DD2434 Machine Learning, Advanced Course

Lecture 12: Method and Model Selection

Hedvig Kjellström
hedvig@kth.se
https://www.kth.se/social/course/DD2434/

Today

Quick recap of course

Variety of models that use different ways to constrain mappings $\mathbf{x} \curvearrowright y$ alt models $p(\mathbf{x}_i|\theta)$ – The No Free Lunch Theorem (Murphy 1.4.9)

Parametric vs Non-parametric (Murphy 1.4.1)

Curse of dimensionality (Murphy 1.4.3)

Occam's razor and over-, underfitting (Murphy 1.4.7-1.4.8, 5.3)

Project presentation, Group E

Quick Recap

Recap from Lecture 1

Supervised/Predictive Learning

Data (**training set**): $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$

features/attributes

response variable

Task: Learn mapping $\mathbf{x} \curvearrowright y$

unknown true function

Functional approximation: $y = f(\mathbf{x})$

Use $\mathcal{D}$ to learn an approximative function $\hat{y} = \hat{f}(\mathbf{x})$

Recap from Lecture 1

Supervised/Predictive Learning

Classification: $y \in \{1, \ldots, C\}$ is discrete and finite

Probabilistic formulation: Model $p(y = 1|\mathbf{x}, \mathcal{D})$, $p(y = 2|\mathbf{x}, \mathcal{D})$, etc…

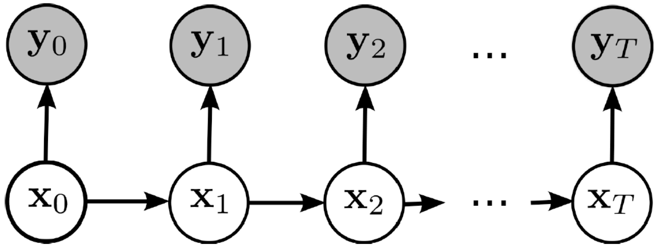
Best $y \equiv$ most probable y:

$$\hat{y} = \hat{f}(\mathbf{x}) = \arg\max_{c=1}^{C} p(y = c|\mathbf{x}, \mathcal{D})$$

Lecture 4: Hidden Markov Model (HMM)

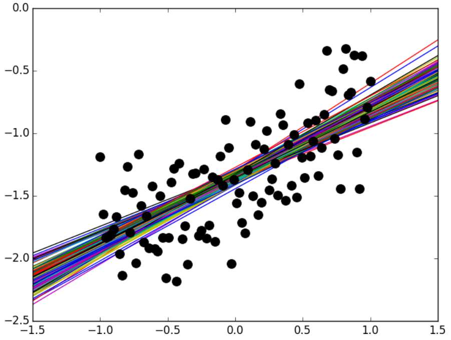
Markov assumption

Chain of observations/feature sets $\mathbf{X}$ generate chain of target values y, infer $\mathbf{X}$ from y



Lecture 5: Linear Regression

Assumption that the function is linear is a constraint in the mapping from $\mathbf{X}$ to y

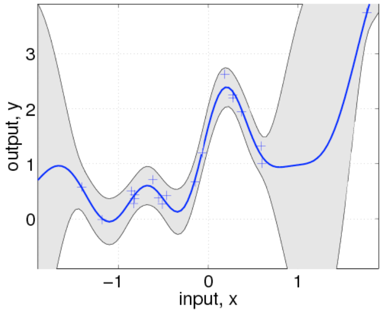


Handle nonlinearity by defining kernel $K(\mathbf{x}_1, \mathbf{x}_2)$ which defines a space with (more) linear y

Lecture 7: Gaussian Processes (GP)

Same kind of structural assumptions as HMM but for continuous data $\mathbf{X}$

"Soft" version of Markov assumption – correlation decreases with distance in $\mathbf{X}$



Lecture 9: Hierarchical Models

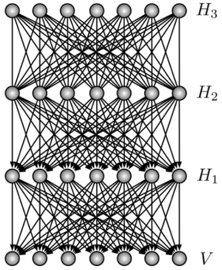
Model is a sequence of nested functions – constraints in the analytical form of the hidden layers:

$f : \mathbf{x} \rightarrow y$ standard functional mapping

$f : \mathbf{x} \rightarrow \mathbf{H} \rightarrow \mathbf{H} \rightarrow y$ hierarchical mapping

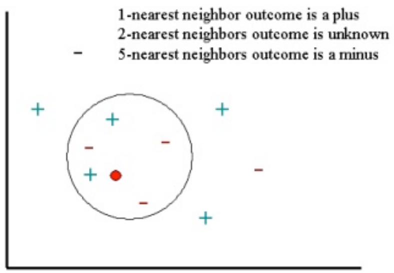
Very flexible model – need much training data, hard optimization problem, but very expressive model

Example: Deep Belief Network



Lecture 10: *k* Nearest Neighbors (*k*NN), Probabilistic Nearest Neighbors (PNN)

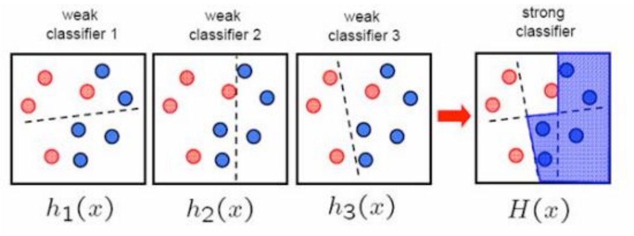
No constraining model – suitable for densely sampled and very non-linear spaces


1-nearest neighbor outcome is a plus
2-nearest neighbors outcome is unknown
5-nearest neighbors outcome is a minus

Lecture 10: AdaBoost

Models the target function as a mixture of linear classifier

Related to e.g., feed forward neural networks, Random Forests etc. (not covered in this course)



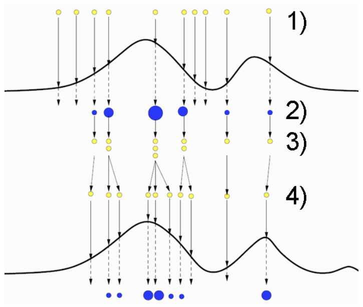
Lecture 10: Particle Filtering

Models distribution evolving over time

Models distribution as a set of samples – particles

Good for non-linear and non-Gaussian distributions

(Kalman filter = same thing but with Gaussian assumption)



Recap from Lecture 1

Unsupervised/Descriptive Learning

Data (**training set**): $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N}$

Task: discover patterns in $\mathcal{D}$

Under-specified problem – what patterns? How measure error?

Recap from Lecture 1

Unsupervised/Descriptive Learning

Probabilistic formulation: Density estimation
Models of the form $p(\mathbf{x}_i|\theta)$

Use $\mathcal{D}$ to maximize the probability $p(\mathbf{x}_i|\theta)$ of seeing each $\mathbf{x}_i$ given the model θ

New obstacles: Multivariate distributions

Unsupervised learning is more similar to how humans and animals learn!

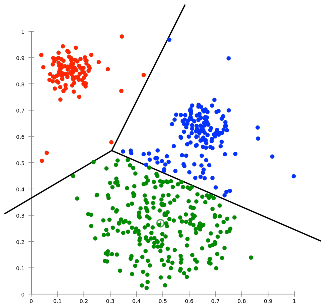
Practical advantage: No labeling of data required!

Lecture 2: *k*-Means Clustering

Model: data generated from *k* different clusters

Assumption: data points from the same cluster are closer than data points from different clusters

(Generalization: EM)



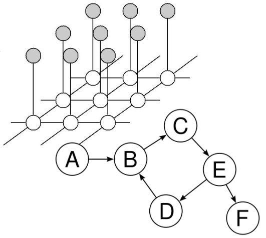
Lectures 2-5: Graphical models

Assumption: some dimensions in the statespace depend on other dimensions, either with a causal relationship or by being correlated. Other dimensions are uncorrelated

Constraint: This is modeled as a graph. (Groups of) statespace dimensions are the nodes, directed edges mean causal relationship, undirected edges mean correlation

Segmentation: remove edges in a Markov Random Field

Structure learning: add/remove edges as correlations/uncorrelations are observed in data



Lecture 7: Kernels

Similarity measure that make the feature space (where data live) as nice as possible

Nice = easy to do clustering, regression etc.

Can learn from data or (more common) define by hand

Do not need to know the space, all we need is a function $K(x_i, x_j)$, the similarity between two points

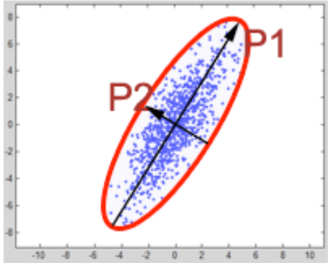
Lecture 8: Principal Component Analysis (PCA)

Purpose: Get a latent lowdim representation

Assumption: Data Gaussian distributed – linear method

SVD – get eigenvectors, project down data on the largest few eigenvectors (captures most variation)

(Extensions: Probabilistic PCA, Kernel PCA)

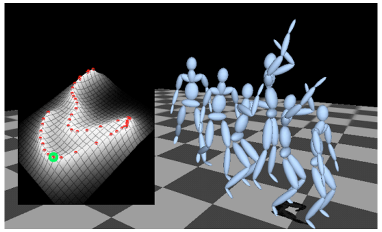


Lecture 8: Gaussian Process Latent Variable Models (GPLVM)

In the spirit of PPCA, latent variable model

Add-on to Gaussian Process, find a low-dim latent representation

Can have different constraints on the latent space



Lecture 9: Hierarchical Models

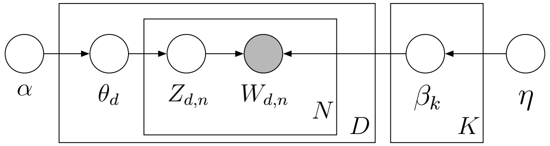
You can use hierarchical models for unsupervised learning tasks as well!

Lecture 11: Latent Dirichlet Allocation (LDA)

A latent variable model designed for discrete data (free text)

Assumption: Text is generated by drawing from the word distributions of one or more **topics**

Constraint: Do not model text grammar or structure, just word frequencies



Parametric vs Non-Parametric

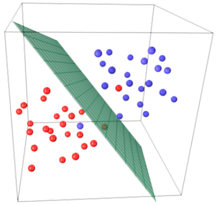
Recap from Lecture 1

Basic concept:
Parametric vs Non-Parametric

Models $p(\mathbf{x})$ and $p(y|\mathbf{x})$

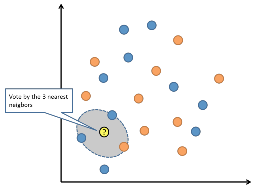
Parametric: Number of parameters constant with more data

E.g., linear classifier



Non-parametric: Number of parameters grows with more data

E.g., *k*NN classifier



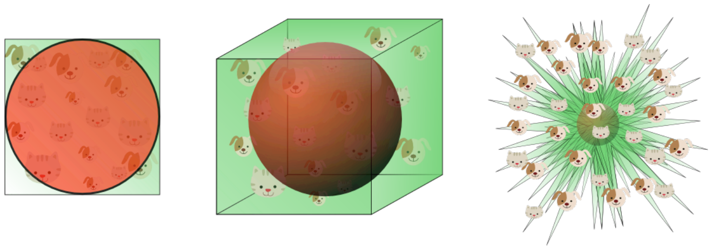
Examples

Parametric regression	vs	Non-parametric regression
Standard		Gaussian processes
Parametric classification	vs	Non-parametric classification
AdaBoost		*k*NN
Parametric sequential estimation	vs	Non-parametric sequential estimation
HMM		Particle Filter

Curse of dimensionality

Recap from Lecture 1

Basic concept: Curse of Dimensionality



2D

$$\text{cube/sphere} = \frac{\pi}{2^2}$$

3D

$$\text{cube/sphere} = \frac{4\pi}{2^3 * 3}$$

8D

$$\text{cube/sphere} = \frac{\pi^4}{2^8 * 24}$$

Adressed by using parametric models (fewer parameters – more robust)

Examples

Graph methods **model (un-)correlations** in data so that each node (with lower dim than entire data) can be processed separately

Latent variable models try to find the inherent (VC) dimension of the data, mapping each datapoint to a **lower dimensional space** where the characteristics of data (pairwise similarity) are maintained

Occam’s Razor and Under- and Overfitting

Recap from Lecture 1

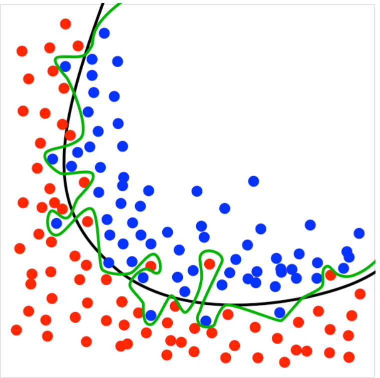
Basic concept: Overfitting

Model fits training data perfectly but not novel data

Reasons: Too little data, to high dimension, too flexible model

Met this problem in Assignment 3, e.g., for kNN with too sparse samples

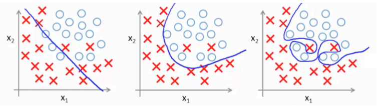
How to find out if a model is too flexible?



Recap from Lecture 1

Basic concept: Model Selection

Overfitting and **underfitting**

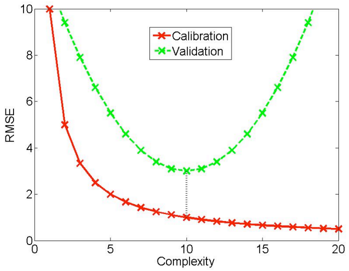


More complex model always have lower training data error

Solution to question on last slide:

Divide data into training set and **validation set**

Evaluate each model, each parameter setting with the validation set



Model Selection – Data Determines Model Complexity

Simple model needed for steering system

Complex model needed for steering system

Bayesian Occam’s Razor

Requires no knowledge about how data was generated

Requires that we know something about the data distribution, e.g., that it is Gaussian or multinomial

Experimental model selection: find optimal parameters $\hat{\theta}$ for each model *m*, comparing performance for different *m*

More principled: Integrate out θ – “average performance of *m* for all possible θ ”

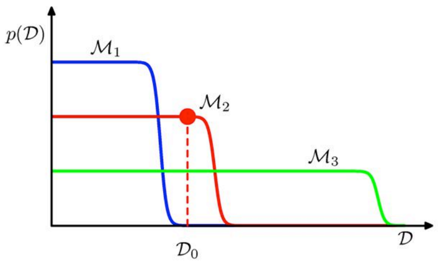
Assume all models *m* equally likely – estimate **marginal likelihood = evidence** for model *m* given the data

$$p(\mathcal{D}|m) = \int p(\mathcal{D}|\theta)p(\theta|m)d\theta$$

(Details on how to do that in Murphy Section 5.3.2)

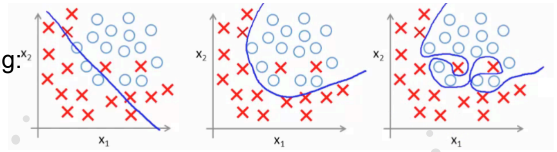
Bayesian Occam's Razor

Intuition: Models with more parameters fit wider range of possible data – less robust since the probability of each possible data value is lower:

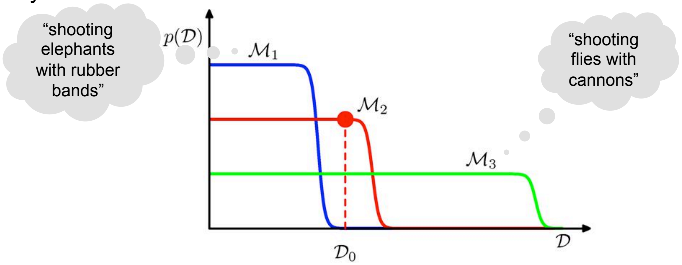


Comparing Over-, Underfitting with Bayesian Occam's Razor

Over/underfitting:



Bayesian Occam's razor:



What is next?

Project – talk to your group and send your supervisor an email about what you plan to do

Mon 19 Jan 9:00-12:00 V3

Presentation of 14 projects – 10 min each

12	9:00	7	9:50	3	11:00
11	9:10	6	10:00	2	11:10
10	9:20	BREAK		1	11:20
9	9:30	5	10:40	D2	11:30
8	9:40	4	10:50	D1	11:40

Happy Holidays and good luck with the projects and the Assignment 3 presentations!

Project presentation, Group E