Maschinelles Lernen und Datenanalyse

In der Mess- und Prüftechnik PD Stefan Bosse

Universität Bremen - FB Mathematik und Informatik

Taxonomie des Maschinellen Lernens

Zielvariablen: Kategorische Klassifikation, Numerische Prädiktorfunktionen, Gruppierung

Modellfunktionen: Mit welchen Daten- und Programmarchitekturen können Eingabevariablen auf Zielvariablen abgebildet werden?

Training und Algorithmen: Wie können die Modellfunktionen an das Problem angepasst werden?

Überwachtes, nicht überwachtes und Agentenlernen

Datenverarbeitung

- Die Daten die als Grundlage für die Induktion (Lernen) und die Deduktion (Applikation/Inferenz der Zielvariablen) müssen i.A. vorverarbeitet werden → **Merkmalsselektion**

[6]

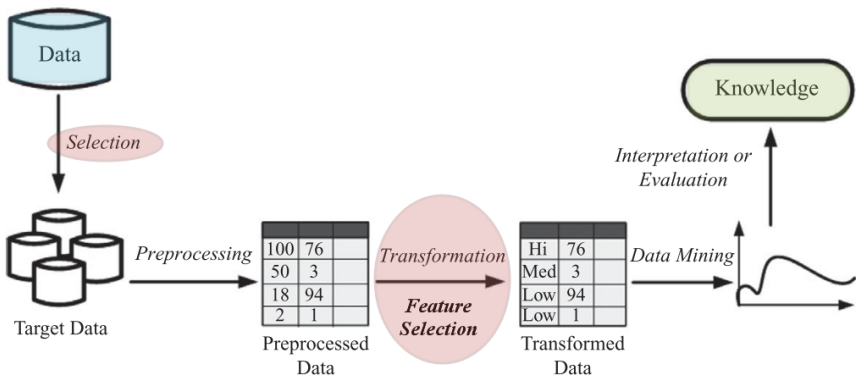


Abb. 1. Maschinelles Lernen ist ein Werkzeug der Datenanalyse und des Data Minings

Die Modellfunktion

- Die Modellfunktion *F* soll möglichst genau und effizient die Eingabedaten ***X*** auf die Zielvariablen ***Y*** abbilden:

$$F(\vec{X}) : \vec{X} \rightarrow \vec{Y},$$

$$X = \begin{cases} \text{diskrete kategorische Werte } \mathbb{C} \\ \text{numerische Werte } \mathbb{N},\mathbb{R} \end{cases},$$

$$Y = \begin{cases} \text{diskrete kategorische Werte } \mathbb{C} \\ \text{numerische Werte } \mathbb{N},\mathbb{R} \\ \text{Gruppen(X), Netzwerke } \mathbb{Q} \end{cases}$$

- Die Modellfunktion *F* **approximiert** eine i.A. nicht bekannte Funktion *M*, d.h. eine axiomatisch oder analytisch abgeleitete Modellfunktion (z.B. phys. Gesetze) → *F* ist **Hypothese** von *M*!

Beispiel

[100]

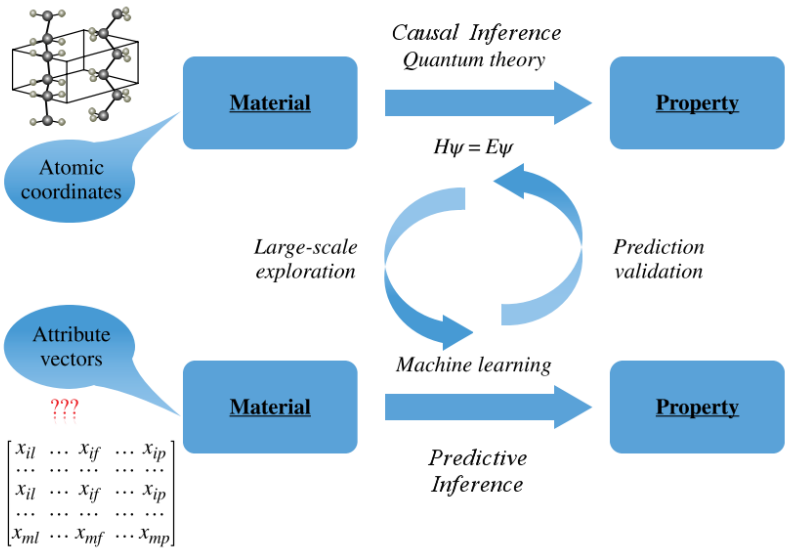


Abb. 2. Kausale vs. Prädiktive Modellbildung und Physikalische Modelle versa algorithmisch bestimmte Modelle (Hypothesen)

Beispiel

[100]

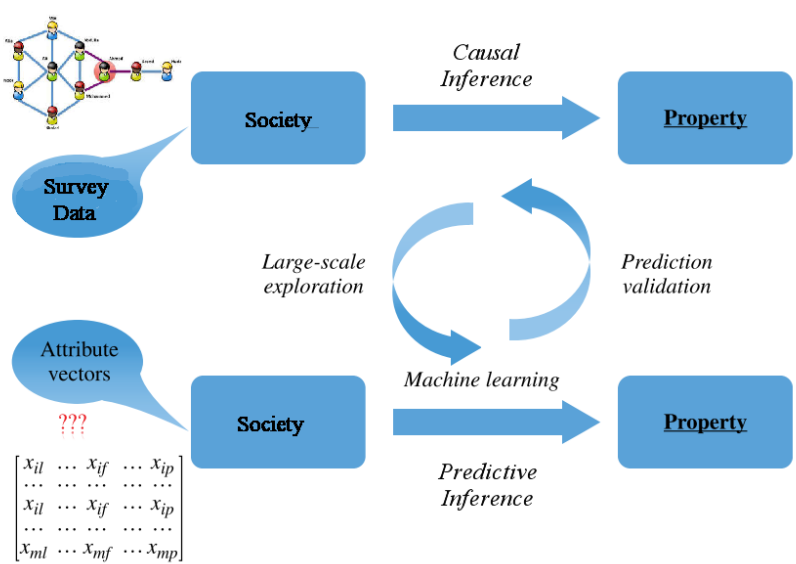


Abb. 3. Kausale vs. Prädiktive Modellbildung und Soziale Netzwerkmodelle versa algorithmisch bestimmte Modelle (Hypothesen)

Lernen

> Lernen bedeutet die unbekannte Modellfunktion M möglichst genau durch F aus Daten so zu approximieren dass $min\ error(|Y_0\text{-}Y|)$ für alle (X,Y_0) Paare gilt (Y_0: Referenzdaten).

- I.a. ist F eine parametrisierbare Funktion $f(\boldsymbol{P})$ oder eine parametrisierbare Datenstruktur
  - Der Parametersatz $\boldsymbol{P}=\{p_1,p_2,..,p_i\}$ bestimmt sowohl Funktion als auch Struktur (z.B. eines Entscheidungsbaumes)
- Es gibt nicht eine Modellfunktion $\boldsymbol{F}$, sondern eine große Menge möglicher Funktionen, genannt **Hypothesen** $\mathbb{H}=\{F_1,F_2,..\}$.

> Lernen bedeutet also die bestmögliche Anpassung der Parametersätze ***P*** um den Fehler zu minimieren und eine geeignet Hypothesenfunktion zu finden.

- Man unterscheidet bekannte Referenzwerte der Zielvariablen (und Beziehung zu X) Y_0, auch **Labels** genannt, und prognostische Werte Y die als Ergebnis von $F(X)$ geliefert werden (Inferenzwerte), d.h. bei der Applikation ist der wahre Wert Y_t unbekannt (**Schätzung** von Y_t)

$$H(\vec{X}) : \vec{X} \rightarrow \vec{Y},$$
$$H \in \mathbb{H} = \{F_1^{P1}, F_2^{P2}, .., F_k^{Pk}\},$$
$$error(X, Y_0, F) = |F(X) - Y_0|$$

Beispiele

[]

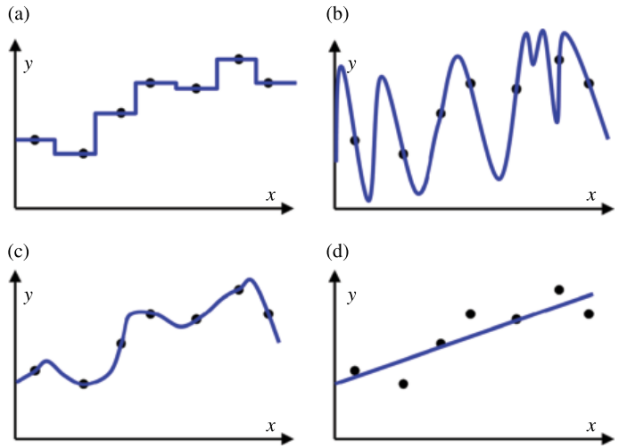
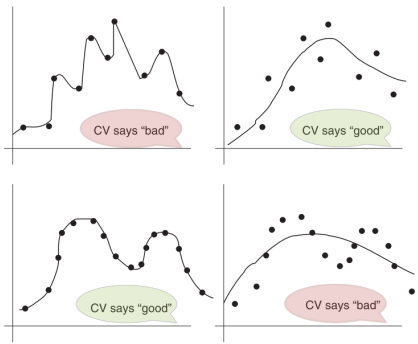


Abb. 4. Verschiedene Modellfunktionen *M* die die (Trainings) Daten repräsentieren

Kreuzvalidierung

- Beim Training wird ein Inferenzfehler zunächst aus Trainingsdaten bestimmt → Trugschluss!
- Stattdessen müssen auch unabhängige Testdaten für eine Kreuzvalidierung herangezogen werden, und dann ...



[13] Abb. 5. Durch Kreuzvalidierung (CV) werden ungeeignete Hypothesenmodelle erkannt

Fehler (Verlust)

Jede Hypothesenfunktion $F \in \mathbb{H}$ führt zu einem Informationsverlust durch Approximation der tatsächlichen und unbekannten Modellfunktion M.

- Es gilt also:

$$M(x) : x \rightarrow y = F(x) + E(x) + S$$

mit E als eine Fehlerfunktion (i.A. zufälliger Fehler) und $\hat{E}$ als mittlerer Prädiktionsfehler und S als systematischer Fehler.

- Die Hypothesenmenge $\mathbb{H}$ ist also tatsächlich eine Approximation eines unbekannten "exakten" Modells (Modellfunktion) M_F, die z.B. mittels physikalischer oder soziologischer Modelle ableitbar wäre.
- Genauso wie eine Sensor eine physikalische Größe nur approximieren kann, der tatsächliche Wert der zu messenden Größe ist nicht bekannt

[11]

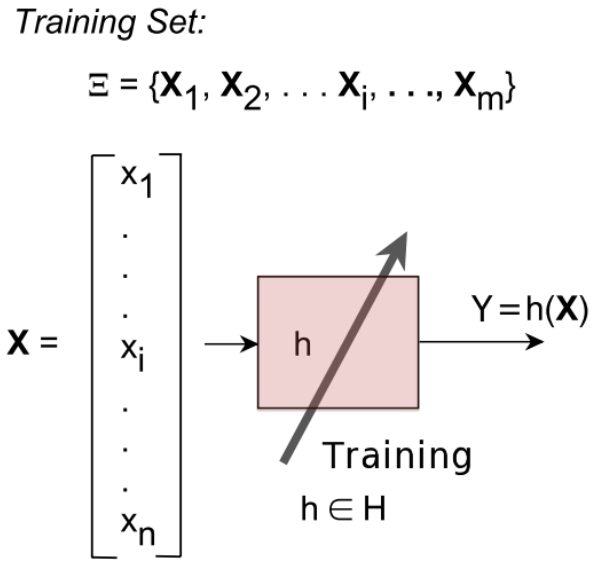


Abb. 6. Training als Anpassung von Hypothesen für die Abbildungsfunktion $X \rightarrow Y$ mit Trainingsdaten

Parametrisierung

Die Paramater in dem Parametersatz $\boldsymbol{P}$ bestehen aus zwei Klassen:

Statische Parameter P_s

Parameter die die Modellimplementierung (Funktion, Datenstruktur, usw.) festlegen und i.A. während des Trainings und der Applikation unverändert bleiben. (Ausnahme: Evolutionäre Algorithmen) → **Konfiguration**

Dynamische Parameter P_d

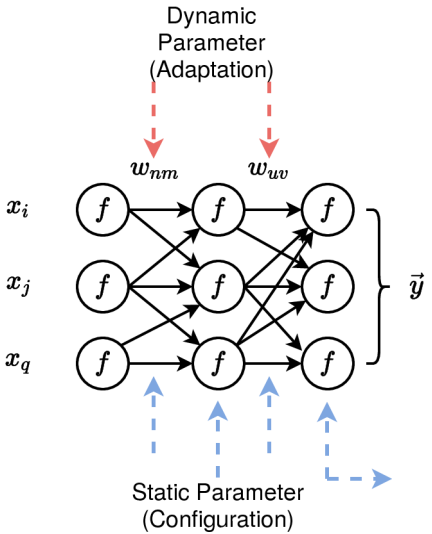
Parameter die während des Trainings verändert (angepasst) werden. Z.B. Funktionsparameter oder Kantengewichte von neuronalen Netzen → **Adaption**

Beispiele

1. Zwei mögliche Numerische Prädiktorfunktionen mit unterschiedlicher Struktur und Parametersätzen, aber gleicher Signatur(T: Temperatur, S: Satisfaction) → Regression

$$f(T) : T \rightarrow S = a + bT + cT^2 + dT^3,$$
$$P_s = \{degr : 3\}, P_d = \{a, b, c, d\}, S = [0, 1]$$
$$f(T) : T \rightarrow S = a + bT + cT^d + eT^f,$$
$$P_s = \{terms : 4, lin : 2, exp : 2\}, P_d = \{a, b, c, d, e, f\}, S = [0, 1]$$

2. Künstliches Neuronales Netzwerk



Daten

Trainingsdaten D_{train}

Datentabellen die aus Zeilen mit einer bekannten Beziehung (X,Y) bestehen und verwendet werden die Modellfunktion M durch Veränderung von $\boldsymbol{P}$ zu approximieren

Testdaten D_{test}

Datentabellen die aus Zeilen mit einer bekannten Beziehung (X,Y) bestehen und verwendet werden die Modellfunktion M auf Genauigkeit und Fehler zu testen. Man spricht auch von einer Kreuzvalidierung da $\boldsymbol{D}_{\text{test}} \cap \boldsymbol{D}_{\text{train}} = \emptyset$ sein sollte.

Inferenzdaten D_{inf}

Datentabellen die nur aus Zeilen $\boldsymbol{X}$ bestehen (Y ist unbekannt)

Es gilt: $\boldsymbol{D}_{\text{train}} \subseteq \boldsymbol{D}_{\text{all}}$, $\boldsymbol{D}_{\text{test}} \subseteq \boldsymbol{D}_{\text{all}}$, $\boldsymbol{D}_{\text{inf}} \subseteq \boldsymbol{D}_{\text{all}}$ aber $\boldsymbol{D}_{\text{train}} \cap \boldsymbol{D}_{\text{test}} = \emptyset$ und $\boldsymbol{D}_{\text{train}} \cap \boldsymbol{D}_{\text{test}} \cap \boldsymbol{D}_{\text{inf}} \neq \emptyset$ (Idealfall!)

Die großen Probleme beim Modellieren aus Daten:

- Die Trainingsdaten sind nicht repräsentativ (Umfang, Varianz, Qualität)
- Die Testdaten sind nicht repräsentativ (Umfang, Varianz, Qualität)
- Die Trainingsdaten enthalten schwache Variablen die nicht entfernt wurden (Inkonsistenz und geringer Informationsgewinn)

> **Generalisierung**. Das gelernte Modell F bildet alle drei Datenmengen gleichermaßen gut ab!

- Ergänzung:

Bewertungsdaten

Beim Einsatz eines gelernten Modells kann eine Evaluierung bezüglich Qualität / Genauigkeit stattfinden. Diese Daten können dann ggfs. für eine Adaption des Modells und dessen Parametersatz ***P*** verwendet werden.

> D.h. bei der Anwendung des Modells können somit auch neue Trainingsdaten gewonnen werden, z.B. im Rahmen eines Produktlebenszyklusmanagements!

Lernverfahren

Überwachtes Lernen

Es gibt Trainingsdaten mit bekannten Beziehungen (X,Y) die verwendet werden um die Modellfunktion mit minimalen Fehler anzupassen. Überwachung benötigt i.A. einen Experten der die Beziehungen (X,Y) erstellt und analytisch den Fehler bewertet.

Unüberwachtes Lernen

Es gibt Trainingsdaten ohne bekannte beziehung (X,Y), d.h., schon das Lernen führt zu einer automatischen Inferenz der Zielvariablen Y, die aber in diesem Fall i.A. nur durch Gruppenmengen $\mathbb{Q}$ bestehen. Eine Gruppenmenge $Q=\{X_i\} \subseteq \mathbb{Q}$ bringt verschiedene Eingabewerte in Beziehung. D.h. $Y \equiv \mathbb{Q}$.

Belohnungs- und Agentenlernen

Die Abbildungsfunktion $f(X)$: $X \rightarrow Y$ wird schrittweise durch eine Evaluierung des inferrierten Y mit einem Belohnungswert r=[0,1] gelernt. Training und Inferenz findet gleichzeitig statt.

[6]

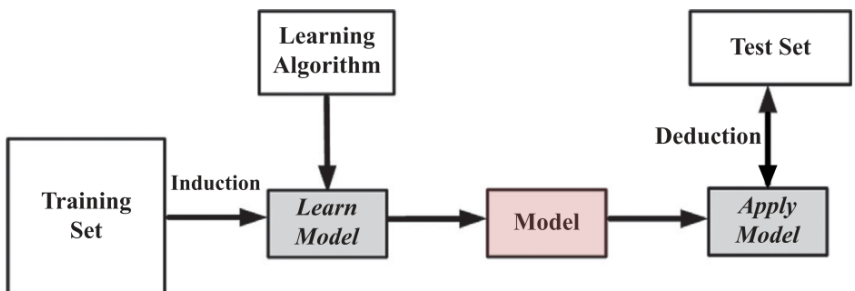
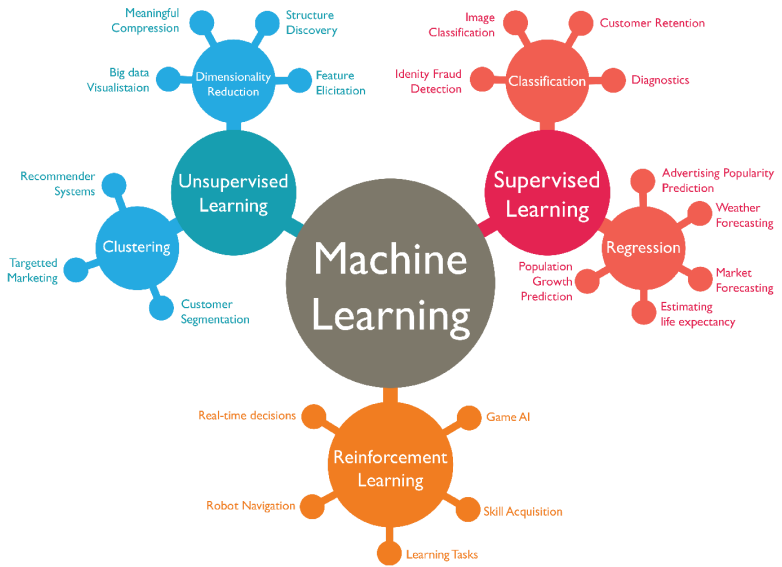


Abb. 7. Ablauf Überwachtes Lernen mit Trainings- (Induktion) und Applikationsphasen (Deduktion)

Taxonomie der Verfahren



[Abdul Rahid,www.wordstream.com]

Überwachte Lernverfahren - Unterklassen

[4]

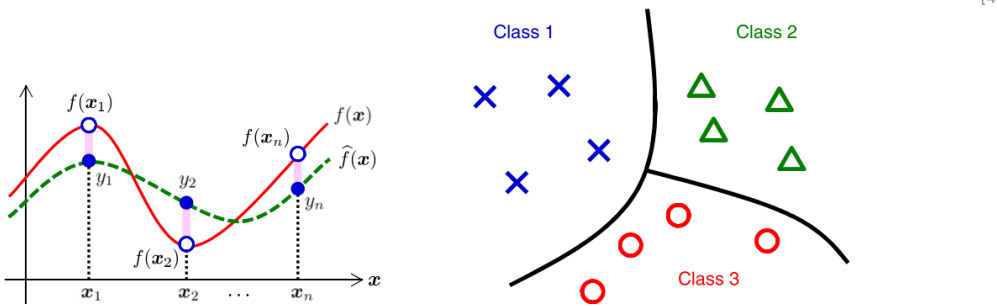


Abb. 8. Zwei wichtige Unterklassen von überwachten Lernen: Regression (Numerische Zielvariablen) und Klassifikation (Kategorische Zielvariablen)

Dimensionalitätsreduktion

- ML kann auch für die Reduktion von Datendimensionalität eingesetzt werden (Informationen sind reduzierte Daten)
  - Beispiele: Principle Component Analysis, Single Value Decomposition, ..

[4]

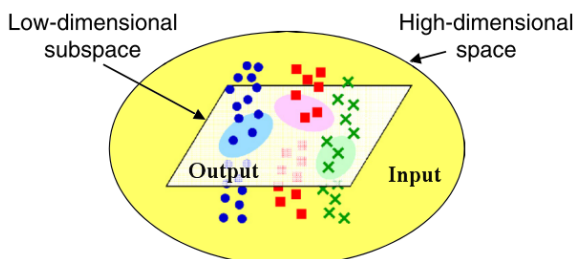


Abb. 9. Abbildung von hochdimensionale Daten X^n auf niederdimensionale X^m mit $m < n$

Unüberwachtes Lernen - Unterklassen

[4]

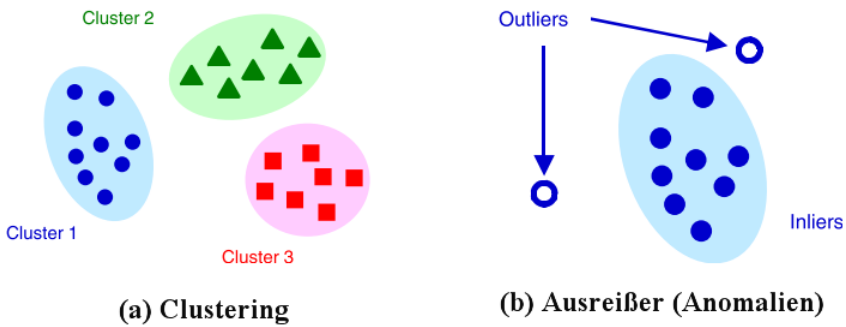


Abb. 10. Zwei wichtige Unterklassen von nicht überwachten Lernen: Clustering (Gruppenbildung) und Ausreißerdetektion

Training

- Das Training einer Modellfunktion *M* kann
  - **monolitisch** (alle Dateninstanzen werden "parallel" verwendet), oder
  - **stapelbasiert** (d.h. Gruppen von Instanzen werden "parallel" verarbeitet), oder
  - **iterativ** (Dateninstanzen werden "sequenziell" verwendet), und
  - **inkrementell** (iterativ mit neuen Daten).

Es gibt beim Training eine Fehlerfunktion (Cost or Loss Function). Der Fehler ergibt sich aus der Anwendung der Trainingsdaten auf das bisherigen Modell. Der Fehler dient zur Korrektur der dynamischen Parameter. Der Fehler kann dann aus einzelnen ("on-line") oder einer Gruppe ("batch") von Trainingsdaten berechnet werden.

- Inkrementelle Trainings- und Anpassungsverfahren können alte Datensätze verwerfen → Stromdatenlernen!
- Nicht jede Modellimplementierung ist geeignet:
  - Graphen (Bäume) können i.A. nur monolithisch trainiert = erzeugt werden!
  - Regression von math. Funktionen kann monolithisch und/oder iterativ erfolgen;
  - Neuronale Netze können monolithisch, stapelbasiert, iterativ, und vor allem inkrementell trainiert werden.

Modellimplementierungen

Es gibt im wesentlichen vier verschiedene Architekturen die Modelle M zu implementieren:

Funktionen

Die Struktur einer mathematischen Funktion wird durch ihre Terme gebildet (Berechnungsknoten), z.B. $ax+bx^2$. Zu jedem Term gehört ein dynamischer Parameter der beim Training angepasst wird um den Fehler zu minimieren. Das Ergebnis ist die Zielvariable y.

Gerichtete Graphen

Gerichtete Graphen (oder Entscheidungsbäume) bestehen aus Knoten und Kanten. Die Knoten repräsentieren eine Eingabevariable (Attribute) $x \in \boldsymbol{X}$. Die Kanten beschreiben die Entwicklung eines Graphen beginnend vom Wurzelknoten hin zu den Blättern. Die Blätter enthalten die Werte der Zielvariable(n) y. Der dynamische Parametersatz ist der Graph (dessen Struktur).

Funktionale Graphen

Hybrid aus gerichteten Graph und Funktion → Künstliche Neuronale Netze. Die Knoten repräsentieren Berechnungsfunktionen, die Kanten verbinden Ausgänge von Funktionen mit Eingängen. Es gibt Eingangsknoten die mit den Eingabevariablen $\boldsymbol{X}$ verbunden sind, und Ausgangsknoten die mit den Ausgangsvariablen $\boldsymbol{Y}$ verbunden sind.

Ungerichtete Graphen

Hier repräsentieren die Knoten Dateninstanzen X, und die Kanten verbinden die nächsten Nachbarn miteinander. Hier geht es um Gruppenbildung (k nächste Nachbarn/kNN Problem).

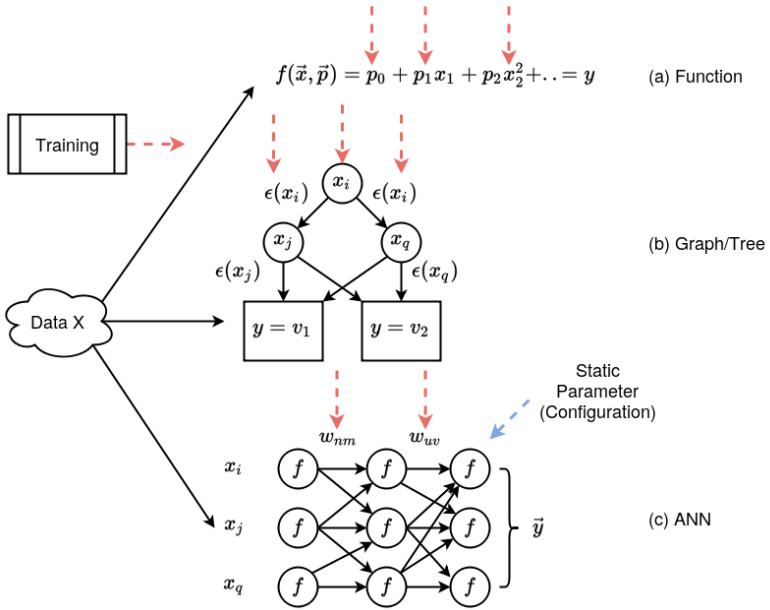


Abb. 11. Verschiedene Modellimplementierungen

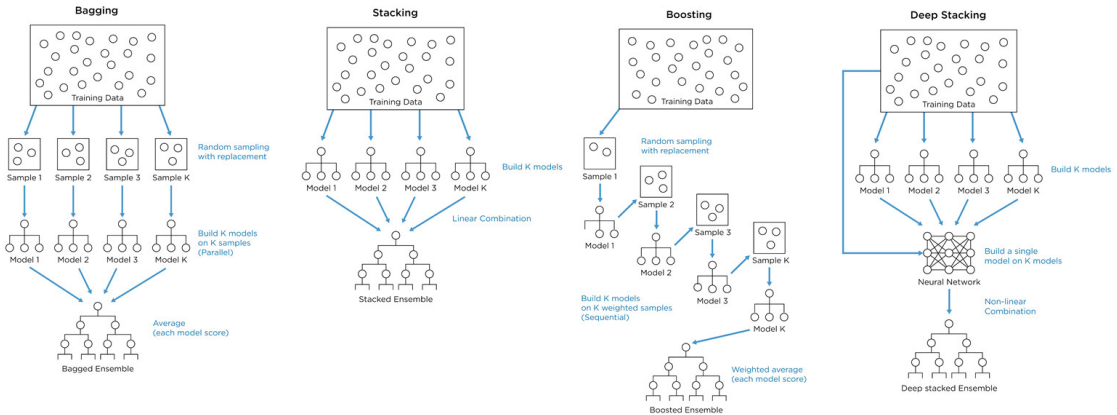
Hybride Modelle

Multiinstanz Modelle

- Ensemblelernen vereint multiple Modelle (gleicher Klasse oder unterschiedlich)

$$M(\vec{X}) : \vec{X} \rightarrow \vec{Y} = \Phi(\{M_1(X), M_2(X), .., M_n(X)\})$$

- Die einzelnen Modellinstanzen arbeiten mit gleichen oder verschiedenen Dateninstanzen
- Es gibt eine Split- und eine Join Schicht (Kombinierer, Modellfusion)



[Jay Budzik, www.thetalkingmachines.com]

Abb. 12. Verschiedene Architekturen für Multiinstanz Lernen und Inferenz

Instanzklassifikation

SLSP
: Einzelinstanz Lernen (auf allen Daten) und Einzelinstanz Prädiktion (Inferenz auf allen Daten)

SLMP
: Einzelinstanz Lernen (auf allen Daten) und replizierte Multiinstanz Prädiktion (Inferenz auf Teildaten mit Modellfusion)

MLSP
: Multiinstanz Lernen (auf Teildaten) mit Modellfusion und Einzelinstanz Prädiktion (Inferenz auf allen Daten)

MLMP
: Multiinstanz Lernen und Multiinstanz Prädiktion (Modellfusion)

Ablauf und Phasen von ML

0. Statistische Analyse und Bewertung der Daten
1. Merkmalsselektion
2. Aufteilung der Daten in Trainings- und Testdaten (i.A. randomisiert)
   $\boldsymbol{D} = \boldsymbol{D}_{\text{train}} \cup \boldsymbol{D}_{\text{test}}$
3. Training einer Modellfunktion F mit bekannten (markierten bei ÜL) Trainingsdaten $\boldsymbol{D}_{\text{train}}$ unter Bewertung des Modellfehlers $E(X)$
4. Test und Bewertung von F mit bekannten Daten $\boldsymbol{D}_{\text{test}}$
5. Applikation (Inferenz) von F auf unbekannten Daten $\boldsymbol{D}$

ML in der Soziologie

- Qualitative und quantitative Sozialwissenschaften wollen aus Daten **erklärbare** Modelle ableiten
  - Die Inferenz von Aussagen mit neuen Daten ist von geringer Bedeutung
  - Das Modell ist das Ziel
- Datenwissenschaften wollen aus Daten (ggfs. Black-Box) Modelle ableiten
  - Die Inferenz von Aussagen mit neuen Daten ist Ziel!
  - Das Modell selber ist nur das Werkzeug

Qualitative Kodierung

Qualitative Kodierung ist eine der wichtigsten Techniken, die in der qualitativen Analyse in den Sozialwissenschaften verwendet werden.

Im Allgemeinen bezieht sich die Kodierung auf den Prozess der Zuweisung beschreibender oder inferentieller Annotierungen zu Datenblöcken, die die Entwicklung von Konzepten oder Theorien unterstützen können.

Kodierung ist in der Regel eine sehr arbeitsintensive und zeitaufwendige Aufgabe.

Einsatz von ML

- ML Verfahren können zur Automatisierung der Q. Kodierung eingesetzt werden [101]

ML in der Soziologie findet sich vor allem in den ersten Stufen der "Wertschöpfungskette" → **Werkzeuge der Datenverarbeitung und Merkmalsselektion**

Soziale Analysen aus Texten

- Rückschlüsse auf soziales Verhalten und Netzwerkbildung können u.A. aus textuellen Quellen gewonnen werden:
  - Soziale Medien (Twitter, Facebook, Blogs, ...)
  - Nachrichten
  - Wissensdatenbanken
- Häufig ist Mustererkennung und Klassifikation zentrale Merkmalsselektion (mit Natural Language Processing NLP)

Einsatz von ML

- Textklassifikation und Vorhersage
- Suche nach Inkonsistenz (z.B. in juristischen Texten)
- Suche nach Textmustern (z.B. Betrug, Hass, usw.)

Soziologische und naturwissenschaftliche Modellinferenz

- Neben der kausalen Modellinferenz können auch prädiktive Modellinferenzverfahren - also ML - eingesetzt werden
- Spannende Frage: Wie ist die Korrelation von kausal und prädiktiv gewonnenen Modellen?
- Was bedeutet eine Abweichung?

Kernfrage ist die Erklärbarkeit von algorithmisch erzeugten Modellen mit ML Verfahren, auch in der Mess- und Prüftechnik!

ML in den Fertigungs- und Materialwissenschaften

- ML ist auch hier ein Werkzeug um analytisch und physikalisch nicht mathematisch modellierbare Zusammenhänge zu approximieren (quasi ein vorläufiger Modellersatz)
- Auch hier kann es Probleme grundsätzlicher Art geben:
  - Fehlende Nachverfolgbarkeit (warum kommt ein Y bei einem X?)
  - Fehlende Erklärbarkeit (wie ist der Zusammenhang Y(X) zu verstehen?)
  - Fehlende Rückverfolgung (welches X aus gegebenen Y?)
- Inverse ML Modellierung ist von großer Bedeutung (z.B. welche Prozessparameter müssen gewählt werden wenn bestimmte Materialparameter als Ergebnis einer Fertigung gegeben sind)

Vorwärts- und Rückwärtsmodellierung

Vorwärtsmodellierung

- Typischerweise schließt man von Eingabedaten (Sensoren) auf Ausgabedaten (Systemvariablen, Aggregatvariablen)
  - Eingabedaten sind i.A. individuell (Einzelfall)
  - Ausgabedaten von Modellfunktionen repräsentieren häufig statistische Ensemblemittelwerte!
  - Viele ML Modelle sind daher Mittelwertbilder!

Eine Funktion $F(\vec{X})$: $\vec{X} \rightarrow \vec{Y}$ bildet i.A. einen hochdimensionalen Eingaberaum $n=|\vec{X}|$ auf einen niederdimensionalen Ausgabe/Ergebnisraum $m=|\vec{Y}|$ mit $m \ll n$ ab

Rückwärtsmodellierung

- Bei der Rückwärtsmodellierung möchte man von den System- und Aggregatvariablen auf die Sensordaten schließen: $G(\vec{Y}):\vec{Y} \rightarrow \vec{X}$
- Die Modellfunktion G kann durch Invertierung des Vorwärtsmodells F gewonnen werden, d.h., $G=F^{-1}$

Kann F noch durch ein vollständig bestimmtes mathematisches Problem beschrieben werden (d.h. Abbildung $\vec{X} \rightarrow \vec{Y}$ ist eindeutig), so ist die Inversion i.A. ein unterbestimmtes Problem (Mehrdeutigkeit aufgrund der Dimensionalitätserhöhung)

Big Data Analysen

- Big Data bedeutet nicht groß (wenn auch meistens), sondern die Eingabevariablen sind scheinbar schwach korreliert, gekennzeichnete durch hohes Rauschen und Verzerrung!
- Aber mit ML kann auch solch schwachen Daten Informationen abgeleitet werden:
  - Genaue Wahlvorhersage
  - Demografische Vorhersagen
- Kritik: Die Datenvoreverarbeitung und ML Datenkette kann (ungewollt) zu Verzerrung und Offset führen.

Daher: Die "Fehler" in der ML Verarbeitungskette bezüglich sozialer Eigenschaften können nicht technisch gelöst und korrigiert werden. Dazu müssen wiederum Modelle der Soziologie verwendet werden. Der "Theorie Rein - Theorie Raus" Ansatz [102]!!

- Die Sozialtheorie hilft bei der Lösung von Problemen, die während des gesamten Aufbaus und der Bewertung von Modellen für maschinelles Lernen für soziale Daten auftreten.

Zusammenfassung Unterschiede Soziologische und naturwissenschaftliche Verfahren vs ML

- Soziologische und naturwissenschaftliche Theorie ist oft hypothesengetrieben, während maschinelles Lernen Daten sind!
- Beim maschinellen Lernen beginnt man mit einem Datensatz, um eine Hypothese aufzustellen, während man in der Soziologie oft mit einer Hypothese beginnt.
- Beide verwenden (oder eher ML, beide sollten zumindest) eine Auswertung außerhalb der Stichprobe, um Ihre Hypothesen zu testen.
- Beim maschinellen Lernen liegt der Fokus im Allgemeinen auf der Vorhersage, in der Soziologie nicht auf der Vorhersage, ohne zu erklären, warum ein Phänomen Auftritt.

- Beim maschinellen Lernen glaubt man nicht, dass das Modell richtig ist, dh. es wird nicht angenommen, dass das Modell der datengenerierende Mechanismus ist.
- Modelle werden nur danach ausgewertet, wie gut Sie anhand von Daten Vorhersagen machen, aus denen Sie selber nicht erstellt wurden, und nicht erklären wie sie zu Stande kommen.
- In der Soziologie betrachtet man allgemein, ob ein Koeffizient eines linearen Modells von null unterscheidbar ist; dies macht starke Annahmen über den datengenerierenden Mechanismus, den maschinelle Lerner nicht für gültig halten würden.
- Der Fokus des maschinellen Lernens lag traditionell nicht auf kausalen Effekten, obwohl Maschinelles lernen bei kausalen inferenzproblemen nützlich sein kann.

Zusammenfassung

Maschinelles Lernen besteht aus:

1. Modellimplementierungen:
   - Funktionen, Gerichtete Graphen, Funktionalen Graphen, Ungerichtete Graphen, also mit/für
   - Regression, Entscheidungsbäume, Neuronale Netze, Clustering (kNN)
2. Aufgaben
   - Regression, Klassifikation, Gruppierung (Clustering), Prognostik

3. Methoden und Verfahren
    - Überwachtes, nicht überwachtes, und rückgekoppeltes Belohnungslernen
    - Monolithisches, stapelbasiertes, iteratives, und inkrementelles Lernen
    - Einzel- versa Multiinstanzlernen
    - Enscheidungsbaumlernen (Konstruktion), Support Vector Machines (Regression), Backpropagation in Neuronalen Netze, usw.
4. ML besteht aus mehreren Phasen:
    - Datenerhebung (Messung), Datenvorverarbeitung, Statistische Bewertung, Merkmalsselektion, Modellerstellung, Training, Test und Analyse (Kreuzvalidierung), Anwendung/Inferenz

5. Daten werden unterteilt in:
    - Trainingsdaten , Testdaten, Anwendungsdaten
    - Trainings- und Testdaten bei ÜL mit (x,y) Beziehungen (Markierung/Labelling)