# Machine Learning Zusammenfassung

Übersicht Lernziele und Review

|  |  |
| --- | --- |
| Ein Bild, das Text enthält.  Automatisch generierte BeschreibungEin Bild, das Text enthält.  Automatisch generierte Beschreibung  **1) Introduction** | Ein Bild, das Text, drinnen, Screenshot enthält.  Automatisch generierte BeschreibungEin Bild, das Text, Papier, Screenshot enthält.  Automatisch generierte Beschreibung  **2) Formulating Learning Prob.** |
| Ein Bild, das Text, drinnen, Screenshot, Pflanze enthält.  Automatisch generierte BeschreibungEin Bild, das Text enthält.  Automatisch generierte Beschreibung  **3) Model Assessement&Selection** | Ein Bild, das Text, drinnen enthält.  Automatisch generierte BeschreibungEin Bild, das Text enthält.  Automatisch generierte Beschreibung  **4) Support Vector Machine** |
| **5) Ensemble Methods** | Ein Bild, das Text, drinnen, Screenshot enthält.  Automatisch generierte BeschreibungEin Bild, das Text enthält.  Automatisch generierte Beschreibung  **6) Best Practices in ML** |
| Ein Bild, das Text, drinnen, Screenshot, Pflanze enthält.  Automatisch generierte BeschreibungEin Bild, das Text enthält.  Automatisch generierte Beschreibung  **7) ML System Design** |  |

1) Introduction

|  |  |  |  |
| --- | --- | --- | --- |
| Definition nach [Mitchell](https://www.youtube.com/watch?v=wKsh8EjUUVo): A computer program is said to learn from experience with respect to some class of tasks and performance measure ,if its **performance at tasks in**  (play game), as **measured by**  (% of won games), improves with **experience**  (games played against itself).  Ingredients to Learning  Eine Learning Solution benötigt eine **Representation**, eine **Evaluationfunktion** und einen **Optimizer**. Diese lassen sich aus der Formulierung eines gut gestellten **Lernproblems ableiten**; Lernproblem als Task , Performance Measure und Training Experience . ➔ Für die Ableitung dieser konkreten Methoden gibt es **keine allgemeine Lösung**. Sie ist problem-(daten-)abhängig und stützt sich auf **Vorwissen**. Gültige Leitfäden sind die Merkmale von **Methoden** (induktive Verzerrung, VC-Theorie), **Erfahrung**/Best Practices und **Vorwissen**.  Unterteilung des ML-Bereichs: | |  | |
| **Inductive** ➔vonBeobachtung zur Schlussfolgerung/Regel | **Transductive** ➔ von Beobachtung zur Beobachtung | | Deductive➔von Schlussfolgerung/Regel zu Beobachtung |
| **Supervised\*** Basis:Trainingsdaten **mit Labeln**; Konzept lernen -> Werte vorhersagen| Bsp.: Regression, Klassifikation | **Reinforcement** Basis: Lernen durch **Belohnungssysteme** (Maximierung); keine Beispieldaten -> lernen in der Simulationsumgebung| Bsp.: Autonomes Fahren, Spiele | | **Unsupervised** Ermittlung **unbekannter Strukturen** in den Daten ohne Zielvariable; Strukturen finden | Bsp.: Clustering, Dimensionsreduktion |
| **\***Ziel des induktiven überwachten Lernens ist, eine Hypothese aus einem durch Theorie und Empirie gewählten **Hypothesenraum** so **anzupassen** (= spezifische Parameter), dass am besten zu den Trainingsdaten passt und damit möglichst genau abbildet; dies wird durch die **Minimierung der Lossfunktion** (empirische Fehler () erreicht. ➔ Ein Modell, das an eine hinreichend große Beispielmenge angepasst ist, lässt sich auf **ungesehene Daten verallgemeinern**; es ist jedoch **nicht wahrheitserhaltend**. | | | |
| Induktive Bias: Alle Modelle/Lernalgorithmen haben je nach Problemstellung Vor- und Nachteile; den «Besten» gibt es nicht. ➔ «No free lunch Theorem»; Letztlich hängt ML von der intelligenten Wahl der Klasse des Hypthesenraums ab; der Lernalgorithmus (Loss-Funktion (MSE) & Optimierer (GD)) optimiert dann die Details. | | | |
| What is learnable (computational learning theory) ➔ Jede Zielfunktion 𝑓 über einer Menge 𝑋 ist lernbar,   |  |  |  | | --- | --- | --- | | …bei hinreichend **aussagekräftigen Hypothesenraum** , | …eine ausreichend **große Trainingsmenge** , | ... Daten in und sind aus **derselben Verteilung** |   **VC-Dimension** ➔ Die comutational learning theory (CoLT) garantiert, dass **die Anzahl der benötigten Trainingsdaten linear mit der VC-Dimension** **wächst**. | | | |

2) Formulating learning problems

|  |  |
| --- | --- |
| Represntation  ➔ **Lineare Regression** | **Hypothesenraum: lineare Regression**  | = Intercept | = Gradient  (Theta) = Variable für alle Parameter der Funktion ➔ Wie wählt man ? |
| Evaluation  ➔ **Mean Squared Error** | |  |  | | --- | --- | | **Cost-Function (Loss): Mean Squared Error**  - Wähle so, dass nahe bei für die Traingingsbeispiele liegt  - Eine Funktion beziffert die Kosten der gewählten Parameter über die Trainingsmenge für bestimmte Hypothese  ➔ Ziel ist es diesen **«Mean Squared Error»** zu minimieren (Ableitung!) |  | |
| Optimization  ➔ **Gradient Descent** | **Numerical Optimizer: Gradient Descent**  Optimierungsalgorithmus, um die **Werte der Parameter**  einer Funktion zu finden, die eine **Kostenfunktion minimiert**: Have – want –> minimize  Der Algorithmus bewegt sich **iterativ** in die Richtung des **steilsten Abstiegs der Kostenfunktion** , der durch den **negativen Wert des Gradienten** definiert ist.  **Starting-Point** = Kleine Änderungen des Ausgangspunkts führen zu unterschiedlichen lokalen Minima.  **Gradient** = partielle Ableitung einer Funktion mit ; gibt die Richtung des steilsten Anstieges an (für Minimum - deshalb negativ).  **Learning-Rate** = Die Lernrate steuert, wie stark wir die Koeffizienten /Gewichte in Bezug auf den Gradienten anpassen sollten. ➔ zu kleiner Wert führt zu extrem langen Rechnendauern | zu gross führt dazu, dass Minimum übersprungen wird   |  |  | | --- | --- | | Die Formeln sagen aus, dass jedes neue Gewicht w und jedes neue Bias b auf Basis des alten w bzw. alten b aktualisiert werden. Von diesen wird Ableitung der Verlustfunktion in Bezug auf den jeweiligen Parameter abgezogen. Zusätzlich wird die diese mit α der Lernrate multipliziert. Nachdem die Parameter aktualisiert wurden, kann wieder mit der Forward Propagation, mit den nun aktualisierten Parameter, begonnen werden und der nächste Trainingsdurchgang kann starten. |  | |

3) Model Performance: evaluating a model’s performance & Model Selection: selecting among competing models

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bias-Variance Trade-Off -   |  |  | | --- | --- | | (Hoher) Bias -Durchschnittliche Fehler zwischen der **Model Prediction und dem «wahren» Wert** ➔ Der Bias der geschätzten Funktion gibt an, wie **gut das Modell die Werte vorhersagen** kann. | • übervereinfachtes Modell  • Underfitting  • Hoher Fehler auf Trainings- und Tesetdaten | | (Hohe) Varinanz - Durchschnittliche Abweichung der **Model Prediction vom Erwartungswert** auf dem Datensatz ➔ Die Varianz der geschätzten Funktion gibt an, wie stark sich das Modell an die **Veränderung des Datensatzes anpassen** kann. | • überkomplexes Modell  • Overfitting (Lernt Noise der Trainingsdaten)  • Niedriger Fehler Trainingsdaten|hoher Testdaten |   ➔ Beste Modell liegt im Minimum des Testfehlers (Generalisierungsfehler (gelb)); es ist ein Kompromiss: Die einzige Möglichkeit, beides zu reduzieren, ist die Vergrößerung der Stichprobe oder durch Ensembling (Boosting). | | Image for post |
| Model Perormance: Bewertung von Classificcation   |  |  | | --- | --- | | **Accuracy** ➔ wie viele Items wurden erkannt?  **Recall** ➔ wie viele der relevanten Items wurden erkannt? (Ausbeute)  **Precision** ➔ wie viele der zurückgegeben Items sind relevant? (Präzision) | ➔ tradeoff, deshalb:  **F-Measure** = harmonisches Mittel | | **receiver operating characteristic curve (ROC-Kurve)**  True Positive Rate = Recall = ➔ wie viele der relevanten Items wurden erkannt? (Ausbeute) False Positive Rate = ➔ wie viele der nicht relevanten Items wurden erkannt? (Fehlalarm)  ➔ Je häufiger man richtig erkennen möchte (true positive), desto häufiger wird es auch Fehlalarm (false positive) geben; das Modell wird damit **sensitiver** (y-Achse). Güte-Mass: Fläche unter der ROC-Kurve (**AUC**). | |   Model Perormance: Bewertung von Prediciton  Mean-squared error | Ausreisser überproportional 🡪 deshalb mean absolute error  Model Selection: selecting among competing models  **Maximum Likelihood (ML)** - Parameter so bestimmen, dass beobachteten Daten am wahrscheinlichsten sind.  1. Das Modell mit dem besten Cross Validation-Wert  2. Ockhams Razor: einfachere Modelle bevorzugen, wenn keine anderen Beweise vorliegen |  | |

4) Support Vector Machines

|  |  |
| --- | --- |
| 1. (Maximal margin) hyperplane classifier ➔ linearly separable data  ➔ Es soll die **Hyperebene** gewählt werden, die den **Abstand** von der Hyperebene zum **nächsten Trainingspunkt** (jeder Seite) **maximiert** (Maximierung der Margin). Die Hyperebene kann als Linearkombination\* von nur wenigen Trainingspunkten dargestellt werden – sogenannte **«Support Vectors»\*\***  \* Linearkombination: Summe von Vektoren (Vektoraddition), wobei jeder Vektor noch mit einer reellen Zahl (Linearfaktor) multipliziert wird.  \*\* Support Vector: Diese Punkte liegen auf dem **Rand** und **definieren damit die «Lösung»**; es sollen die Punkte gefunden werden, die Marge maximieren.  2. Support vector classifier ➔ almost linearly separable data  ➔ **Soft Margin**, die einige **Fehlklassifizierungen** zulässt. Manchmal ist ein größerer Spielraum es wert, falsch klassifizierte Beobachtungen zu haben. Die Anzahl zulässiger Misclassifications wird durch **Penalty Factor**  kontrolliert. stellt das Budget für Misclassification dar; der Parameter misst für jeden Punkt den Abstand, den er von der jeweiligen Support-Hyperebene zur anderen Klasse überschreitet (grüne Linien).  Der **Hyperparameter** definiert die **Toleranz gegenüber Fehlklassifikationen** ➔ **Generell gilt: steuert den Bias-Varianz-trade-off in der SVM!**  **• klein:** breites Band / geringe Strafe ➔ zu klein: Übergeneralisierung (underfitting) | **• gross:** schmales Band / hohe Strafe ➔ größer: konvertiert zu Lösung  Größere führen zu kleinerer Margin ➔ führt weniger Support Vektoren (Lösung hängt von weniger Punkte ab) ➔ daher höhere Varianz  3. Support vector machine ➔ non-linearly separable data  ➔ Die Idee ist, den Vektorraum und damit auch die darin befindlichen Trainingsvektoren in einen **höherdimensionalen Raum** zu überführen (**Kernel-Trick**).  ist eine lineare Gleichung. Das bedeutet, dass SVM am besten funktioniert, wenn man die Daten linear klassifizieren kann.  **Beispiel**: Der **RBF-Kernel** (Gaussian Kernel) projiziert die und in einen **höherdimensionalen Raum** und findet eine Hyperebene, die die Farben (rot/violet) sauber trennt. Bei der Projektion zurück in den 2d-Raum bildet die Grenze ungefähr einen Kreis; dies **ermöglicht nicht-lineare Entscheidungsgrenzen**.  Der **Hyperparameter**  des RBF-Kernel definiert, **wie weit der Einfluss eines einzelnen Punktes reicht**, wobei niedrige Werte "weit" und hohe Werte "nah".  **• klein:** bezieht mehr Punkte ein und hat glattere Grenzen | **• gross:** bezieht weniger Punkte ein und passt sich enger an die Daten an  \***Kernel**: Algorithmen, die werden verwendet, um ein **nichtlineares Problem** mit einem linearen Klassifikator zu lösen. Der SVM verwendet einen sogenannten "Kernel-Trick", bei dem die Daten transformiert werden und eine optimale Grenze für die möglichen Ausgaben gefunden wird. |  |

5) Ensemble Methods

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Meta-Learning** = basiert auf Kombination unterschiedlicher Modelle - nicht auf Daten selbst; mit dem **Ziel**: Kombination **mehrerer komplementärer\* (ergänzender) Modelle** zur Steigerung der Performance; \***komplementär** = Modelle können - müssen aber nicht - aus gleichen Hypothesenraum sein, **müssen sich aber ergänzen**!   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Statistik**  Mittelwertbildung kann eine bessere **Annäherung** an das wahre sein |  | **Rechnerisch**  Vermeidung **lokaler Minima**: Wiederholung mit zufälligen Neustarts |  | **Repräsentation**  Wahre lässt sich evtl. nicht über gewählten **Hypothesenraum** abbilden |  | | Da wir das beste nicht kennen, deshalb **Durchschnitt** | | Möglicherweise finden wir nicht das Beste, deshalb **Durchschnitt** | | Da wir das beste nicht finden können, deshalb **Durchschnitt** | | |

|  |  |
| --- | --- |
| Bagging (parallel) basiert auf der Idee einer **kollektiven Entscheidung** von Classifiern und macht sich Bootstrapping\* zur Grundlage.  **Vorgehen**: Durch **paralleles** Trainieren von  **Modellen auf den unterschiedlichen Trainingsdaten** (Bootstrapping) entstehen komplementäre Modelle. Die Entscheidung basiert dann auf **Mehrheitsentscheid** (Klassifikation) oder **Durchschnitt** (Regression). \*Bootstrapping: dabei werden durch **Ziehen mit Zurücklegen** zunächst **unterschiedliche Trainingsdatensätze** erzeugt.  **Impact**: **reduziert die Varianz – nicht Bias** | durch Mittelwertbildung der Performance wird die **Streuung (Varianz) reduziert; der Mittelwert bleibt gleich** (Bias) |  |
| Adaptives Boosting (sequenziell) basiert auf der Idee, **schwache Classifier sequenziell zu kombinieren** (geboostet).  **Vorgehen**: Es wachsen **sequenziell** schwache Entscheidungsbäume. Dabei **bestraft** die Methode **falsch vorhergesagte Stichproben**, indem ihnen nach jeder Vorhersagerunde ein **größeres Gewicht** zugewiesen wird. Auf diese Weise lernt der Algorithmus aus früheren Fehlern. Die endgültige Vorhersage ist die **gewichtete Mehrheit**.  **Impact**: **reduziert die Varianz und Bias**; immun für Overfitting aber sehr anfällig für Ausreisser | Es wird ein Classifier trainiert und sich dann darauf konzentriert, wo er Verzerrungen (Bias) aufweist. Durch das höhere Gewichten von Fehlern wird durch **jede Iteration der Bias reduziert**.  Beim AdaBoost werden falsch vorhergesagte Datensätze im jeweils nächsten Durchlauf höher gewichtet. Beim **Gradient Boosting**, werden Gewichtungen explizit in Gegenrichtung des Prädiktionsfehlers angepasst. Die Methode ermöglicht die Berücksichtigung anderer Lossfunktionen. |  |

6) Debugging machine learning algorithms

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Folk Knowledge **• explorative Datenanalyse •** schnelles **Prototyping •** Entscheidungen **experimentell verifizieren** («no free lunch») **• Ensembles** verschiedener (komplementärer) Modelle (immer Bagging des endgültigen Modells) **•** Nehme **Algorithmus mit geringem Bias** und füttern ihn mit **Tonnen von Daten** (sorgt für geringe Varianz) ➔ kleiner Testfehler **•** Probiere **einfache Algorithmen** (z.B. naiveBayes vor logistischer Regression, kNN vor SVM); probiere verschiedene Algorithmen **• Ockhams Razor** **• genügend Trainingsdaten** (Verhältnis zur Flexibilität des Lerners / VC-Dimension) | | | | | | | |
| Hochdimensionale Räume • **weniger dicht besiedelter Raum** bei gleichem N aber höherem • **Ähnlichkeiten werden verfälscht** • Problematisch, wenn **irrelevante Dimensionen** existieren  Fluch der Dimensionalität • **Irrelevante** Dimensionen schaden • Bedarf an Trainingsdaten wächst **exponentiell** mit 𝑝 • DL kann gut damit umgehen; ML mit Distanzen weniger (z.B kNN)  Segen der Dimensionalität • Mehr (**komplementäre**) Messungen = mehr Daten Komplexe • nichtlineare Beziehungen werden einfacher und linear in einer höherdimensionalen Darstellung | | | | | | | |
| Debugging Machine Learning Algorithms | | Underfitting | **Just right** | | Overfitting | |
| • high Bias ➔ is high | • training error etwas niedriger als der test error | | • high Variance ➔ is high | |
| **Reg|Clust|DL**  **Lerning Curve**  (Error vs Size N) |  |  | Ein Bild, das Transport, Rad enthält.  Automatisch generierte Beschreibung | | Ein Bild, das Zubehör enthält.  Automatisch generierte Beschreibung | |
| Learning curve | | • Plot training and test (CV) errors vs. **training set size 𝑁’** | • Vary 𝑁′ = 1. . 𝑁 (test/validation set stays full size) | | | | |
| Actions | | • neue Features • komplexeres Modell • Regularization lockern | | • weniger Features • mehr Daten • Regularization • Ensembles (wenig Daten) | | |
| Regularization - Regularisierung **bekämpft Overfitting**, indem es eine hohe Flexibilität bestraft (aber dennoch zulässt). **Regularisierungsparameter** für Learner :   |  |  |  | | --- | --- | --- | | **Regression** ➔Bestrafung hoher über Loss | • **SVM** ➔ bestraft Fehlklassifikation | • **NN** ➔ Dropout (zuf. weglassen von Neuronen) | | | | | | |  | |

7) ML System Design

Leistungsnachweis

* Pseudo-Code?
* Lernziele anschauen… Educational objectives (Verstehen ist das spannende!) -> Überlegung von Fragen, die auf Verstehen abzielen…
* Sprache Prüfung?