Machine Learning 1
Lecture 11.1 - Kernel Methods
Kernelizing Linear Models

Erik Bekkers

(Bishop 6.0, 6.1)
So Far: Parametric Models

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

\[
y(x, W^{(1)}, W^{(2)}) = \sum_{m=0}^{M} w_m^{(2)} h\left( \sum_{d=0}^{D} w_{md}^{(1)} x_d \right)
\]

- Training:
  - MLE, MAP: use training data to obtain point estimate of \( w \)
  - Full Bayesian: use training data to obtain posterior \( p(w | X, t) \)
- Test time: Discard training data, only need \( w \) or \( p(w | X, t) \)
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 $M \gg N$

- Linear parametric models:
  - Can be re-cast into equivalent ‘dual representation’
  - Predictions are based on linear combinations of the kernel function evaluated at training data points

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

$$k(x, x') = \phi(x)^T \phi(x')$$

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

$$k(x, x') = k(x', x)$$
Kernelized Ridge Regression

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

\[ J(w) = \frac{1}{2} \sum_{n=1}^{N} \{ w^T \phi(x_n) - t_n \}^2 + \frac{\lambda}{2} w^T w \]

- Solution: Solve

\[ \frac{\partial J(w)}{\partial w} = 0: \]

\[ \frac{\partial J(w)}{\partial w} = \sum_{n=1}^{N} \{ w^T \phi(x_n) - t_n \} \phi(x_n)^T + \lambda w^T I = 0 \]

\[ \Leftrightarrow \quad w^T \left( \sum_{n=1}^{N} \phi(x_n) \phi(x_n)^T + \lambda I \right) = \sum_{n=1}^{N} t_n \phi(x_n)^T \]

\[ \Leftrightarrow \quad \left( \sum_{n=1}^{N} \phi(x_n) \phi(x_n)^T + \lambda I \right) w = \sum_{n=1}^{N} t_n \phi(x_n) \]

\[ w = \left( \sum_{n=1}^{N} \phi(x_n) \phi(x_n)^T + \lambda I \right)^{-1} \sum_{n=1}^{N} t_n \phi(x_n) = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T t \]
Kernelized Ridge Regression

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

\[ J(w) = \frac{1}{2} \sum_{n=1}^{N} \{ w^T \phi(x_n) - t_n \}^2 + \frac{\lambda}{2} w^T w \]

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

\[ w = (\Phi^T \Phi + \lambda I_M)^{-1} \Phi^T t, \]

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

\[ (P^{-1} + B^T R^{-1} B)^{-1} B^T R^{-1} = P B^T (BPB^T + R)^{-1} \]

\[ \left( \lambda I_m + \Phi^T I_N \Phi \right)^{-1} = \Phi \Phi^T \]

- Allows us to alternatively obtain \( w \) via

\[ w = \Phi^T (\Phi \Phi^T + \lambda I_N)^{-1} t = \Phi^T (K + \lambda I_N)^{-1} t \]

- With Gramm matrix \( K = \Phi \Phi^T \) with \( K_{ij} = \phi^T(x_i)\phi(x_j) \)
Kernelized Ridge Regression

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

\[ J(w) = \frac{1}{2} \sum_{n=1}^{N} \{ w^T \phi(x_n) - t_n \}^2 + \frac{\lambda}{2} w^T w \]

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

\[ w = \Phi^T (K + \lambda I_N)^{-1} t \]

\[ K = \Phi \Phi^T, \quad K_{ij} = \phi^T(x_i) \phi(x_j) \]

- Primal/dual viewpoint

  - Primal variable: \( w = \Phi^T a \)

  - Dual variable: \( a = (K + \lambda I_N)^{-1} t \)

- Predictive mean of primal viewpoint \( y(x', w) = w^T \phi(x') \)

- Predictive mean of dual viewpoint \( y(x', a) = \sum_{n=1}^{N} a_n k(x_n, x') \)
Primal vs Dual/Kernel Approach

- Computational cost (closed form solutions):
  - The dual variables: \( a = (K + \lambda I_N)^{-1} t \) \( O(N^3) \)
  - The primal variables: \( w = (\Phi^T \Phi + \lambda I_M)^{-1} \Phi^T t \) \( O(M^3) \)

- Computational cost (predictions):
  - Dual case: \( y(x', a) = \sum_{n=1}^{N} \alpha_n k(x_n, x') \) \( O(NM) \)
  - Primal case: \( y(x', w) = w^T \phi(x') \) \( O(M) \)

- But… dual approach:
  - No explicit parameters (implicitly many!) \(-\) nonparametric model
  - Does not rely on explicit features but on similarity kernel function.
  - Can be slow at prediction
  - Upcoming: **Kernel methods with sparse solutions**! \( \mathcal{O}(N'M) \)