ASSIGNMENT 5: VC-DIMENSION

Institute for Machine Learning

JOHANNES KEPLER UNIVERSITY LINZ

Institute for Machine Learning

Contact

Heads: Johannes Kofler,
Markus Holzleitner

Institute for Machine Learning
Johannes Kepler University
Altenberger Str. 69
A-4040 Linz

E-Mail: theoretical@ml.jku.at
Only mails to this list are answered!
Institute Homepage

Copyright statement:

Agenda: Assignment 5 – VC-Dimension

- Statistical learning: main ideas/intuition
- Statistical learning: more formal
- Shattering and VC-dimension

Statistical Learning Theory (SLT)

- **Learning from data**
  With finitely many training examples we can select a model close to the optimal model for future data.
- **Empirical Risk Minimization (ERM)**
  With increasing number of training examples a model converges to the best model for future data.
- How close we can get depends on:
  - complexity of the task
  - number of training examples
  - complexity of the model (and model class)
- SLT provides mathematical framework to analyze performance of machine learning models, taking into account a tradeoff between these three building blocks

Empirical Risk Minimization

- If the training set is explained by the model then the model generalizes to future data
  $\Longrightarrow$ **minimize the training error**
- Risk R

$$\mathrm{R} = E_{\mathbf{z}}(L(g(\mathbf{x}), y))$$

- Empirical Risk R_{emp}

$$\mathrm{R}_{\text{emp}} = \frac{1}{n} \sum_{i}^{n} L(g(\mathbf{x}^i), y^i)$$

Finite Set Error Bound

- We choose a model g from a finite set of functions $\{\mathrm{g}_1, \ldots, \mathrm{g}_M\}$, then using the *union bound* and *Hoeffding's inequality*, we get

$$\mathrm{R}(\mathrm{g}) \leq \mathrm{R}_{\textsf{emp}}(\mathrm{g}, l) + \underbrace{\epsilon(l, M, \delta)}_{\text{complexity}}$$

$$\epsilon(l, M, \delta) = \sqrt{\frac{ln(M) - ln(\delta)}{2l}}$$

l … number of training examples

M … number of possible functions

δ … prob of bound being violated

- However, in most machine learning applications M is not finite!

VC-Dimension

- **Idea:** on the training set only a finite number of functions can be distinguished
- $\implies$ Complexity measure: number of data points for which a model class can assign all possible binary labeling vectors
- We only consider classification, but with a slight tweak applicable to regression
- If a model class can produce all 2^l possible labellings we call it *shattering*

VC-Dimension

- **Shattering Coefficient**: Given a dataset $\{\mathbf{x}_1, \dots, \mathbf{x}_l\}$, we can define how many labellings a binary model class $\mathcal{F}$ can produce

$$\mathrm{N}_{\mathcal{F}}(\{\mathbf{x}_1, \dots, \mathbf{x}_l\})$$

- **Growth Function**

$$\mathrm{G}_{\mathcal{F}}(l) = \ln \sup_{\{\mathbf{x}_1, \dots, \mathbf{x}_l\}} \mathrm{N}_{\mathcal{F}}(\{\mathbf{x}_1, \dots, \mathbf{x}_l\})$$

- We call the maximal number of points l for which a model class can produce all 2^l labels the **VC-dimension** d_{VC}:

$$d_{VC} = \max_{l} \{l | G_{\mathcal{F}}(l) = \ln 2^l\}$$

- If this maximum does not exist we set $d_{VC} = \infty$

VC-Dimension

- d_{VC} bounds the Growth Function

$$G_{\mathcal{F}}(l) \begin{cases} = \ l \ \ln(2) & \text{if } l \leq d_{VC} \\ \leq \ d_{VC}(1 + \ln \frac{l}{d_{VC}}) & \text{if } l > d_{VC} \end{cases}$$

- $\implies$ a model class with finite VC-dimension is consistent and converges fast since

$$\lim_{l \to \infty} \frac{G_{\mathcal{F}}(l)}{l} = 0$$

is necessary and sufficient for fast convergence

Error Bound

- We can now replace M in the previous error bound with d_{VC} for a more useful bound
- Finite Set Error Bound

$$\mathrm{R}(\mathrm{g}) \leq \mathrm{R}_{\mathsf{emp}}(\mathrm{g}, l) + \epsilon(l, M, \delta)$$

- Vapnik Bound

$$\mathrm{R}(\mathrm{g}) \leq \mathrm{R}_{\mathsf{emp}}(\mathrm{g}, l) + \sqrt{\epsilon(l, \delta)}$$

with

$$\epsilon(l, \delta) = \frac{8}{l}\left(d_{VC}\left(\ln\left(\frac{2l}{d_{VC}}\right) + 1\right) + \ln\frac{4}{\delta}\right)$$

Remarks

- Model classes with finite *VC-dimension* are consistent and converge fast.
- A finite VC-dimension can be used to calculate a theoretical bound on the risk.
- However, in practice this bound is often not useful because only for large l it gives a non-trivial value

Structural Risk Minimization

- It makes sense to minimize not only R_{emp} but also $\epsilon(l, \delta)$, i.e. control the complexity of the model
- Structural Risk Minimization
  1. Using a priori knowledge of the domain, choose a class of functions
  2. Divide the class of functions into a hierarchy of nested subsets in order of increasing complexity
  3. Perform ERM on each subset
  4. Select the model in the series with the minimal sum of R_{emp} and $\epsilon(l, M)$

Structural Risk Minimization