MachineLearning for Visual Computing
Aufgabenblock 1

Christian Brändle - Gruppe 5

November 26, 2014

0.1 Einfaches Perceptron - Datengeneration

Gegeben sind vier Datensets a 100 Beobachtungen von 2-dimensionalen Eingangsdaten, welche normalverteilt sind. Die Figuren in Tabelle 1 verdeutlichen dies.

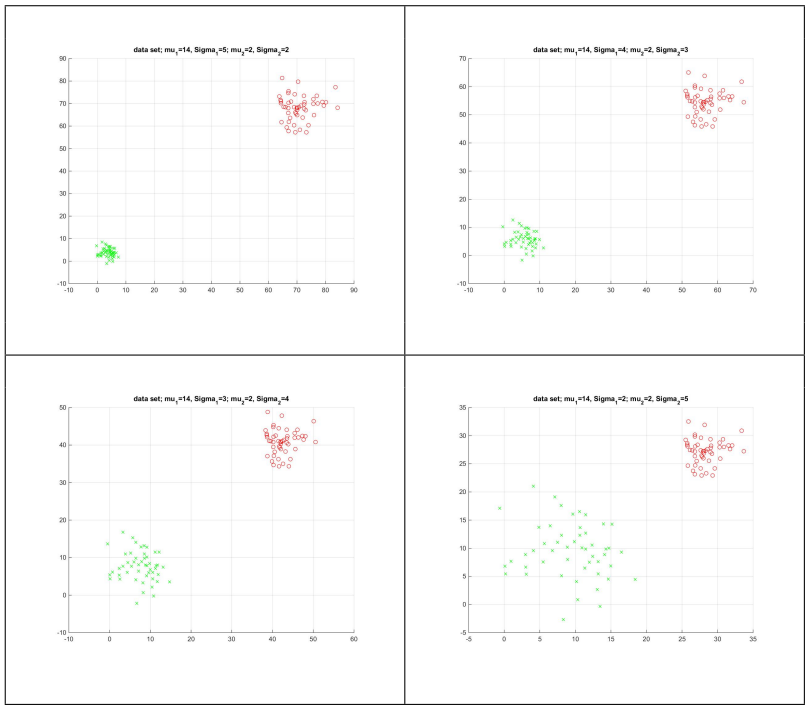


Table 1: Iteration distribution over gamma

Chapter 1

Einfaches Perceptron - Perceptrontraining

1.1 Anzahl Iterationen

Die Anzahl der Iterationen hängt maßgeblich von der *margin* der separierbaren Daten ab. Je kleiner die margin, das heißt der Minimalabstand zwischen den beiden Mengen, desto mehr Iterationen sind notwendig, bis der Algorithmus terminiert. Dies ist aus der Angabe der Iterationen aus 1.2 ersichtlich.

1.2 Welchen Einfluß hat die Schrittweite

Der Einfluß der Schrittweite hängt sowohl von den Eingangsdaten als auch vom gewählten Algorithmus ab. Es ist zu beobachten, daß der batch-Algorithmus keine großen Abweichungen zeigt, egal welches γ gewählt wird. Beim online-Verfahren ist der Einfluß größer, das heißt die Varianz in den ermittelten Iterationen ist höher, aber auch hier ist kein eindeutiger Bereich über alle Simulationen auszumachen, wo ein gegebenes γ die Iterationen des Algorithmus wesentlich reduziert. Zu sehen ist die Verteilung der Iterationen über ein γ von 0.1 bis 4 für die jeweiligen unterschiedlichen Datensätze in Tabelle 1.1.

1.3 Daten und Entscheidungsgrenzen im $\mathbb{R}^2$

Die ermittelten Entscheidungsgrenzen des batch-learning sowie des online-learning Algorithmus sind für die einzelnen Datensätze und ein gegebenes γ von 1 in Tabelle 1.2 zu sehen.

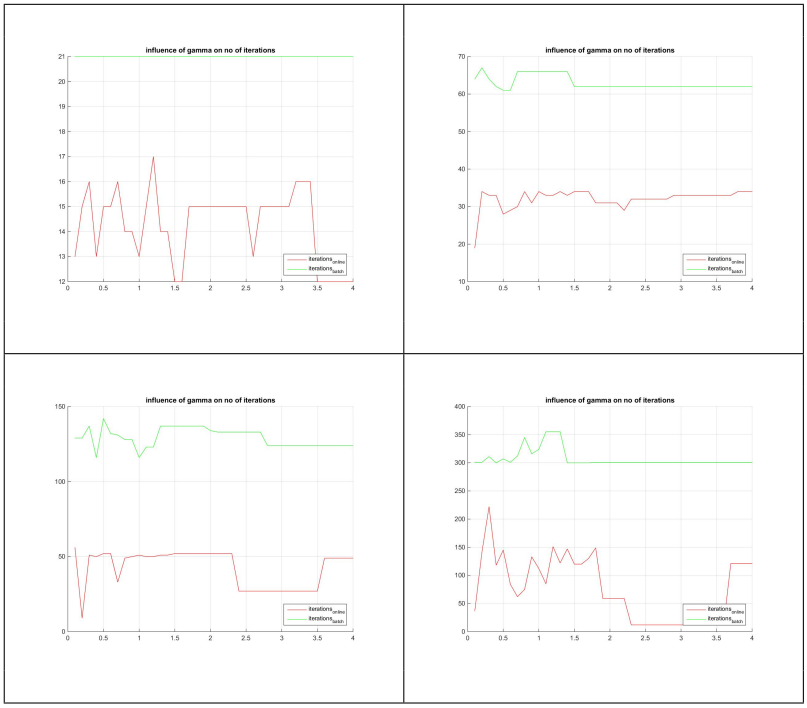


Table 1.1: iteration distribution over gamma

1.4 Wie ist das Verhalten bei nicht linear separierbaren Daten

Der Algorithmus terminiert nicht da bei jedem Durchlauf Daten gefunden werden, welche in der falschen Klasse landen.

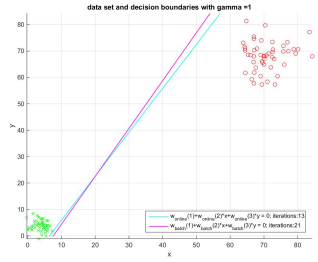
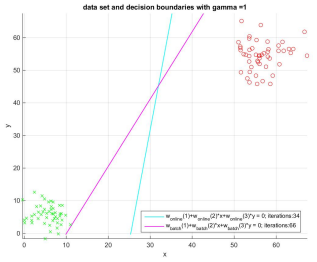
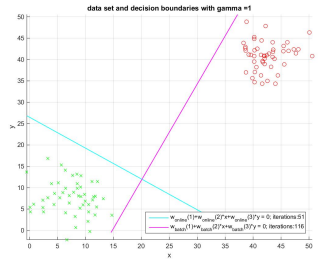
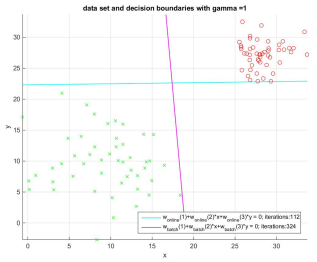
 set_1: $\mu_1 = 10$, $\Sigma_1 = 5$ set_2: $\mu_2 = 2$, $\Sigma_2 = 2$	 set_1: $\mu_1 = 10$, $\Sigma_1 = 4$ set_2: $\mu_2 = 2$, $\Sigma_2 = 3$
 set_1: $\mu_1 = 10$, $\Sigma_1 = 3$ set_2: $\mu_2 = 2$, $\Sigma_2 = 4$	 set_1: $\mu_1 = 10$, $\Sigma_1 = 2$ set_2: $\mu_2 = 2$, $\Sigma_2 = 5$

Table 1.2: iteration distribution over gamma

Chapter 2

Lineare Regression

Wir erstellen Daten im Bereich von $[0, 5]$ mit einer Schrittweite von 0.1 aus der Funkion:

$$y = x^2 - Gx + 1 \tag{2.1}$$

wobei $G = 10$ angenommen wird.

Des weiteren erstellen wir eine Trainingsmenge aus jedem sechsten Datenpunkt und fügen je einen Zufallswert aus $N(\mu = 0, \Sigma = 0.7)$ hinzu.

2.1 Gewichtsvektor mittels Gradientenabstieg bei quadratischer Fehlerfunktion

Der Gewichtsvektor w_{online} welcher aufgrund des online Verfahrens für *LMS* (least mean square) ermittelte wurde kommt in der Anwendung auf die Funktion $f(x) = x^2 - Gx + 1$ dem optimalen w^* zwar nahe, erreicht dessen Qualität aber nicht. Eine Verringerung der Fehlertoleranz würde hier sicher Abhilfe schaffen. In der Anwendung auf die Funktion $f(x)$ ergeben sich wie in Figur 2.1 gezeigt für die Dimensionalität 3 die gezeigten Kurven.

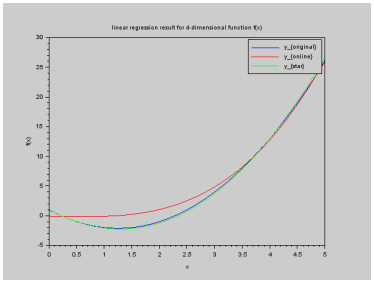


Figure 2.1: online vs batch LMS

2.2 Optimaler Gewichtsvektor w^*

Den optimalen Gewichtsvektor w^* kann man über die Bestimmung der Pseudoinversen A^+ für die Gleichung $Aw^* = b$ ermitteln. Die Gleichung $w^* = A^+b$ bestimmt das optimale w^*. Dabei berechnet sich A^+ wie folgt:

$$A^+ = (AA^T)^{-1} \quad \text{für invertierbare } AA^T \tag{2.2}$$

$$A^+ = (AA^T)^{-1} + \lambda I \quad \text{für nicht invertierbare} AA^T \text{ mit } \lambda << 1 \tag{2.3}$$

Das ermittelte Ergebnis für w^* mit normalisierten Daten aus dem $\mathbb{R}^2$ lautet: $w^* = (1.0106678, -10.0086, 1.8960336, 0.0262634)$.

2.3 Konvergenzverhalten in Abhängigkeit von γ

In Figur 2.2 ist zu sehen, daß die Anzahl der notwendigen Iterationen um zu einem Optimum zu gelangen mit der Größe von *lambda* exponentiell abnimmt bis sie bei einem Wert von $\lambda \approx 0.0001$ ein Minimum erreichen. Danach steigt die Anzahl der notwendigen Iterationen wieder an.

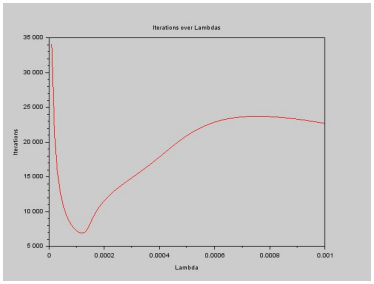
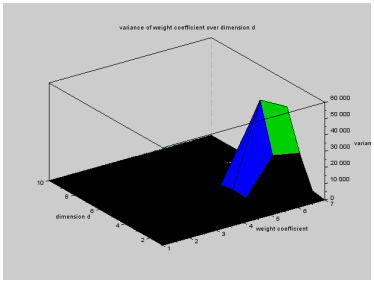


Figure 2.2: convergence over gamma

2.4 Erwartungswert und Varianz von w^* im Bezug zu Dimensionalitt d von $f(x)$

Erwartungswert und Varianz von w^* im Bezug zu Dimensionalitt d von $f(x)$ zeigt, dass je höher die Dimensionalitt d steigt, desto höher auch die Varianz *Sigma* der ermittelten Werte für w^* für verrauschte Eingangsdaten wird. Dies ist auch klar, da hherdimensionale Koeffizienten von w^* höhere Frequenzen repräsentieren, welche durch das Verrauschen der Eingangsdaten stärker betroffen sind als niedrigere Frequenzen. Interessanterweise steigt die Varianz aller Koeffizienten von w^* in Tabelle 2.1 nur bis zu einer Dimensionalität des Problems von 6, danach fällt sie wieder. Es ist aber anzunehmen, dass dies zum Einen aufgrund von Unzulänglichkeiten beim generieren der Pseudoinversen zur

Berechnung von w^* liegt und zum Anderen von der fehlenden Betrachunng noch höherer Dimensionen, wo sich globale Anstieg der Varianz sicherlich fortsetzt.



variance of weight coeffinients | deviation of weight coeffinients

Table 2.1: variance and deviantion of w^* over dimension

Des weiteren ist in Figur 2.3 festzustellen, dass für höherdimensionale Komponenten für w^* durch overfitting der Kurve $f(x)$ die Varianz zunimmt, zu sehen in den Schwingungen der Kurven.

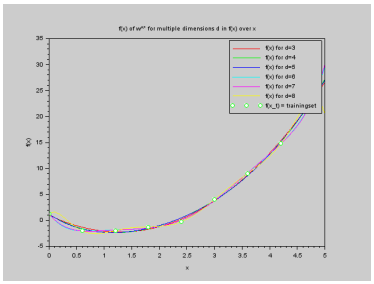


Figure 2.3: overfitting of $f_{w^*}(x)$ curves over dimension

2.5 Mittlere quadratische Abweichung von $f_{w^*}(x^*)$

Wie erwartet, steigt auch die mittlere quadratische Abweichung von $f_{w^*}(x^*)$ mit der Zunahme der Dimension d an, wie in Figur 2.4 zu sehen ist.

2.6 Gewichtsvektor w^* bei ungestörter Trainingsmenge über Dimension d

Die resultierenden Kurven für $f_{w^*}(x)$ welche aus der ungestörten Trainingsmenge ermittelt wurden fitten die Originalpunkte aus der Originalkurve $y =$

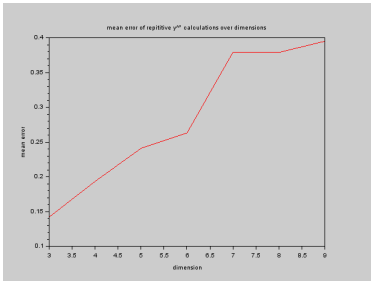
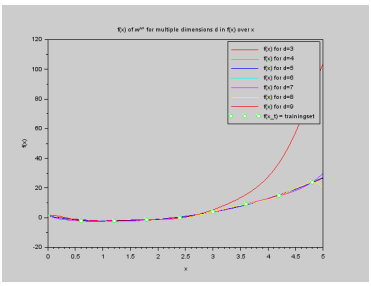
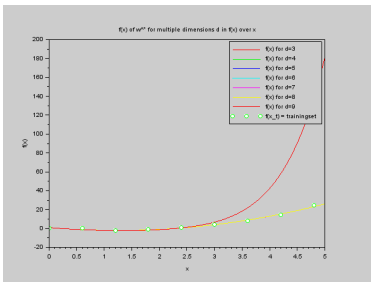
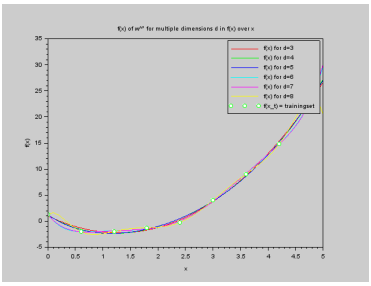


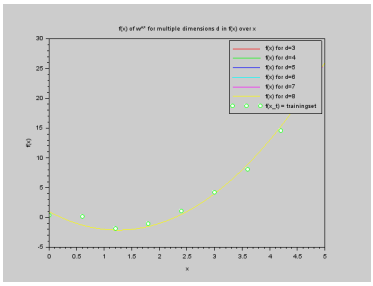
Figure 2.4: mean error of $f_{w^*}(x^*)$ curves over dimension

$2x^2 - Gx + 1$ naturgemäß wesentlich besser als die Kurven, welche auf der verrauschen Trainingsmenge basieren. Dies ist in Tabelle 2.2 zu sehen. Der Ausreißer bei Dimension 9 ist wiederum auf Schwächen in der Bestimmung der Pseudoinversen zurückzuführen.









w^* based on preturbed trainingsset	w^* based on not preturbed trainingsset

Table 2.2: $f_{w^*}(x)$ over dimension