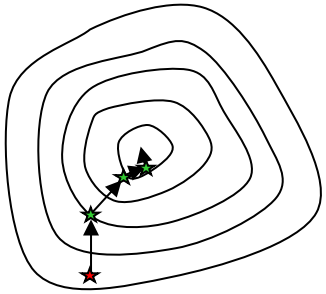
Optimierung

Vorlesung 7

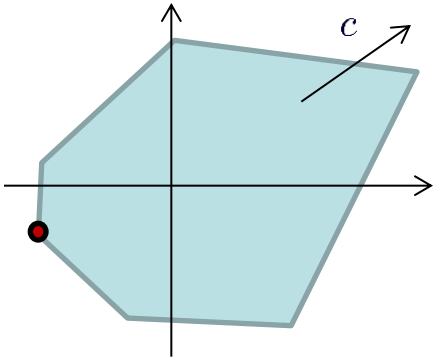
Nichtlineare Programmierung

Übersicht der bisherigen Verfahren



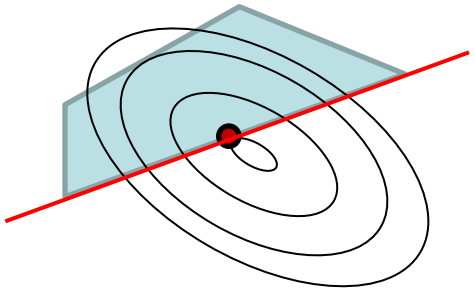
Optimierung ohne Nebenbedingungen

- Gradientenabstieg, Quasi-Newton, Newton
- Beliebige glatte, nichtlineare Funktionen



Lineare Programme (Simplex-Verfahren)

- Lineare Funktion, lineare Nebenbedingungen



Quadratische Programme (Active-Set-Verfahren)

- Konvexe quadratische Funktion,
  lineare Nebenbedingungen

Optimierung ohne Nebenbedingungen bisher so viel allgemeiner

- Lineare und quadratische Programme sind sehr eingeschränkte Problemklassen

- Die Active-Set-Verfahren waren sehr speziell auf diese Problemklassen ausgelegt.

- Wie lösen wir Probleme mit allgemeineren Funktionen und allgemeineren Nebenbedingungen?

- Es gibt zwei Arten von Ansätzen:
  1. Sequentielle quadratische Programmierung (SQP)
     (Lösen einer Reihe quadratischer Programme)
  2. Sequentielle Approximation durch Formulierungen ohne Nebenbedingungen
     (Projektionsverfahren, Strafverfahren, Barriereverfahren)

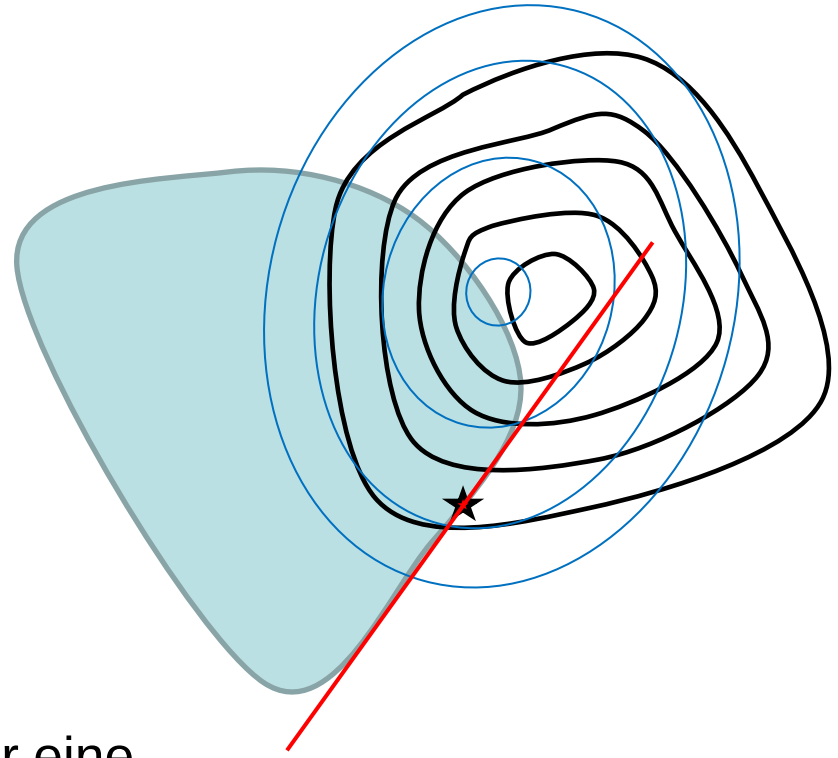
Grobe Idee sequentieller quadratischer Programmierung

- Hauptidee: lokale quadratische Approximation der Lagrange-Funktion und lokale lineare Approximation der aktiven Nebenbedingungen

- Motivation ähnlich wie bei der Optimierung ohne Neben-bedingungen:

  Quadratische Approximation entspricht Newton-Verfahren

  Ausführen eines Schritts, danach erneute Approximation

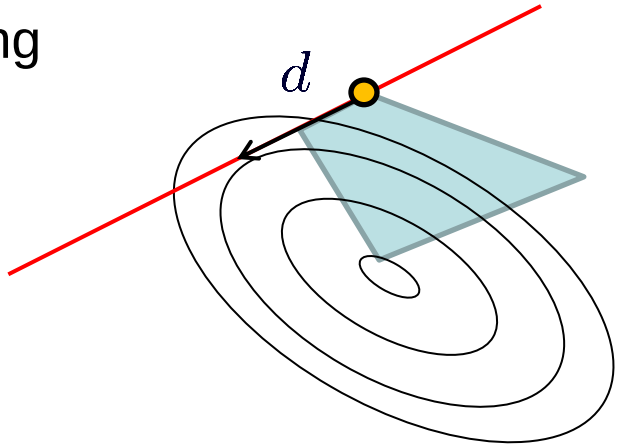


- Die Nebenbedingungen werden über eine Arbeitsmenge (wie bei der quadratischen Programmierung) berücksichtigt

  Die Arbeitsmenge wird immer wieder aktualisiert.

Projektionsmethoden

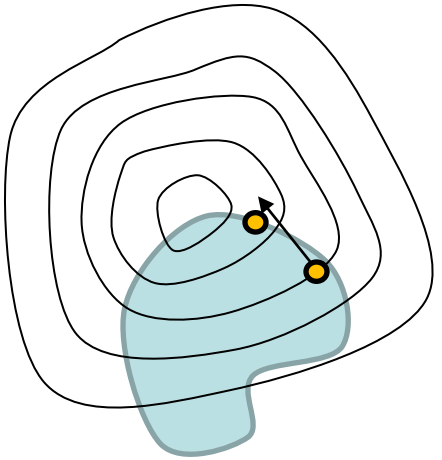
- Das Verfahren zur quadratischen Programmierung in der letzten Vorlesung verletzte zunächst Nebenbedingungen, die nicht in der Arbeitsmenge enthalten waren.



- Wir verhinderten dies, indem wir die Schrittlänge passend verkürzten.

- Könnten wir die Nebenbedingungen zunächst nicht einfach ignorieren, einen Gradientenschritt berechnen und den Schritt dann so anpassen, dass alle Bedingungen eingehalten werden?

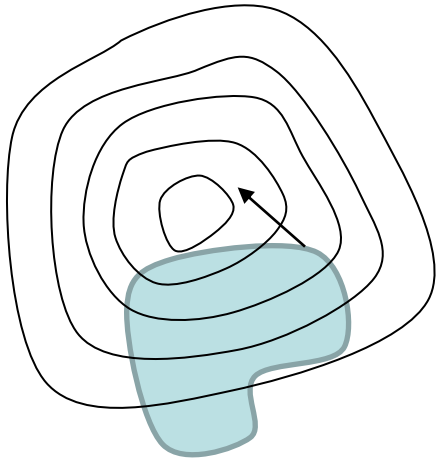
- Dies ist die allgemeine Idee von Projektionsmethoden mit verschiedenen Vor- und Nachteilen.



Schrittverkürzung reicht nicht

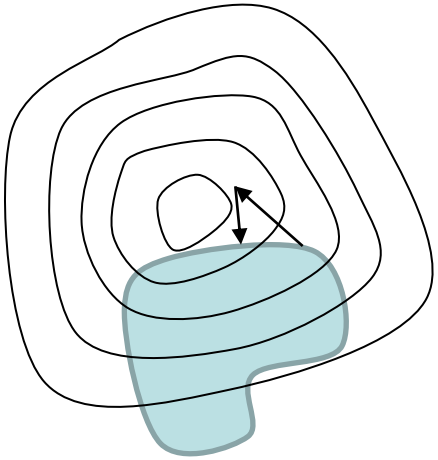
- Die Idee, den Schritt einfach so zu verkürzen, dass alle Bedingungen eingehalten werden, führt irgendwann zu Schritten der Länge 0.

  → Das Verfahren stoppt in einem Punkt, der die KKT-Bedingungen normalerweise nicht erfüllt.



- Um dies zu verhindern, müssten wir den Gradienten bereits unter Berücksichtigung der aktiven Bedingungen berechnen (Active-Set-Methode).

- Wenn wir zunächst alle Nebenbedingungen ignorieren möchten, müssen wir stattdessen die neue Lösung auf die gültige Menge zurückprojizieren.



- Aber wie projizieren?

Projektion des Gradienten

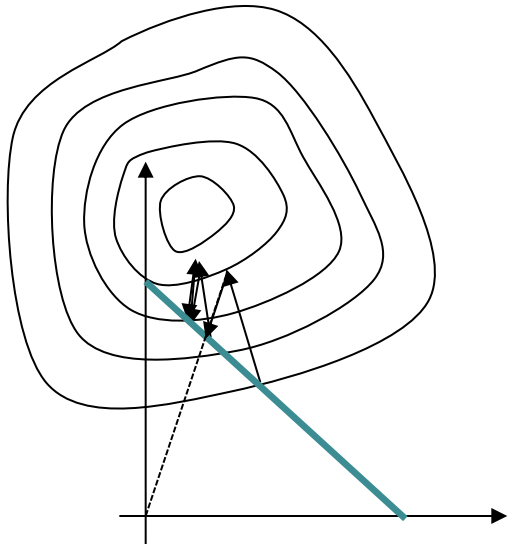
- Nicht jede Projektion ist geeignet.

- Beispiel: Funktion mit zwei Variablen x_1, x_2 und den Bedingungen

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$x_1 + x_2 = 1$$



- Projizieren wir nach jedem Schritt mittels

$$x_1 \leftarrow \frac{x_1}{x_1 + x_2} \qquad x_2 \leftarrow \frac{x_2}{x_1 + x_2}$$

konvergiert das Verfahren nicht zum Optimum.

- Der Schritt entlang der gültigen Menge wird irgendwann komplett von der schlechten Projektion kompensiert.

Beispiel kommt in einem Interpolationsproblem vor

- Gegeben: Einzelne Bildpunkte mit Labels
  Gesucht: Labels auf allen Bildpunkten

- Optimierungsproblem:

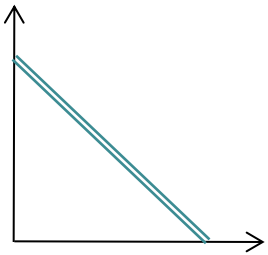
$$f(x) = \sum_{i=1}^{n} \left(c_i^\top x_i + \mathcal{R}(x_i) \right)$$

Kosten $c_i > 0$ wenn ein Bildpunkt nicht das vorgegebene Label annimmt plus ein Strafterm $\mathcal{R}$, für benachbarte Punkte mit unterschiedlichen Labels

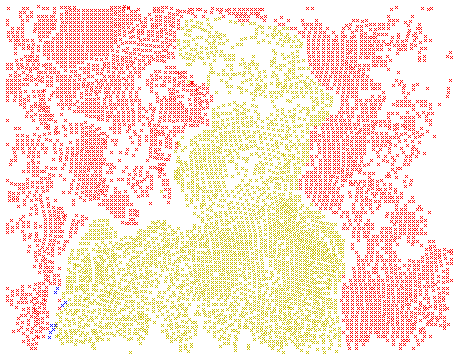
Nebenbedingungen:

$$\sum x_{il} = 1 \quad \forall l$$

$$x_{il} \in [0, 1] \quad \forall i, l$$



Resultat



Eingabe

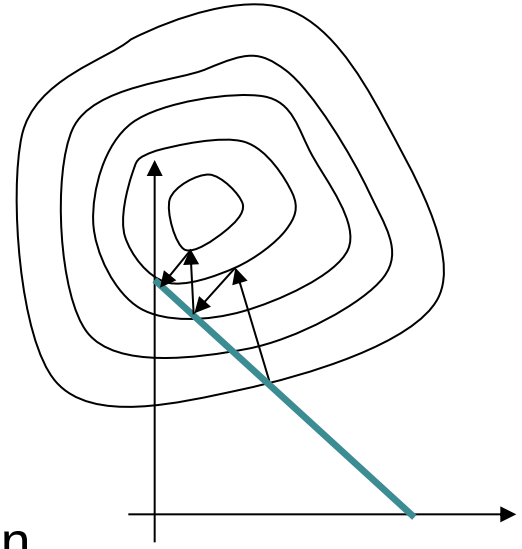
Iterationen

Indikatorvektor der roten Region (weiß=1, schwarz=0)

Orthogonale Projektion

- Um Konvergenz zu einem Optimum sicherzustellen, müssen wir orthogonal zur Oberfläche der gültigen Menge projizieren.

- Der Anteil entlang der Oberfläche wird durch diese Projektion nicht mehr beeinflusst.



- Problem: Allgemein ist es sehr schwierig eine orthogonale Projektion auf die gültige Menge zu bestimmen.

- Dazu müssten die aktiven Bedingungen bekannt sein, was wieder zu einem Active-Set-Verfahren führt.

- Projektionsmethoden eignen sich daher nur für Probleme mit einfachen Nebenbedingungen, bei denen die orthogonale Projektion einfach ist.

Duales Problem

- Bei einigen quadratischen Programmen kann eine einfachere Form der Nebenbedingungen durch Betrachtung des dualen Problems erreicht werden.

  Quadratisches Programm:

$$\min_x \frac{1}{2} x^\top Q x + c^\top x, \quad Ax \geq b$$

  Duales quadratisches Programm:

$$\max_\lambda -\frac{1}{2}(A^\top \lambda - c)^\top Q^{-1}(A^\top \lambda - c) + b^\top \lambda, \quad \lambda \geq 0$$

- Das duale Problem hat sehr einfache Nebenbedingungen, auf die man leicht projizieren kann.

- Sehr effiziente Lösungsstrategie, wenn Q einfach zu invertieren ist (siehe Beispiel der Support Vector Machine wo Q die Einheitsmatrix ist)

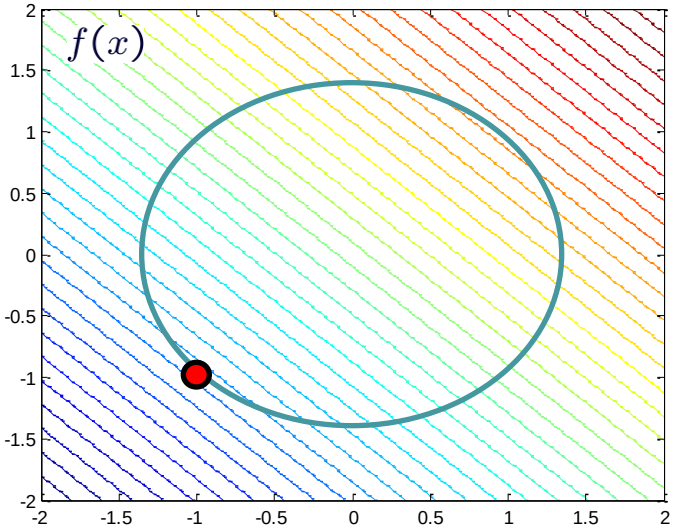
Strafverfahren

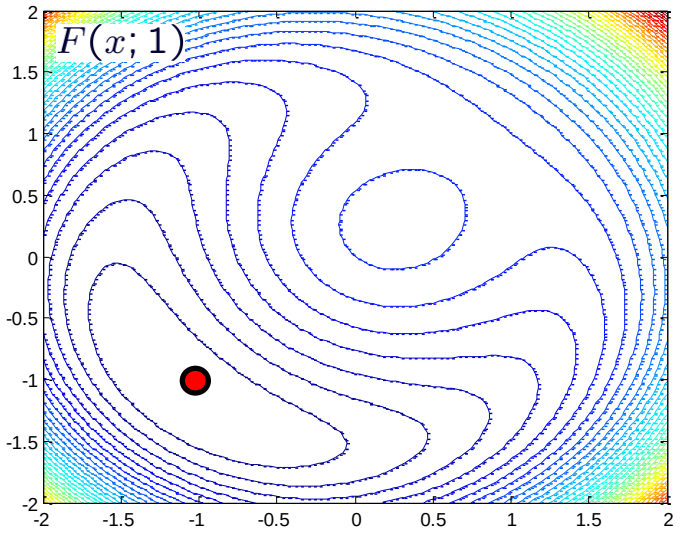
- Idee: Modifiziere die Zielfunktion f so, dass Abweichungen von den Nebenbedingungen bestraft werden:

$$F(x;\mu) = f(x) + \frac{\mu}{2}\sum_i c_i^2(x)$$

- Dies ist nicht zu verwechseln mit der Lagrange-Funktion, die notwendige Bedingungen für ein Optimum definiert, aber selbst nicht minimiert wird.

- Im Optimum von F werden alle Nebenbedingungen eingehalten sobald μ groß genug gewählt wird.

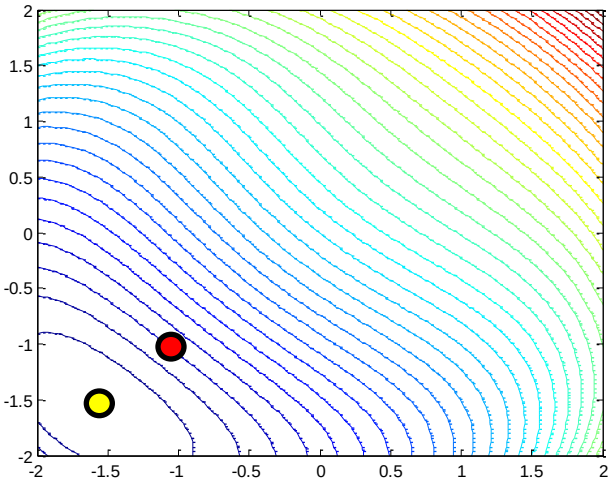




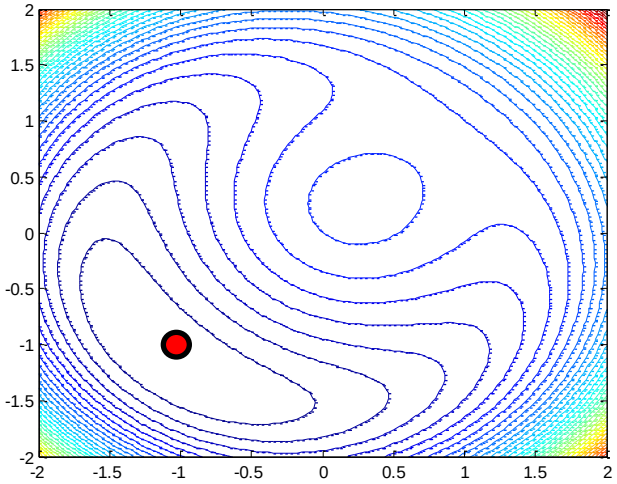
Wahl des Strafparameters

- Ist der Parameter μ zu klein, erfüllt das Optimum von F nicht die Nebenbedingungen

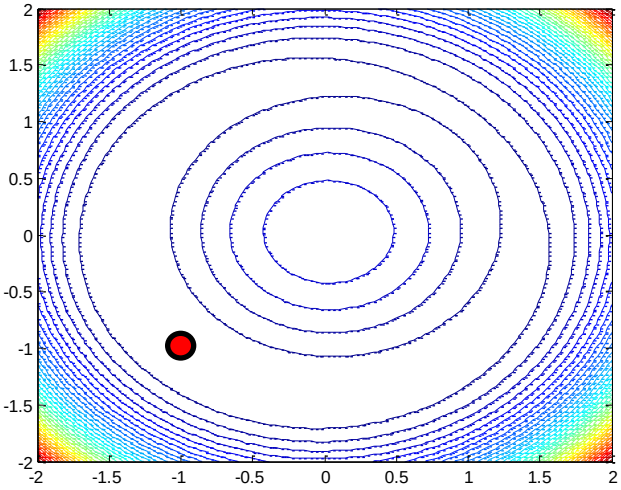
- Mit größerem Einfluss des Strafterms steigt jedoch die Konditionszahl der Hesse-Matrix → numerische Probleme



$\mu = 0.1$



$\mu = 1$



$\mu = 10$

- Man löst daher eine Sequenz von Minimierungsproblemen mit größer werdenden μ

Ungleichheitsbedingungen

- Auch bei Ungleichheitsbedingungen

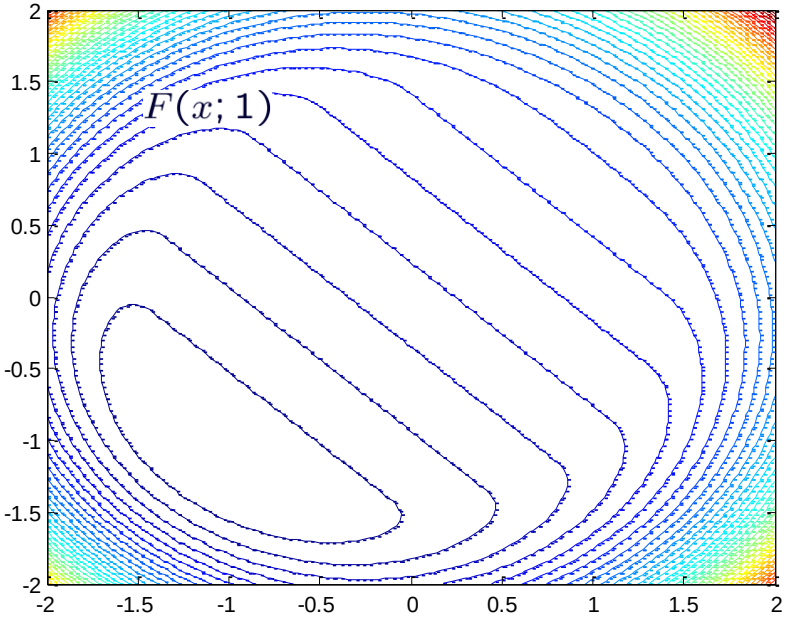
  $c_i \geq 0$

  können Strafterme eingesetzt werden:

$$F(x;\mu) = f(x) + \frac{\mu}{2} \sum_i (\max(-c_i(x), 0))^2$$

- Hier tritt ein weiteres Problem auf:
  Die Funktion ist nur noch einmal differenzierbar.

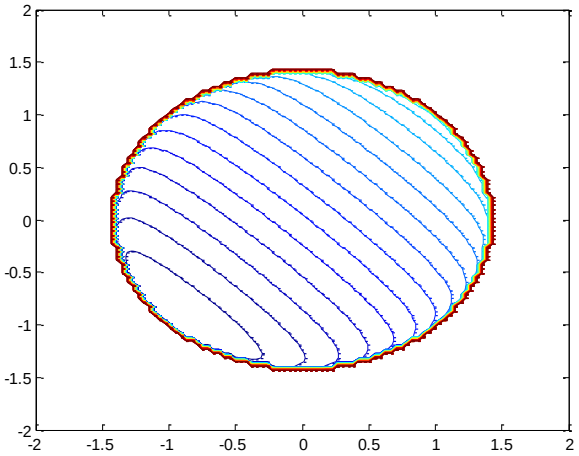
- Problem für das Newton-Verfahren, das wiederum mit der ungleichmäßigen Skalierung am besten umgehen könnte



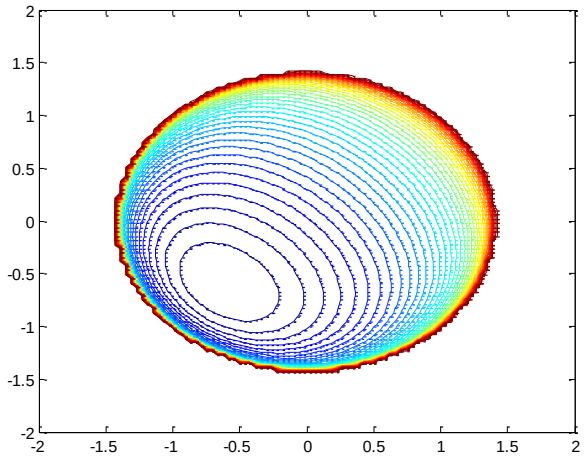
Log-Barrier-Verfahren

- Eine sehr ähnliche Motivation liegt den Log-Barrier-Verfahren zugrunde

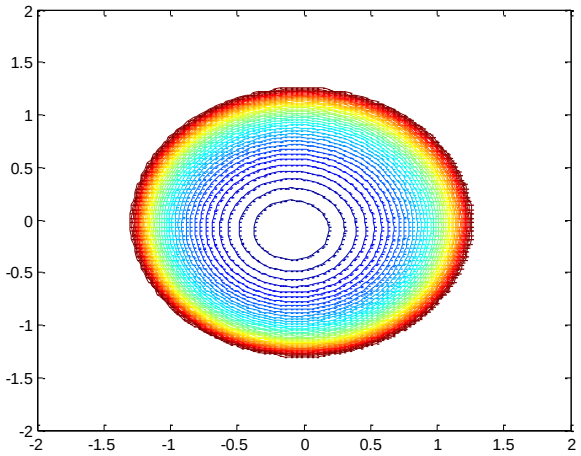
$$F(x;\mu) = f(x) - \mu \sum_i \log c_i(x)$$



$\mu = 0.1$



$\mu = 1$

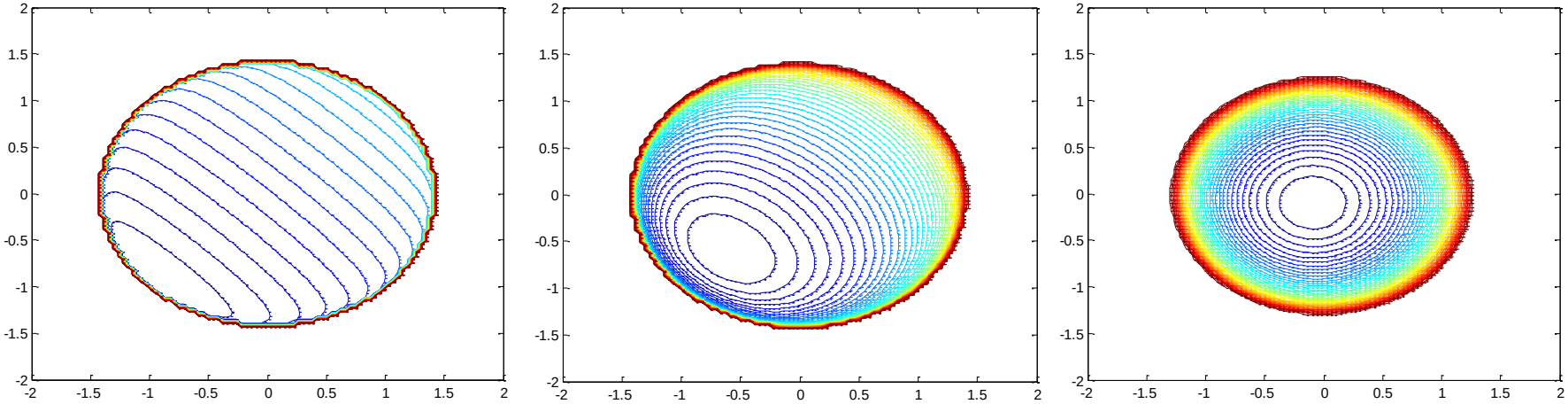


$\mu = 10$

- Da $\log(c) \to -\infty$ für $c \to 0$, ist für alle Minima von F mit $\mu > 0$ sichergestellt, dass die Nebenbedingungen eingehalten werden.

- Man verwendet diesmal eine Sequenz von kleiner werdenden μ

Innere-Punkt-Methode



$\mu = 0.1$ $\mu = 1$ $\mu = 10$

- Die durch den Logarithmus aufgebaute Barriere verhindert, dass wir uns beim Optimieren dem Rand der gültigen Menge zu sehr nähern.

  → **Innere-Punkt-Methode**

- Dies steht im Gegensatz zur Strategie der Active-Set-Verfahren, die vermehrt am Rand der gültigen Menge entlanglaufen (z.B. Simplex-Verfahren, SQP)

Nachteile des Log-Barrier-Verfahrens

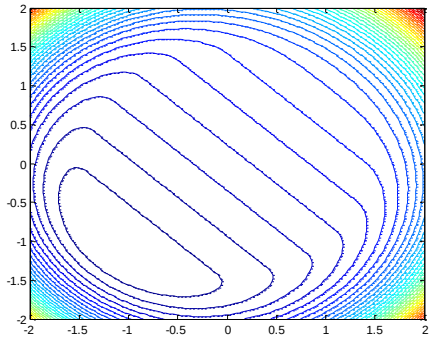
- Wie der quadratische Strafterm, führt auch dieser Strafterm zu schlecht konditionierten Hesse-Matrizen
  → Newton-Verfahren empfohlen

- Die Funktion F lässt sich schlecht quadratisch approximieren. Darauf basiert aber das Newton-Verfahren.

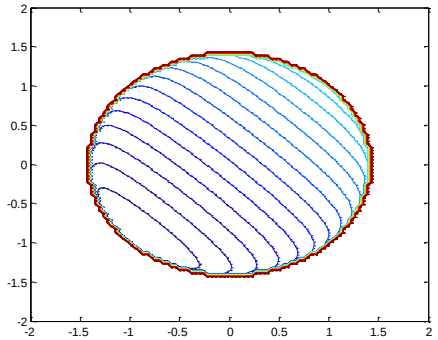
  → Meist niedrige Konvergenzraten

- Gleichheitsbedingungen bedürfen eines quadratischen Strafterms, mit den entsprechenden Nachteilen.

- Innere-Punkt-Methoden waren daher lange Zeit unbeliebt bis in den 80/90er Jahren Primal-Dual-Verfahren (zunächst für lineare Programme) entwickelt wurden.



Quadratische Strafe



Log-Barrier

Primal-Dual Innere-Punkt-Verfahren

- Primal-Dual-Verfahren lösen das (leicht modifizierte) KKT-System, welches die primalen und dualen Variablen enthält.

- Betrachten wir zunächst lineare Programme

$$\min_x c^\top x, \quad Ax = b, x \geq 0$$

- Die KKT-Bedingungen:

$$\begin{aligned} A^\top \lambda + s &= c \\ Ax &= b \\ x_i s_i &= 0, \ \forall i \\ x, s &\geq 0 \end{aligned}$$

- Wir möchten das Gleichungssystem unter Einhaltung von $x, s \geq 0$ lösen.

- Hierfür können wir die Gleichungen auch als Residuum eines Least-Squares-Problems formulieren (siehe Vorlesung 3)

Iteratives Lösen des Least-Squares-Problems

- Wir haben das Residuum

$$r(x, \lambda, s) = \begin{pmatrix} A^\top \lambda + s - c \\ Ax - b \\ XSe \end{pmatrix}$$

mit Diagonalmatrizen X und S mit Einträgen x_i bzw. s_i Und $e = (1, ..., 1)^\top$

- Linearisierung für den Gauss-Newton-Schritt

$$\underbrace{r(x^{k+1}, \lambda^{k+1}, s^{k+1})}_{=0} = r(x^k, \lambda^k, s^k) + J(x^k, \lambda^k, s^k)(\Delta x, \Delta\lambda, \Delta s)^\top$$

Jacobi-Matrix

Schritt

- Wir erhalten den Schritt durch Lösen von

$$\begin{array}{r} \\ \nabla(A^\top \lambda + s + c) \\ \nabla(Ax - b) \\ \nabla XSe \end{array} \begin{array}{c} \begin{array}{ccc} \nabla x & \nabla \lambda & \nabla s \end{array} \\ \begin{pmatrix} 0 & A^\top & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{pmatrix} \end{array} \begin{pmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{pmatrix} = \begin{pmatrix} c - A^\top \lambda^k - s^k \\ b - Ax^k \\ -X^k S^k e \end{pmatrix}$$

Jacobi-Matrix

Schritt

Einhalten der Nebenbedingungen

- Der Newton-Schritt kann die Bedingungen $x, s \geq 0$ verletzen. Um diese einzuhalten, müssten wir den Schritt entsprechend verkürzen.

  Führt meist zu sehr kleinen Schritten entlang des Rands.

- Idee der Inneren-Punkt-Methode: der Rand der gültigen Menge ist tabu!

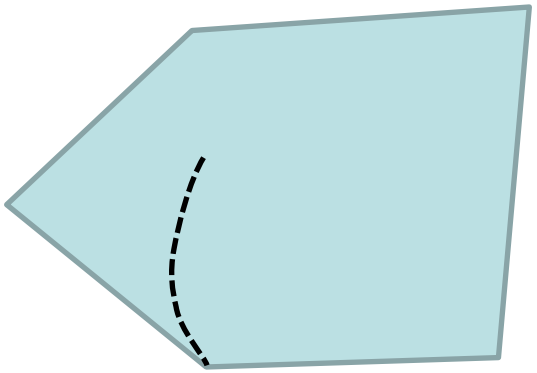
- Statt $XSe = 0$ zu fordern (damit landen wir direkt auf dem Rand), fordern wir $XSe = \mu e$

  Dabei entspricht μ dem **Dualitätsmaß**

  $$\mu = \frac{1}{n} x^\top s$$

  also der aktuellen durchschnittlichen Abweichung von der Komplementarität.

  Verfahren erreicht den Rand erst im Optimum



Primale Darstellung des Pfads (ohne duale Variablen λ, s)

Nichtlineare Programme

- Dieses Verfahren lässt sich für nichtlineare Programme verallgemeinern

$$\min_{x,z} f(x) \qquad g_E(x) = 0 \qquad g_I(x) - z = 0,\ z \geq 0$$

(Schlupfvariablen z zur Darstellung der Ungleichheitsbedingungen)

- KKT-Bedingungen

$$\nabla f(x) - A_E^\top(x)\lambda - A_I^\top(x)s = 0 \qquad \text{Gradient der Lagrange-Funktion}$$

$$ZSe = 0 \qquad \text{Komplementarität}$$

$$g_E(x) = 0$$

$$g_I(x) - z = 0$$

$$z, s \geq 0$$

- Auch hier können wir $ZSe = 0$ zu $ZSe = \mu e$ ändern und die entsprechenden Newton- oder Gauss-Newton-Schritte berechnen.

Allgemeine Schwierigkeiten in der kontinuierlichen Optimierung

- Numerische Instabilität
  hohe Konditionszahlen, (fast) singuläre Matrizen

- Größe der Probleme
  viele Variablen, viele Nebenbedingungen
  → Speicher- und Rechenzeitbeschränkungen

- Lineare/quadratische Approximation ist unpassend
  → ineffiziente Schritte

- Fehlende Glattheit
  Gradient/Hesse-Matrix kann nicht berechnet werden
  → Subgradienten-Verfahren

- Nicht-konvexe Funktionen
  → lokale Minima

Zusammenfassung

- Die Active-Set-Verfahren können auf allgemeine nichtlineare Programme erweitert werden, indem eine Sequenz quadratischer Programme gelöst wird.

- Projektionsmethoden eigenen sich, wenn die orthogonale Projektion einfach vorzunehmen ist.

- Strafmethoden sind intuitiv einleuchtend aber schwer zu kontrollieren und numerisch instabil.

- Primal-Dual Innere-Punkt-Methoden lösen sequentiell das KKT-System, das in jedem Schritt so modifiziert wird, dass der Rand der gültigen Menge erst im Optimum erreicht wird.

Übungsaufgabe

1. Diesmal möchten wir den kleinen Hund in `puppy.png` vom Hintergrund segmentieren. Das Optimierungsproblem ist sehr ähnlich zu dem, das wir bei der Bildverbesserung bekommen haben:

$$f(x) = \sum_{i,j} \left(\left((y_{ij} - 128)^2 - (y_{ij} - 255)^2 \right) x_{ij} + \frac{1}{2} \sqrt{(x_{ij} - x_{i+1j})^2 + (x_{ij} - x_{ij+1})^2 + 1} \right)$$

   allerdings haben wir noch Nebenbedingungen

$$x_{i,j} \in [0, 1], \ \forall i, j$$

   Berechnen Sie den Gradienten.

2. Implementieren Sie wieder einen Gradientenabstieg. Dazu müssen Sie Ihre Implementierung aus der zweiten Übung nur geringfügig anpassen. Nach jedem Iterationsschritt müssen Sie diesmal jedoch noch auf die gültige Menge zurückprojizieren.

3. Wenden Sie das Verfahren auf `puppy.png` an. Verwenden Sie 0,5 als Startwerte. Sie sollten am Ende für jeden Bildpunkte Werte nahe 0 oder 1 bekommen. Diese stehen für Hintergrund bzw. Vordergrund. Genaugenommen haben Sie damit ein kombinatorisches Problem gelöst, das wir uns in der nächsten Vorlesung noch einmal genauer ansehen werden.