Aufgabe 1: (Suche)

a) **(1 Punkt)**
Ist Tiefensuche optimal, wenn Aktionskosten als identisch angenommen werden?

Nein.

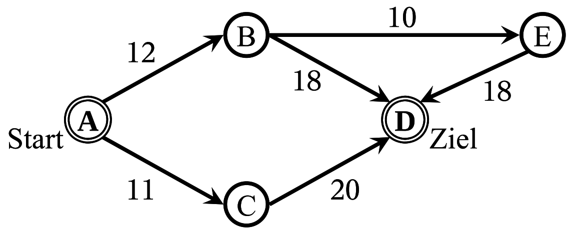
b) **(1 Punkt)**
Geben Sie eine Bedingung an, die ein Problemraum erfüllen muss, damit Tiefensuche vollständig ist.

Der Problemraum ist endlich.

c) **(1 Punkt)**
Welches Speicherprinzip verwendet Breitensuche?

Queue / FIFO

d) **(2 Punkte)**
Gegeben sei folgender Problemraum:



Es sei A der Startzustand und D der Zielzustand. Gerichtete Kanten repräsentieren die Aktionen, d.h. die gerichtete Kante (A, B) bedeutet, dass es eine Aktion gibt, die Zustand A in Zustand B überführt. Die Umkehrung gilt jedoch nicht.

Die Restwegkostenschätzung sei

h(A) = h(C) = h(E) = 17
h(B) = 24
h(D) = 0

Warum kann A* mit der angegebenen Restwegkostenschätzung nicht mit einer optimalen Lösung terminieren?

A garantiert nur dann eine optimale Lösung, wenn die Restwegkostenschätzung die tatsächlichen Kosten des Restwegs unterschätzen. Hier ist h(B) = 24 > 18 jedoch eine Überschätzung.*

Korrigieren Sie die Restwegkostenschätzung, so dass gilt:
- Die Schätzungen h(A), h(B), h(C), h(E) sind größer 0,
- A* garantiert eine optimale Lösung.

Korrigierte Restwegkostenschätzung:

h(A) = h(C) = h(E) = 17
h(B) = 18
h(D) = 0

e) **(2 Punkte)**

Wenden Sie den A* Algorithmus auf das in Aufgabe 1d) gegebene Problem an. Verwenden Sie dabei die von Ihnen korrigierte Restwegkostenschätzung. Notieren Sie in unten stehender Tabelle für jeden Suchschritt die Queue von Teilpfaden. Die Kosten müssen Sie nicht angeben.

Iteration	**Queue**
0	*(A, 17)*
1	*(AC, 28), (AB, 30)*
2	*(AB, 30), (ACD, 31)*
3	*(ABD, 30), (ACD, 31), (ABE, 39)*

Aufgabe 2: (Planen)

Modellierung eines Verschiebepuzzles in STRIPS

C	**A**
B	

Startzustand

A	**B**
C	

Ziel

a) **(1 Punkt)**

Definieren Sie geeignete Konstantensymbole, Prädikatensymbole und Variablen für die Modellierung des Verschiebepuzzles in STRIPS. Wählen Sie dabei Namen für die Symbole, aus denen sich sofort die beabsichtigte Bedeutung ablesen lässt. Tipp: Verwenden Sie Positionen und Nachbarschaftsbeziehungen und beachten Sie die Aufgabenstellung 2c).

- *Konstantensymbole: K = {A; B, C, *, 1, 2, 3, 4}*
  - *A, B, C, stehen für die Steine*
  - ** steht für das „freie Feld“*
  - *1, 2, 3, 4, stehen für die Spielfeldpositionen, durchnummeriert von links oben nach rechts unten.*
- *Prädikatensymbole: P = {neighbor, symbolAt}*
  - *neighbor steht für die Nachbarschaftsbeziehung zweier Felder*
  - *symbolAt steht für die Position eines Steins oder des freien Felds.*
- *Variablen V = {x, y}*

b) **(2 Punkte)**

Formulieren Sie den Startzustand (linke Konfiguration) und das Ziel (rechte Konfiguration) in STRIPS. Sie dürfen dabei nur die in Aufgabenteil a) definierten Konstanten, Prädikate und Variablen verwenden.

Startzustand:

symbolAt(C, 1), symbolAt(A, 2), symbolAt(B, 3), symbolAt(, 4),*
neighbor(1, 2), neighbor(1, 3), neighbor(2, 1), neighbor(2, 4),
neighbor(3, 1), neighbor(3, 4), neighbor(4, 2), neighbor(4, 3)

Ziel:

symbolAt(A, 1), symbolAt(B, 2), symbolAt(C, 3)

c) **(3 Punkte)**

Formulieren Sie ein Aktionsschema `move`, das die Vorbedingungen und Effekte eines Zuges beschreibt. Der Zug soll die Verschiebung des leeren Feldes beschreiben. Das Aktionsschema soll so formuliert werden, dass man einen Plan angeben kann, der den Startzustand in den Zielzustand überführt. Wie in den vorigen Aufgabenteilen dürfen Sie nur die in Aufgabenteil a) definierten Konstanten, Prädikate und Variablen verwenden.

ACT move(x, y, z)
PRE neighbor(x, y), symbolAt(, x), symbolAt(z, y)*
ADD symbolAt(, y), symbolAt(z, x)*
DEL symbolAt(, x), symbolAt(z, y)*

Bemerkung: Es wird die freie Stelle statt eines Steins verschoben. Dabei ist

- *x = aktuelle Position der freien Stelle*
- *y = Position der freien Stelle nach Anwendung von move*
- *z = Stein an Position y*

d) **(1 Punkt)**

Geben Sie einen Plan an, der den Startzustand in einen Zielzustand überführt.

move(4, 3, B), move(3, 1, C), move(1, 2, A), move(2, 3, B)

Aufgabe 3: (Resolution) **(8 Punkte)**

Beweisen Sie mit der Resolutionsmethode folgenden Schluss:

$$\forall x\ [(t(x) \vee o(x)) \rightarrow (m(x) \wedge g(x))]\ \vdash\ \forall x\ [(t(x) \vee o(x)) \rightarrow m(x)]$$

Beschreiben Sie Ihre Schritte und gestalten Sie alle Beweisteile nachvollziehbar.

Sei $F1 = \forall x\ [(t(x) \vee o(x)) \rightarrow (m(x) \wedge g(x))]$

Und $F2 = \forall x\, [(t(x) \vee o(x)) \rightarrow m(x)]$

Um F1 |- F2 zu beweisen ist eine Wissensbasis WB = $F1 \cup \neg F2$ *aufzubauen und daraus der Widerspruch* □ *per Resolution herzuleiten.*

<u>*Zuerst die KNF für F1 und ¬F2 bilden*</u> *(es gibt auch einen anderen Weg, der ohne Existenzquantoren auskommt, wenn man F2 umformt und erst nach Entfernen des Quantors die Negation bildet und daraus die KNF bildet):*

$\neg F2 = F3 = \neg \forall x\, [\neg(t(x) \vee o(x)) \vee m(x)]$

<u>*1. Implikation eliminieren:*</u>

$F1 = \forall x\, [\neg(t(x) \vee o(x)) \vee (m(x) \wedge g(x))]$

$F3 = \neg \forall x\, [\neg(t(x) \vee o(x)) \vee m(x)]$

<u>*2. negierte Quantoren auflösen*</u>

$F3 = \exists x\, \neg[\neg(t(x) \vee o(x)) \vee m(x)]$ $\qquad \neg \forall x\, P \Leftrightarrow \exists x\, \neg P$

<u>*3. existenzquantifizierte Variablen skolemisieren*</u>

$F3 = \neg[\neg(t(A) \vee o(A)) \vee m(A)]$ *Skolemkonstante A*

<u>*4. Variablen standardisieren entfällt.*</u>

<u>*5. Universalquantoren entfernen*</u>

$F1 = \neg(t(x) \vee o(x)) \vee (m(x) \wedge g(x))$

$F3 = \neg[\neg(t(A) \vee o(A)) \vee m(A)]$

<u>*6. KNF-Konnektoren bilden*</u>

$F1 = (\neg t(x) \wedge \neg o(x)) \vee (m(x) \wedge g(x))$ *(Negation nach innen)*

$\quad = (\neg t(x) \vee m(x)) \wedge (\neg t(x) \vee g(x)) \wedge (\neg o(x) \vee m(x)) \wedge (\neg o(x) \vee g(x))$ *(DeMorgan)*

Klauselmenge F1 = $\{ \neg t(x) \vee m(x),\ \neg t(x) \vee g(x),\ \neg o(x) \vee m(x),\ \neg o(x) \vee g(x) \}$

$F3 = \neg\neg\ (t(A) \vee o(A))] \wedge \neg m(A)$ *(Negation nach innen)*

$\quad = (t(A) \vee o(A)) \wedge \neg m(A)$ *(ist KNF nach Entfernen der doppelten Negation)*

Klauselmenge F3 = $\{ t(A) \vee o(A),\ \neg m(A) \}$

<u>*Resolutionsbeweis:*</u>

$WB = F1 \cup \neg F2 =$

$\{ \underbrace{\neg t(x) \vee m(x)}_{(1)},\ \underbrace{\neg t(x) \vee g(x)}_{(2)},\ \underbrace{\neg o(x) \vee m(x)}_{(3)},\ \underbrace{\neg o(x) \vee g(x)}_{(4)},\ \underbrace{t(A) \vee o(A)}_{(5)},\ \underbrace{\neg m(A)}_{(6)} \}$

In den folgenden Anwendungen der Resolutionsregel (4 Schritte, von links nach rechts) wird die Variable x in den ersten drei Schritten mit der Konstante A unifiziert.

1. $\dfrac{(3) \wedge (6)}{\neg o(A)}$ 2. $\dfrac{\neg o(A) \wedge (5)}{t(A)}$ 3. $\dfrac{t(A) \wedge (1)}{m(A)}$ 4. $\dfrac{m(A) \wedge (6)}{\Box}$

Aufgabe 4: (Unifikation) **(3 Punkte)**

Bilden Sie mit Hilfe des Unifikationsalgorithmus den allgemeinsten Unifikator für

f(x, g(x), y) und f(g(A), g(h(z)), g(z))

x,y,z sind Variablen, A eine Konstante. Gestalten Sie die Schritte nachvollziehbar.

```
f(x,g(x,y)        f(g(A),g(h(z)),g(z))

1 2 3 4 5         6 7 8  9 A B   C D       ← Positionen
```

1. Positionspaar = (2,7)

Θ = { x\g(A) }; Aθ = { f(g(A),g(g(A),y)), f(g(A),g(h(z)),g(z)) }

2. Positionspaar = (4,A)
g und h sind verschieden und keine Variablen. An dieser Stelle terminiert der Unifikationsalgorithmus, die Ausdrücke sind nicht unifizierbar.

Aufgabe 5: (Satz von Bayes, Bayes Netze)

Der Satz von Bayes lautet für zwei Ereignisse A und B:

$$P(A|B) = (\ P(B|A) * P(A)\ )\ /\ P(B).$$

a) **(2 Punkte)**
Benutzen Sie die Definition der bedingten Wahrscheinlichkeit um den Satz von Bayes zu beweisen!

Bedingte Wahrscheinlichkeit:	*P(A|B) = P(A,B) / P(B).*	*(1)*
(1) →	*P(A,B) = P(A|B) * P(B)*	*(2) und auch*
(1) →	*P(A,B) = P(B|A) * P(A)*	*(3)*
(2), (3) →	*P(A|B) * P(B) = P(B|A) * P(A)*	*(4)*
(4) →	*P(A|B) = P(B|A) * P(A) / P(B)*	*q.e.d.*

b) **(3 Punkte)**
Ihr Arzt informiert Sie, dass ein Test T für Krankheit K positiv ausfiel.
Wie hoch ist in Anbetracht dieses Testergebnisses die Wahrscheinlichkeit, dass Sie die Krankheit haben?

Es gilt: Der Test erkennt die Krankheit zu 98% (d.h., die Wahrscheinlichkeit, dass der Test positiv ausfällt, wenn die Krankheit vorliegt, ist 0.98, $P(\,T=w \mid K=w\,) = 0.98$) und weist die Abwesenheit der Krankheit zu 99% nach (d.h., die Wahrscheinlichkeit, dass er negativ ausfällt, wenn die Krankheit nicht vorliegt, ist 0.99, $P(\,T=f \mid K=f\,) = 0.99$). Die Krankheit ist allerdings selten, und betrifft nur einen aus 5000 Menschen Ihrer Altersgruppe.
[Den Rechenweg nachvollziehbar darstellen! Falls kein Taschenrechner vorhanden, sind Formeln, Einsetzen der Zahlenwerte und ggf. Kürzen ausreichend.]

gegeben: $P(\,T=w \mid K=w\,) = 0.98, \quad P(\,T=f \mid K=f\,) = 0.99, \quad P(K=w) = 0.0002$
gesucht: $P(K=w \mid T=w\,)$

$$P(K=w \mid T=w\,) = P(\,T=w \mid K=w\,)\ P(K=w) / P(T=w).$$

$$\begin{aligned} P(T=w) &= P(T=w \mid K=w)\ P(K=w) + P(T=w \mid K=f)\ P(K=f) \\ &= P(T=w \mid K=w)\ P(K=w) + (1 - P(T=f \mid K=f))\ P(K=f) \\ &= 0.98 * 0.0002 + 0.01 * 0.9998 \\ &\approx 0.01. \end{aligned}$$

Also

$$\begin{aligned} P(K=w \mid T=w\,) &\approx 0.98 * 0.0002 / 0.01 \\ &\approx 0.02. \end{aligned}$$

c) **(1 Punkt)**

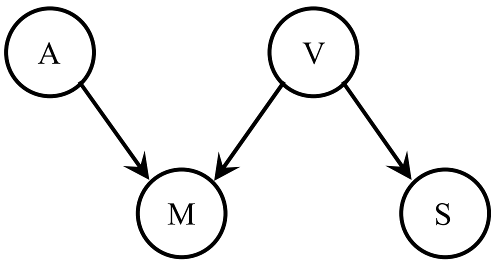
Konstruieren Sie ein Bayes Netz, das folgende Zusammenhänge über die Erbkrankheit Chorea Huntington *kausal* darstellt:

- Die Krankheit kann vom Vater an den Sohn weitergegeben werden
- Ihre Symptome zeigen sich meist zwischen dem 30. und dem 45. Lebensjahr.

Benutzen Sie dazu folgende Codierung für die Knoten:

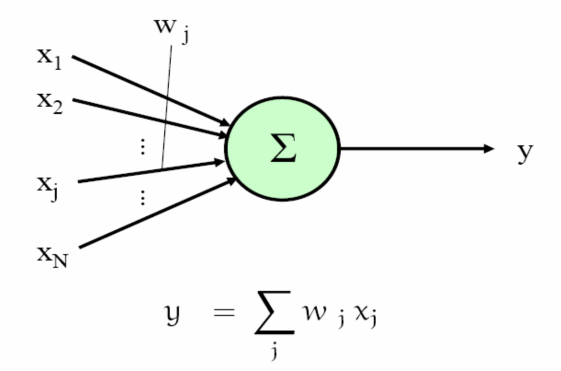
- V – Vater hat die Krankheit
- S – Sohn hat die Krankheit
- M – Der Vater zeigt Symptome der Krankheit
- A – Alter des Vaters

Lösung:



Aufgabe 6: (Neuronale Netze)

Die folgende Abbildung zeigt ein lineares konnektionistisches Neuron y mit Eingaben $x_1, x_2, \ldots, x_N$. Die Menge $\{\vec{x}^{(\alpha)}\}$ enthalte Vektoren $(x_1^{(\alpha)}, x_2^{(\alpha)}, \ldots, x_N^{(\alpha)})$ von Beobachtungen.



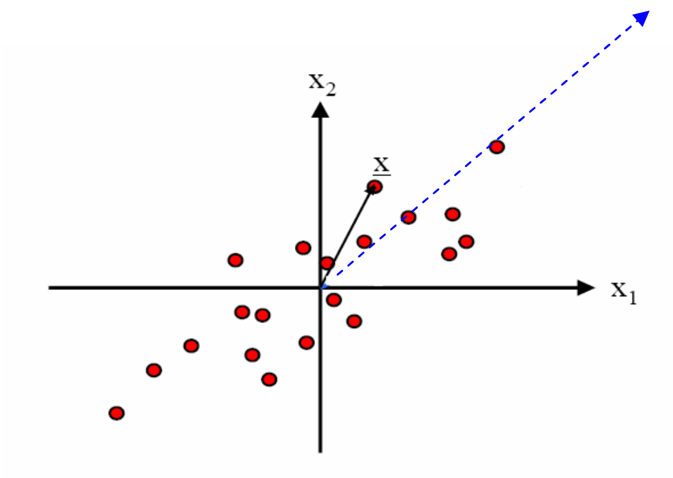
a) **(1 Punkt)**

Geben Sie die Änderung eines Gewichtes Δw_j für das Hebb'sche Lernen einer einzelnen Beobachtung $\vec{x}$ an!

$$\Delta w_j = \eta \; x_j \; y$$

b) **(2 Punkte)**

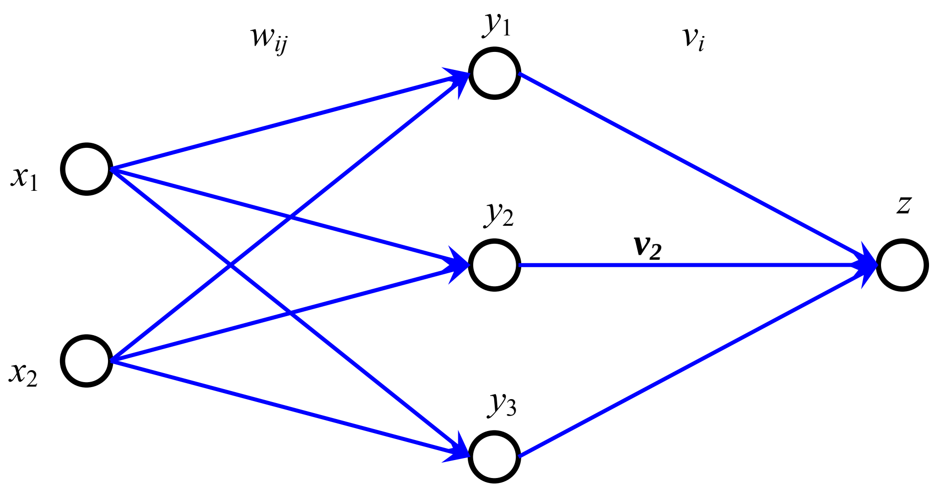
Beschreiben Sie in Worten, was die Konsequenz Hebb'schen Lernens ist, wenn es für zufällige Elemente aus der Menge $\{\vec{x}^{(\alpha)}\}$ von Beobachtungen iterativ durchgeführt wird! Zeichnen Sie den resultierenden Gewichtsvektor für die in der Skizze gezeigten Beobachtungen zweidimensionaler reeller Merkmale ein!



Der Gewichtsvektor (blauer Pfeil) konvergiert gegen die Richtung des Eigenvektors zum größten Eigenwert der Korrelationsmatrix; die Gewichte selbst divergieren.

c) **(5 Punkte)**

Betrachten Sie nun das folgende Multilagen-Perzeptron mit einer versteckten Schicht:



Die Eingaben $\vec{x} = (x_1, x_2)$ seien reellwertig. Die Transferfunktion aller Neuronen y_i und z sei die logistische Funktion $f(h) = 1 / (1 + \exp(-\beta h))$, wobei h die Gesamteingabe des jeweiligen Neurons ist. Die Kostenfunktion für ein einzelnes Trainingsbeispiel $(\vec{x}^*, z^*)$ ist der quadratische Fehler $e^* = ½ \ (z^* - z(\vec{x}^*, \vec{w}))^2$.

Für das gezeichnete Netzwerk und ein Trainingsbeispiel $(\vec{x}^*, z^*)$, schreiben Sie die Gewichtsveränderung Δv_2 des eingezeichneten Gewichtes v_2 explizit auf! Verwenden Sie Online-Gradientenabstieg mit Backpropagation. Benennen und interpretieren Sie die einzelnen Bestandteile des erhaltenen Ausdrucks!

$$\Delta v_2 = -\eta \frac{\partial e^*}{\partial v_2} = -\eta \frac{\partial e^*}{\partial z} \frac{\partial z}{\partial h} \frac{\partial h}{\partial v_2}, \qquad \textit{wobei}$$

$$\frac{\partial e^*}{\partial z} = (z(\vec{x}^*, \vec{w}) - z^*) \qquad \textit{die Kostenfunktion,}$$

$$\frac{\partial z}{\partial h} = f'(h) = (-\exp(-\beta h) \ / \ (1 + \exp(-\beta h))^2 \qquad \textit{die lokale Fehlergröße}$$

und

$$\frac{\partial h}{\partial v_2} = y_2 \qquad \textit{die Aktivität am Neuron } y_2.$$

d) **(2 Punkte)**

Warum benutzt man Kreuzvalidierung? Wie führt man Sie durch?

Kreuzvalidierung ist eine Methode der Abschätzung des Generalisierungsfehlers. Dieser kann zum Beispiel durch den Vorhersagefehler des trainierten Netzwerkes auf Testdaten geschätzt werden. Da aber in der Regel nicht viele Daten zur Verfügung stehen, kann man dazu Kreuzvalidierung benutzen:

Der Datensatz D *wird in* n *Teile* D_i *unterteilt. Das Netzwerk wird jeweils auf den verkleinerten Datensätzen* D/D_i *trainiert und der jeweilige Fehler bei der Vorhersage der Testdaten* D_i *gemessen. Der Generalisierungsfehler wird dann als der mittlere Vorhersagefehler auf den Testdatensätzen abgeschätzt.*

Aufgabe 7: (Bayes'sche Inferenz)

Bei Bayes'scher Inferenz wird eine Wahscheinlichkeitsverteilung für Zustände y über Attribute $\vec{x}$ durch eine Marginalisierung über die betrachtete Modellklasse generiert:

$$P(y \mid \vec{x}; Y, X) = \int P(y \mid \vec{x}, \vec{w}) P(\vec{w} \mid Y, X) d^N \vec{w},$$

wobei $\vec{w}$ der N-dimensionale Parametervektor der Modellklasse und (X, Y) die vorhandenen Trainingsdaten sind.

a) **(2 Punkte)**
Geben Sie eine Möglichkeit an, $P(\vec{w} \mid Y, X)$, also die Wahrscheinlichkeit eines bestimmten Modells aus der Modellklasse, gegeben die Daten (X, Y), zu berechnen! Kennzeichnen Sie in dem erhaltenen Ausdruck den Prior, Posterior, sowie die Likelihood der Daten!

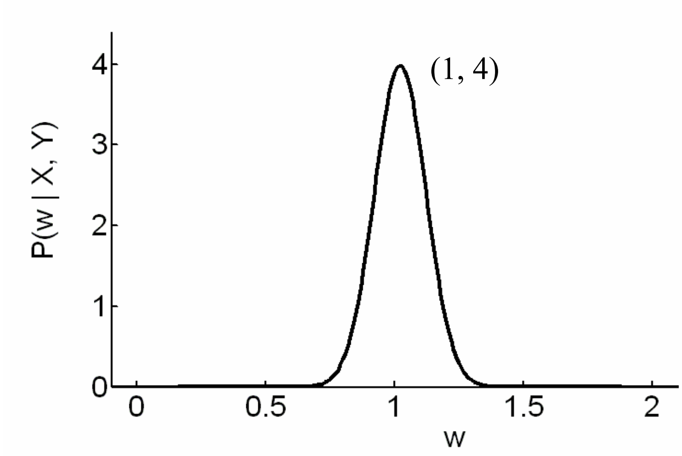
$P(\vec{w} \mid Y, X) \sim P(Y \mid X; \vec{w}) \quad P(\vec{w})$ (nach Satz von Bayes), wobei

$P(\vec{w} \mid Y, X)$ *der Posterior,*
$P(Y \mid X; \vec{w})$ *die Likelihood der Daten und*
$P(\vec{w})$ *der Prior sind.*

b) **(1 Punkt)**
Wie würden Sie den Prior wählen, wenn Sie über keinerlei Vorwissen verfügt?

Flat Prior / konstanter Prior / $P(\vec{w}) = konst$

c) **(3 Punkte)**
Es sei nun der reellwertige Zustand y aus dem reellwertigen Attribut x zu bestimmen. Die gewählte Modellklasse sei $y = w * x + \eta$ mit Gaußschem Rauschen η, also $P(\hat{y} \mid x, w) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\hat{y} - y(x, w) \right)^2 \right)$. Der Posterior $P(w \mid X, Y)$, erhalten aus den Trainingsdaten (X, Y) ist im folgenden Diagramm abgebildet.



Beschreiben Sie, wie Bayes'sche Inferenz durch das Maximum-A-Posteriori Prinzip vereinfacht wird!
Wenden Sie die Maximum-A-Posteriori Methode auf das oben abgebildete Beispiel an, um das wahrscheinlichste y zum Messwert $x = 3$ vorherzusagen!

Das MAP Prinzip geht davon aus, dass der Posterior ein scharfes Maximum hat, dessen Position durch das wahrscheinlichste Model w geschätzt werden kann. Bei einem schafen Maximum vereinfacht sich der Ausdruck*

$$P(y \mid \vec{x}; Y, X) = \int P(y \mid \vec{x}, \vec{w}) P(\vec{w} \mid Y, X) d^N \vec{w} \quad \textit{zu} \quad P(y \mid \vec{x}; Y, X) = P(y \mid \vec{x}, \vec{w}^*) .$$

Im oben gegebenen Fall also:
$w^* = 1$, also $y = w\,x + \eta = 1 * 3 + \eta$, das wahrscheinlichste y nach dem MAP Prinzip ist damit $y = 3$.

Aufgabe 8: (Lerntheorie / SVM)

In der Vorlesung wurde die VC-Dimension zur Beurteilung des Potentials einer Modelklasse für induktives Lernen eingeführt.

a) **(1 Punkt)**
Was gibt die VC-Dimension an?

Die VC-Dimension ist ein Kapazitätsmaß, welches die Komplexität einer Modellklasse bzw. die Schwierigkeit eines Lernproblems quantifizieren kann.

b) **(2 Punkte)**
Welche Vorteile hat die C-„Support Vector" Maschine gegenüber der Standard SVM?

Eine Standard SVM kann nicht trennbare Probleme nicht behandeln und neigt bei dicht beieinander liegenden Klassen zu hoher Modellkomplexität durch einstellen einer sehr kleinen Margin. Die C-SVM löst dies durch einen Kompromiss zwischen Modellkomplexität und Margin-Fehler.

(ÄLTERE VERSION DER AUFGABE)
b) **(2 Punkte)**

Sie haben zwei Modelle mit verschiedenen VC-Dimensionen, die beide den gleichen Trainingsfehler für einen bestimmten Datensatz aufweisen. Welches der beiden Modelle ist besser als Prädiktor für diese Art Daten geeignet und warum?

Das Modell mit der geringeren VC Dimension ist für Vorhersagen vorzuziehen, da induktives Lernen nur für $p > d_{VC}$ *möglich ist, also für Lernprobleme deren Schwierigkeit größer als die Mächtigkeit der Modellklasse ist.*