BagBoosting für Tumor-Klassifizierung mit Gen-Expressions-Daten

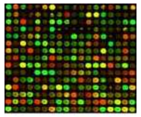
Marcel Dettling

Seminar für Statistik
ETH Zürich

dettling@stat.math.ethz.ch
http://stat.ethz.ch/∼dettling

Microarrays und Gen-Expressions-Daten

Microarray-Technologie & Pre-Processing $\rightarrow$ **Gen-Expressions-Matrix**



$$\rightarrow (x_{ij}) = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

- n Experimente, welche die *Expression*, bzw. die Aktivität von p Genen zeigen.
- In der Regel Daten von 2'000–20'000 Genen, aber nur für 20–200 Experimente.
- Die Experimente sind in mehrere *Populationen* gruppiert, z.B. in verschiedene *Krebsklassen* oder *Tumortypen*. Diese zusätzliche Information ist als kategoriale Zielvariable $y = (y_1, y_2, \dots, y_n)^T$ gegeben.

Notation, Ziel und Herausforderung

Mathematik:

- Gegeben sind n *Beobachtungen* $(x_i, y_i)_{i=1,\dots,n}$ als *i.i.d. Trainingsdaten*
- Verwende das *Gen-Expressions-Profil* $x_i \in R^p$ als *erklärende Variable*
- Die *Klassenzugehörigkeit* $y_i \in \{0, 1\}$ ist die *Zielvariable*

Absicht: Gegeben das Gen-Expressions-Profil x_{neu} eines zusätzlichen Patienten

a) *Mache eine Vorhersage seiner Klassenzugehörigkeit* y_{neu}

b) *Schätze die Wahrscheinlichkeit dieser Vorhersage* $p(x_{neu}) = P[Y = 1|X = x_{neu}]$

- a) & b) sind Standard-Aufgaben in der Statistik
- Da $p \gg n$ ist, funktionieren viele bewährte statistische Verfahren nicht (gut)

$\rightarrow$ Boosting-Methoden sind eine vielversprechende Alternative in dieser Situation

Boosting: Grundlagen

- *Boosting* ist eine Ensemble-Technik, die einen Basis-Prädiktor aggregiert
- Der Basis-Prädiktor (z.B. CART) wird auf gewichteten Daten geschätzt

$$\begin{array}{ccccccc} \boxed{\text{Trainingsdaten}} & \longrightarrow & \boxed{\text{gewichtete Daten}} & \longrightarrow & \dots & \longrightarrow & \boxed{\text{gewichtete Daten}} \\ \downarrow & & \downarrow & & & & \downarrow \\ f_1(x) & & f_2(x) & & & & f_M(x) \end{array}$$

$$\widehat{F}_{Boost}(x) = \sum_{m=1}^{M} \alpha_m f_m(x)$$

- Das ursprüngliche Boosting-Rezept von Freund & Schapire (1996) war:
  - *Erhöhe die Gewichte für wiederholt falsch klassifizierte Beobachtungen*
  - *Das Aggregations-Gewicht α_m ist gross, falls $f_m(\cdot)$ erfolgreich war*
- Die genaue Definition von Daten- und Aggregationsgewichten ist die Kunst in diesem Geschäft

Tumor-Klassifizierung mit Boosting

Boosting funktioniert(e) speziell gut in hochdimensionalen Klassifizierungs-Problemen
→ *Auch geeignet für die Tumor-Klassifizierung mit Gen-Expressions-Daten!?!*

Aber: "Schlechte" Resultate in *Ben-Dor et al. (2000)* und *Dudoit et al. (2002)*.

Unser Beitrag: Diverse Modifikationen, um aus Boosting einen guten Klassifizierer für Gen-Expressions-Daten zu machen *(MD & Bühlmann, Bioinformatics 2003)*

1. *Angepasste Gen-Vorselektion mit der Wilcoxon-Statistik anstelle des t-Tests*
   - Beschränkung auf relevante Gene und Verkürzung der Rechenzeit
   - Die besten Resultate werden idR mit 50 bis 200 Genen erzielt
2. *Der Basis-Prädiktor $f_m(x)$*
   - Beschränkung auf "Stumps", das sind Bäume mit 1 Split und 2 Endknoten
   - Grössere Bäume, die nicht-additive Modelle erlauben, bringen nichts
3. *Umgewichtung der Daten und Aggregierung*
   - Benütze ein robusteres Verfahren, d.h. LogitBoost anstelle von AdaBoost
   - Von Vorteil bei Daten-Inhomogenität und falschen Klassen-Zugehörigkeiten y_i
4. *Nicht-simultane Behandlung von Mehrklassen-Problemen*
   - Codierung auf mehrere binäre Probleme erlaubt unterschiedliche Komplexität

BagBoosting: Grundlagen

Idee und Nomenklatur durch *Peter Bühlmann & Bin Yu (Annals of Statistics, 2000)*

$$\begin{array}{ccccc} \boxed{\text{Trainingsdaten}} & \longrightarrow & \boxed{\text{gewichtete Daten}} & \longrightarrow \dots \longrightarrow & \boxed{\text{gewichtete Daten}} \\ \downarrow & & \downarrow & & \downarrow \\ \boxed{\text{Bagging}} & & \boxed{\text{Bagging}} & & \boxed{\text{Bagging}} \\ \downarrow & & \downarrow & & \downarrow \end{array}$$

$$\underbrace{f_1^*(x) \qquad\qquad f_2^*(x) \qquad\qquad\qquad f_M^*(x)}_{\widehat{F}_{BagBoost}(x) = \sum_{m=1}^{M} \alpha_m f_m^*(x)}$$

- Verwende Bagging als Modul im modifizierten Boosting-Algorithmus
- Der Basis-Prädiktor für den Boosting-Algorithmus ist nun ein "gebaggter Stump"
- Die Umgewichtung der Daten sowie die Aggregierung der Basis-Klassifizierer, d.h. der (modifizierte) Boosting-Algorithmus an sich, bleiben unverändert

Boosting: Theorie

Theorem: Boosting entspricht einem Gradientenverfahren zur Minimierung einer empirischen Verlustfunktion und fittet ein additives logistisches Regressions-Modell

Herleitung: Wir suchen $F_{opt}(x) = \arg\min_{F(x)} E[L(y, F(x))]$ für den *binomialen log-Likelihood Verlust* $L(y, F(x)) = \log\left(1 + e^{-2yF(x)}\right)$. Die Lösung ist

$$F_{opt}(x) = \frac{1}{2} \log\left(\frac{p(x)}{1 - p(x)}\right).$$

Boosting approximiert diese optimale Lösung durch ein *Gradientenverfahren zur Minimierung der empirischen Verlustfunktion*

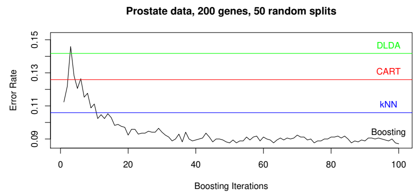
$$R(x_i, y_i, F) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, F(x_i)).$$

Bis auf Details ergibt dies den LogitBoost-Algorithmus, welcher einer *additiven Expansion auf der logistischen Skala* entspricht

$$\widehat{F}_{Boost}(x) = \sum_{m=1}^{M} \alpha_m f_m(x) \approx \frac{1}{2} \log\left(\frac{p(x)}{1 - p(x)}\right).$$

Modifiziertes Boosting: Resultate

Erkenntnisse aufgrund empirischer Resultate von mehr als 10 Gen-Expressions-Datensätzen, basierend auf je 50 zufälligen Einteilungen in Lern- und Testdaten



Zusammenfassung:

- Boosting ist deutlich besser als CART, d.h. einzelne Klassifizierungsbäume
- Ähnliche Fehlerraten wie kNN & DLDA in "einfachen" binären Problemen
- Vorteile vs. kNN & DLDA in "schwierigen" Mehrklassen-Problemen

Das ist nicht das Ende der Geschichte:

- Weitere Verbesserung ist möglich mit einem neuartigen Basis-Prädiktor

BagBoosting: Resultate (I)

Ziel: Vergleich von Klassifizierern durch Analyse ihrer Misklassifikationsraten

Schema:

1) Mache 50 zufällige Einteilungen in $\frac{2}{3}$ Trainings- und $\frac{1}{3}$ Testdaten
2) Wähle 50 Mal die 200 "besten" Gene gemäss der Wilcoxon-Test-Statistik
3) Fitte 50 Mal 5 verschiedene Klassifizierer und generiere Vorhersagen

Klassifizierer:

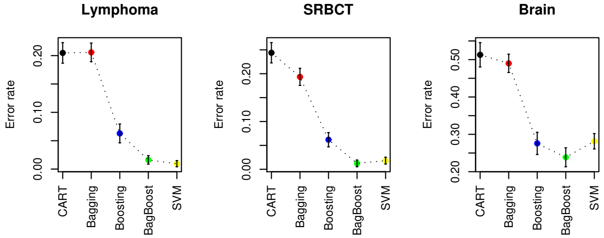
1. *CART:* Standard-Implementation gemäss der R-Funktion `rpart()`
2. *Bagging:* Aggregiere $B = 50$ Classification-Trees aus `rpart()` durch Voting
3. *Boosting:* Verwende den modifizierten LogitBoost-Algorithmus
4. *BagBoosting:* Verwende modifizierten LogitBoost mit 50 "gebaggten" Stumps
5. *SVM:* Standard-Implementation gemäss der R-Funktion `svm()`

BagBoosting: Resultate (II)

Zusammenfassung einer empirischen Studie auf Mehrklassen-Datensätzen
Die Fehlerraten basieren auf 50 zufälligen Einteilungen in Trainings- und Testdaten

50 Splits	Lymphoma	SRBCT	Brain
CART	20.48%	24.38%	51.29%
Bagging	20.57%	19.33%	49.00%
Boosting	6.29%	6.19%	27.57%
BagBoost	1.62%	1.24%	23.86%
SVM	0.95%	1.81%	28.14%

Darstellung mit Pseudo-KI's von jeweils ± 2 Standard-Errors vom Mittelwert



Bagging: Worum geht es?

Breiman (1996): *Bagging ist ein Akronym für Bootstrap Aggregating und ist nützlich zur Verbesserung von instabilen Basis-Prädiktoren (z.B. CART)*

Der Algorithmus lautet wie folgt:

1) Konstruiere B Bootstrap-Datensätze $(x_i^{(b)}, y_i^{(b)})_{i=1,\dots,n}$ mit je n Beobachtungen durch Ziehen mit Zurücklegen aus den Originaldaten
2) Fitte den Basis-Prädiktor $f^{(b)}(x)$ auf jedem der B Bootstrap-Datensätze
3) Der Bagging-Prädiktor ist dann gegeben durch $f^*(x) = \frac{1}{B}\sum_{b=1}^{B} f^{(b)}(x)$

In Worten:

Der Bagging-Prädiktor $f^*(\cdot)$ ist das Mittel über die B Basis-Prädiktoren $f^{(b)}(\cdot)$, die auf den zufälligen Bootstrap-Datensätzen angepasst wurden

Bemerkungen:

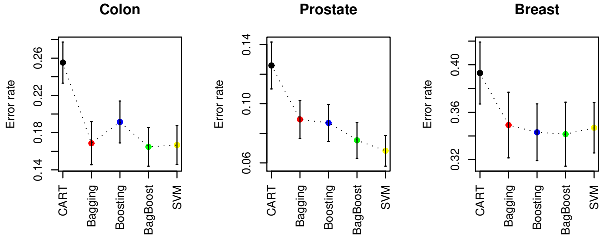
- Der theoretische Bagging-Prädiktor ist $f^*(\cdot) = E^*[f^{(b)}]$ und entspricht $B = \infty$
- Wahl von $B < \infty$ steuert die Approximation, ist aber kein Tuning-Parameter
- Für BagBoosting verwenden wir $B = 50$, was ausreichend genau ist

BagBoosting: Resultate (II)

Zusammenfassung einer empirischen Studie auf binären Microarray-Datensätzen
Die Fehlerraten basieren auf 50 zufälligen Einteilungen in Trainings- und Testdaten

50 Splits	Colon	Prostate	Breast
CART	25.52%	12.59%	39.31%
Bagging	16.86%	8.94%	34.92%
Boosting	19.14%	8.71%	34.31%
BagBoost	16.48%	7.53%	34.15%
SVM	16.67%	6.82%	34.69%

Darstellung mit Pseudo-KI's von jeweils ± 2 Standard-Errors vom Mittelwert



BagBoosting: Was steckt dahinter?

- *BagBoosting mit aggregierten Stumps fittet ein linear logistisches Modell, in dem die Prädiktor-Variablen (Gene) additive Beiträge haben*
  - Bagging & Boosting (mit Stumps) sind Linearkombinationen von Indikatoren
  - Neue Notation: $\widehat{F}_{BagBoost}(x) = \sum_{j=1}^{p} \theta_j g_j(x_j)$, wo $g_j(\cdot)$ eine Treppenfunktion ist
- *Das lineare, additive Modell ist aber stark regularisiert. Es benützt zahlreiche Gene, fittet aber nur wenige Koeffizienten*
  - Jede Variable im Bagging-Prädiktor hat dasselbe Gewicht im Boosting-Algorithmus
  - Der Einfluss zusätzlicher Terme nimmt mit der Anzahl Boosting-Iterationen ab
  - Boosting addiert Komplexität sequenziell: Nur dort, wo es etwas bringt
- *Es findet eine Variablenselektion in einem multivariaten Rahmen statt*
  - In Bagging und Boosting werden einzelne Gene gar nie gewählt
  - BagBoosting funktioniert (ziemlich) gut ohne Gen-Vorselektion
  - Nötig ist die Vorselektion höchstens aus rechentechnischen Gründen
- *Bagging bringt “Randomness” in den Boosting-Algorithmus*
  - Bei Ensemble-Methoden hilfreich (Random Forests, Stochastic Boosting)
  - BagBoosting bietet mehr: Es steckt auch theoretische Legitimation dahinter

Zusammenfassung der Resultate

Erkenntnisse aus 10 Datensätzen, davon 7 binäre und 3 Mehrklassen-Probleme

Rangliste: (in Rangpunkten)

1.	BagBoosting	1.7
2.	SVM	2.3
3.	Boosting	2.8
4.	Bagging	3.6
5.	CART	4.6

Bemerkungen:

- CART und Bagging sind oft massiv schlechter als die anderen 3 Klassifizierer
- BagBoosting ist stets besser als CART und Bagging, meist auch als Boosting
- Wenn BagBoosting gegenüber Boosting nichts bringt, ist es nicht schlechter
- Gute Performance auch im Vergleich zu Benchmarks wie SVM, DLDA und kNN

$\Rightarrow$ Der (einzige) Preis für BagBoosting ist der grössere Rechenaufwand

Diskussion

Vorteile von BagBoosting für die Tumor-Klassifizierung

- *BagBoosting schätzt Wahrscheinlichkeiten* $P[Y = 1|X = x]$
  - Natürliche Quantifizierung der Unsicherheit einer 0/1-Klassifizierung
  - Erlaubt die Klassifikation unter ungleichen Missklassifikationskosten
- *Einfache und “sichere” Implementation*
  - braucht weder kompliziertes Fein-Tuning, noch nicht-lineare Optimierung
  - Resistenz gegenüber Over-Fitting dank sequenzieller Regularisierung

Take-Home-Message:

Modifizierte Boosting-Algorithmen, insbesondere BagBoosting, sind vielversprechend für die Tumor-Klassifizierung mit Gen-Expressions-Daten, wie auch für andere hochdimensionale Probleme

Zusatzmaterial: Manuskripte und Software über (Bag)Boosting sind erhältlich bei

`dettling@stat.math.ethz.ch`

Warum funktioniert BagBoosting?

Theorie:

- *Bagging reduziert die Varianz instabiler Prädiktoren, erhält aber den Bias*
- *Boosting reduziert den Prädiktor-Bias, vergrössert aber die Varianz leicht*
- *!!! Diese beiden Resultate gelten nur in Spezialfällen !!!*

Kurz & prägnant:

Boosting mit einem Basis-Prädiktor, der kleinere Varianz hat, ist besser

Heuristik:

Die Kombination von Boosting und Bagging vereinigt die Vorteile der beiden Ensemble-Methoden und resultiert in einem Algorithmus mit kleinerem Bias und kleinerer Varianz, d.h. kleinerem MSE

Aber:

Eine mathematisch sauberer Beweis dieser Aussage ist (zu) schwierig
Simulationen können eine empirische Bestätigung der Behauptung liefern