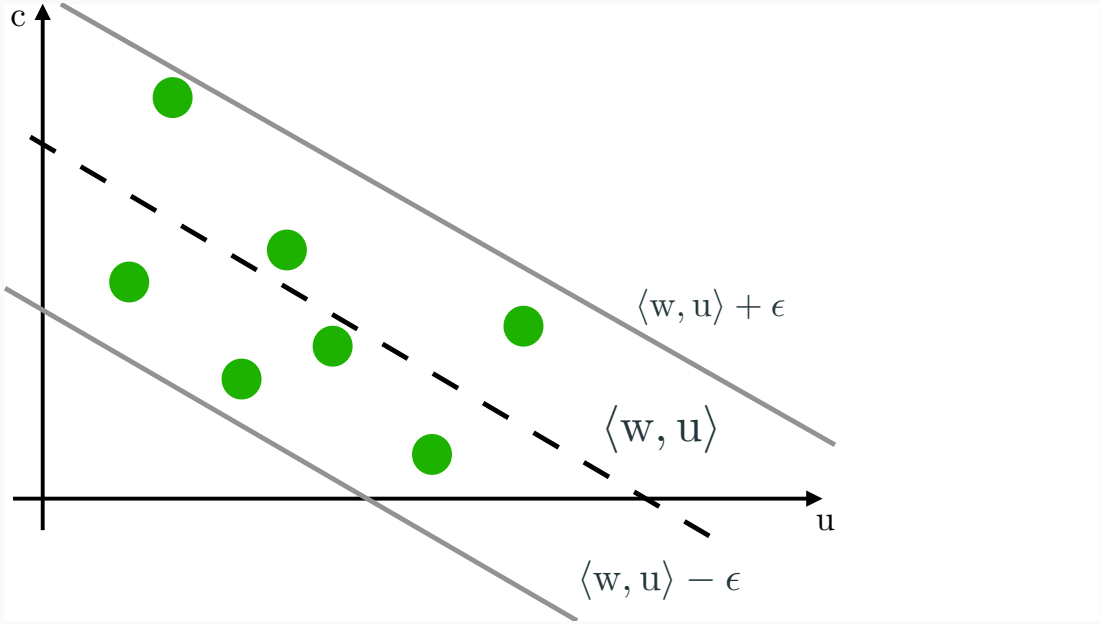
From linear regression to SVR

- Database : dataset $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathbb{R}^N \times \mathbb{R} \;\middle|\; \ell \in \{1, \ldots, L\}\right\}$.
- **Linear regression**:

$$\min_{\mathrm{w}} \sum_{\ell} (\mathrm{c}_\ell - \langle \mathrm{w}, \mathrm{u}_\ell \rangle)^2$$

- **Support Vector Regression**:

$$\min_{\mathrm{w}} \frac{1}{2} \|\mathrm{w}\|^2 \quad \text{s.t.} \quad (\forall \ell) \;\; |\mathrm{c}_\ell - \langle \mathrm{w}, \mathrm{u}_\ell \rangle| \leq \epsilon$$



Linearization

- **Database**: $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{H} \times \mathcal{G} \;\middle|\; \ell \in \{1, \ldots, L\}\right\}$

examples: $\mathrm{u}_\ell =$  and $z_\ell = 2$

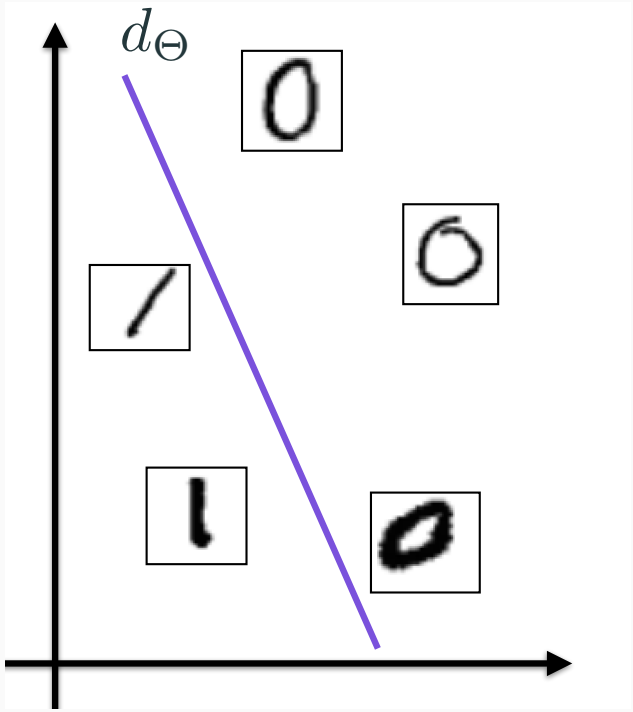
$\mathrm{u}_\ell =$  and $z_\ell = 9$

- $\varphi(\mathrm{u})\colon \mathbb{R}^N \to \mathbb{R}^M$: mapping from the input space onto an arbitrary feature space with $M > N$

  $\Rightarrow$ **linearization**

  examples: convolution networks.

  scattering coefficients.



Objective

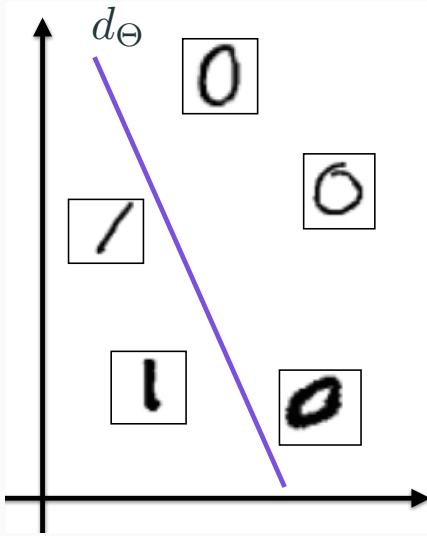
- The predictor relies on K different discriminating functions $D_{\theta^{(k)}} : \mathbb{R}^N \to \mathbb{R}$:

$$\mathrm{D}_{\theta^{(k)}}(\mathrm{u}) = (\mathrm{w}^{(k)})^\top \varphi(\mathrm{u}) + b^{(k)} \qquad \text{where} \qquad \theta_k = \{\mathrm{w}^{(k)}, b^{(k)}\}$$

with $\phi(\mathrm{u}) = [\varphi(\mathrm{u})^\top 1]^\top$ and $\Theta = [\underbrace{(\mathrm{w}^{(1)})^\top, b^{(1)}}_{\theta^{(1)}}, \ldots, \underbrace{(\mathrm{w}^{(K)})^\top, b^{(K)}]^\top}_{\theta^{(K)}}$

- The predictor selects the class that best matches an observation

$$d_\Theta(\mathrm{u}) = \arg \max_{1 \le k \le K} D_{\theta^{(k)}}(\mathrm{u})$$



Objective

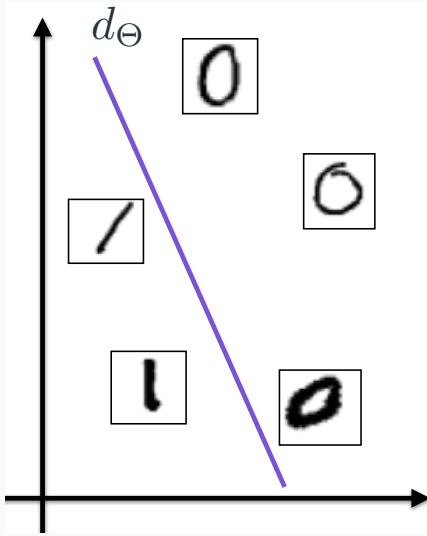
- The predictor relies on K different discriminating functions $D_{\theta^{(k)}} : \mathbb{R}^N \to \mathbb{R}$:

$$\mathrm{D}_{\theta^{(k)}}(\mathrm{u}) = (\mathrm{w}^{(k)})^\top \varphi(\mathrm{u}) + b^{(k)} \quad \Leftrightarrow \quad D_k(\mathrm{u}) = (\theta^{(k)})^\top \phi(\mathrm{u})$$

with $\phi(\mathrm{u}) = [\varphi(\mathrm{u})^\top 1]^\top$ and $\Theta = [\underbrace{(\mathrm{w}^{(1)})^\top, b^{(1)}}_{\theta^{(1)}}, \ldots, \underbrace{(\mathrm{w}^{(K)})^\top, b^{(K)}]^\top}_{\theta^{(K)}}$

- The predictor selects the class that best matches an observation

$$d_\Theta(\mathrm{u}) = \arg \max_{1 \leq k \leq K} D_{\theta^{(k)}}(\mathrm{u})$$



Objective

- Objective of the learning stage: estimate Θ to correctly predict the input-output pair $(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{S}$ for every $\ell \in \{1, \ldots, L\}$,

$$d_\Theta(\mathrm{u}) = \arg \max_{1 \le k \le K} D_{\theta^{(k)}}(\mathrm{u}) : \qquad \mathrm{c}_\ell = \arg \max_{1 \le k \le K} (\theta^{(k)})^\top \phi(\mathrm{u}_\ell)$$

Objective

- Objective of the learning stage: estimate Θ to correctly predict the input-output pair $(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{S}$ for every $\ell \in \{1, \ldots, L\}$,

$$d_\Theta(\mathrm{u}) = \arg \max_{1 \le k \le K} D_{\theta^{(k)}}(\mathrm{u}) : \qquad \mathrm{c}_\ell = \arg \max_{1 \le k \le K} (\theta^{(k)})^\top \phi(\mathrm{u}_\ell)$$

$$\Leftrightarrow \quad \max_{k \neq \mathrm{c}_\ell} (\theta^{(k)} - \theta^{(\mathrm{c}_\ell)})^\top \phi(\mathrm{u}_\ell) < 0$$

Objective

- Objective of the learning stage: estimate Θ to correctly predict the input-output pair $(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{S}$ for every $\ell \in \{1, \ldots, L\}$,

$$d_\Theta(\mathrm{u}) = \arg \max_{1 \le k \le K} D_{\theta^{(k)}}(\mathrm{u}) : \qquad \mathrm{c}_\ell = \arg \max_{1 \le k \le K} (\theta^{(k)})^\top \phi(\mathrm{u}_\ell)$$

$$\Leftrightarrow \quad \max_{k \neq \mathrm{c}_\ell} (\theta^{(k)} - \theta^{(\mathrm{c}_\ell)})^\top \phi(\mathrm{u}_\ell) < 0$$

[relax the strict ineqality] $$\Leftrightarrow \quad \max_{k \neq \mathrm{c}_\ell} (\theta^{(k)} - \theta^{(\mathrm{c}_\ell)})^\top \phi(\mathrm{u}_\ell) \le -\mu_\ell$$

with $\mu_\ell > 0$

Objective

- Objective of the learning stage: estimate Θ to correctly predict the input-output pair $(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{S}$ for every $\ell \in \{1, \ldots, L\}$,

$$d_\Theta(\mathrm{u}) = \arg \max_{1 \le k \le K} D_{\theta^{(k)}}(\mathrm{u}) : \qquad \mathrm{c}_\ell = \arg\max_{1 \le k \le K} (\theta^{(k)})^\top \phi(\mathrm{u}_\ell)$$

$$\Leftrightarrow \quad \max_{k \neq \mathrm{c}_\ell} (\theta^{(k)} - \theta^{(\mathrm{c}_\ell)})^\top \phi(\mathrm{u}_\ell) < 0$$

[relax the strict ineqality] $$\Leftrightarrow \quad \max_{k \neq \mathrm{c}_\ell} (\theta^{(k)} - \theta^{(\mathrm{c}_\ell)})^\top \phi(\mathrm{u}_\ell) \le -\mu_\ell$$

[deal with unfeasible constraints] $$\Leftrightarrow \quad \max_{k \neq \mathrm{c}_\ell} (\theta^{(k)} - \theta^{(\mathrm{c}_\ell)})^\top \phi(\mathrm{u}_\ell) \le \zeta_\ell - \mu_\ell$$

with $\mu_\ell > 0$ and $\zeta_\ell \ge 0$.

Multiclass SVM

$$\underset{(\Theta,\xi)\in\mathbb{R}^{(M+1)K}\times\mathbb{R}^{L}}{\text{minimize}} \quad \sum_{k=1}^{K}\|\theta^{(k)}\|_2^2+\lambda\sum_{\ell=1}^{L}\xi^{(\ell)} \quad \text{subj. to}$$

$$\begin{cases}(\forall \ell\in\{1,...,L\}) & \max\limits_{k\neq \mathrm{c}_\ell}\,(\theta^{(k)}-\theta^{(\mathrm{c}_\ell)})^\top\phi(\mathrm{u}_\ell)\leq \xi_\ell-\mu_\ell,\\ (\forall \ell\in\{1,...,L\}) & \xi_\ell\geq 0,\end{cases}$$

or equivalently

$$\underset{\Theta\in\mathbb{R}^{(M+1)K}}{\text{minimize}} \sum_{k=1}^{K}\|\theta^{(k)}\|_2^2+\lambda\sum_{\ell=1}^{L}\max\left\{0,\mu_\ell+\max_{k\neq \mathrm{c}_\ell}\,(\theta^{(k)}-\theta^{(\mathrm{c}_\ell)})^\top\phi(\mathrm{u}_\ell)\right\}.$$

Alternative to standard SVM data-term

$$h(\Theta) = \sum_{\ell=1}^{L} \max\left\{0, \mu_\ell + \max_{k \neq \mathrm{c}_\ell} (\theta^{(k)} - \theta^{(\mathrm{c}_\ell)})^\top \phi(\mathrm{u}_\ell)\right\}.$$

- Multiclass SVM [Blondel et al.]
  $h(\Theta) = \sum_{\ell=1}^{L} \sum_{k \neq \mathrm{c}_\ell} \left(\max\left\{0, \mu_\ell + (\theta^{(k)} - \theta^{(\mathrm{c}_\ell)})^\top \phi(\mathrm{u}_\ell)\right\}\right)^2$

- Multinomial logistic regression [Krishnapuram et al.]
  $h(\Theta) = \sum_{\ell=1}^{L} \log\left(1 + \sum_{k \neq \mathrm{c}_\ell} \exp\left\{\mu_\ell + (\theta^{(k)} - \theta^{(\mathrm{c}_\ell)})^\top \phi(\mathrm{u}_\ell)\right\}\right)$

- "one-vs-all" strategy binary SVM [Laporte et al.] with $\widetilde{\mathrm{c}}_\ell$ being equal to 1 if $\mathrm{c}_\ell = k$, and -1 otherwise $h(\Theta) = \sum_{\ell=1}^{L} \left(\max\left\{0, \mu_\ell + \widetilde{\mathrm{c}}_\ell \, (\theta^{(k)})^\top \phi(\mathrm{u}_\ell)^\top\right\}\right)^2$

Example: Supervised learning

- **Database**: $\mathcal{S} = \left\{ (\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{H} \times \mathcal{G} \;\middle|\; \ell \in \{1, \dots, L\} \right\}$
- **Goal**: Learn a prediction function $d_\Theta : \mathcal{H} \to \mathcal{G}$
- Learning procedure relies on a minimization problem:

$$\underset{\Theta}{\text{minimize}} \quad \underbrace{\frac{1}{L} \sum_{\ell=1}^{L} F(\mathrm{c}_\ell, d_\Theta(\mathrm{u}_\ell))}_{\text{Data-term}} \quad + \quad \lambda \underbrace{R(\Theta)}_{\text{Prior}}$$

Example: Supervised learning

- **Database**: $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{H} \times \mathcal{G} \;\middle|\; \ell \in \{1, \dots, L\}\right\}$
- **Goal**: Learn a prediction function $d_\Theta : \mathcal{H} \to \mathcal{G}$
- Learning procedure relies on a minimization problem:

$$\underset{\Theta}{\text{minimize}} \quad \underbrace{\frac{1}{L}\sum_{\ell=1}^{L} F(\mathrm{c}_\ell, d_\Theta(\mathrm{u}_\ell))}_{\text{Data-term}} \;+\; \lambda \underbrace{R(\Theta)}_{\text{Prior}}$$

- **Linear predictor**: $d_\Theta(u) = \Theta^\top u$

  - ⊙ Ridge regression: $\underset{\Theta \in \mathcal{H}}{\text{minimize}} \;\; \frac{1}{L}\sum_{\ell=1}^{L} (\mathrm{c}_\ell - \Theta^\top \mathrm{u}_\ell)^2 + \lambda \|\Theta\|_2^2$
  - ⊙ Logistic classification: $\underset{\Theta \in \mathcal{H}}{\text{minimize}} \;\; \frac{1}{L}\sum_{\ell=1}^{L} \log\left(1 + e^{-\mathrm{c}_\ell \Theta^\top \mathrm{u}_\ell}\right) + \lambda \|\Theta\|_2^2$

Example: Supervised learning

- **Database**: $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{H} \times \mathcal{G} \;\middle|\; \ell \in \{1, \dots, L\}\right\}$
- **Goal**: Learn a prediction function $d_\Theta : \mathcal{H} \to \mathcal{G}$
- Learning procedure relies on a minimization problem:

$$\underset{\Theta}{\text{minimize}} \quad \underbrace{\frac{1}{L}\sum_{\ell=1}^{L} F(\mathrm{c}_\ell, d_\Theta(\mathrm{u}_\ell))}_{\text{Data-term}} \;+\; \lambda \underbrace{R(\Theta)}_{\text{Prior}}$$

- **Linear predictor**: $d_\Theta(u) = \Theta^\top u$

  - ⊙ Ridge regression: $\underset{\Theta \in \mathcal{H}}{\text{minimize}} \;\; \frac{1}{L}\sum_{\ell=1}^{L} (\mathrm{c}_\ell - \Theta^\top \mathrm{u}_\ell)^2 + \lambda \|\Theta\|_2^2$
  - ⊙ Logistic classification: $\underset{\Theta \in \mathcal{H}}{\text{minimize}} \;\; \frac{1}{L}\sum_{\ell=1}^{L} \log\left(1 + e^{-\mathrm{c}_\ell \Theta^\top \mathrm{u}_\ell}\right) + \lambda \|\Theta\|_2^2$

Example: Supervised learning

- **Database**: $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{H} \times \mathcal{G} \;\middle|\; \ell \in \{1, \dots, L\}\right\}$
- **Goal**: Learn a prediction function $d_\Theta : \mathcal{H} \to \mathcal{G}$
- Learning procedure relies on a minimization problem:

$$\underset{\Theta}{\text{minimize}} \quad \underbrace{\frac{1}{L}\sum_{\ell=1}^{L} F(\mathrm{c}_\ell, d_\Theta(\mathrm{u}_\ell))}_{\text{Data-term}} \; + \; \lambda \underbrace{R(\Theta)}_{\text{Prior}}$$

- **Linear predictor**: $d_\Theta(u) = \Theta^\top u$

  $\Rightarrow$ can be extended to $d_\Theta(u) = \Theta^\top \phi(u)$ (e.g. ϕ scattering transform)

  - Ridge regression: $\underset{\Theta \in \mathcal{H}}{\text{minimize}} \; \frac{1}{L}\sum_{\ell=1}^{L} (\mathrm{c}_\ell - \Theta^\top \mathrm{u}_\ell)^2 + \lambda \|\Theta\|_2^2$
  - Logistic classification: $\underset{\Theta \in \mathcal{H}}{\text{minimize}} \; \frac{1}{L}\sum_{\ell=1}^{L} \log\left(1 + e^{-\mathrm{c}_\ell \Theta^\top \mathrm{u}_\ell}\right) + \lambda \|\Theta\|_2^2$

Example: Supervised learning

- **Database**: $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{H} \times \mathcal{G} \;\middle|\; \ell \in \{1, \dots, L\}\right\}$
- **Goal**: Learn a prediction function $d_\Theta : \mathcal{H} \to \mathcal{G}$
- Learning procedure relies on a minimization problem:

$$\underset{\Theta}{\text{minimize}} \quad \underbrace{\frac{1}{L}\sum_{\ell=1}^{L} F(\mathrm{c}_\ell, d_\Theta(\mathrm{u}_\ell))}_{\text{Data-term}} \; + \; \lambda \underbrace{R(\Theta)}_{\text{Prior}}$$

- **Linear predictor**: $d_\Theta(\mathrm{u}) = \Theta^\top \mathrm{u}$
  - ⊙ Lasso: $\underset{\Theta \in \mathcal{H}}{\text{minimize}} \;\; \frac{1}{L}\sum_{\ell=1}^{L} (\mathrm{c}_\ell - \Theta^\top \mathrm{u}_\ell)^2 + \lambda \|\Theta\|_1$
  - ⊙ Sparse logistic classification: $\underset{\Theta \in \mathcal{H}}{\text{minimize}} \;\; \frac{1}{L}\sum_{\ell=1}^{L} \log\left(1 + e^{-\mathrm{c}_\ell \Theta^\top \mathrm{u}_\ell}\right) + \lambda \|\Theta\|_1$
  - ⊙ SVM classification: $\underset{\Theta \in \mathcal{H}}{\text{minimize}} \;\; \frac{1}{L}\sum_{\ell=1}^{L} \max\left(0, 1 - \mathrm{c}_\ell \Theta^\top \mathrm{u}_\ell\right) + \lambda \|\Theta\|_2^2$

Example: Supervised learning

- **Database**: $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{H} \times \mathcal{G} \;\middle|\; \ell \in \{1, \dots, L\}\right\}$
- **Goal**: Learn a prediction function $d_\Theta : \mathcal{H} \to \mathcal{G}$
- Learning procedure relies on a minimization problem:

$$\underset{\Theta}{\text{minimize}} \quad \underbrace{\frac{1}{L}\sum_{\ell=1}^{L} F(\mathrm{c}_\ell, d_\Theta(\mathrm{u}_\ell))}_{\text{Data-term}} \; + \; \lambda \underbrace{R(\Theta)}_{\text{Prior}}$$

- **Linear predictor**: $d_\Theta(\mathrm{u}) = \Theta^\top \mathrm{u} \quad \Rightarrow$ **Convex non-smooth problems**
  - ⊙ Lasso: $\underset{\Theta \in \mathcal{H}}{\text{minimize}} \; \frac{1}{L}\sum_{\ell=1}^{L} (\mathrm{c}_\ell - \Theta^\top \mathrm{u}_\ell)^2 + \lambda \|\Theta\|_1$
  - ⊙ Sparse logistic classification: $\underset{\Theta \in \mathcal{H}}{\text{minimize}} \; \frac{1}{L}\sum_{\ell=1}^{L} \log\left(1 + e^{-\mathrm{c}_\ell \Theta^\top \mathrm{u}_\ell}\right) + \lambda \|\Theta\|_1$
  - ⊙ SVM classification: $\underset{\Theta \in \mathcal{H}}{\text{minimize}} \; \frac{1}{L}\sum_{\ell=1}^{L} \max\left(0, 1 - \mathrm{c}_\ell \Theta^\top \mathrm{u}_\ell\right) + \lambda \|\Theta\|_2^2$

Experiment 2

MNIST database

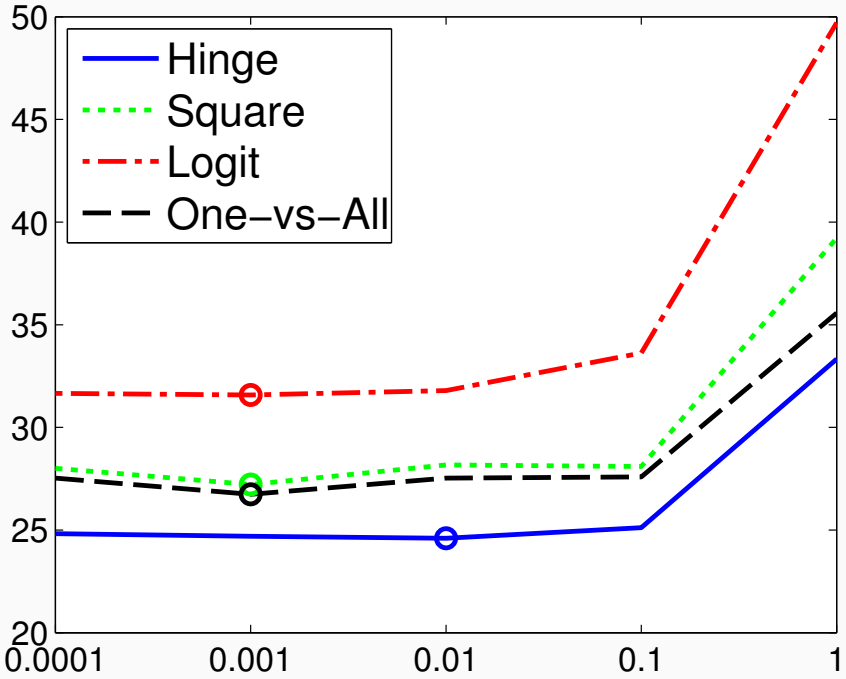
- $N = 28 \times 28$
- $K = 10$
- 60000 training images
- 10000 test images

Scattering convolution network [Bruna & Mallat, 2013]

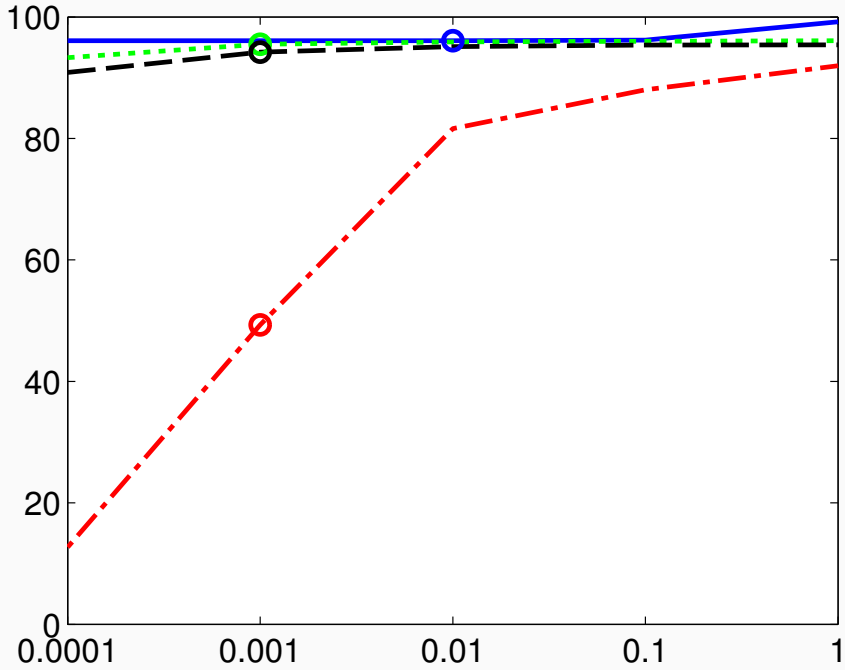
- 2 wavelet layers
- 4 scales DWT
- Feature mapping: ϕ: $\mathbb{R}^{28\times28} \mapsto \mathbb{R}^{14\times14\times81}$

Experiment 2

Classification results for $L/K = 3$



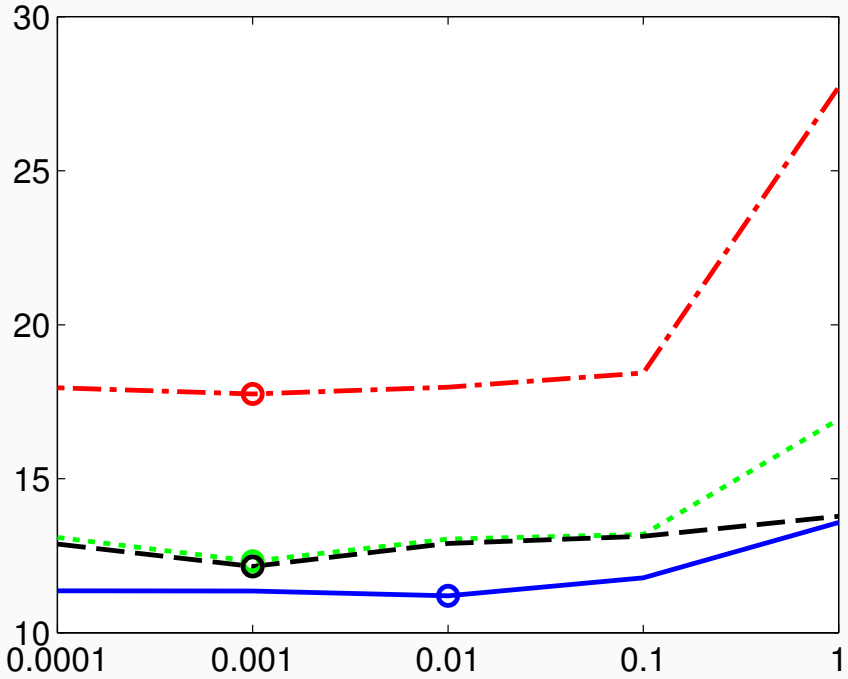
Classification error



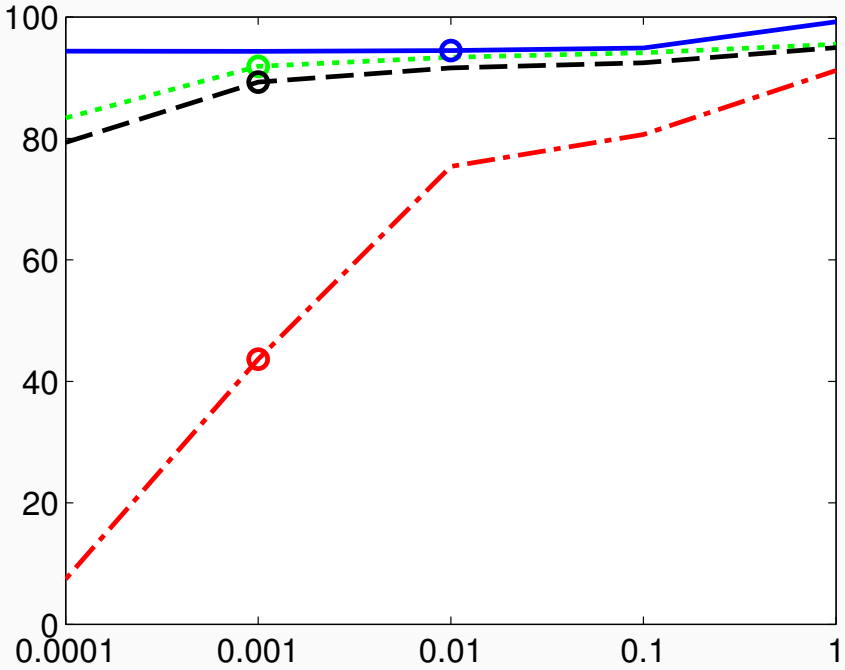
Percentage of zero coefficients

Experiment 2

Classification results for $L/K = 10$



Classification error



Percentage of zero coefficients

Example: Supervised learning

- **Database**: $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{H} \times \mathcal{G} \;\middle|\; \ell \in \{1, \dots, L\}\right\}$
- **Goal**: Learn a prediction function $d_\Theta : \mathcal{H} \to \mathcal{G}$
- Learning procedure relies on a **minimization problem**:

$$\underset{\Theta}{\text{minimize}} \quad \underbrace{\frac{1}{L}\sum_{\ell=1}^{L} F(\mathrm{c}_\ell, d_\Theta(\mathrm{u}_\ell))}_{\text{Data-term}} \; + \; \lambda \underbrace{R(\Theta)}_{\text{Prior}}$$

- **Neural network predictor**:

$$d_\Theta(\mathrm{u}) = \eta^{[K]}(W^{[K]}\eta^{[K-1]}(W^{[K-1]} \dots \eta^{[2]}(W^{[2]}\eta^{[1]}(W^{[1]}u)) \dots))$$

  - ⊙ Linear operators: $W^{[1]}, W^{[2]}, \dots, W^{[K]}$
  - ⊙ Activation functions: $\eta^{[1]}, \eta^{[2]}, \dots, \eta^{[K]}$

Example: Supervised learning

- **Database**: $\mathcal{S} = \left\{(\mathrm{u}_\ell, \mathrm{c}_\ell) \in \mathcal{H} \times \mathcal{G} \;\middle|\; \ell \in \{1, \dots, L\}\right\}$
- **Goal**: Learn a prediction function $d_\Theta : \mathcal{H} \to \mathcal{G}$
- Learning procedure relies on a **minimization problem**:

$$\underset{\Theta}{\text{minimize}} \quad \underbrace{\frac{1}{L}\sum_{\ell=1}^{L} F(\mathrm{c}_\ell, d_\Theta(\mathrm{u}_\ell))}_{\text{Data-term}} \; + \; \lambda \underbrace{R(\Theta)}_{\text{Prior}}$$

- **Neural network predictor**: $\Rightarrow$ **Non-convex problems**

$$d_\Theta(\mathrm{u}) = \eta^{[K]}(W^{[K]}\eta^{[K-1]}(W^{[K-1]} \dots \eta^{[2]}(W^{[2]}\eta^{[1]}(W^{[1]}u)) \dots))$$

  - ⊙ Linear operators: $W^{[1]}, W^{[2]}, \dots, W^{[K]}$
  - ⊙ Activation functions: $\eta^{[1]}, \eta^{[2]}, \dots, \eta^{[K]}$