University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 11.1 - Kernel Methods
Kernelizing Linear Models

Erik Bekkers

(Bishop 6.0, 6.1)

Slide credits: Patrick Forré and Rianne van den Berg

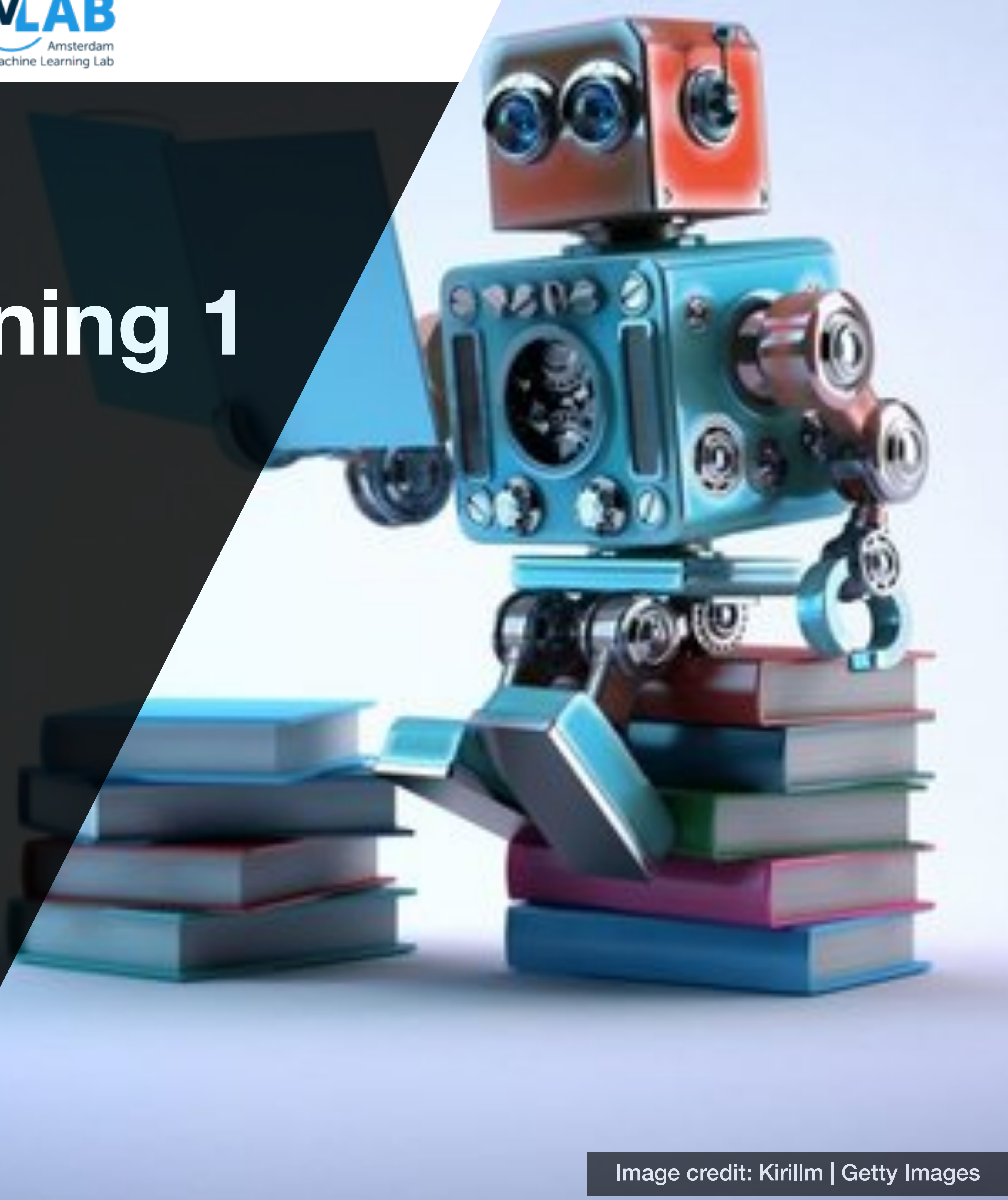
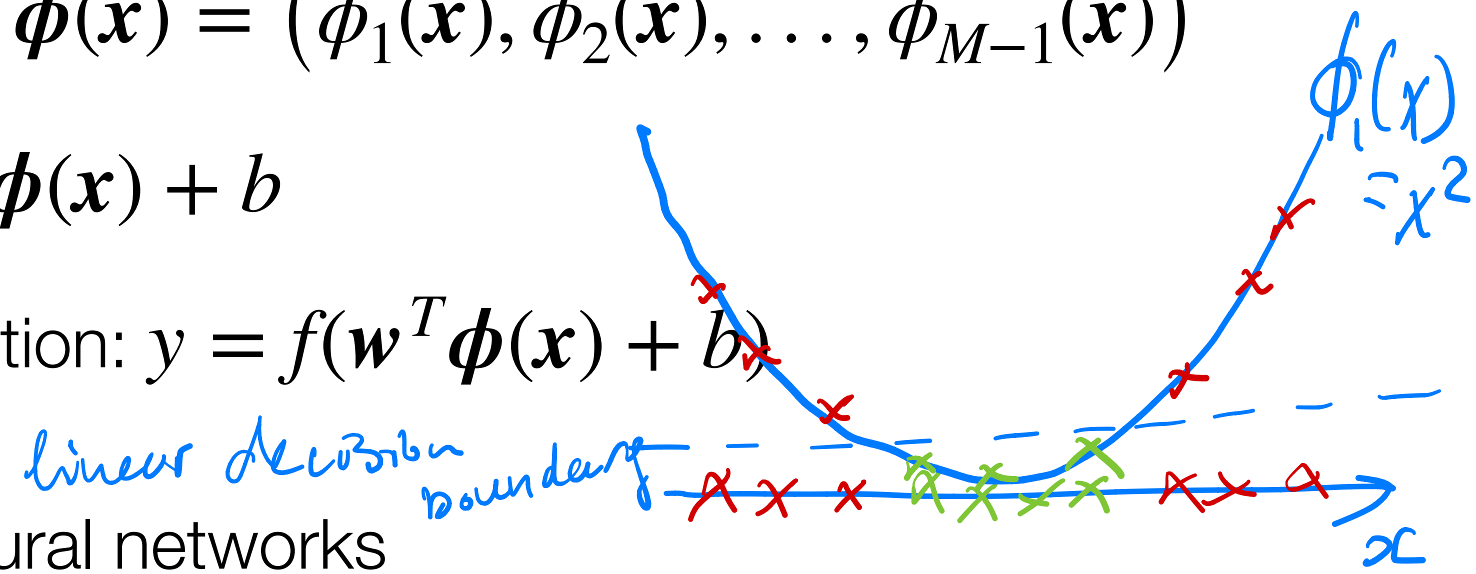


Image credit: Kirillm | Getty Images

So Far: Parametric Models

- Fixed basis function methods: $\boldsymbol{\phi}(\boldsymbol{x}) = \left(\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots, \phi_{M-1}(\boldsymbol{x})\right)^T$
  - Linear regression: $y = \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}) + b$
  - Linear models for classification: $y = f(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}) + b)$



- Learnable basis functions: Neural networks

$$y(\mathbf{x}, \mathbf{W}^{(1)}, \mathbf{w}^{(2)}) = \sum_{m=0}^{M} w_m^{(2)} h(\underbrace{\sum_{d=0}^{D} w_{md}^{(1)} x_d}_{\phi_m(x)})$$

- Training:
  - MLE, MAP: use training data to obtain point estimate of $\mathbf{w}$
  - Full Bayesian: use training data to obtain posterior $p(\mathbf{w} \mid \mathbf{X}, \mathbf{t})$

- Test time: Discard training data, only need $\mathbf{w}$ or $p(\mathbf{w} \mid \mathbf{X}, \mathbf{t})$

Parametric vs Non-Parametric Models

- Parametric models = models with a finite number of parameters
- Non-Parametric models = models with no explicitly defined parameters (but *implicitly still* work with (finite) or infinite number of parameters)
- Parametric methods:
  - Working in the (finite dimensional) parameter space
- Non-Parametric methods
  - Directly working in possibly infinite dimensional function spaces
  - Typically we have $M \gg N$

Non-Parametric Kernel Methods

- Kernel methods: Use (subset) of training points for predictions (test time!). Useful if $M \gg N$

parametrized by $\underline{w}$

- Linear parametric models:
  - Can be re-cast into equivalent 'dual representation'

  $\underline{a}$
  - Predictions are based on linear combinations of the *kernel function* evaluated at training data points

- For linear models with fixed feature vectors $\boldsymbol{\phi}(\mathbf{x})$ we will encounter

$$\boxed{k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{x}')}$$

- Kernel measures similarity between $\mathbf{x}$ and $\mathbf{x}'$ in feature space defined by mapping $\boldsymbol{\phi}(\mathbf{x})$

$$k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$$

Kernelized Ridge Regression

- Goal: Minimize sum of squared errors with quadratic weight penalty

$$J(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - t_n\}^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

$$\Phi = \begin{pmatrix} \phi_0(x_1) & \phi_1(x_1) & \dots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \dots & \phi_{M-1}(x_2) \\ \vdots & & \ddots & \vdots \\ \phi_0(x_N) & \phi_1(x_N) & \dots & \phi_{M-1}(x_N) \end{pmatrix} \in \mathbb{R}^{N \times M}$$

- Solution: Solve $\dfrac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 0$:

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \sum_{n=1}^{N}\{\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - t_n\}\boldsymbol{\phi}(\mathbf{x}_n)^T + \lambda\mathbf{w}^T I = \mathbf{0}$$

Transpose on both sides, using $(a^T B)^T = B^T a$

$$\Leftrightarrow \quad \mathbf{w}^T\left(\sum_{n=1}^{N}\boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^T + \lambda\mathbf{I}\right) = \sum_{n=1}^{N} t_n\boldsymbol{\phi}(\mathbf{x}_n)^T$$

$$\Leftrightarrow \quad \left(\sum_{n=1}^{N}\boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^T + \lambda\mathbf{I}\right)\mathbf{w} = \sum_{n=1}^{N} t_n\boldsymbol{\phi}(\mathbf{x}_n)$$

$Ax = b$, $x = A^{-1}b$

$$\mathbf{w} = \left(\sum_{n=1}^{N}\boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^T + \lambda\mathbf{I}\right)^{-1}\sum_{n=1}^{N} t_n\boldsymbol{\phi}(\mathbf{x}_n) = \underbrace{(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\mathbf{I})}_{M \times M}{}^{-1}\boldsymbol{\Phi}^T\boldsymbol{t}$$

Kernelized Ridge Regression

- Goal: Minimize sum of squared errors with quadratic weight penalty

$$J(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - t_n\}^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

- Solution: Solve $\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 0$:

$$\mathbf{w} = \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\mathbf{I}_M\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{t},$$

$$\boldsymbol{\Phi}^T\boldsymbol{\Phi} \in \mathbb{R}^{MxM}$$

- Use matrix inversion lemma (see e.g. Bishop C.5):

$$\left(P^{-1} + B^TR^{-1}B\right)^{-1}B^TR^{-1} = PB^T\left(BPB^T + R\right)^{-1}$$

$$\begin{pmatrix} P^{-1} = \lambda\mathbf{I}_M \\ B = \boldsymbol{\Phi} \\ R = \mathbf{I}_N \end{pmatrix}$$

$(\lambda I_m + \Phi^T I_N \Phi)^{-1} \ldots$ $\Phi\Phi^T$

- Allows us to alternatively obtain $\mathbf{w}$ via

$$\mathbf{w} = \boldsymbol{\Phi}^T\left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \lambda\mathbf{I}_N\right)^{-1}\boldsymbol{t} = \boldsymbol{\Phi}^T\left(\boldsymbol{K} + \lambda\mathbf{I}_N\right)^{-1}\boldsymbol{t}$$

- With Gramm matrix $\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$ with $K_{ij} = \boldsymbol{\phi}^T(\boldsymbol{x}_i)\boldsymbol{\phi}(\boldsymbol{x}_j)$

Kernelized Ridge Regression

- Goal: Minimize sum of squared errors with quadratic weight penalty

$$J(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - t_n\}^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

- Solution: Solve $\dfrac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 0$:

$$\boxed{\mathbf{w} = \boldsymbol{\Phi}^T \underbrace{\left(\boldsymbol{K} + \lambda \mathbf{I}_N\right)^{-1} \boldsymbol{t}}_{a}}$$

$$\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T, \; K_{ij} = \boldsymbol{\phi}^T(\boldsymbol{x}_i)\boldsymbol{\phi}(\boldsymbol{x}_j)$$

- Primal/dual viewpoint
  - Primal variable: $\mathbf{w} = \boldsymbol{\Phi}^T\mathbf{a}$

    $\arg\min_{\mathbf{w},\mathbf{z}} \frac{1}{2}||\mathbf{z}||^2$ with constraints $\mathbf{z} = \boldsymbol{\Phi}\mathbf{w} - \mathbf{t}$ and $\frac{1}{2}||\mathbf{w}||^2 \leq R^2$
  - Dual variable: $\mathbf{a} = \left(\boldsymbol{K} + \lambda \mathbf{I}_N\right)^{-1} \boldsymbol{t}$

    $\arg\min_{\mathbf{a}} \frac{1}{2}\mathbf{a}^T\mathbf{K}\mathbf{a} - \mathbf{a}^T\mathbf{t} + \frac{\lambda}{2}||\mathbf{a}||^2$

- Predictive mean of primal viewpoint $y(\mathbf{x}', \mathbf{w}) = \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}')$

  of dual viewpoint $y(\mathbf{x}', \mathbf{a}) = \sum_{n=1}^{N} a_n k(\mathbf{x}_n, \mathbf{x}')$

Primal vs Dual/Kernel Approach

- Computational cost (closed form solutions):
  - The dual variables. $\boldsymbol{a} = (\underbrace{\boldsymbol{K} + \lambda \mathbf{I}_N}^{N \times N})^{-1} \boldsymbol{t}$ $\qquad O(N^3)$
  - The primal variables $\mathbf{w} = (\underbrace{\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \lambda \mathbf{I}_M}_{M \times M})^{-1} \boldsymbol{\Phi}^T \boldsymbol{t}$ $\qquad O(M^3)$
- Computational cost (predictions):
  - Dual case: $y(\mathbf{x}', \mathbf{a}) = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \mathbf{x}')$ $\qquad O(NM)$
  - Primal case: $y(\mathbf{x}', \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}')$ $\qquad O(M)$
- But… dual approach:
  - No explicit parameters (implicitly many!) -> nonparametric model $\quad M = \infty$
  - Does not rely on explicit features but on similarity kernel function.
  - Can be slow at prediction $\quad N' \ll N$
  - Upcoming: **Kernel methods with sparse solutions**! $\quad O(N'M)$