Ausarbeitung

zum Vortrag

Adaptive Selektivitätsschätzung durch Polynom-Regression

Johannes John

im Rahmen des Proseminars

Datenreduktionstechniken

Prof. Dr.-Ing. Gerhard Weikum

Betreuer:
Dipl.Inform. Arnd Christian König

Sommersemester 1999
Universität des Saarlandes

Inhalt

1 Einleitung

Die vorliegende Arbeit befaßt sich mit dem Thema Regression zur Selektivitätsschätzung in Datenbanken.
Ich habe mich bemüht die Grundlagen des Themas, Regression und Abfrage Optimierung, besonders herauszuarbeiten, da die Unterschiede in den Vorkenntnissen der Teilnehmer am Proseminar ziemlich groß waren.
Das erste Kapitel beginnt mit der Vorstellung der Theorie der klassischen Regression. Im zweiten Kapitel wird eine Verwendungsmöglichkeit von Regression vorgestellt, und in den letzten beiden Kapiteln werden, erst theoretisch, dann sehr konkret, Algorithmen vorgestellt, die Regression auf dem Computer umsetzen.

Ich danke vor allem Christian König für seine Unterstützung und Prof. Weikum für seine konstruktive Kritik.

1.1 Schreibweise:

Im wesentlichen gilt in den Formeln folgende Schreibweise:

Große Buchstaben X für Matrizen
Kleine Buchstaben x für Skalare
Unterstrichene Buchstaben $\underline{x}$ für Vektoren
Variablen mit Dach $\hat{x}$ für Schätzwerte.

2 Statistische Grundlagen der Regression

2.1 Das klassische lineare Regressionsmodell

Die Regression ist ein statistisches Werkzeug, bei dem man die Korrelation zweier oder mehrerer Merkmale funktional darstellt.

Seien $\underline{x} = (x_1,...,x_n)$ und y die Merkmale.

Nach einer Stichprobe $\underline{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_T \end{bmatrix}$ von y und $X = \begin{bmatrix} x_{10} & x_{11} & \cdots & x_{1n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{T0} & x_{T1} & \cdots & x_{Tn} \end{bmatrix}$ von $\underline{x}$ vom Umfang T, nimmt man an, daß y von $(x_1,...,x_n)$ linear abhängt, in der Form: $\underline{y} = X\underline{c} + \underline{u}$.

$\underline{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}$ ist dabei die Steigung einer Geraden durch die Punkte $(y_t, x_{t,1},...,x_{t,n})$

$\underline{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_T \end{bmatrix}$ sind die Störungen, die bei der Messung von $(y_0,..., y_T)$ aufgetreten sind .

Man nennt

$$\underline{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_T \end{bmatrix}$$ Regressant, abhängige Zufallsariable

$$X = \begin{bmatrix} x_{10} & x_{11} & \cdots & x_{1n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{T0} & x_{T1} & \cdots & x_{Tn} \end{bmatrix}$$ Regressoren, erklärende Veriablen

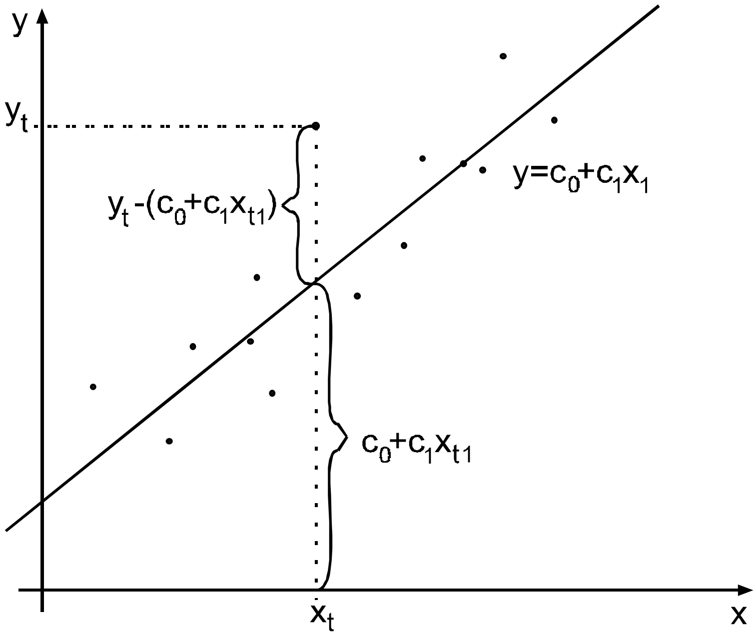
$$\underline{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}$$ Regressionskoeffizienten

$$\underline{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_T \end{bmatrix}$$ Stör- oder Fehlervariable

$$\underline{y} = X\underline{c} + \underline{u}$$ <u>multiples lineares Regressionsmodell.</u>

Beispiel: das einfache lineare Regressionsmodell

Im Fall n=1 spricht man vom einfachen Modell.



Die Punkte in dieser Graphik (y_t, $x_{t,1}$) enstsprechen unseren Meßwerten[1]

c_1 entspricht der Steigung einer Geraden durch die Punkte, und c_0 ist der Y-Achsen-Abschnitt.

Die Methode der kleinsten Quadrate

Gegeben seien die Beobachtungswerte y_t und $x_{t,i}$. und die Schätzfunktion $\underline{\hat{y}} = X\underline{\hat{c}}$.

Die „ideale" Gerade durch die Punkte $(y_t, x_{t,0}, x_{t,1}, \dots, x_{t,N})$ ist diejenige, bei der die Differenzen aus unseren Schätzwerten $\hat{y}_t$ und unseren gemessenen Werten y_t insgesamt am geringsten ausfällt (s. Abb. 1).

Gesucht: Die idealen Regressionskoeffizienten $\underline{\hat{c}} = (\hat{c}_1, \dots, \hat{c}_N)$

Lösung: Die Koeffizienten ergeben sich durch Lösen des Problems:

$$\min_{\hat{c}_i}\left[\sum_{t=1}^{T}\left(y_t - \sum_{i=0}^{n}\hat{c}_i x_{t,i}\right)^2\right]$$

Man betrachtet hierbei die Quadratische Differenz, dadurch heben sich positive und negative Differenzen nicht gegenseitig auf.

$\underline{\hat{c}}$ heißt : Kleinst-Quadrat-Schätzung (ordinary least square estimator).

---

[1] $x_{t,0}$ wird immer als 1 angenommen, das sog. inhomogene Modell

Zusammenfassung

Vorteile von Regression:

- Die Regression eignet sich zur Veranschaulichung von Korrelation, vor allem bei ein bis zwei erklärenden Variablen.
- Es können Vorraussagen über nicht erfasste Realisierungen von y, z.B. Werte in der Zukunft oder Vergangenheit gemacht werden.
- Die Schätzwerte $\underline{\hat{y}}$ sind zum Teil genauer als die Meßwerte $\underline{y}$, weil die Regressionsgerade Meßfehler ausgleicht.

Schwächen:

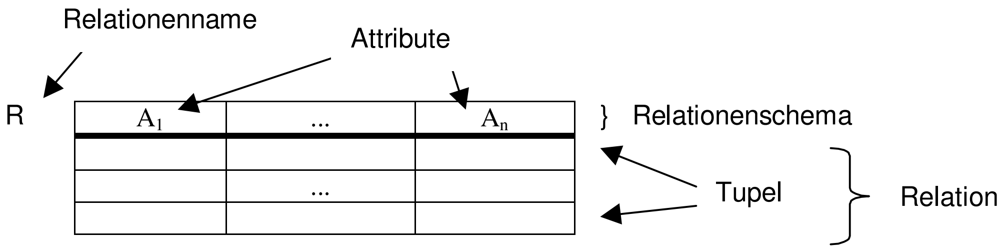
- Wenn das Regressionsmodell schlecht gewählt ist, also kein tatsächlicher linearer Zusammenhang zwischen den Merkmalen besteht, kann man mit der Schätzfunktion völlig falsche Werte erhalten, die eventuell sogar außerhalb des Wertebereichs von y liegen, z.B. negative Häufigkeit oder negatives Gewicht.

In den folgenden Kapiteln wird gezeigt, wie man Regression zur Schätzung von Häufigkeitsverteilungen in Datenbanken benutzen kann.
Es fehlt dabei ein wichtiger Aspekt der statistischen Regression: Es gibt in Datenbanken keine Meßfehler bei der Bestimmung von Selektivität.
Bei den nachfolgenden Algorithmen steht die statistische Genauigkeit jedoch im Hintergrund. Regression wird vor allem benutzt, weil sie sehr schnell berechenbar ist.

3 Regression und Datenbanksysteme

3.1 Das relationale Datenbankmodell

Eine relationale Datenbank spiegelt seine Daten in Relationen wieder, die man sich als Tabellen vorstellen kann



Um Daten aus Relationen auszuwählen, gibt es drei grundlegende Funktionen:

<u>Projektion</u>: π

Bei der Projektion werden ein oder mehrere Attribute einer Relation ausgewählt.

z.B. $\pi_{Vorname,Ort}(Personen)$

Personen

PANr	Vorname	Name	Ort	Straße
4711	Andreas	Heuer	DBR	BHS
5588	Gunter	Saake	MD	STS
6834	Michael	Korn	MD	BS
8832	Tamara	Jagellovsk	BS	GS
9912	Antje	Hellhof	HRO	AES
9999	Christa	Loeser	HD	TS

$\Downarrow$

Vorname	Ort
Andreas	DBR
Gunter	MD
Michael	MD
Tamara	BS
Antje	HRO
Christa	HD

2. Selektion: σ

Bei der Selektion werden bestimmte Tupel anhand einer Bedingung aus einer Relation ausgewählt.

z.B. $\sigma_{5000 \le PANr \le 9000}(Personen)$

Personen

PANr	Vorname	Name	Ort	Straße
4711	Andreas	Heuer	DBR	BHS
5588	Gunter	Saake	MD	STS
6834	Michael	Korn	MD	BS
8832	Tamara	Jagellovsk	BS	GS
9912	Antje	Hellhof	HRO	AES
9999	Christa	Loeser	HD	TS

⇩

PANr	Vorname	Name	Ort	Straße
5588	Gunter	Saake	MD	STS
6834	Michael	Korn	MD	BS
8832	Tamara	Jagellovsk	BS	GS

3. Join: $\bowtie$

Éin Join fügt zwei Relationen über ihre gemeinsamen Attribute zusammen.

z.B. $\bowtie_{Personen.PANr=Telefon.PANr}$ oder $Personen \bowtie Telefon$

Personen

PANr	Vorname	Name	Ort	Straße
4711	Andreas	Heuer	DBR	BHS
5588	Gunter	Saake	MD	STS
6834	Michael	Korn	MD	BS
8832	Tamara	Jagellovsk	BS	GS
9912	Antje	Hellhof	HRO	AES
9999	Christa	Loeser	HD	TS

$\bowtie$

Telefon

PANr	Tel
4711	12230
4711	3401
4711	3427
5588	345677
5588	3800
9999	400177

⇩

PANr	Vorname	Name	Ort	Straße	Tel
4711	Andreas	Heuer	DBR	BHS	12230
4711	Andreas	Heuer	DBR	BHS	3401
4711	Andreas	Heuer	DBR	BHS	3427
5588	Gunter	Saake	MD	STS	345677
5588	Gunter	Saake	MD	STS	3800
9999	Christa	Loeser	HD	TS	400177

3.2 Abfrage Optimierung

Ein Hauptproblem bei relationalen Datenbanken ist, daß man sehr häufig Joins braucht, um zusammenhängende Daten darzustellen.

Joins sind leider sehr teuer; im schlimmsten Fall, d.h. wenn die Relationen keine gemeinsamen Attribute haben, erhält man als Resultat das Kreuzprodukt aus beiden Relationen.

Man muß deshalb darauf achten, daß die zu bearbeitenden Relationen möglichst klein sind und daß beim Join keine Tabellen ohne gemeinsame Attribute verknüpft werden.

Beispiel:

Angenommen wir wollen Adresse und Telefonnummer von Christa Loeser bestimmen, dann gibt es zwei Möglichkeiten:

$\sigma_{PANr=9999}(\triangleright\triangleleft_{Personen.PANr=Telefon.PANr})$ und

$\triangleright\triangleleft_{\sigma_{PANr=9999}(Personen)).PANr=Telefon.PANr}$

Das obere Beispiele ist offensichtlich: es ist immer besser eine Selektion vor einem Join durchzuführen, da dadurch die Daten nur verringert werden können.

Wenn man allerdings die Selektivität von seinen Daten kennt, kann man noch eine Reihe weiterer Optimierungen vornehmen:

- Die Reihenfolge von Joins durch die Größe der Relationen festlegen
- Die Reihenfolge von Selektionen durch die Selektivität der Attribute festgelegen

Es ist dabei wichtig, das die Berechnung der Selektivität so schnell geht, daß sich diese Optimierungen lohnen.

Mit Regression kann man die Selektivität sehr elegant schätzen.

4 Selektivitätsschätzung durch Polynom-Regression

4.1 Schätzen einfacher Selektionen

Speziell für die Abfrage Optimierung braucht man sehr schnell Infomationen über die Häufigkeitsverteilung bestimmter Attribute.Dabei spielt die Genauigkeit der Berechnung keine so große Rolle; man ist nur an der Größenordnung der Selektivität interessiert.

Die Idee des folgenden Algorithmus' ist, die Verteilungsfunktion f(x) eines Attributes X mit einem Polynom[2] näherungsweise zu bestimmen [1].

Benutzt werden dabei die Methoden der Regression, wie im Kapitel 1 beschrieben. Die Selektivität f ist der Regressant, und die Ausprägungen des Attributs sind die Regressoren.

---

[2] streng gesehen handelt es sich um gebrochenrationale Funktionen

Als Schätzfunktion g(x) wählt man ein Polynom.

$g(x) = \sum_{i=-n_2}^{n_1} c_i x^i$	g(x)	die geschätzte Selektivität
	x	Regressor
	c_i	Regressionskoeffizienten
	n_1,n_2 :	größter, kleinster Exponent

Tests ergaben sinnvolle Grenzwerte für n_1 und n_2: n_1 <=6, n_2 <=4

Polynome höheren Grades neigen zum Oszillieren; sie verkomplizieren die Berechnungen aber erhöhen die Genauigkeit der Schätzung nicht.

Die Regressionskoeffizientsn werden wie im linearen Modell durch KQ-Schätzung bestimmt.

Ausgehend von einer Menge Werte x_i eines Attributs X, deren Selektivität $f(x_i)$ wir schon kennen, suchen wir die Lösung $\underline{\hat{c}} = \{\hat{c}_{-n2}, \dots, \hat{c}_{n1}\}$ des Problems:

$$\min_{\underline{\hat{c}}} \sum (f(x_i) - \sum \hat{c}_j x_i^j)^2$$

Diese Gleichung kann man mit linearer Algebra umformen.
Die optimalen Koeffizienten ergeben sich dann aus:

$$\underline{\hat{c}} = (X^T X)^{-1} X^T \underline{f}$$

mit

$$X = \begin{bmatrix} x_1^{-n2} & \dots & x_1^{-1} & 1 & x_1 & \dots & x_1^{n1} \\ x_2^{-n2} & \dots & x_2^{-1} & 1 & x_2 & \dots & x_2^{n1} \\ \dots & \dots & \dots & 1 & \dots & \dots & \dots \\ x_m^{-n2} & \dots & x_m^{-1} & 1 & x_m & \dots & x_m^{n1} \end{bmatrix} \text{ und } \underline{f} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \dots \\ f(x_m) \end{bmatrix}$$

Die Selektivität einer einfachen Selektion, wie $x_{l1} \le R.X \le x_{l2}$, kann folgendermaßen abgeschätzt werden:

Die Wahrscheinlichkeit p, mit der ein Tupel die Bedingung erfüllt:

$$p(x_{l1} \le R.X \le x_{l2}) = \frac{\int_{xl1}^{xl2} g(x)dx}{\int_{x\min}^{x\max} g(x)dx}$$

$x_{\max}$, $x_{\min}$: Wertebereich von X

Die Anzahl der Tupel zwischen_ $x_{l1} \le R.X \le x_{l2}$:

$$\sum_{i=l1}^{l2} f(x_i) \approx p \cdot \sum_{i=1}^{m} f(x_i)$$

Diesen Berechnung liegt eine Annahme über die vorliegende Datenverteilung zugrunde: wir nehmen an, daß der Abstand $\Delta x = x_{i+1} - x_i$ zweier beliebiger benachbarter Ausprägungen von X in der Datenbank überall ungefähr gleich ist (Die sogenannte Equi-Spread Bedingung, die auch bei der Berechnung von Histogrammen gebraucht wird).

Die obigen Formeln ergeben sich dann durch den Zusammenhang:

$$\frac{\sum_{i=l1}^{l2} f(x_i)}{\sum_{i=1}^{m} f(x_i)} \approx \frac{\sum_{i=l1}^{l2} g(x_i)\cdot\Delta x}{\sum_{i=1}^{m} g(x_i)\cdot\Delta x} \approx \frac{\int_{xl1}^{xl2} g(x)dx}{\int_{x\min}^{x\max} g(x)dx}$$

4.2 Schätzen komplexer Selektionen

Seien R.X und R.Z zwei Attribute einer Relation.

Eine komplexe Selektion der Form $(x_{l1} \leq R.X \leq x_{l2}) \wedge (z_{l1} \leq R.Z \leq z_{l2})$ kann analog mit der Schätzfunktion:

$$g(x,z) = \sum_{i=-n2}^{n1} \sum_{j=-n2}^{n1} c_{ij} x^i z^j$$

bestimmt werden.

Die Koeffizienten werden bestimmt durch:

$$\min_{\tilde{c}} \left(\sum_{l=1}^{m} \sum_{i=-n2}^{n1} \sum_{j=-n2}^{n1} (\tilde{c}_{ij} x_l^i z_l^j - f(x_l, z_l))^2 \right)$$

Die Schätzfunktion liegt dreidimensional im Raum. Um die Selektivität einer Selektion zu berechnen, betrachten wir nicht mehr die Fläche unter dem Graphen, sonder den Raum:

Die Wahrscheinlichkeit, mit der ein Tupel die Selektion erfüllt, beträgt:

$$p(x_{l1} \leq x \leq x_{l2} \wedge z_{k1} \leq z \leq z_{k2}) = \frac{\int_{xl1}^{xl2} \int_{zk1}^{zk2} g(x,z)dxdz}{\int_{x\min}^{x\max} \int_{z\min}^{z\max} g(x,z)dxdz}$$

4.3 Schätzen von Joins

Man berechnet die Selektivität eines Joins der Form Sei R.X = S.Z mit der gleichen Schätzfunktion wie bei der komplexen Selektion:

$$g(x,z) = \sum_{i=-n2}^{n1} \sum_{j=-n2}^{n1} c_{ij} x^i z^j$$

Man benutzt eine neue Verteilungsfunktion $ff(x,z) = f(x)\cdot f(z)$, die Join Verteilungsfunktion.

$ff(x, z)$ ist das Produkt der Anzahl der Tupel aus R mit X=x und der Anzahl der Tupel aus S mit Z=z.

Die optimalen Koeffizienten berechnen sich aus:

$$\min_{\tilde{c}} \left(\sum_{l=1}^{mx} \sum_{k=1}^{my} \sum_{i=-n2}^{n1} \sum_{j=-n2}^{n1} (\tilde{c}_{ij} x_l^i z_k^j - ff(x_l, z_k))^2 \right)$$
$$= \min_{\tilde{c}} \left(\sum_{l=1}^{mx} \sum_{k=1}^{my} \sum_{i=-n2}^{n1} \sum_{j=-n2}^{n1} (\tilde{c}_{ij} x_l^i z_k^j - f(x_l) \cdot f(z_k))^2 \right)$$

m_x : Anzahl der bekannten Häufigkeiten von X

m_z : Anzahl der bekannten Häufigkeiten von Z

Die Schätzfunktion schätzt damit die Häufigkeit aller möglicher Kombinationen von Tupeln aus dem Kreuzprodukt von R und S.

Die Verteilung eines bestimmten Joins wird durch die Funktion

$g_{join}(x) = g(x, x) = \sum_{i=-n2}^{n1} \sum_{j=-n2}^{n2} \hat{c}_{ij} x^{i+j}$ brechnet.

Die Wahrscheinlichkeit p des Joins R.X=S.Z berechnet sich durch:

$$p(R.X = S.Z) = \frac{\int_{d1}^{d2} g_{join}(x) dx}{\int_{x\min}^{x\max} \int_{z\min}^{z\max} g(x, z) dxdz}$$

mit $d_1 = \max(x_{\min}, z_{\min})$ und $d_2 = \min(x_{\max}, z_{\max})$

Anmerkung: Da der Grad der g-Funktion sehr hoch ist, und die Selektivität bei Joins gewöhnlich sehr stark schwankt kommt man mit dieser Methode leicht zu Rundungsfehlern.
Deshalb berechnet man die Schätzfunktion $g_{\log}$ mit dem Logarithmus der Funktion ff, anstatt mit ff selbst, und $g(x, z) = \exp(g_{\log}(x, z))$.

5 Ein praktischer Ansatz: ASE

Adaptive Selectivity Estimation [2] ist eine einfache Methode zur Abschätzung von Selektionen, die die Vorteile der Regression, nämlich schnelle Abschätzung der Bezugsgrößen ohne rechnerischen Overhead, noch weiter hervorhebt.

5.1 RLSE (Recursive Least Square Error)

Recursive Least Square Error ist der Kern des Algorithmus‘. Es handelt sich dabei um eine normale KQ-Schätzfunktion, die die Regressionskoeffizienten rekursiv an neue Daten anpasst.

Bei jeder Selektion $\sigma_{l \le A \le h}$ merkt man sich 3 Parameter: $\xi(l, h, f)$, das „Query Feedback“. Dabei ist f = f($\sigma_{l \le A \le h}$) die Selektivität der Selektion σ.

Die Regressionskoeffizienten werden mit einer rekursiven Methode an das neue Feedback angepasst, und man erhält eine Schätzfunktion, die sich immer mehr an die tatsächliche Datenverteilung, und speziell an häufig abgefragte Datenbereiche, anpaßt.

Man erhält eine sinnvolle Abschätzung quasi „nebenbei“, ohne Voranalysen.

Um den Algorithmus allgemein zu halten, legt man sich bei der Schätzfunktion nicht direkt auf ein Polynom fest, sondern verwendet beliebige Modell-Funktionen ϕ_i:

$$g(x) = \sum_{i=0}^{n} c_i \phi_i(x)$$

Der Schätzwert für die Selektivität einer Selektion der Form $l \le R.X \le h$ ist dann:

$$\hat{f} = \int_{l}^{h+1} g(x)dx = G(h+1) - G(l) = \sum_{j=0}^{n} c_j [\Phi_j(h+1) - \Phi_j(l)]$$

Die Großbuchstaben G und Φ stehen hierbei für die Stammfunktionen von g und ϕ.

Für die Koeffizienten $\underline{\hat{c}}$ ergibt sich, nachdem $m \ge n$ query feedbacks gesammelt worden sind:

$$\min_{\underline{\hat{c}}} \left[\sum_{i=1}^{m} (\hat{f}_i - f_i)^2 \right]$$
$$= \min_{\underline{\hat{c}}} \left[\sum_{i=1}^{m} (\sum_{j=0}^{n} \hat{c}_j [\Phi_j(h_i + 1) - \Phi_j(l_i)] - f_i)^2 \right]$$

m: Anzahl der Feedbacks
n: Anzahl der Koeffizienten

Die gleiche Formel in linearer Algebra:

$$\min_{\underline{\hat{c}}} \left(X \cdot \underline{\hat{c}} - \underline{f} \right)$$

$$\underline{f} = \begin{bmatrix} f_1 \\ f_2 \\ \cdots \\ f_m \end{bmatrix} \quad \underline{\hat{c}} = \begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \cdots \\ \hat{c}_n \end{bmatrix} \quad X = \begin{bmatrix} \Phi_0(h_1+1) - \Phi_0(l_1) & \cdots & \Phi_n(h_1+1) - \Phi_n(l_1) \\ \Phi_0(h_2+1) - \Phi_0(l_2) & \cdots & \Phi_n(h_2+1) - \Phi_n(l_2) \\ \cdots & \cdots & \cdots \\ \Phi_0(h_m+1) - \Phi_0(l_m) & \cdots & \Phi_n(h_m+1) - \Phi_n(l_m) \end{bmatrix}$$

Auflösen nach $\underline{\hat{c}}$ ergibt:

$$\underline{\hat{c}} = (X^T X)^{-1} X^T \underline{f}$$

Obwohl die Modellfunktionen ϕ_i nicht genau spezifiziert sind, ergeben sich keinerlei Komplikationen bei der Berechnung der Koeffizienten.

Diese Berechnungsvorschrift hat den Nachteil, das X eine $m \times n$ Matrix ist, d.h. bei jedem neuen Query Feedback muß der Computer eine größere Berechnung durchführen. Außerdem ist die Matrixinversion $(X^T X)^{-1}$ eine teure Operation.

Man kann aber mit einigen kleinen Umformungen die Funktion in konstanter Zeit berechnen.

Im wesentlichen faßt man dabei $\left(X^T X\right)^{-1}$ zu einer neuen $n \times n$ Matrix G zusammen. G und $\underline{\hat{c}}$ werden rekursiv berechnet, d.h. bei jedem neuen Feedback $\xi_m(l_m, h_m, f_m)$ werden die alten Werte von G und $\underline{\hat{c}}$, G_{m-1} und $\underline{\hat{c}}_{m-1}$, mit dem neuen Feedback modifiziert:

$$\underline{\hat{c}}_m = \underline{\hat{c}}_{m-1} - G_m X_m^T (X_m \underline{\hat{c}}_{m-1} - f_m)$$
$$G_i = G_{m-1} - G_{m-1} X_m^T (1 + X_m G_{m-1} X_m^T)^{-1} X_m G_{m-1}$$

(X_i ist hier ausnahmsweise die i-te Zeile von X und nicht “X Index i”, im Gegensatz zu G_i)

Dieser Algorithmus hat eine Laufzeit von $O(n^2)$. n ist die Anzahl der Regressionskoeffizienten. n ist konstant, und bei einer gebrochenrationalen Funktion ist n < 10 (s. Kapitel 4). Der Algorithmus läuft also in konstanter Zeit ab.

<u>Initialisierung von $\underline{\hat{c}}_0$ und G_0</u>

Der rekursive Algorithmus RLSE braucht Anfangswerte für G und $\underline{\hat{c}}$. Theoretisch kann man G_0 und $\underline{\hat{c}}_0$ beliebig belegen. Man kann den Anpassungsprozeß der Schätzfunktion aber beschleunigen, indem man eine sinnvolle Startverteilung vorgibt, z.B. die Gleichverteilung.

Die Anfangswerte von RLSE, $\underline{\hat{c}}_0$ und G_0, werden mit künstlichen Feedbacks berechnet:

$$l_i = h_i = x_{\min} + (i-1) \cdot \frac{x_{\max} - x_{\min}}{(n-1)}, \quad f_i = \frac{|R|}{(x_{\max} - x_{\min})}$$

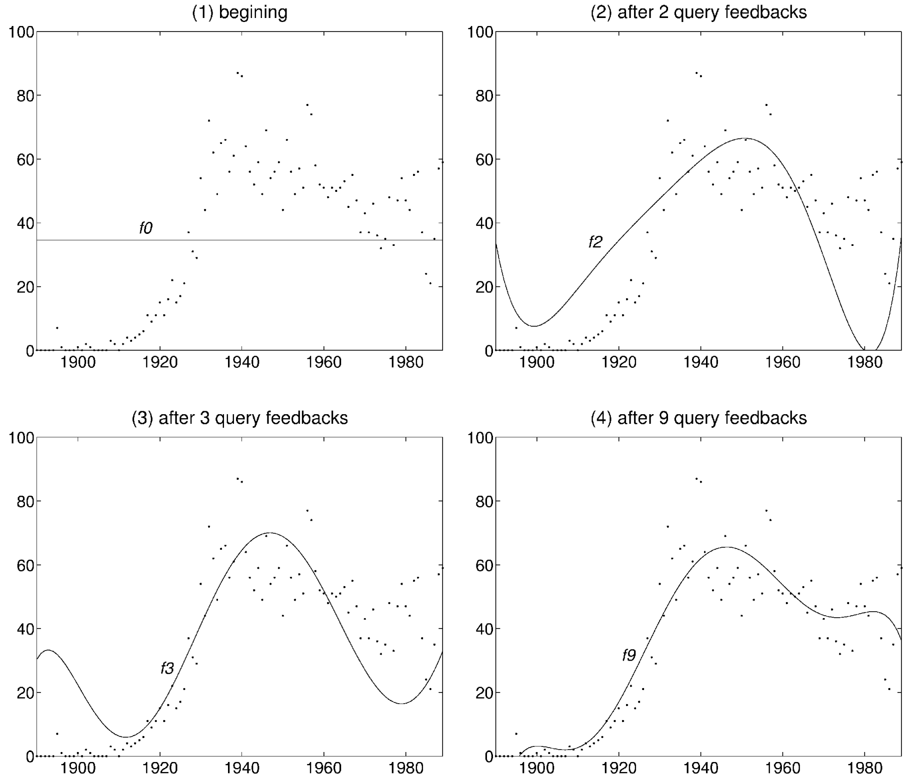
<u>Auswahl der Schätzfunktion ϕ</u>

Wie im letzten Kapitel erörtert, eignet sich für ϕ ein Polynom 6. Grades, also $\phi_i(x) = x^i$.

Die Schätzfunktion hat dann die Form: $g(x) = \sum_0^6 c_i x^i$.

Ein Nachteil des Polynoms als Modell-Funktion ist, das negative Funktionswerte möglich sind. In diesem Fall wird 0 zurückgegeben.

Abb. 2: Anpassung von ASE an eine Datenverteilung



5.2 Anpassung an Datenbankveränderungen

Eine Datenbank unterliegt Veränderungen. Neue Datensätze kommen hinzu, alte werden gelöscht. Damit verlieren die alten Koeffizienten unserer Schätzfunktion an Gültigkeit.

Die Schätzfunktion muß an diese Veräderungen angepasst werden.

Dazu führt man zwei neue Werte ein: Das "Fading Weight" α_i $(\alpha_i \leq 1)$, das das Alter von Koeffizienten im KQ-Schätzer berücksichtigt und das "Importance Weight" β_i, das bestimmte Feedbacks in unserer Schätzung hervorhebt.

Der KQ-Schätzer wird modifiziert: $\min\limits_{\underline{\hat{c}}} \left[\sum_{i=1}^{m} \left((\prod_{j=i+1}^{m} \alpha_j) \cdot \beta_i \cdot (\hat{f}_i - f_i) \right)^2 \right]$

Der RLSE Algorithmus verändert sich dadurch nicht wesentlich:

$$\underline{\hat{c}}_m = \underline{\hat{c}}_{m-1} - \beta_m^2 G_m X_m^T (X_m \underline{\hat{c}}_{m-1} - f_m)$$

$$G_i = (\frac{1}{\alpha_m})^2 G_{m-1} - (\frac{\beta_m}{\alpha_m})^2 G_{m-1} X_m^T (\alpha_m^2 + \beta_m^2 X_m G_{m-1} X_m^T)^{-1} X_m G_{m-1}$$

α_i und β_i können frei gewählt werden.

Das Problem ist, α_i an die Datenfluktuation der Datenbank anzupassen.

Eine sinnvolle Belegung für α_i wäre z.B. : α_i immer gleich 1, außer bei erstem Feedback nach einer Datenbankveränderung, dann $\alpha_i < 1$.

5.3 Zusammenfassung:

ASE erweist sich als ein wirksames Werkzeug zur Abfrage Optimierung,, vor allem wegen seiner herausragenden Geschwindigkeit (konstant in Bezug auf die Anzahl der Feedbacks).

Weitere Vorzüge von ASE:

- ASE ist anpassungsfähig an Datenbankänderungen, sowohl bei inserts wie auch bei deletes
- ASE macht im Voraus keine Annahmen über die vorliegende Datenverteilung
- Der Algorithmus ist durch seinen allgemeinen Aufbau sehr erweiterungsfähig, z.B. durch freie Wählbarkeit von ϕ, G_0, $\hat{\underline{c}}_0$, α_i und β_i

Der größte Nachteil von ASE, und allen in dieser Ausarbeitung vorgestellten Algorithmen zur Schätzung von Selektivität, ist die Beschränkung der Schätzfunktion auf Polynome mit $n \leq 10$ Koeffizienten.
Ein Polynom n-ten Grades, hat höchstens n-1 verschiedene Biegungen. Sehr unregelmäßige Verteilungen mit hoher Varianz lassen sich damit nur schlecht approximieren. Im schlechtesten Fall schätzt der Algorithmus die Gleichverteilung.

Literatur

[1] Wei Sun, Yibei Ling, ...
An Instant an Accurate Size Estimation Method for Joins and Selection in a Retrieval-Intensive Environment
ACM SIGMOD Conference, 1993

[2] Chungmin Melvin Chen, Nick Roussopoulos
Adaptive Selectivity Estimation Using Query Feedback
ACM SIGMOD Conference , 1994

[3] Daniel Barbará, William DuMouchel, ...
The New Jersey Data Reduction Report
IEEE Bulletin of the Technical Committee on Data Engineering, 1997

[4] Andreas Heuer, Gunter Saake
Datenbanken, Konzepte und Sprachen
International Thomson Publishing, 1997

[5] Prof. Dr. Volker Steinmetz
Kompaktskript zu den Vorlesungen Statistik Teil A und B
Universität des Saarlandes, 1998