Technische Hochschule Rosenheim

Modul - Unsupervised and Reinforcement Learning (URL)

Bachelor Programme AAI

12 - Summary

Prof. Dr. Marcel Tilly

Faculty of Computer Science, Cloud Computing

Agenda

Technische Hochschule Rosenheim

On the menu for today:

- Summary

> Code can be found on Github
>
> - Github
> - hosted on myBinder

00: Introduction

Lecture

Goals (formal*)

Technische Hochschule Rosenheim

Students ...

- ... know fundamentals and details of unsupervised and reinforcement learning in contrast to supervised learning.
- ... know various techniques in unsupervised and reinforcement learning.
- ... can use and implement algorithms for unsupervised and reinforcement learning.
- ... can decide which approach can be applied to which problem.
- ... can analyse and measure the quality of various models for unsupervised and reinforcement learning.

(*taken from module handbook)

What does it really mean?

Technische Hochschule Rosenheim

Students of AAI should definitely ...

- ... know k-means and other clustering algorithms
- ... know about feature reduction
- ... know about reinforcement and unsupervised learning
- ... know about Q-learning and Deep Q-Networks
- ... be able to code all these things in Python!

Machine Learning

Technische Hochschule Rosenheim

Supervised

- Features and labels are provided
- Finds pattern in existing structure
- Uses techniques such as regression and classification
- What can we do with this?

Unsupervised

- No labels are provided
- Finds structure in unlabeled data
- Uses techniques such as clustering and dimensionality reduction
- What can we do?

Machine Learning

Technische Hochschule Rosenheim

Supervised

- We are given input/output samples (X, y) which we relate with a function y = f(X). We would like to “learn” f, and evaluate it on new data.
- Types:
  - **Classification**: y is discrete (class labels).
  - **Regression**: y is continuous, e.g. linear regression.

Unsupervised

- Given only samples X of the data. we compute a function f such that y = f(X) is in a “group”.
- Types:
  - **Clustering**: y is discrete
  - Y is continuous: **Matrix factorization, Kalman filtering, unsupervised neural networks**.

01: Clustering

Lecture

Distance Function for Vectors

Technische Hochschule Rosenheim

Suppose there are two points x and y in R^n:

- $x = (x_1, x_2, \ldots, x_n)^T$
- $y = (y_1, y_2, \ldots, y_n)^T$
- **Minkowski Distance**: $d(x, y) = \sqrt[p]{\sum_{i=1}^{n} (x_i - y_i)^p}$
- **Euclidean Distance**: p=2 $\Rightarrow d(x, y) = \sqrt[2]{\sum_{i=1}^{n} (x_i - y_i)^2}$
- **Manhattan Distance**: p=1 $\Rightarrow d(x, y) = \sum_{i=1}^{n} (x_i - y_i)$

k-means

Technische Hochschule Rosenheim

Let the set of data points D be $(X_1, X_2, \ldots, X_n)$, where $X_i = (x_1, x_2, \ldots, x_r)^T$ is a vector in $X \in R^r$, and r is the number of dimensions.

- The k-means algorithm partitions the given data into k clusters:
  - Each cluster has a cluster center, called centroid.
  - k is specified by the user

1. Choose k (random) data points (seeds) to be the initial centroids, cluster centers
2. Assign each data point to the closest centroid
3. Re-compute the centroids using the current cluster memberships
4. If a convergence criterion is not met, repeat steps 2 and 3

k-means Convergence

Technische Hochschule Rosenheim

Stopping criterion

- no (or minimum) re-assignments of data points to different clusters, or
- no (or minimum) change of centroids, or
- minimum decrease in the sum of squared error (SSE),

$$SSE = \sum_{j}^{k} \sum_{x} d(x, m_j)$$

– C_j is the jth cluster,

– m_j is the centroid of cluster C_j (the mean vector of all the data points in C_j),

– $d(x, m_j)$ is the (Euclidean) distance between data point x and centroid m_j.

Weaknesses of k-means

Technische Hochschule Rosenheim

- The user needs to specify k. How to find k?

A simple rule of thumb (but not great!):

$$k = \sqrt{n/2}$$

with n=number of samples

> BETTER: Elbow- or Silhouette methods!

- The algorithm is sensitive to outliers
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

Picking k: Elbow Method

Technische Hochschule Rosenheim

Basic idea:

1. specify k = 2 as the initial optimal cluster number k and then increase k by step 1 to the maximal specified k
2. find centroids for given k
3. calculate the cost C as the mean distortion
    - the sum of the squared Euclidean distances (SSE) divided by N (number of data points) per each cluster
4. increase k and repeat 2.- 4.

- The optimal cluster number k is the point from where the cost C remains almost unchanged
- This point is called *cost peak value*

Silhouette Score

Technische Hochschule Rosenheim

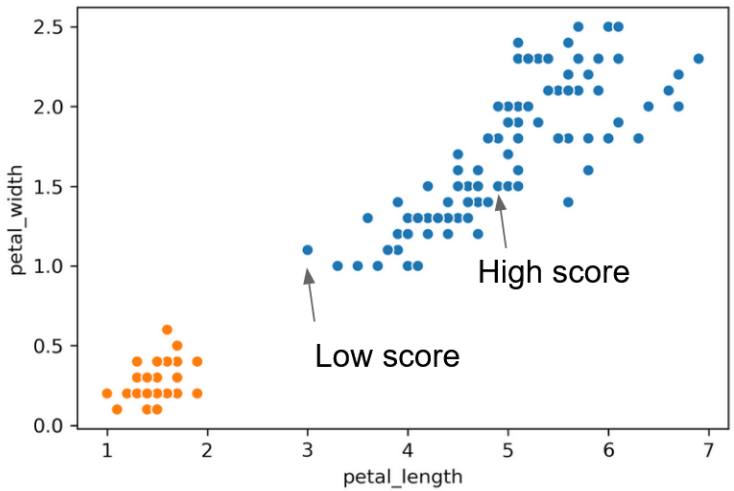
To evaluate how "well clustered" a specific data point is, we can use the "silhouette score", a.k.a. The "silhouette width".

- **High score**: Near the other points in its X's cluster.
- **Low score**: Far from the other points in its cluster.

For a data point X the *Silhouette score* S is:

- A = average distance to other points in the cluster.
- B = average distance to points in the closest cluster.

$$S = \frac{B - A}{max(A, B)}$$



Hierarchical Clustering

Technische Hochschule Rosenheim

- As with *regression* and *classification*, there are many ways to do clustering.
- So far we've seen K-Means, which attempts to minimize distortion.
  - Results not guaranteed to optimize distortion.
  - Even global optimum may not match our intuition of the best result.
- Let's discuss an alternate idea known as *hierarchical clustering*.
- Basic idea:
  - Every data point starts as its cluster.
  - Join clusters with neighbors until we have only k clusters left.

Let's see an example for k = 2.

02: DBSCAN

Lecture

Density-Based Clustering

Technische Hochschule Rosenheim

Basic Concept

- Intuition for the formalization of the basic idea
  - For any point in a cluster, the local point density around that point has to exceed some threshold
  - The set of points from one cluster is spatially connected
- Local point density at a point q defined by two parameters
  - **ε–radius (eps)** for the neighborhood of point q:

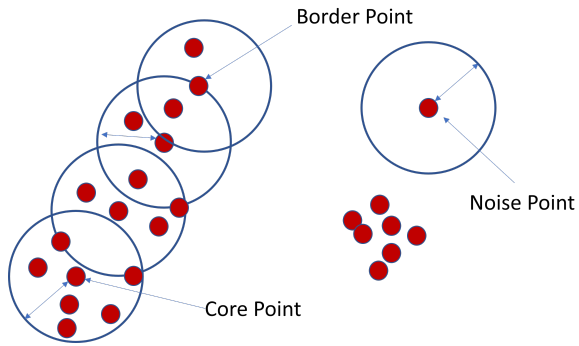
$$N_\varepsilon(q) := \{p \in D | dist(p,q) \leq \varepsilon\}$$

  - **MinPts** – minimum number of points in the given neighborhood $N_\varepsilon(q)$
- q is called a **core object** (or **core point**) if $|N_\varepsilon(q)| \geq MinPts$

Types of Points in DBSCAN Clustering

Technische Hochschule Rosenheim

- **Core point**: A core point is one in which at least have minPts number of points (including the point itself) in its surrounding region within the radius eps: $if |N (p)|≥ MinPts§
- **Border point**: has fewer than MinPts within its ϵ-neighborhood (N), but it lies in the neighborhood of another core point.
- **Noise point**: A noise point (**outlier**) is neither a core point nor is it reachable from any core points.



k-mean vs. HClustering vs. DBSCAN

Technische Hochschule Rosenheim

- kmeans and hierarchical clustering are good when you have some idea regarding the number of clusters in your data.
- DBSCAN, instead, takes a more bottom-up approach by working with your hyperparameters and finding the clusters it views as important. It is robust to noise and can detect arbitrarily-shaped clusters.
- Compared to k-means and hierarchical clustering, DBSCAN can be seen as being potentially more efficient, since it only has to look at each data point once.
- K-means and hierarchical clustering require multiple iterations of finding new centroids and evaluating where their nearest neighbors are, once a point has been assigned to a cluster in DBSCAN, it does not change cluster membership.
- No need to pass several clusters to DBSCAN and hierarchical clustering both share, in comparison with k-means.

03: Data Preparation

Lecture

Encoding

- LabelEncoding
- One-hot Encoding

Label Encoding

Technische Hochschule Rosenheim

- The categorical value represents the numerical value of the entry in the dataset.
  - For example: if there were to be another company in the dataset, it would have been given a categorical value of 3. As the number of unique entries increases, the categorical values also proportionally increase.

CompanyName	Categoricalvalue	Price
VW	1	20000
Acura	2	10011
Honda	3	30000
Honda	3	10000

- The categorical values start from 0 goes all the way up to N-1 categories.

> Why it can be problematic to use label encoding for model training?

One-hot Encoding

Technische Hochschule Rosenheim

- One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction.

CompanyName	Categoricalvalue	Price
VW	[1,0,0]	20000
Acura	[0,1,0]	10011
Honda	[0,0,1]	30000
Honda	[0,0,1]	10000

What to do with different data?

Technische Hochschule Rosenheim

- **Scale** generally means to change the range of the values. The shape of the distribution doesn't change. Think about how a scale model of a building has the same proportions as the original, just smaller. That's why we say it is drawn to scale. The range is often set at 0 to 1.
- **Standardize** generally means changing the values so that the distribution's standard deviation equals one. Scaling is often implied.
- **Normalize** can be used to mean either of the above things (and more!). I suggest you avoid the term normalize because it has many definitions and is prone to creating confusion.

Technische Hochschule Rosenheim

04/05: Dimensionality Reduction and PCA

04-Lecture

05-Lecture

Dimensionality Reduction

Technische Hochschule Rosenheim

What is the objective?

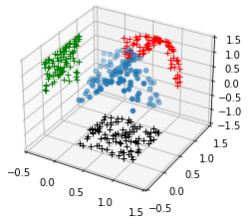
- Depending upon the problem being solved, or the origin of this dataset, we may want to reduce the number of dimensions per sample without losing the provided information.
- Choose an optimum set of features of lower dimensionality to improve classification accuracy.
- By using dimensionality reduction, we are able to remove much of the noise from the signal, which, in turn, will assist with the performance of the algorithms that are applied to the data as well as reduce the size of the dataset to allow for reduced hardware requirements.

Principal Component Analysis (PCA)

Technische Hochschule Rosenheim

Principal Component Analysis (PCA) is an exploratory approach to reduce the data set's dimensionality to 2D or 3D, used in exploratory data analysis for making predictive models. Principal Component Analysis is a linear transformation of data set that defines a new coordinate rule such that:

- The highest variance by any projection of the data set appears to laze on the first axis.
- The second biggest variance on the second axis, and so on.



Methodology (simplified!)

- Start from m x n data matrix X
  - Standardize and **Recenter**
- Compute covariance matrix
- Find eigenvectors and eigenvalues
- Principal components
  - PC1: eigenvector with highest eigenvalues
  - PC2: eigenvector with highest eigenvalues

06: t-SNE

Lecture

Stochastic Nearest Neighbor Embeddings

Technische Hochschule Rosenheim

- Stochastic Nearest Neighbor Embeddings is an alternative dimensionality reduction algorithm
  - t-SNE is the most popular one
- PCA tries to find a global structure
  - Low dimensional subspace
  - Can lead to local inconsistencies
  - Far away point can become nearest neighbors
- t-SNE tries to preserve local structure
  - Low dimensional neighborhood should be the same as original neighborhood.
- Unlike PCA almost only used for visualization

t-SNE - Idea

Technische Hochschule Rosenheim

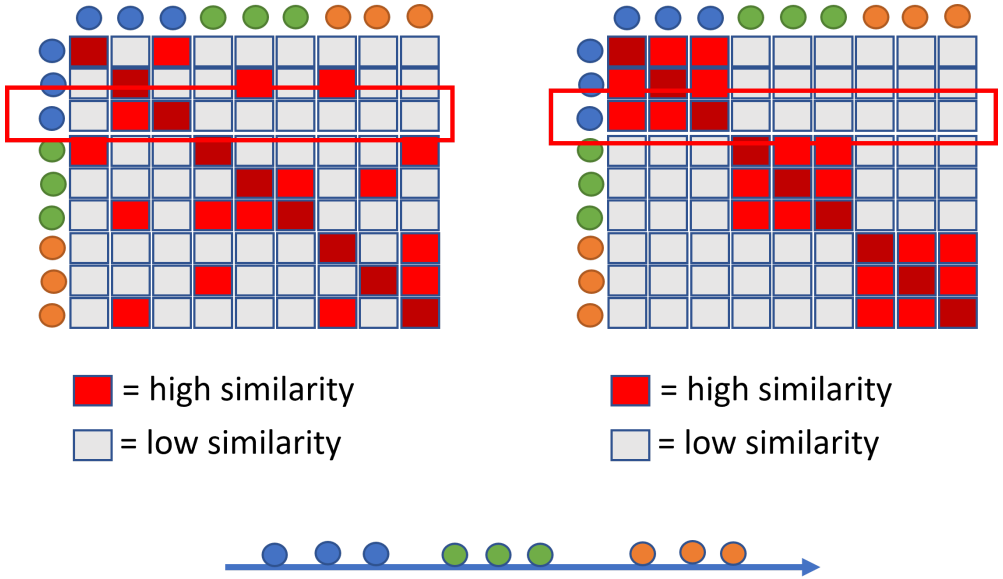
> Main Idea: Map near points of high-dimensional space to a near position in low-dimensional space.

1. Measure euclidean distance in high dim and convert to probability of picking a point as a neighbor (similarity is proportional to probability)
    - use Gaussian distribution for density of each point
2. In low-dimension do same as under 1. in low dimensionality (for t-SNE use t-distribution - has heavier tails)
3. Minimize the difference of the conditional probabilities (KL-divergence)

- **Input space**: building P using probability distributions so that picking two similar points is much more probable than picking two points which are well far apart.
- **Output space**: building Q using probability distributions to judge distances between points

t-SNE Algorithm step 5

- t-SNE tries to create a matrix of scores which looks like the original by moving each point a bit at a time
- Each step t-SNE chooses a direction that makes the matrix on the left more like the matrix on the right



Technische Hochschule Rosenheim

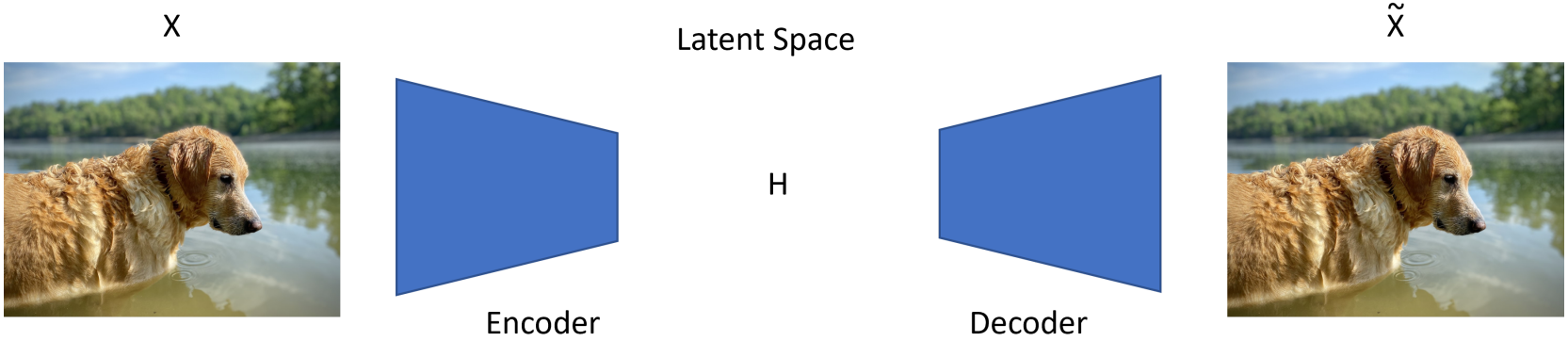
07: Autoencoder

Lecture

Autoencoder

Technische Hochschule Rosenheim

We train the two NN by minimizing the reconstruction **loss function** $L = \sum (x_i - \tilde{x}_i)^2$

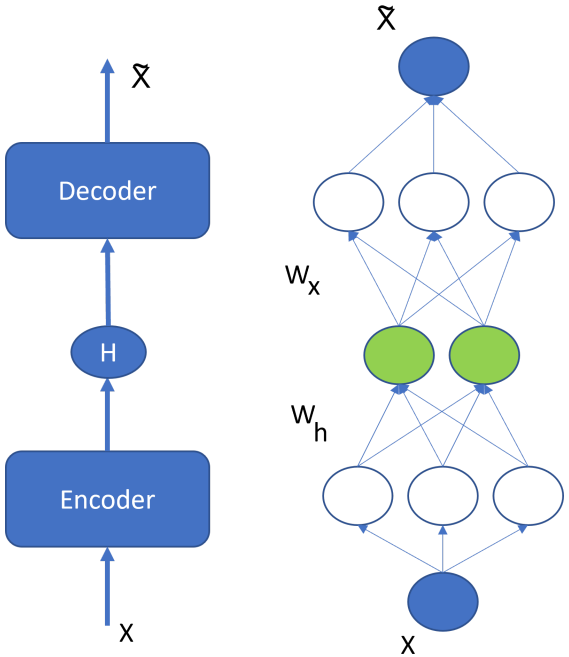


An *autoencoder* finds the best way to encode the input so that the decoded version is as close as possible to the input.

Autoencoder

- The idea is to learn how to encode the input into a *compressed* form, and then re-generate the input from the encoding.
- An encoder network learns how to map the input x to a code h in the *latent space* of smaller dimension.
- And at the same time, a *decoder* network learns to map the code h back to an *approximate* from of the input $\tilde{x}$

> This is an *unsupervised learning* task!



08/09/10: Reinforcement Learning

08- Lecture

09- Lecture

10- Lecture

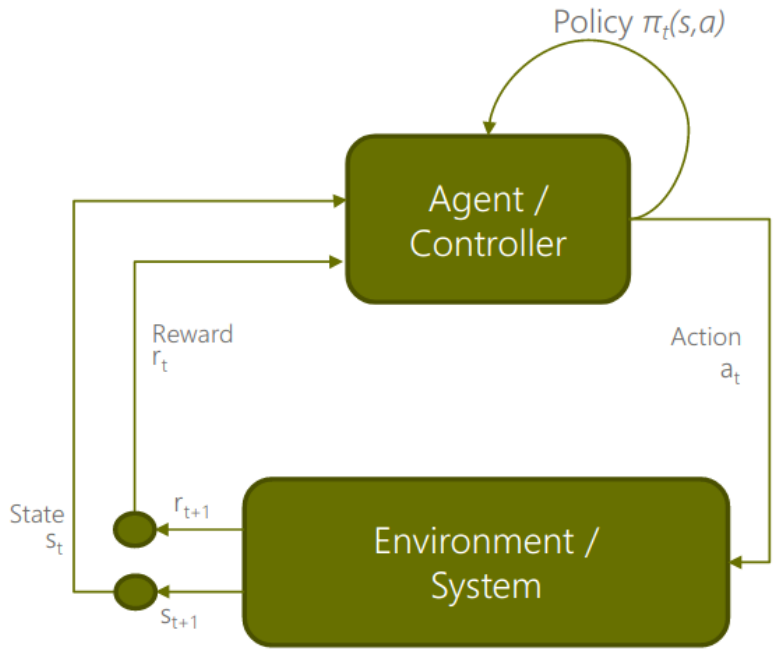
Agent and Environment

At each step the agent

- receives state observations S_t
- receives scalar reward R_t
- executes action A_t

The environment

- receives action A_t
- emits state observation S_{t+1}
- emits scalar reward R_{t+1}



Markov Decision Process (MDP)

Technische Hochschule Rosenheim

MDP is represented by five important elements:

- A set of states S the agent can actually be in
- A set of actions A that can be performed by an agent, for moving from one state to another
- A transition probability $P^{a}_{ss'}$, which is the probability of moving from one state S to another S' by performing an action a
- A reward probability $R^{a}_{ss'}$, which is the probability of a reward acquired by the agent for moving from one state S to another S' by performing some action a
- A discount factor γ, which controls the importance of immediate and future rewards

MDPs, MDP free and RL

Technische Hochschule Rosenheim

Known MDP: Offline Solution

Goal	Technique
Compute V*, Q*, π*	Value / policy iteration
Evaluate a fixed policy π	Policy evaluation

Unknown MDP: Model-Based

Goal	Technique
Compute V*, Q*, π*	VI/PI on approx. MDP
Evaluate a fixed policy π	PE on approx. MDP

Unknown MDP: Model-Free

Goal	Technique
Compute V*, Q*, π*	Q-learning
Evaluate a fixed policy π	Value Learning

Summary

Technische Hochschule Rosenheim

Lessons learned:

- Clustering
  - k-means, hierarchical clustering, DBSCAN
- Data Preparation and Normalization
- Dimensionality Reduction
  - PCA, t-sne, Autoencoder
- RL
  - MDPs, Value iteration, Policy iteration and policy optimization, Q-Learning

> Code can be found on Github
>
> - Github
> - hosted on myBinder