

Models, Data, Learning Problems

Tobias Scheffer

Todo

- Include <http://tylervigen.com/spurious-correlations>

Overview

- Types of learning problems:
 - ◆ Supervised Learning (Classification, Regression, Ordinal Regression, Recommendations, Sequences und Structures)
 - ◆ Unsupervised Learning
 - ◆ Reinforcement-Learning (Exploration vs. Exploitation)
- Models
- Regularized empirical risk minimization
 - ◆ Loss functions,
 - ◆ Regularizer
- Evaluation

Supervised Learning: Basic Concepts

- Instance: $\mathbf{x} \in X$
 - ◆ In statistics: independent variable
 - ◆ X could be a vector space over *attributes* ($X = \mathbb{R}^m$)
 - ◆ An instance is then an assignment to the attributes.
 - ◆ $\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$ feature vector
- Target variable: $y \in Y$
 - ◆ In Statistics: dependent variable
- A model maps instances to the target variable.

$$\mathbf{x} \xrightarrow{\text{Model}} y$$

Supervised Learning: Classification

- Input: Instance $\mathbf{x} \in X$.
 - ◆ e.g., a feature vector

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$$

- Output: Class $y \in Y$; finite set Y .
 - ◆ The class is also referred to as the target attribute.
 - ◆ y is also called the (class) label



Classification: Example

- Input: Instance $\mathbf{x} \in X$
 - ◆ X : the set of all possible combinations of regiment of medication

Attribute	Instance \mathbf{x}
Medication #1 included?	$\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$ <div>Attribute values Feature vector</div>
⋮	
Medication #6 included?	

Medication
combination



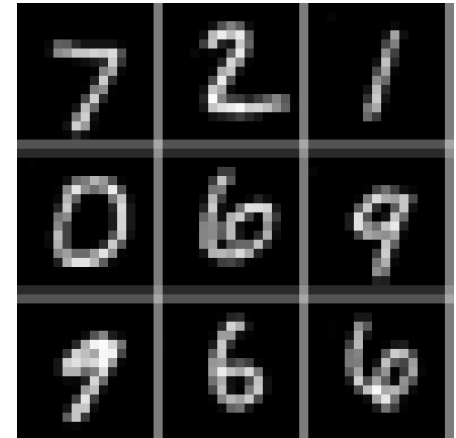
- Output: $y \in Y = \{\text{toxic}, \text{ok}\}$ 😞 / 😊



Classification: Example

- Input: Instance $\mathbf{x} \in X$
 - ◆ X : the set of all 16×16 pixel bitmaps

Attribute	Instance \mathbf{x}
Gray value of pixel 1	$\begin{pmatrix} 0.1 \\ 0.3 \\ 0.45 \\ \vdots \\ 0.65 \\ 0.87 \end{pmatrix}$ 256 pixel values
\vdots	
Gray value of pixel 256	



- Output: $y \in Y = \{0,1,2,3,4,5,6,7,8,9\}$: recognized digit



Classification: Example

- Input: Instance $\mathbf{x} \in X$
 - ◆ X : bag-of-words representation of all possible email texts

Attribute	Instance \mathbf{x}	Email
Word #1 occurs?	(0) Aardvark	<div> <p>Dear Beneficiary,</p> <p>your Email address has been picked online in this years MICROSOFT CONSUMER AWARD as a Winner of One Hundred and Fifty Five Thousand Pounds Sterling...</p> </div>
	1 Beneficiary	
⋮	0 Friend	
	⋮	
	1 Sterling	
Word # m occurs?	(0) Science	
$m \approx 1,000,000$		

- Output: $y \in Y = \{\text{spam, ok}\}$

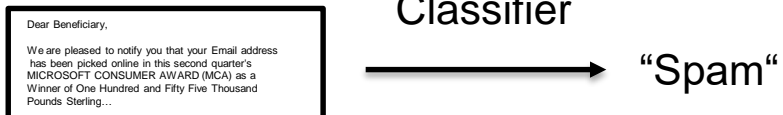
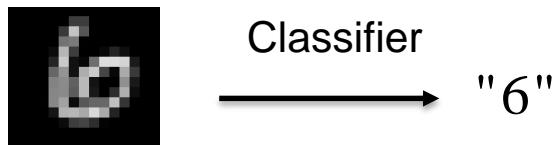
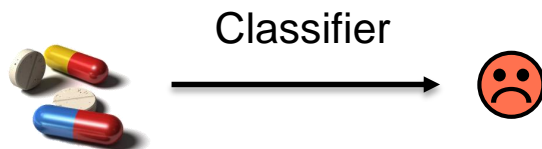
Dear Beneficiary,

We are pleased to notify you that your Email address has been picked online in this second quarter's MICROSOFT CONSUMER AWARD (MCA) as a Winner of One Hundred and Fifty Five Thousand Pounds Sterling...

→ **classifier** → "Spam"



Classification



- Classifier should be learned from training data.



Classifier Learning

- Input to the Learner:
Training data T_n .

$$\diamond \mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix}$$



$$\diamond \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$



- Training Data:
 $T_n = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$

- Output: a Model

$$\diamond \begin{matrix} \text{pill icons} \end{matrix} \mapsto \text{frowny face icon} \\ y_{\theta} : X \rightarrow Y$$


- for example:


$$y_{\theta}(\mathbf{x}) = \begin{cases} \text{frowny face icon} & \text{if } \mathbf{x}^T \boldsymbol{\theta} \geq 0 \\ \text{smiley face icon} & \text{otherwise} \end{cases}$$

Linear classifier with
parameter vector $\boldsymbol{\theta}$.

Classifier Learning

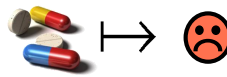
- Input to the Learner:
Training data T_n .

$$\diamond \mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix}$$


$$\diamond \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$


- Training Data:
 $T_n = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$

- Output: a model

$$\diamond y_\theta : X \rightarrow Y$$


- Model classes

- (Generalized) linear model
 - Decision tree
 - Ensemble classifier
 - ...

Supervised Learning: Regression

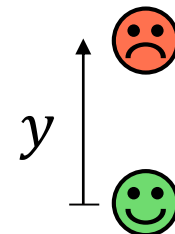
- Input: Instance $\mathbf{x} \in X$.
 - ◆ e.g., feature vector

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$$




How toxic is a combination?


- Output: continuous (real) value, $y \in \mathbb{R}$
 - ◆ e.g., *toxicity*.



Regressor Learning

- Input to the Learner:
Training data T_n .

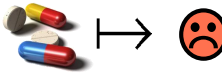
$$\mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix}$$


$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$


The vector \mathbf{y} is annotated with smiley and frowny faces: y_1 has a frowny face, y_2 has a smiley face, y_n has a frowny face, and y_{n+1} has a smiley face.

- Training Data:
 $T_n = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$

- Output: a model

$$y_\theta : X \rightarrow Y$$


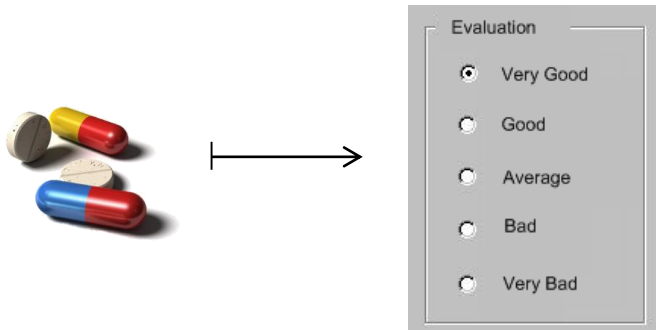
For example

$$y_\theta(\mathbf{x}) = \mathbf{x}^T \boldsymbol{\theta}$$

Generalized linear model with
parameter vector $\boldsymbol{\theta}$.

Supervised Learning: Ordinal Regression

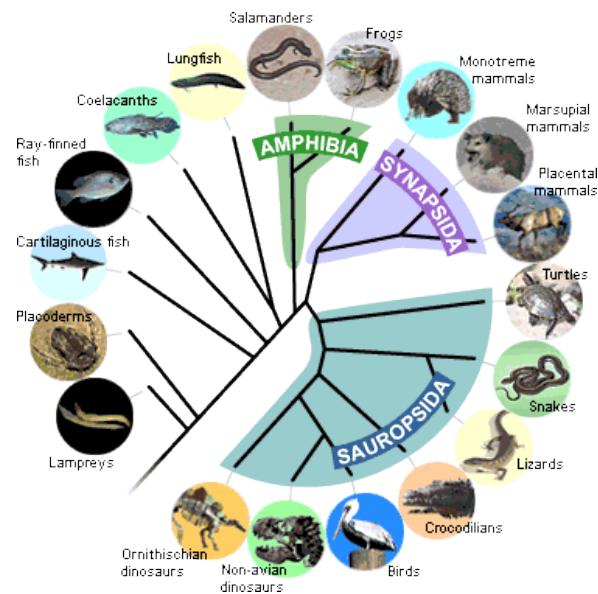
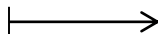
- Input: Instance $x \in X$.
- Output: discrete value $y \in Y$ like classification, but there is an ordering on the elements of Y .
- A large discrepancy between the model's prediction and the true value is worse than a small one.



Satisfied with the outcome?

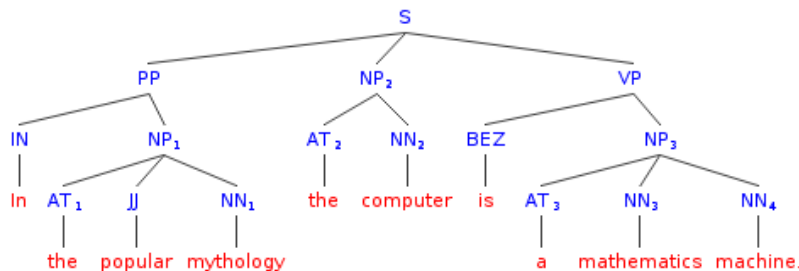
Supervised Learning: Taxonomy Classification

- Input: Instance $x \in X$.
- Output: discrete value $y \in Y$ like classification, but there is a tree-based ordering on the elements of Y .
- The prediction is worse the farther apart the predicted and actual nodes are.

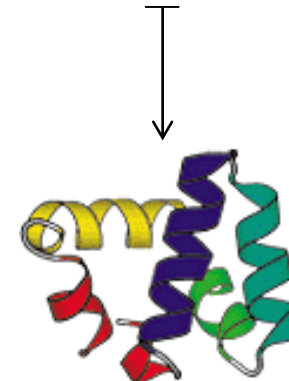


Supervised Learning: Sequence and Structure Prediction

- Input: Instance $x \in X$.
- Output: Sequence, Tree or Graph $y \in Y$.
- Example applications:
 - ◆ Parse natural languages
 - ◆ Protein folding



...AAGCTTGCACTGCCGT...



Supervised Learning: Rankings

- Input: query q and list of items I_1, \dots, I_n .
- Output: a sorting of the items
- Training data: user clicks on I_j after querying with q :
 - ◆ The selected item should be ranked higher than those listed higher that were not clicked.

Web Images Videos Maps News Shopping Mail more ▼

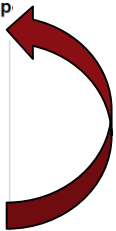
Google potsdam

Web Show options... Results 1 - 10 of about 14,500,000 for p

[Welcome to SUNY Potsdam - SUNY Potsdam](#)
 A liberal arts college. Information for prospective students, current students, faculty and staff, alumni and friends.
[Current Students](#) - [Faculty & Staff](#) - [Athletics](#) - [Majors & Minors](#)
www.potsdam.edu/ - Cached - Similar

[Potsdam - Wikipedia, the free encyclopedia](#)
Potsdam (German pronunciation: [ˈpɒtsdam]) is the capital city of the German federal state of Brandenburg and is part of the Metropolitan area of ...
[Geography](#) - [History](#) - [Politics](#) - [Education and research](#)
en.wikipedia.org/wiki/Potsdam - Cached - Similar

[Potsdam Conference - Wikipedia, the free encyclopedia](#)
 Harry Truman and Joseph Stalin meeting at the **Potsdam** Conference
 left to right, first row: Stalin, Truman, Soviet Ambassador Andrei ...
en.wikipedia.org/wiki/Potsdam_Conference - Cached - Similar



Supervised Learning: Recommendations

- Input: users, items, contextual information.
- Output: How much will a user like a recommendation?
- Training data: ratings, sales, page views.





ALUMINIUM Baseballschläger 30' American Baseball
 von [Outdoor 4 You - Shop](#)
 ★★★★★ (4 Kundenrezensionen) [Mehr zu diesem Artikel](#)

Preis: **EUR 17,58**

Auf Lager.
 Verkauf und Versand durch **NORMANI**.
 Noch 5 Stück auf Lager.
 4 neu ab EUR 17,58

Marken-Uhren mit Tiefpreis-Garantie finden Sie im [Uhren-Shop](#) bei Amazon.de/Uhren.

Kunden, die diesen Artikel gekauft haben, kauften auch Seite 1 von 23

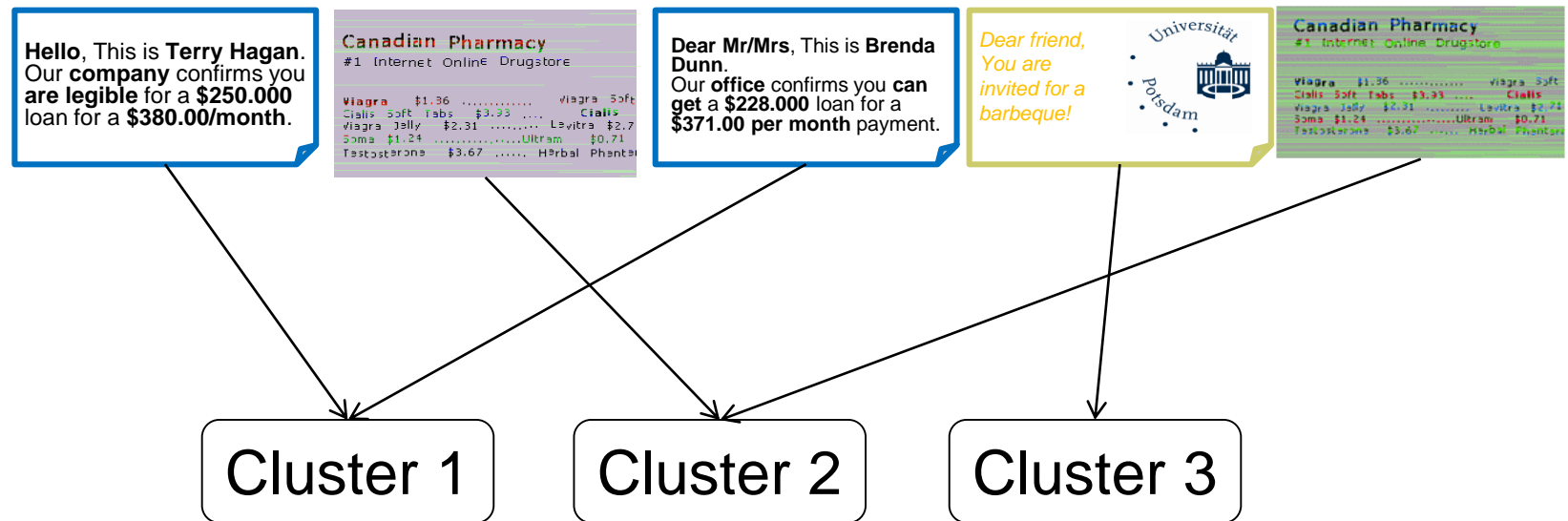
			
Leder Quarzsandhandschuhe schwarz S-XXL	Balaclava 3-Loch ★★★★☆ (4) EUR 3,50	Pfefferspray KO-FOG 40ML ★★★★★ nur EUR 5,95	Baseballschläger Holz 32' American Baseball natur

Unsupervised Learning

- Training data for unsupervised learning: Set of instances $\mathbf{x} \in X$.
- Additional assumptions about the data formation process; for example, independence of random variables.
- The goal is the detection of structure in the data:
 - ◆ For example, find the most likely grouping into clusters of instances that share certain properties, which were not directly observable in the data.

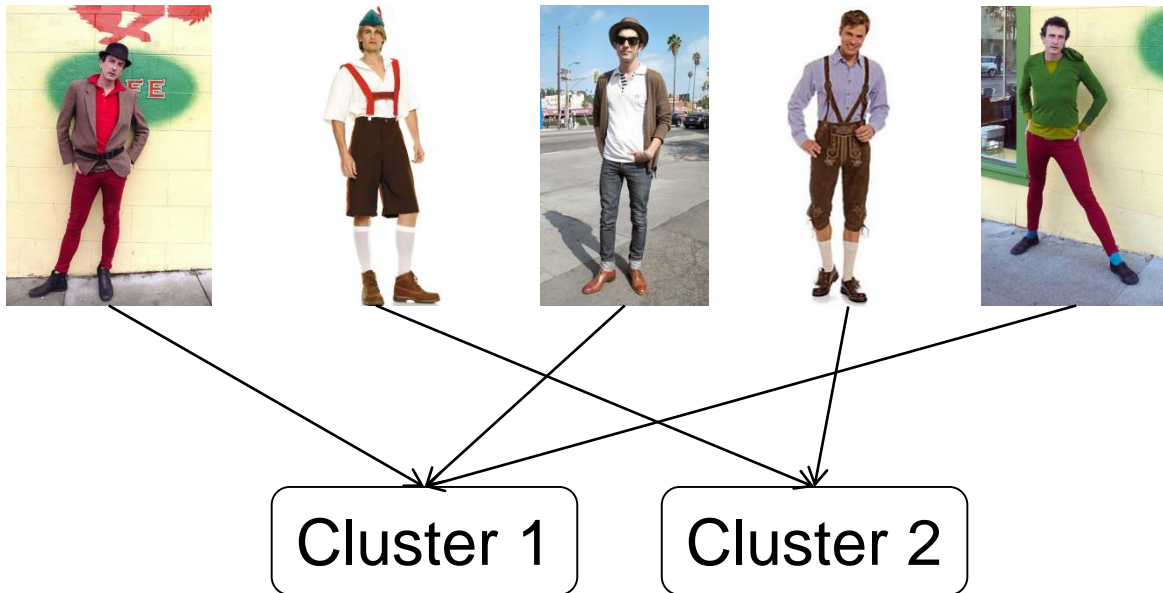
Cluster Analysis: Email Campaign Example

- Input: a stream of emails.
- Output: a partitioning into subsets that belong to the same email campaign.



Cluster Analysis: Market Segmentation Example

- Input: data over customer behavior.
- Output: a partitioning into clusters of customers who have similar product preferences.



Reinforcement Learning: Learning to Control a System

- Suppose there is a system with control parameters.
- A utility function describes the desired system behavior.
- Changes of the control parameters may have time-lagging effects.
- The Learner must experiment with the system to find a model that achieves the desired behavior (Exploration).
- At the same time, the system should be kept in the best state that is possible (Exploitation).



Reinforcement Learning: Learning to Control a System: Example

- Advertisement (Ad) placement.
- To learn which ads the user clicks, the learner must experiment.
- However, when the learner experiments with using ads other than the most popular ones, sales are lost.



Taxonomy of Learning Problems

- Supervised: Training data contain values for variable that model has to predict
 - ◆ Classification: categorical variable
 - ◆ Regression: continuous variable
 - ◆ Ordinal regression, finite, ordered set of values
 - ◆ Rankings: ordering of elements
 - ◆ Structured prediction: sequence, tree, graph, ...
 - ◆ Recommendation: Item-by-user matrix

Taxonomy of Learning Problems


- Unsupervised: discover structural properties of data
 - ◆ Clustering
 - ◆ Unsupervised feature learning: find attributes that can be used to describe the data well
- Control / reinforcement learning: learning to control a dynamical system


Overview

- Types of learning problems:
 - ◆ Supervised Learning (Classification, Regression, Ordinal Regression, Recommendations, Sequences und Structures)
 - ◆ Unsupervised Learning
 - ◆ Reinforcement-Learning (Exploration vs. Exploitation)
- Models
- Regularized empirical risk minimization
 - ◆ Loss functions,
 - ◆ Regularizer
- Evaluation

Classifier Learning

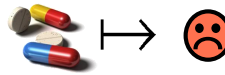
- Input to the Learner:
Training data T_n .

$$\diamond \mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix}$$


$$\diamond \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$


- Training Data:
 $T_n = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$

- Output: a classifier

$$\diamond y_\theta : X \rightarrow Y$$


- How can a learning algorithm learn a model (classifier) from the training data?
- This is a search problem in the space of all models.

Model or Parameter Space

- Model space, parameter or hypothesis space Θ :
 - ◆ The classifier has parameters $\theta \in \Theta$.
 - ◆ Θ is a set of models (classifiers), which are suitable for a learning method.
 - ◆ The model space is one of the degrees of freedom for machine learning; there are many commonly used spaces.
 - ◆ Also called *Language Bias*
- Example:
 - ◆ Linear models

$$y_{\theta}(\mathbf{x}) = \begin{cases} \text{☹️} & \text{if } \sum_{j=1}^m x_j \theta_j \geq \theta_0 \\ \text{😊} & \text{otherwise} \end{cases}$$

Loss Function, Optimization Criterion

- Learning problems will be formulated as optimization problems.
 - ◆ The *loss function* measures the goodness-of-fit a model has to the observed training data.
 - ◆ The *regularization function* measures, whether the model is *likely* according to our prior knowledge.
 - ◆ The *optimization criterion* is a (weighted) sum of the losses for the training data and the regularizer.
 - ◆ We seek the model that minimizes the optimization criterion.
- Learning finds the overall most likely model given the training data and prior knowledge.

Loss Function

- Loss function: How bad is it if the model predicts value $y_{\theta}(\mathbf{x}_i)$ when the true value of the target variable is y_i ?

$$\ell(y_{\theta}(\mathbf{x}_i), y_i)$$

- We average the loss over the entire training data T_n :

- ◆ Empirical risk $\hat{R}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n \ell(y_{\theta}(\mathbf{x}_i), y_i)$

- Example: Binary classification problem with positive class (+1) and negative class (-1). False positives and false negatives are equally bad.

- ◆ Zero-One Loss: $\ell_{0/1}(y_{\theta}(\mathbf{x}_i), y_i) = \begin{cases} 0 & \text{if } y_{\theta}(\mathbf{x}_i) = y_i \\ 1 & \text{otherwise} \end{cases}$

Loss Function

- Example: in diagnostic classification problems, an overlooked illness (false negative) is worse than an incorrectly diagnosed one (false positive).
 - ◆ Cost matrix

$$\ell_{c_{FP}, c_{FN}}(y_{\theta}(\mathbf{x}_i), y_i) = \begin{cases} y_{\theta}(\mathbf{x}_i) = +1 \\ y_{\theta}(\mathbf{x}_i) = -1 \end{cases}$$

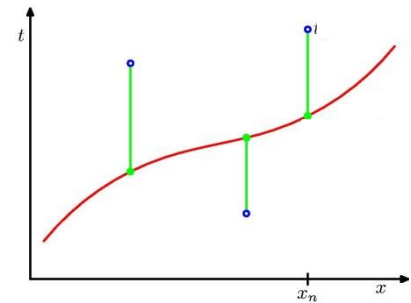
	$y_i = +1$	$y_i = -1$
$y_{\theta}(\mathbf{x}_i) = +1$	0	c_{FP}
$y_{\theta}(\mathbf{x}_i) = -1$	c_{FN}	0

Loss Function

- Example of a loss function for regression: the prediction should be as close as possible to the actual value of the target attribute.

- ◆ Quadratic error:

$$\ell_2(y_{\theta}(\mathbf{x}_i), y_i) = (y_{\theta}(\mathbf{x}_i) - y_i)^2$$



Loss Function

- How bad is it if the model predicts value y' when the true value of the target variable is y ?
 - ◆ Loss: $\ell(y', y)$
- The selected loss function is motivated by the particular application.



Search for a Model

- Search for a Classifier of “toxic combinations”.

- Model space:

$$y_{\theta}(\mathbf{x}) = \begin{cases} \text{☹️} & \text{if } \sum_{j=1}^m x_j \theta_j \geq \theta_0 \\ \text{😊} & \text{otherwise} \end{cases}$$

Sample
Combinations

Training data

Medications in the combination

	x_1	x_2	x_3	x_4	x_5	x_6	y
\mathbf{x}_1	1	1	0	0	1	1	☹️
\mathbf{x}_2	0	1	1	0	1	1	☹️
\mathbf{x}_3	1	0	1	0	1	0	😊
\mathbf{x}_4	0	1	1	0	0	0	😊

- Approach: empirical risk should be minimal

$$\theta^* = \operatorname{argmin}_{\theta} \sum_{i=1}^n \ell_{0/1}(y_{\theta}(\mathbf{x}_i), y_i)$$

- Is there such a model? Are there many?

Search for a Model

- Search for a Classifier of “toxic combinations”.

- Model space:

$$y_{\theta}(\mathbf{x}) = \begin{cases} \text{☹️} & \text{if } \sum_{j=1}^m x_j \theta_j \geq \theta_0 \\ \text{😊} & \text{otherwise} \end{cases}$$

- Models with 0 loss:

- ◆ $y_{\theta}(\mathbf{x}) = \text{☹️}$, if $x_6 \geq 1$
- ◆ $y_{\theta}(\mathbf{x}) = \text{☹️}$, if $x_2 + x_5 \geq 2$
- ◆ $y_{\theta}(\mathbf{x}) = \text{☹️}$, if $2x_4 + x_6 \geq 1$
- ◆ ...

Training data

Medications in the combination

Sample Combinations		x_1	x_2	x_3	x_4	x_5	x_6	y	
		\mathbf{x}_1	1	1	0	0	1	1	☹️
		\mathbf{x}_2	0	1	1	0	1	1	☹️
		\mathbf{x}_3	1	0	1	0	1	0	😊
		\mathbf{x}_4	0	1	1	0	0	0	😊

Search for a Model

- Search for a Classifier of “toxic combinations”.

- Model space:

$$y_{\theta}(\mathbf{x}) = \begin{cases} \text{☹️} & \text{if } \sum_{j=1}^m x_j \theta_j \geq \theta_0 \\ \text{😊} & \text{otherwise} \end{cases}$$

- Models with 0 loss:

- ◆ $y_{\theta}(\mathbf{x}) = \text{☹️}$, if $x_6 \geq 1$
- ◆ $y_{\theta}(\mathbf{x}) = \text{☹️}$, if $x_2 + x_5 \geq 2$
- ◆ $y_{\theta}(\mathbf{x}) = \text{☹️}$, if $2x_4 + x_6 \geq 1$
- ◆ ...

Training data

Medications in the combination

Sample Combinations	Medications in the combination						y
	x 1	x 2	x 3	x 4	x 5	x 6	
x1	1	1	0	0	1	1	☹️
x2	0	1	1	0	1	1	☹️
x3	1	0	1	0	1	0	😊
x4	0	1	1	0	0	0	😊

- The models with an empirical risk of 0 form the *version space*.
- The version space is empty for a set of contradictory data.

Search for a Model

- Search for a Classifier of “toxic combinations”.

- Model space:

$$y_{\theta}(\mathbf{x}) = \begin{cases} \text{☹️} & \text{if } \sum_{j=1}^m x_j \theta_j \geq \theta_0 \\ \text{😊} & \text{otherwise} \end{cases}$$

- Models with 0 loss:

- ◆ $y_{\theta}(\mathbf{x}) = \text{☹️}$, if $x_6 \geq 1$
- ◆ $y_{\theta}(\mathbf{x}) = \text{☹️}$, if $x_2 + x_5 \geq 2$
- ◆ $y_{\theta}(\mathbf{x}) = \text{☹️}$, if $2x_4 + x_6 \geq 1$
- ◆ ...

Training data

Medications in the combination

Sample Combinations	Medications in the combination						y
	x 1	x 2	x 3	x 4	x 5	x 6	
x1	1	1	0	0	1	1	☹️
x2	0	1	1	0	1	1	☹️
x3	1	0	1	0	1	0	😊
x4	0	1	1	0	0	0	😊

- The models with an empirical risk of 0 form the *version space*.
- The version space is empty for a set of contradictory data.

Search for a Model

- Search for a Classifier of “toxic combinations”.

- Model space:

$$y_{\theta}(\mathbf{x}) = \begin{cases} \text{☹️} & \text{if } \sum_{j=1}^m x_j \theta_j \geq \theta_0 \\ \text{😊} & \text{otherwise} \end{cases}$$

- Models with 0 loss:

- ◆ $y_{\theta}(\mathbf{x}) = \text{☹️}$, if $x_6 \geq 1$
- ◆ $y_{\theta}(\mathbf{x}) = \text{☹️}$, if $x_2 + x_5 \geq 2$
- ◆ $y_{\theta}(\mathbf{x}) = \text{☹️}$, if $2x_4 + x_6 \geq 1$
- ◆ ...

Training data

Medications in the combination

Sample Combinations	Medications in the combination						y
	x 1	x 2	x 3	x 4	x 5	x 6	
x1	1	1	0	0	1	1	☹️
x2	0	1	1	0	1	1	☹️
x3	1	0	1	0	1	0	😊
x4	0	1	1	0	0	0	😊

- The models of the version space differ in their predictions of some instances, which do not appear in the training set.
- Which is the correct one?

Uncertainty

- In practice, one can never be certain whether a correct model has been found.
- Data can be contradictory (e.g. due to measurement errors)
- Many different models may achieve a small loss.
- The correct model perhaps may not even lie in the model space.

- Learning as an *optimization problem*
 - ◆ Loss function: Degree of consistency with the training data
 - ◆ Regularizer: a priori probability of a model

Regularizer

- Loss function expresses how good the model fits the data.
- Regularisierer $\Omega(\theta)$:
 - ◆ Expresses assumptions about whether the model θ is a *priori* probable.
 - ◆ Ω is independent from the training data.
 - ◆ The higher the regularization term is for a model, the less likely the model is.
- Often the assumptions express that few attributes should be sufficient for a suitable model.
 - ◆ Count of the non-zero weights, L_0 -Regularization
 - ◆ Sum of the attribute weights, L_1 -Regularization
 - ◆ Sum of the squared attribute weights, L_2 -Regularization.

Regularizer

■ Candidates:

$(\theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$

- ◆ $y_{\theta}(\mathbf{x}) = \text{☹}$, if $x_6 \geq 1$ $\theta_1 = (1, 0, 0, 0, 0, 0, 1)^T$
- ◆ $y_{\theta}(\mathbf{x}) = \text{☹}$, if $x_2 + x_5 \geq 2$ $\theta_2 = (2, 0, 1, 0, 0, 1, 0)^T$
- ◆ $y_{\theta}(\mathbf{x}) = \text{☹}$, if $2x_4 + x_6 \geq 1$ $\theta_3 = (1, 0, 0, 0, 2, 0, 1)^T$

■ Regularizer:

L_0 -Regularisierung L_1 -Regularisierung L_2 -Regularisierung

$$\Omega_0(\theta_1) = 2$$

$$\Omega_1(\theta_1) = 2$$

$$\Omega_2(\theta_1) = 2$$

$$\Omega_0(\theta_2) = 3$$

$$\Omega_1(\theta_2) = 4$$

$$\Omega_2(\theta_2) = 6$$

$$\Omega_0(\theta_3) = 3$$

$$\Omega_1(\theta_3) = 4$$

$$\Omega_2(\theta_3) = 6$$

Optimization Criterion

- Regularized empirical risk: Trade-off between average loss and regularizer

$$\frac{1}{n} \sum_{i=1}^n \ell(y_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i) + \lambda \Omega(\boldsymbol{\theta})$$

- The parameter $\lambda > 0$ controls the trade-off between loss and the regularizer.

Optimization Problem

- Is there a reason to use this optimization criterion (a *regularized* empirical risk)?
- There are several justifications and derivations:
 - ◆ Most probable (a posteriori) model (*MAP-Model*).
 - ◆ One can obtain a smaller upper bound for the error on future data depending on $|\theta|$ (*SRM*).
 - ◆ Learning without regularization is an *ill-posed* problem; there is no unique solution or it is strongly influenced by minimal changes of the data.

Regularized Empirical Risk Minimization

- Search for a Classifier of “toxic combinations”.

- Model space:

$$y_{\theta}(\mathbf{x}) = \begin{cases} \text{☹️} & \text{if } \sum_{j=1}^m x_j \theta_j \geq \theta_0 \\ \text{😊} & \text{otherwise} \end{cases}$$

- Regularized empirical risk:

$$\frac{1}{n} \sum_{i=1}^n \ell_{0/1}(y_{\theta}(\mathbf{x}_i), y_i) + 0.1 \Omega_0(\theta)$$

- Model with minimal regularized empirical risk

$$\diamond y_{\theta}(\mathbf{x}) = \begin{cases} \text{☹️} & \text{if } x_6 \geq 1 \\ \text{😊} & \text{otherwise} \end{cases}$$

Training data

Medications in the combination

Sample Combinations		x_1	x_2	x_3	x_4	x_5	x_6	y	
		\mathbf{x}_1	1	1	0	0	1	1	☹️
		\mathbf{x}_2	0	1	1	0	1	1	☹️
		\mathbf{x}_3	1	0	1	0	1	0	😊
		\mathbf{x}_4	0	1	1	0	0	0	😊

Evaluation of Models

- How good will a model function in the future?
- Future instances will be drawn according to an (unknown) probability distribution $p(\mathbf{x}, y)$.
- Risk: expected loss under distribution $p(\mathbf{x}, y)$.

$$R(\boldsymbol{\theta}) = \sum_y \int \ell(y_{\boldsymbol{\theta}}(\mathbf{x}), y) p(\mathbf{x}, y) d\mathbf{x}$$

- Is the empirical risk on the training data a useful estimator for the risk?

Evaluation of Models

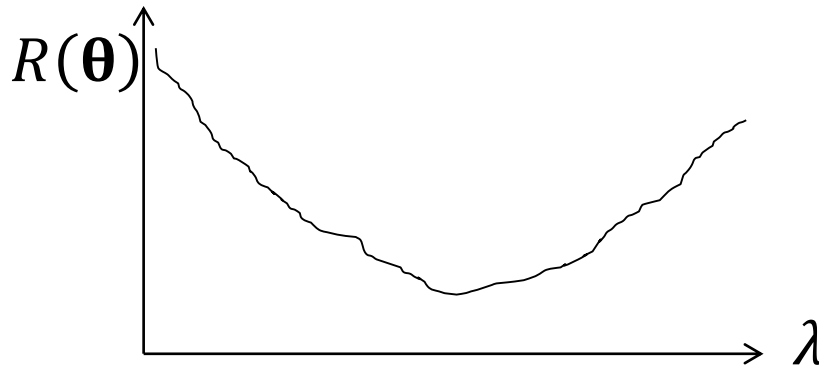
- How good will a model function in the future?
- Future instances will be drawn according to an (unknown) probability distribution $p(\mathbf{x}, y)$.
- Risk: expected loss under distribution $p(\mathbf{x}, y)$.
- Is the empirical risk on the training data a useful estimator for the risk?
 - ◆ Problem: All models in the version space have an empirical risk of 0 on the training data.
 - ◆ A classifier can achieve 0 empirical risk on the training data by simply storing each training instance in a table and reproducing the stored label when queried.

Evaluation of Models

- How good will a model function in the future?
- Future instances will be drawn according to an (unknown) probability distribution $p(\mathbf{x}, y)$.
- Risk: expected loss under distribution $p(\mathbf{x}, y)$.
- Empirical risk on the training data is an extremely optimistic estimator for the risk.
- Risk is evaluated using instances that were not used for training.
 - ◆ Training and Test datasets.
 - ◆ N -fold cross validation.

Optimization Problem

- How should λ be set?
- Divide available data into training and test data.
- Iterate over values of λ
 - ◆ Train on training data to find a model
 - ◆ Evaluate it on the test data
- Choose the value of λ giving minimal loss
- Train with all data



Data, Models, & Learning Problems



- Supervised Learning: Find the function most likely to have generated the training data.
- Loss function: Measures the agreement between the model's predictions and the values of the target variable in the training data.
- Regularizer: Measures the agreement to prior knowledge.
- Unsupervised Learning: with no target variable, discover structure in the data; e.g., by dividing the instances into clusters with common properties.
- Reinforcement Learning: Control of processes.