# Optimisation methods for computational imaging

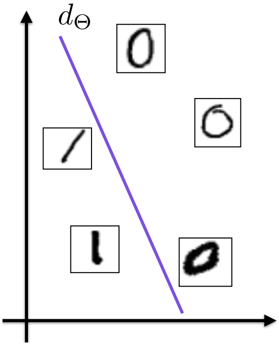
## Chapter 3 - Variational approaches in supervised learning

Nelly Pustelnik - ENS Lyon
Audrey Repetti - Heriot-Watt University

Journées SMAI-MODE 2022 – Limoges

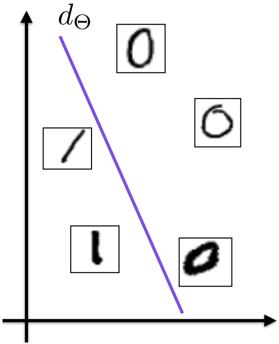
# Example: Supervised learning

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



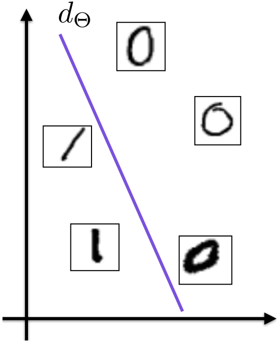
# Example: Supervised learning

- **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 $d_\Theta \colon \mathcal{H} \to \mathcal{G}$



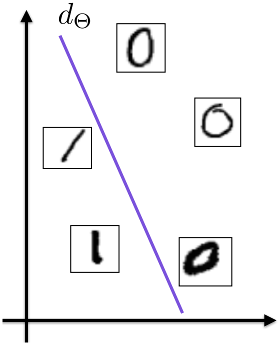
# Example: Supervised learning

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



# 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 $d_\Theta \colon \mathcal{H} \to \mathcal{G}$
- **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;



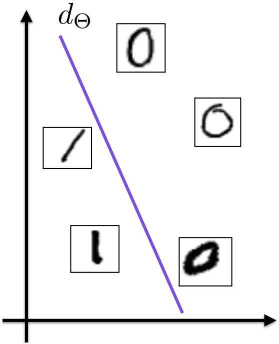
# Example: Supervised learning

- **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;

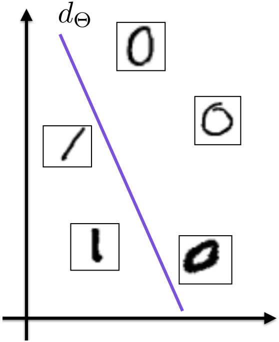


# Example: Supervised learning

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

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

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



## Example: Supervised learning

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

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

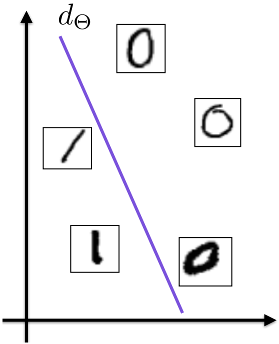
  $\mathrm{u}_\ell =$  and $z_\ell = 9$

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

  $\Rightarrow$ **linearization**

  examples: convolution networks.

  scattering coefficients.



# Objective

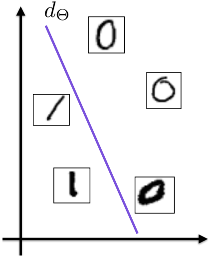
- The predictor relies on $K$ different discriminating functions $D_{\theta^{(k)}} : \mathbb{R}^N \to \mathbb{R}$ :

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

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

- The predictor selects the class that best matches an observation

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



# Objective

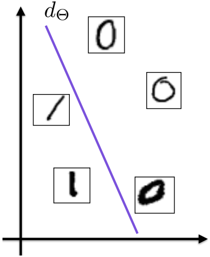
- The predictor relies on $K$ different discriminating functions $D_{\theta^{(k)}} : \mathbb{R}^N \to \mathbb{R}$ :

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

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

- The predictor selects the class that best matches an observation

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



# Objective

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

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

# Objective

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

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

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

# Objective

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

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

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

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

with $\mu_\ell > 0$

# Objective

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

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

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

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

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

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

# Multiclass SVM

$$\underset{(\Theta,\xi)\in\mathbb{R}^{(M+1)K}\times\mathbb{R}^L}{\text{minimise}} \quad \sum_{k=1}^{K} \|\theta^{(k)}\|_2^2 + \lambda \sum_{\ell=1}^{L} \xi^{(\ell)} \quad \text{subj. to}$$
$$\begin{cases} (\forall \ell \in \{1,\dots,L\}) & \max\limits_{k\neq c_\ell} \left(\theta^{(k)} - \theta^{(c_\ell)}\right)^\top \phi(u_\ell) \leqslant \xi_\ell - \mu_\ell, \\ (\forall \ell \in \{1,\dots,L\}) & \xi_\ell \geqslant 0, \end{cases}$$

or equivalently

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

# Alternative to standard SVM data-term

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

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

# 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 $d_\Theta \colon \mathcal{H} \to \mathcal{G}$
- Learning procedure relies on a minimization problem:

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

# Example: Supervised learning

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

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

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

# Example: Supervised learning

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

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

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

# 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 $d_\Theta \colon \mathcal{H} \to \mathcal{G}$
- Learning procedure relies on a minimization problem:

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

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

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

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

## Example: Supervised learning

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

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

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

# Example: Supervised learning

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

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

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

# Experiment 2

**MNIST database**

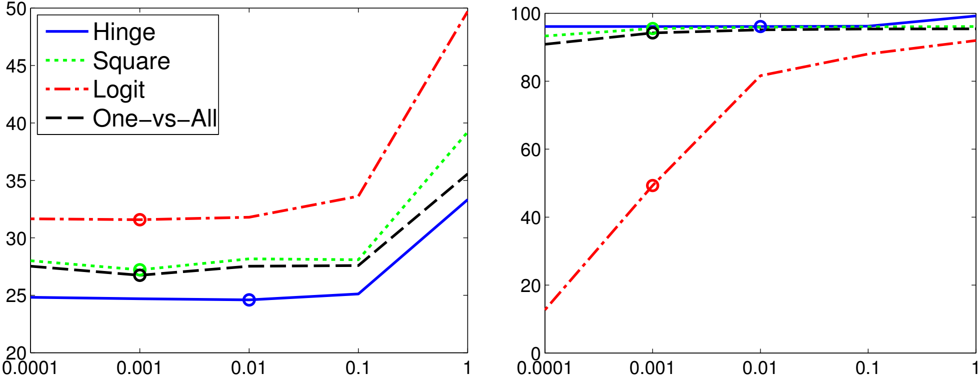
- $N = 28 \times 28$
- $K = 10$
- 60000 training images
- 10000 test images

**Scattering convolution network** [Bruna & Mallat, 2013]

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

# Experiment 2

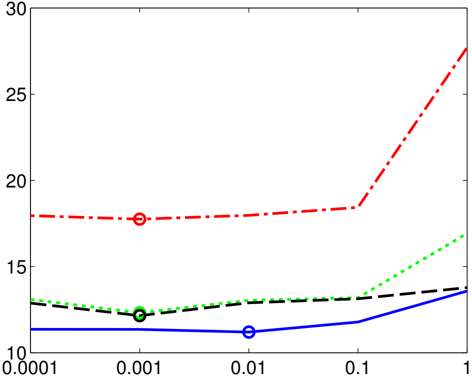
**Classification results for $L/K = 3$**



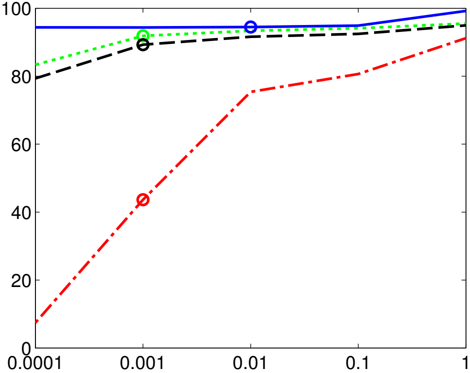

Classification error    Percentage of zero coefficients

# Experiment 2

**Classification results for $L/K = 10$**




Classification error




Percentage of zero coefficients

# Example: Supervised learning

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

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

- **Neural network predictor**:

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

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

# Example: Supervised learning

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

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

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

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

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

# Proximal algorithms

**General objective function**

$$\text{Find} \qquad \widehat{y} \in \underset{y \in \mathcal{H}}{\operatorname{Argmin}} \sum_{j=1}^{J} f_j(H_j y)$$

where $H_j$ linear operator from $\mathcal{H}$ to $\mathcal{G}_j$ and $(f_j)_{1 \leqslant j \leqslant J}$ belong to the class of convex functions, l.s.c., and proper from $\mathcal{G}_j$ to $]-\infty, +\infty]$.

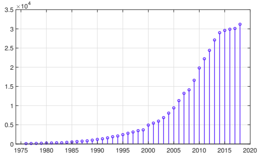
# Proximal algorithms

**General objective function**

$$\text{Find} \qquad \widehat{y} \in \operatorname*{Argmin}_{y\in\mathcal{H}} \sum_{j=1}^{J} f_j(H_j y)$$

where $H_j$ linear operator from $\mathcal{H}$ to $\mathcal{G}_j$ and $(f_j)_{1\leqslant j\leqslant J}$ belong to the class of convex functions, l.s.c., and proper from $\mathcal{G}_j$ to $]-\infty,+\infty]$.

- Numerous proximal algorithms [Bauschke-Combettes, 2017]
  - Forward-Backward
  - Douglas-Rachford
  - ADMM
  - Primal-dual ...




Number of articles per year on Google scholar containing "proximal algorithms" since 1997.