

MACHINE LEARNING

Overview

Key Concepts

Formalism:

- Be capable of giving formal definitions of a pdf, marginal, likelihood
- Be capable of giving principle of basic ML algorithms such as maximum likelihood and E-M, Maximum A Posteriori (MAP)

Taxonomy:

- Know the difference between supervised / unsupervised learning, reinforcement learning and be able to give examples of algorithms in each case
- Be able to discuss concepts such a parametric vs. non-parametric, generative vs. discriminative methods

Principles of evaluation:

- Know the basic principles of evaluation of ML techniques: training vs. testing sets, crossvalidation, ground truth, ROC curve

Key Concepts

- For each algorithm, be able to explain:
 - what it can do: classification, regression, structure discovery / reduction of dimensionality
 - what one should be careful about (limitations of the algorithm, choice of hyperparameters) and how does this choice influences the results.
 - the key steps of the algorithm, its hyperparameters, the variables it takes as input and the variables it outputs

Key Concepts: Example

In red: what you should know; in blue, what would be good to know / bonus.

- For each algorithm, be able to explain:

SVM

– what it can do: classification, regression, structure discovery / reduction of dimensionality

Performs *binary classification*; can be extended to multi-class classification; can be extended to regression (SVR)

– what one should be careful about (limitations of the algorithm, choice of hyperparameters)

e.g. choice of kernel; too small kernel width in Gaussian kernels may lead to overfitting; one can proceed to iterative estimation of the kernel parameters

– the key steps of the algorithm, its hyperparameters, the variables it takes as input and the variables it outputs

Exam Format

The exam lasts a total of 40 minutes:

- Upon entering the room, you get a choice of 3 questions; you can get rid of 1 question!
- Spend 20 minutes in the back of the room preparing answers to the two questions you have picked
 - When needed make schematic or prepare an example
- Present for 20 minutes your answers on the black board.

Exam is closed book but you can bring one A4 page with personal notes written recto-verso.

Example of exam question - I

Exam questions will entail two parts: one conceptual and one algorithmic

- i. Explain backpropagation in feedforward ANN
(to answer this type of question, you should make a schematic that includes all variables and explain with the schematic or with equations how the weights are updated via backpropagation)*
- ii. Is one constrained to using solely continuous variables in input / output? If yes, explain why. If not, explain how to do it.*

Example of question - II

Exam questions will entail two parts: one conceptual and one algorithmic

- ii. What are the pros and the cons of Boosting compared to bagging ?*
- ii. How can we derive the AdaBoost algorithm from a loss minimization?*

Class Overview

This overview is meant to solely highlight similarities and differences across the different methods presented in class.

To be well prepared to the exam, read carefully the lecture notes and the slides.

Class Overview

This class has presented groups of methods for doing the classification, regression, structure discovery, estimation of time series.

Note that several algorithms do more than one of these types of computation.

Structure Discovery

Kernel – PCA, ICA, CCA

GPLVM

HMM

Regression

SVR
GMR
GPR
LWPR
ANN

Classification / Clustering

Decision Trees
+ boosting/bagging

SVM

ANN

GMM + Bayes

Time Series

RL

RNN / TDNN

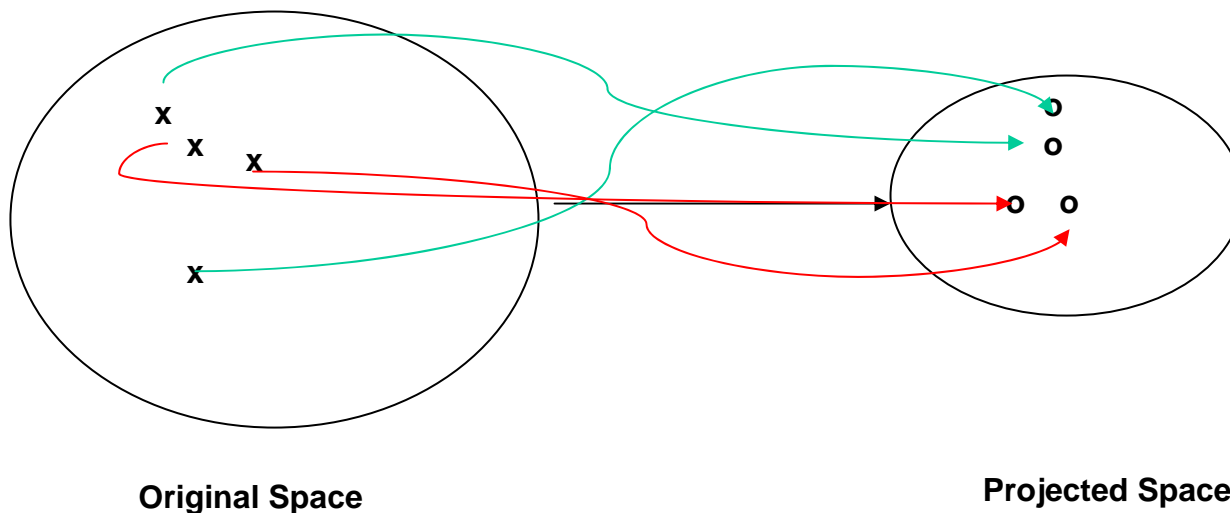
HMM

Overview: Finding Structure in Data

Techniques for finding structure in data proceed by projecting the data from the original space into another space of either lower dimension or higher dimension.

The projected space is chosen so as to highlight particular features common to subsets of datapoints.

The found structure may be exploited in a second stage by another algorithm for regression, classification, etc.



Overview: Finding Structure in Data

Classical techniques for finding structure in data:

- Linear PCA & Probabilistic PCA
- Non-linear PCA / kernel PCA
- kernel CCA
- Sparse methods, kernel k-means (not presented in class)

But also implicit extraction of structure in more complex techniques such as:

- GMM, GP
- GPLVM, SVM, LWPR
- HMM

WHICH METHOD, WHEN?

We have seen several techniques all based on the same problem:

Given a dataset X , find a projection $Z=WX \leftrightarrow X=W^T Z$.

PCA

PPCA

CCA

ICA

The kernel based methods performs the same computation as the linear methods but in *feature space*.

Given a dataset X , find a non-linear projection $Y=\phi(X)$ and then search for $Z=WY$

Kernel PCA

Kernel CCA

Kernel ICA

Each of these methods optimize for something else and hence must be used appropriately!

Overview: Finding Structure in Data

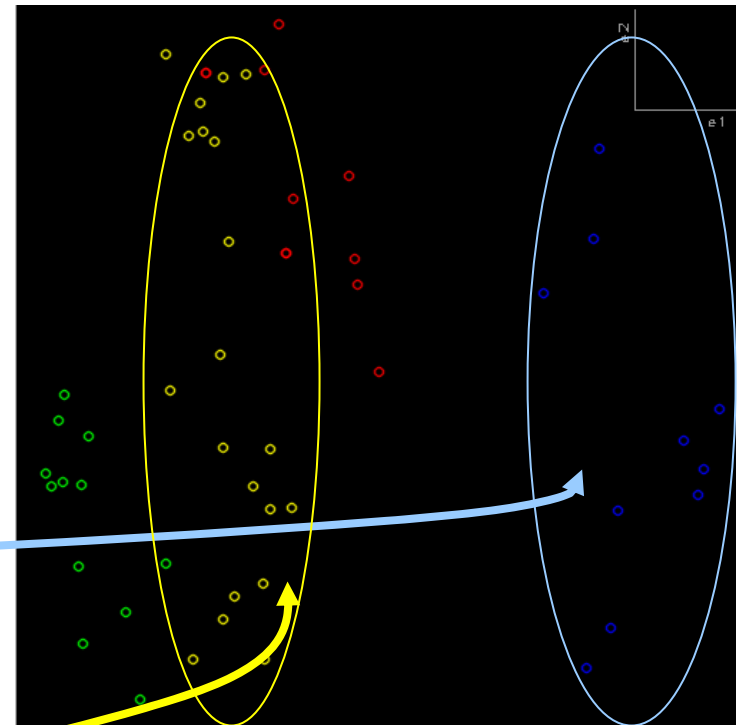
Principal Component Analysis (PCA)



$$x \in \mathbb{R}^N$$

$$Z = WX$$

$$z \in \mathbb{R}^q, \quad q \leq N$$



- Determines what is most common across datapoints.
- Projects onto axes that maximize correlation (eigenvectors of covariance matrix).
- Discard projections with the smallest eigenvalues.

Overview: Finding Structure in Data

Probabilistic Principal Component Analysis (PPCA)

Probabilistic PCA, like PCA, find projections that lead to minimal reconstruction error.

Differences:

- assumes z follows a probabilistic distribution
- assumes data x are subjected to additive noise
- does not assume data are zero-mean

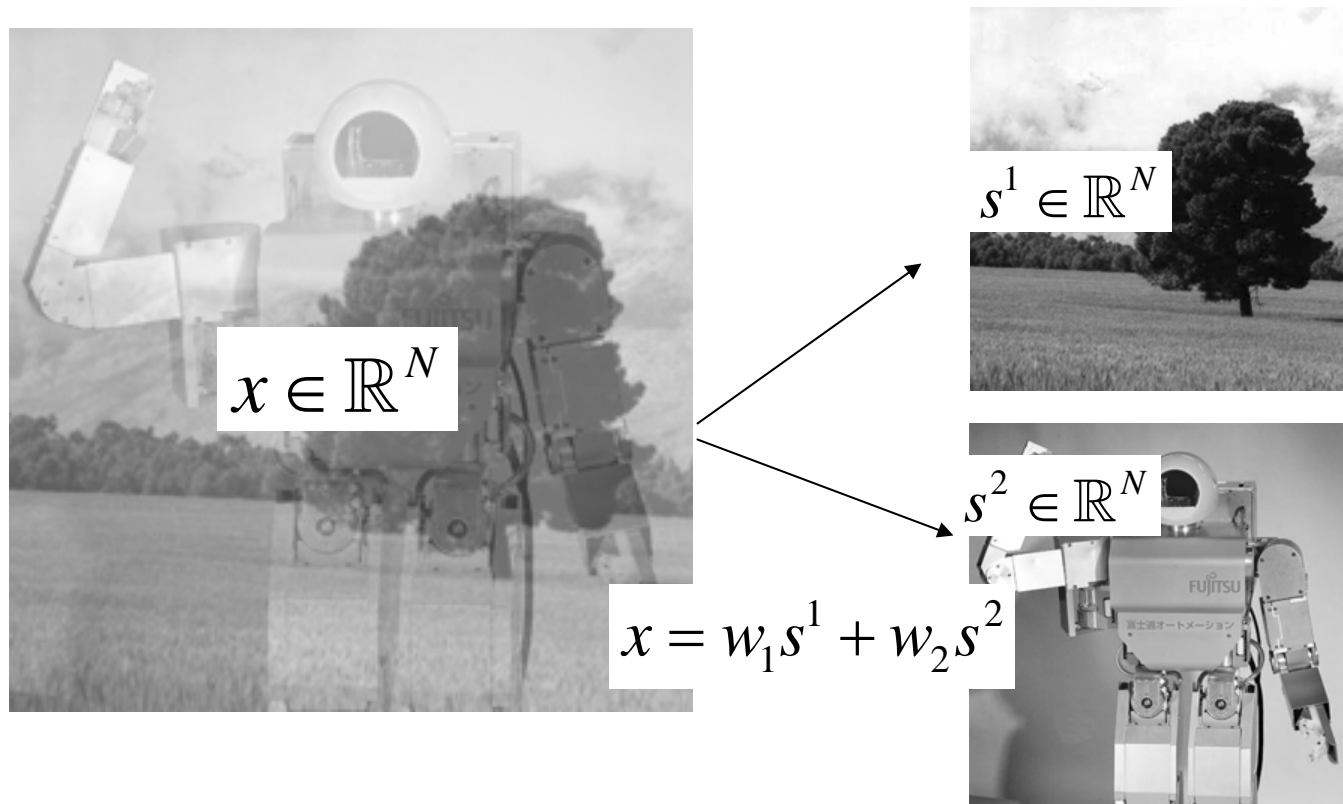
$$x = W^T z + \mu + \varepsilon, \quad z \sim N(0, I), \quad \varepsilon \sim N(0, \sigma_\varepsilon)$$

$$x \in \mathbb{R}^N \quad z \in \mathbb{R}^q, \quad q \leq N$$

- Find W and μ, σ_ε by optimizing likelihood of the model given the data.
- With missing data, exploits expectation-maximization.

Overview: Finding Structure in Data

Independent Component Analysis (ICA)



- Extract from the data the set of statistically independent sources:
- Determine the number of sources (e.g. with PCA first).
- Maximize statistical independence across projections: $p(s_1, s_2) = p(s_1)p(s_2)$
- Find each source iteratively (minimization of negentropy and orthog. of source)

Overview: Finding Structure in Data

Canonical Correlation Analysis (CCA)

$$x \in \mathbb{R}^N$$

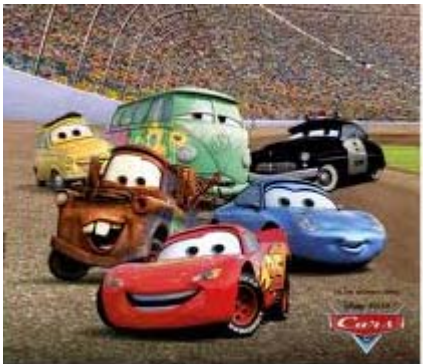
$$y \in \mathbb{R}^P$$



$$\{x^1, y^1\}$$

$$\max_{w^x, w^y} \text{corr}(w^x x, w^y y)$$

$$\{x^2, y^2\}$$



Video description

Audio description

Determine features in two (or more) separate descriptions of the dataset that best explain each datapoint. Extract hidden structure that maximize correlation across two different projections.

Overview: Finding Structure in Data

Kernel Methods for Determining Structure in Data

- allow to extract *non-linear* types of correlations.

- exploit the **Kernel Trick**:

Is based on the observation that all *linear* methods for finding structure in data are based on computing an *inner product* across variables.

This inner product can be replaced by the kernel function if known. The problem becomes then *linear in feature space*.

$$k : X \times X \rightarrow \mathbb{R}$$

$$k(x^i, x^j) = \langle \phi(x^i), \phi(x^j) \rangle.$$

Metric of similarity across datapoints



Kernel methods for determining structure in Data seen in class

Kernel PCA

Kernel ICA

Kernel CCA

In the lecture notes but not presented in class (kernel K-Means)

Overview: Regression Techniques

We have seen several methods for non-linear regression.

They proceed using the same idea as kernel-based methods for finding structure in data:

Aim:

Given a dataset (X, Y) , find the relation $y=f(x)$ that explains best all pairs in (X, Y) .

Principle:

Find a projection $z=\phi(x)$ that reduces the problem to a *linear* regressive problem of the form: $y=w^T z$

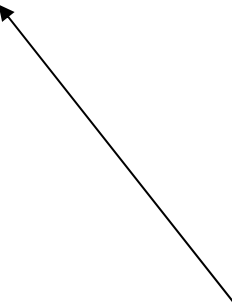
Take a probabilistic view and compute $y=E\{p(y|x)\}$. The pdf $p(y|x)$ embeds the non-linearities.

Overview: Regression Techniques

SVM, GMR and GPR (and similarly the variants on these, i.e. GPLVM, LWPR) lead to the following regressive model:

For a query point x^* , predict the associated output y^*

$$y^* = \sum_{i=1}^M \alpha_i k(x^*, x^i)$$



In SVR, the sum is reduced to summing only over the support vectors (a subset of datapoints)

In GPR, the sum is over the whole dataset!

In GMR, the sum is over the set of Gaussians. The centers of the Gaussians are usually not located on any particular datapoint.

Overview: Regression Techniques

SVR, GMR and GPR are based on same probabilistic regressive model, but do not optimize for the same objective function.

- SVR:
 - expresses $p(x|y)$ as a linear combination of known probabilistic models;
 - minimizes reconstruction error through convex optimization
 - finds a nm of models \leq nm of datapoints (support vectors)
- GMR:
 - learns $p(x,y)$ through likelihood maximization and then compute $p(y|x)$;
 - starts with a low nm of models \ll nm of datapoints
- GPR:
 - expresses $p(x|y)$ as a full density model
 - nm of models = nm of datapoints!

Overview: Techniques for Processes Evolving in Time

Some processes are evolving in time. This evolution can depend on many variables and be highly nonlinear. It can be an either explicit or implicit function of time.

We saw four machine learning approaches to do this:
GMR, RL, RNN/TDNN and HMM

Markov Processes ~ discretization of dynamical system (RL, HMM)

$$x_{t+1} = f(x_t) \sim p(x_{t+1} | x_t)$$

Take a density estimate approach (GMR):

$$\dot{x} = f(x) \sim E\{p(\dot{x} | x)\}$$

Model directly the dynamical system through sets of non-linear systems (TDNN):

$$\dot{x} = f(x) \sim \sum_i \text{sigmoid}(w_i^T x)$$

Overview of Topics Covered

This course covered a variety of topics that are core to Machine Learning. It gives you the basis to go and read recent advances in each of these topics.

We hope that you will find this material useful and that you will use some of these algorithms in the future.

If you do so, drop us a note and we would be glad to include your application in future lectures as examples!