9. Additive Models, Trees, and Related Methods

Krystyna Grzesiak
feat Tomasz Kulik

Generalized Additive Models (GAM)

- GAMs - elastyczne metody statystyczne służące do identyfikacji i charakterystyki nieliniowych efektów
- Forma modelu addytywnego:

$$\mathrm{E}(Y|X_1, X_2, \ldots, X_p) = \alpha + f_1(X_1) + f_2(X_2) + \cdots + f_p(X_p).$$

gdzie fj są gładkimi (nieparametrycznymi) funkcjami.

Rozszerzenie funkcji bazowych

- nieparametryczne funkcje fj mogą zostać zapisane jako kombinacje liniowe pewnych funkcji bazowych
- zmienna X jest rozszerzana do jej przekształceń
- estymacja metodą najmniejszych kwadratów

Estymacja wszystkich funkcji jednocześnie

- fitting za pomocą scatterplot smoother (np. cubic smoothing spline lub kernel smoother
- algorytm symultanicznej estymacji wszystkich funkcji fj

Przykłady uogólnionych modeli addytywnych

$$\log\left(\frac{\mu(X)}{1-\mu(X)}\right) = \alpha + \beta_1 X_1 + \cdots + \beta_p X_p.$$

Addytywny model regresji logistycznej zastępuje liniowe składniki uogólnioną, funkcyjną postacią:

$$\log\left(\frac{\mu(X)}{1-\mu(X)}\right) = \alpha + f_1(X_1) + \cdots + f_p(X_p),$$

Przykłady uogólnionych modeli addytywnych

- modele są addytywne, co pozwala na prostą interpretację
- warunkowa wartość oczekiwana Y jest zależna od pewnej addytywnej funkcji predyktorów poprzez funkcję linkującą g:

$$g[\mu(X)] = \alpha + f_1(X_1) + \cdots + f_p(X_p).$$

- uogólnione modele liniowe mogą być w łatwy sposób rozszerzone do uogólnionych modeli addytywnych

Szczegóły:

- scatterplot smoother
- estymatory funkcji pokazują możliwy brak liniowości efektów Xj
- nie wszystkie fj muszą być nieliniowe - można mieszać zarówno liniowe, jak i inne parametryczne formy z nieliniowymi efektami
- nieliniowe składniki także nie muszą być ograniczone do głównych efektów - możemy mieć osobne krzywe dla jednej zmiennej w zależności od innej zmiennej (jakościowej)

Zastosowania:

- $g(\mu) = X^T\beta + \alpha_k + f(Z)$
- $g(\mu) = f(X) + g_k(Z)$
- $g(\mu) = f(X) + g(Z, W)$

Modele addytywne mogą również zastąpić modele liniowe w przypadku addytywnej dekompozycji szeregów czasowych

$$Y_t = S_t + T_t + \varepsilon_t,$$

gdzie St jest składnikiem sezonowym, a Tt to trend czasowy.

Dopasowanie modeli addytywnych

- Założenia

$$Y = \alpha + \sum_{j=1}^{p} f_j(X_j) + \varepsilon,$$

gdzie epsilon jest błędem losowym ze średnią 0.

Dla tego problemu optymalizacyjnego definiuje się sumę kwadratów z dodatkowa karą:

$$\mathrm{PRSS}(\alpha, f_1, f_2, \ldots, f_p) = \sum_{i=1}^{N} \left(y_i - \alpha - \sum_{j=1}^{p} f_j(x_{ij}) \right)^2 + \sum_{j=1}^{p} \lambda_j \int f_j''(t_j)^2 dt_j,$$

- Można pokazać, że powyższe równanie minimalizuje cubic spline model, w którym każda z funkcji fj jest krzywą złożoną z wielomianów sześciennych z węzłami w każdym punkcie xij.

- Aby zapewnić jedyność rozwiązania zakłada się, że dla każdego j :

$$\sum_1^N f_j(x_{ij}) = 0$$

Algorithm 9.1 *The Backfitting Algorithm for Additive Models.*

1. Initialize: $\hat{\alpha} = \frac{1}{N}\sum_1^N y_i$, $\hat{f}_j \equiv 0, \forall i, j$.

2. Cycle: $j = 1, 2, \ldots, p, \ldots, 1, 2, \ldots, p, \ldots,$

$$\hat{f}_j \leftarrow \mathcal{S}_j\left[\{y_i - \hat{\alpha} - \sum_{k \neq j} \hat{f}_k(x_{ik})\}_1^N\right],$$

$$\hat{f}_j \leftarrow \hat{f}_j - \frac{1}{N}\sum_{i=1}^N \hat{f}_j(x_{ij}).$$

until the functions $\hat{f}_j$ change less than a prespecified threshold.

Omówiony algorytm może zostać dostosowany do innych metod dopasowywania modeli poprzez zadanie odpowiedniego operatora wygładzającego Sj:

- inne jednoczynnikowe wygładzacze takie jak regresja wielomianowa czy metody jądrowe
- operatory regresji liniowej dające wielomianowe dopasowanie, takie jak piecewise constant fits, parametric spline fits, series and Fourier fits
- bardziej skomplikowane operatory takie jak powierzchniowe wygładzanie dla wysokorzędowych interakcji lub okresowe wygładzacze dla sezonowych efektów

Przykład c.d.: Addytywna regresja logistyczna

- zrozumienie roli każdego z faktorów raczej ważniejsze od klasyfikowania nowych zmiennych
- uogólniona struktura addytywnego modelu logistycznego:

$$\log \frac{\Pr(Y=1|X)}{\Pr(Y=0|X)} = \alpha + f_1(X_1) + \cdots + f_p(X_p).$$

- Backfitting algorithm within a Newton–Raphson procedure

Algorithm 9.2 *Local Scoring Algorithm for the Additive Logistic Regression Model.*

1. Compute starting values: $\hat{\alpha} = \log[\bar{y}/(1-\bar{y})]$, where $\bar{y} = \text{ave}(y_i)$, the sample proportion of ones, and set $\hat{f}_j \equiv 0 \; \forall j$.

2. Define $\hat{\eta}_i = \hat{\alpha} + \sum_j \hat{f}_j(x_{ij})$ and $\hat{p}_i = 1/[1+\exp(-\hat{\eta}_i)]$.

   Iterate:

   (a) Construct the working target variable

   $$z_i = \hat{\eta}_i + \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1-\hat{p}_i)}.$$

   (b) Construct weights $w_i = \hat{p}_i(1-\hat{p}_i)$

   (c) Fit an additive model to the targets z_i with weights w_i, using a weighted backfitting algorithm. This gives new estimates $\hat{\alpha}, \hat{f}_j, \; \forall j$

3. Continue step 2. until the change in the functions falls below a prespecified threshold.

Przykład: klasyfikowanie spamu

Dane: (ftp.ics.uci.edu)

- 4601 wiadomości mailowych
- 57 zmiennych (w tym 48 zmiennych jakościowych)

Podział:
zbiór testowy: 1536
zbiór treningowy: 3065

Dopasowanie krzywą składającą się z wielomianów sześciennych

Test error rates:

TABLE 9.1. *Test data confusion matrix for the additive logistic regression model fit to the spam training data. The overall test error rate is* 5.5%.

	Predicted Class	
True Class	email (0)	spam (1)
email (0)	58.3%	2.5%
spam (1)	3.0%	36.3%

liniowa regresja logistyczna: 7.6%.

TABLE 9.2. *Significant predictors from the additive model fit to the spam training data. The coefficients represent the linear part of $\hat{f}_j$, along with their standard errors and Z-score. The nonlinear P-value is for a test of nonlinearity of $\hat{f}_j$.*

Name	Num.	df	Coefficient	Std. Error	Z Score	Nonlinear P-value
			Positive effects			
our	5	3.9	0.566	0.114	4.970	0.052
over	6	3.9	0.244	0.195	1.249	0.004
remove	7	4.0	0.949	0.183	5.201	0.093
internet	8	4.0	0.524	0.176	2.974	0.028
free	16	3.9	0.507	0.127	4.010	0.065
business	17	3.8	0.779	0.186	4.179	0.194
hpl	26	3.8	0.045	0.250	0.181	0.002
ch!	52	4.0	0.674	0.128	5.283	0.164
ch$	53	3.9	1.419	0.280	5.062	0.354
CAPMAX	56	3.8	0.247	0.228	1.080	0.000
CAPTOT	57	4.0	0.755	0.165	4.566	0.063
			Negative effects			
hp	25	3.9	−1.404	0.224	−6.262	0.140
george	27	3.7	−5.003	0.744	−6.722	0.045
1999	37	3.8	−0.672	0.191	−3.512	0.011
re	45	3.9	−0.620	0.133	−4.649	0.597
edu	46	4.0	−1.183	0.209	−5.647	0.000

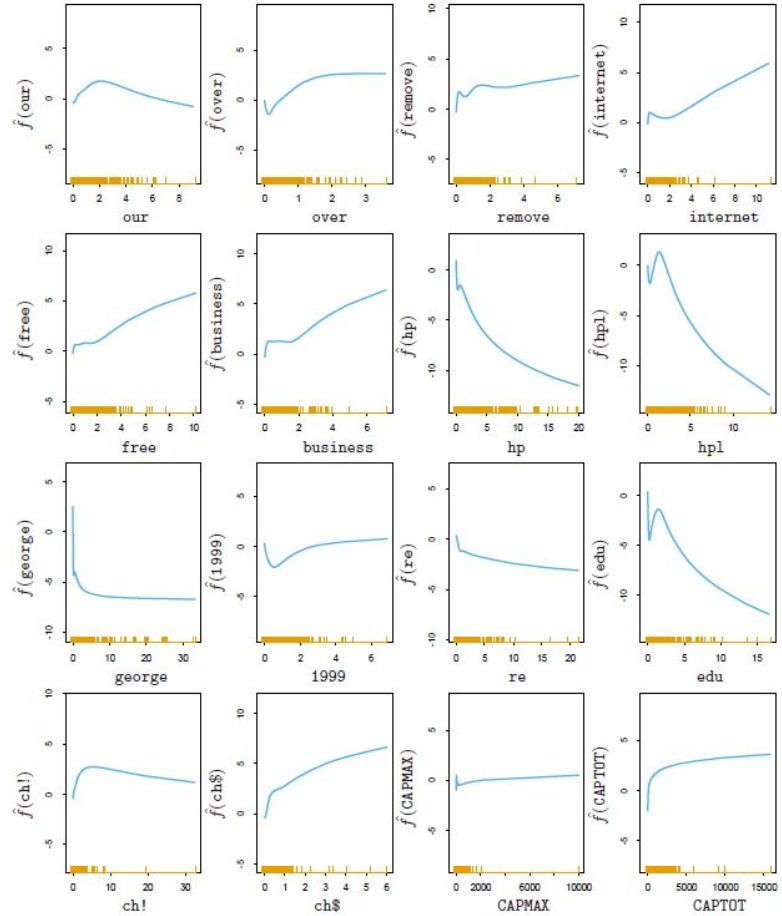


FIGURE 9.1. *Spam analysis: estimated functions for significant predictors. The* rug plot *along the bottom of each frame indicates the observed values of the corresponding predictor. For many of the predictors the nonlinearity picks up the discontinuity at zero.*

Podsumowanie

Zalety:

- Modele addytywne są przydatnym rozszerzeniem zwykłych modeli liniowych
- Nadal możemy używać podobnych narzędzi jak w przypadku modeli liniowych
- Dopasowywanie modeli addytywnych metodą backfitting jest proste i można je łatwo modyfikować
- wyniki są łatwe do interpretacji - szeroko używane w środowisku statystycznym

Wady:

- ograniczenia dla dużej eksploracji danych - algorytm backfitting dopasowuje wszystkie zmienne, co jest trudne do osiągnięcia w przypadku dużych zbiorów danych (procedura BRUTO łączy backfitting z selekcją zmiennych żeby poradzić sobie z tym problemem)

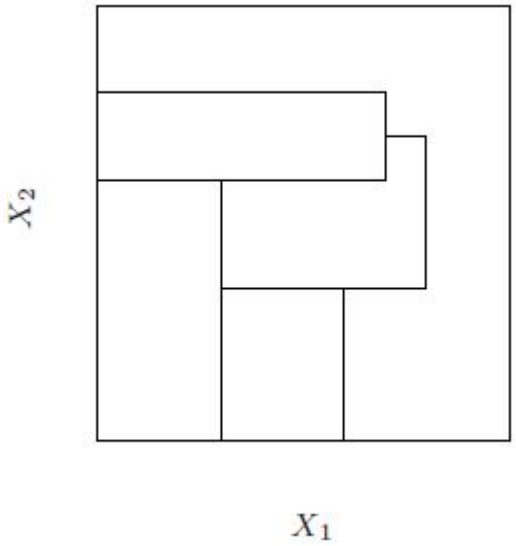
- dla dużych danych efektywniejsze są podejścia takie jak boosting

Modele oparte o drzewa

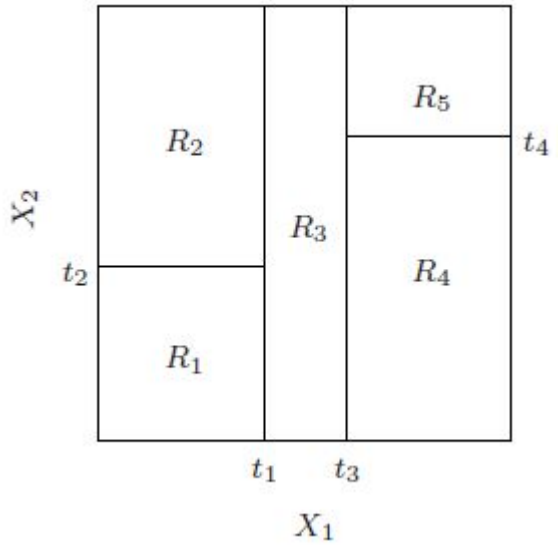
- Podział przestrzeni predyktorów na prostokąty
- Dopasowanie prostego modelu (np. stałej) w każdym prostokącie
- Proste, ale skuteczne
- Przykładowe modele: CART, C4.5

- Problem regresji:
  - Y - ciągła zmienna odpowiedzi
  - $X_1, X_2 \in [0,1]$ - predyktory

- Podział przestrzeni liniami równoległymi do osi układu współrzędnych
- W każdym nowo powstałym obszarze możemy przybliżać Y przez inną wartość
- Problem: niektóre obszary są trudne do zdefiniowania

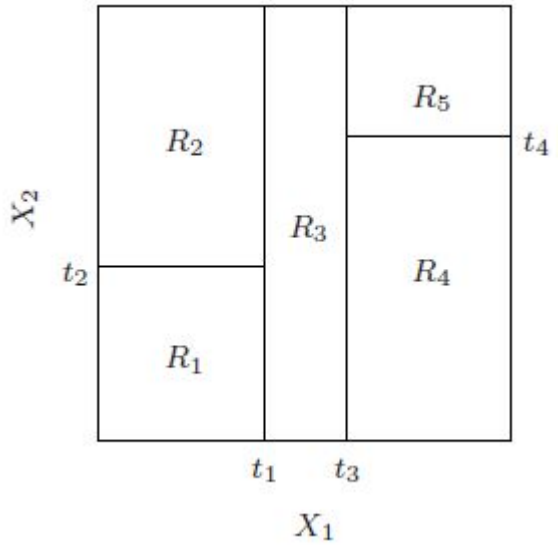


- Rozwiązanie: rekurencyjne podziały binarne:
  - w pierwszym kroku dzielimy przestrzeń na dwie części i przybliżamy zmienną odpowiedzi średnią z wartości Y należących do danego podzbioru
  - zmienna (X□) i punkt podziału dobierane tak, by przybliżenie było jak najdokładniejsze
  - podział jest kontynuowany aż do spełnienia warunku stopu

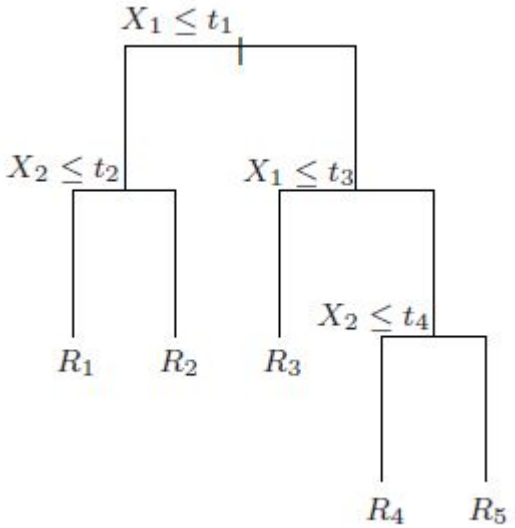


- Przykład:
  - Podział w $X_1=t_1$
  - Obszar $X_1 \leq t_1$ podzielony w $X_2=t_2$
  - Obszar $X_1 > t_1$ podzielony w $X_1=t_3$
  - Obszar $X_1 > t_3$ podzielony w $X_2=t_4$
- Rezultat: podział przestrzeni na pięć obszarów $R_1, R_2, \ldots, R_5$
- Model elementom zbioru R□ przypisuje stałą c□
- Formalnie:

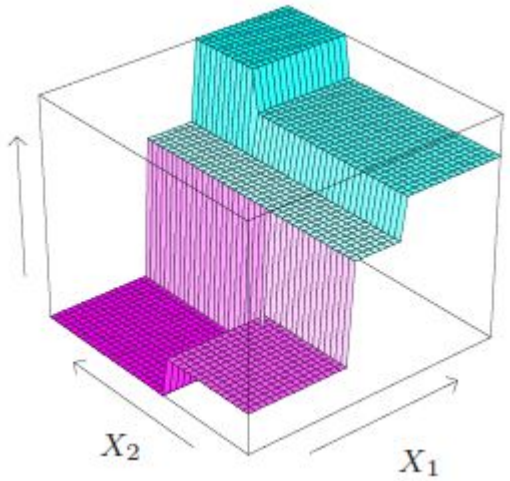
$$\hat{f}(X) = \sum_{m=1}^{5} c_m I\{(X_1, X_2) \in R_m\}$$



- Model można zaprezentować także w formie drzewa binarnego
- Pełny zbiór danych u góry
- Obserwacje spełniające warunek są przypisywane do lewej gałęzi, pozostałe do prawej
- Końcowe węzły (liście) odpowiadają obszarom R_1, R_2,..., R_5



- Graficzna prezentacja modelu regresji dla przykładowych wartości c_1=-5, c_2=-7, c_3=0, c_4=2, c_5=4



Drzewa regresyjne

- Dane:
  - N obserwacji
  - każda obserwacja składa się z p zmiennych oraz zmiennej odpowiedzi
  - oznaczenia:

$$(x_i, y_i), i = 1, 2, \ldots, N$$

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$$

Drzewa regresyjne – podział danych w węźle

- Chcemy, aby algorytm decydował o wyborze zmiennej do podziału oraz punktu, w którym ten podział nastąpi
- Załóżmy, że mamy dane podzielone na M obszarów R_1, R_2,..., R_M
- Modelujemy zmienną odpowiedzi przez stałą c_m w każdym z obszarów, tj.:

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

- Kryterium podziału - suma kwadratów błędów: $\sum (y_i - f(x_i))^2$
- Wówczas najlepszym estymatorem $\hat{c}_m$ jest średnia wartość y w zbiorze R_m:

$$\hat{c}_m = \text{ave}(y_i | x_i \in R_m)$$

Drzewa regresyjne – podział danych w węźle

- Znalezienie najlepszego w sensie SSE binarnego podziału jest najczęściej obliczeniowo niewykonalne
- Rozwiązaniem jest zastosowanie algorytmu zachłannego
- Zaczynając od całego zbioru danych, dla zmiennej podziału *j* oraz punktu podziału *s* możemy zdefiniować parę podprzestrzeni:

$$R_1(j,s) = \{X|X_j \leq s\} \text{ and } R_2(j,s) = \{X|X_j > s\}$$

- Szukamy takiej zmiennej *j* oraz punktu *s*, które zminimalizują wyrażenie:

$$\min_{j,\,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right]$$

- Dla dowolnych j oraz s rozwiązaniem problemu w nawiasie jest:

$$\hat{c}_1 = \text{ave}(y_i|x_i \in R_1(j,s)) \text{ and } \hat{c}_2 = \text{ave}(y_i|x_i \in R_2(j,s))$$

- Dla każdej zmiennej *j*, można szybko określić optymalny punkt *s*. Iteracja po wszystkich *p* zmiennych pozwala wyznaczyć najlepszą parę (*j*,*s*)
- Po wyznaczeniu pary (*j*,*s*), dzielimy przestrzeń na dwie podprzestrzenie i powtarzamy proces na nowych zbiorach

Drzewa regresyjne – rozmiar drzew

- Rozmiar drzewa określa złożoność modelu
- Bardzo duże drzewo może skutkować nadmiernym dopasowaniem (overfitting)
- Małe drzewo może nie uwzględniać ważnych zależności
- Kryteria stopu:
  - dzielenie węzła tylko, jeśli zysk w SSE przekracza pewną ustaloną wartość
    - pozornie niepotrzebny podział może prowadzić do dobrego podziału w niższej części drzewa
  - dzielenie węzła, jeśli jego wielkość przekracza określoną wartość (np. 5 obserwacji)
    - duże drzewo T_0, które trzeba przyciąć w oparciu o kryterium kosztu złożoności (*cost-complexity pruning, CCP)*

Drzewa regresyjne - Cost-complexity pruning

- Niech $T \subset T_0$ oznacza dowolne drzewo powstałe przez przycięcie T_0
- Niech *m* indeksuje końcowe (najniższe) węzły (węzeł *m* odpowiada zbiorowi R_m)
- Niech |T| oznacza liczbę końcowych węzłów w T
- Przyjmując oznaczenia:

$$N_m = \#\{x_i \in R_m\},$$
$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i,$$
$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2$$

możemy zdefiniować kryterium kosztu złożoności (CCP) jako:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|$$

Drzewa regresyjne – weakest link pruning

- Parametr $\alpha \geq 0$ odpowiada za balans między rozmiarem drzewa a dokładnością dopasowania do danych
  - duże wartości α skutkują małymi drzewami i odwrotnie
  - α=0 daje pełne drzewo T_0
- Chcemy dla każdego α znaleźć drzewo $T_\alpha \subseteq T_0$ minimalizujące $C_\alpha(T)$
- Można pokazać, że dla każdego α najmniejsze takie drzewo jest jedyne
- Aby je znaleźć, korzystamy z przycinania najsłabszego ogniwa (weakest link pruning)
  - usuwamy wewnętrzny węzeł, który odpowiada za najmniejszy przyrost wyrażenia: $\sum_m N_m Q_m(T)$
  - ustalamy α taką, aby CCP poddrzewa było mniejsze niż drzewa
    - pełne drzewo ma CCP=SSE dla α=0
  - powtarzamy aż otrzymamy pojedynczy węzeł (korzeń)
  - otrzymujemy w ten sposób skończoną sekwencję poddrzew (i ciąg wartości α), która zawiera T_α
  - Estymator α uzyskujemy poprzez pięcio- lub dziesięciokrotną walidację krzyżową
  - Wybieramy wartość odpowiadającą poddrzedwu minimalizującemu SSE

Drzewa klasyfikacyjne

- Dla zmiennej odpowiedzi przyjmującej wartości 1,2,...,K, potrzeba jedynie zmienić kryteria podziału węzłów i przycinania drzewa
- Dla *m*-tego węzła odpowiadającemu podprzestrzeni R□ z N□ obserwacjami, definiujemy proporcję obserwacji należących do klasy *k* w węźle *m*:

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

- Klasyfikujemy obserwacje w węźle m do najliczniejszej klasy

$$k(m) = \arg\max_k \hat{p}_{mk}$$

Drzewa klasyfikacyjne - impurity measures

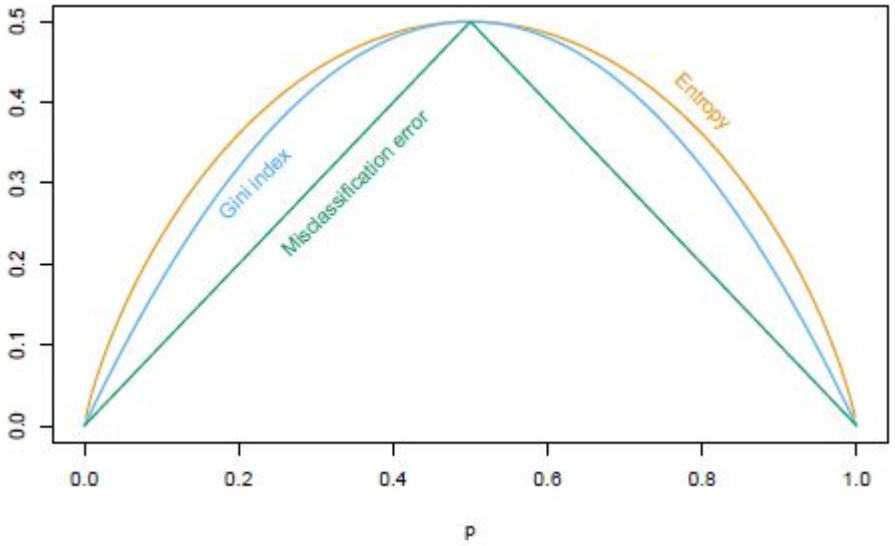
- Popularne metody mierzenia “zanieczyszczenia węzła” (node impurity) Q□(T):

Misclassification error:	$\frac{1}{N_m}\sum_{i\in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk(m)}$
Gini index:	$\sum_{k\neq k'} \hat{p}_{mk}\hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1-\hat{p}_{mk})$.
Cross-entropy or deviance:	$-\sum_{k=1}^{K} \hat{p}_{mk}\log\hat{p}_{mk}$.

- Przy dwóch klasach, jeśli przez *p* oznaczymy proporcję w drugiej klasie, powyższe wyrażenia przyjmują odpowiednio postaci:
  - 1-max(*p*, 1-*p*)
  - 2*p*(1-*p*)
  - -*p*log(*p*)-(1-*p*)log(1-*p*)

Drzewa klasyfikacyjne – impurity measures

- Wykres przedstawia porównanie tych trzech metod w zależności od parametru *p*
- Wszystkie są podobne, ale Gini index i entropia są różniczkowalne, a tym samym bardziej odpowiednie do numerycznej optymalizacji



Drzewa klasyfikacyjne - impurity measures

- Entropia i Gini Index są bardziej wrażliwe na proporcje w węźle
- Przykład:
  - dwie klasy, w każdej 400 obserwacji (400,400)
  - I podział: węzły (300,100) i (100,300)
  - II podział: węzły (200,400) i (200,0)
  - w obu przypadkach 25% obserwacji zostało błędnie sklasyfikowanych
  - w II podziale mamy “czysty” węzeł, który jest bardziej pożądany
  - podobnie jak w przypadku drzew regresyjnych, aby obliczyć impurity węzła, mnożymy jej (impurity)wartość przez jego (węzła) liczebność

Drzewa klasyfikacyjne - impurity measures

- Gini Index oraz Entropia mają niższe impurity dla drugiego podziału, zatem przy tworzeniu drzew lepiej korzystać z jednego z nich
- Do przycinania drzewa w oparciu o kryterium kosztu złożoności można użyć dowolnej z powyższych metod (typowo jest to misclassification rate)

	I podział	II podział
Misclass. error	200	200
Gini Index	300	266.67
Entropy	449.87	381.91

Drzewa klasyfikacyjne - Gini Index

- Gini Index może być interpretowany na dwa interesujące sposoby
  - Zamiast klasyfikować obserwacje na podstawie najliczniejszej klasy w węźle, możemy klasyfikować do klasy *k* z prawdopodobieństwem $\hat{p}_{mk}$
    - Błąd treningowy jest wówczas postaci $\sum_{k \neq k'} \hat{p}_{mk}\hat{p}_{mk'}$ - Gini Index.
  - Jeśli w danym węźle oznaczymy przynależność do klasy *k* przez 1, a do pozostałych klas przez 0, wówczas wariancja takiej zerojedynkowej zmiennej odpowiedzi jest postaci $\hat{p}_{mk}(1 - \hat{p}_{mk})$. Suma po wszystkich *k* klasach da ponownie Gini Index.

Drzewa klasyfikacyjne – macierz straty

- W niektórych przypadkach konsekwencje błędnego zaklasyfikowania obserwacji są większe niż w innych
  - np. predykcja, że pacjent nie będzie miał ataku serca, w sytuacji kiedy jednak go dostanie
- Aby temu przeciwdziałać definiujemy macierz straty L rozmiaru *K*x*K*, gdzie $L_{kk'}$ oznacza stratę spowodowaną zaklasyfikowaniem obserwacji pochodzącej z klasy *k* do klasy *k'*
- Zwykle przyjmuje się, że poprawnym klasyfikacjom nie przypisuje się kary, tj. $L_{kk} = 0 \ \forall k.$
- Aby uwzględnić ten pomysł w procesie modelowania możemy zmodyfikować Gini Index do postaci: $\sum_{k \neq k'} L_{kk'} \hat{p}_{mk} \hat{p}_{mk'}$
- Sposób ten działa w przypadku wielu klas, ale nie powoduje żadnego efektu w przypadku dwóch klas, gdyż każdorazowo dostaniemy współczynnik $L_{kk'} + L_{k'k}$
  - Rozwiązaniem może być przypisanie wagi tylko jednej klasie
- Efektem przypisania obserwacjom wag jest zmodyfikowanie rozkładu a priori
- W ujęciu bayesowskim, w końcowym węźle klasyfikujemy do klasy $k(m) = \arg\min_k \sum_\ell L_{\ell k} \hat{p}_{m\ell}.$

Brakujące dane

- Usunięcie obserwacji z brakującymi danymi
  - może powodować duże straty w zbiorze treningowym
- Uzupełnienie danych, np. przez średnią, medianę lub bardziej zaawansowaną metodą
- W modelach opartych o drzewa istnieją dwa lepsze sposoby:
  - W przypadku zmiennych kategorycznych tworzymy dodatkową wartość, jaką zmienna może przyjmować, oznaczającą brak informacji (“brak”, 0)
    - może się zdarzyć, że obserwacje z brakującymi wartościami będą się w modelu zachowywać inaczej niż te z kompletną informacją
  - Bardziej ogólnym podejściem jest określenie zmiennych zastępczych (surrogate variables)
    - rozważając daną zmienną jako kandydata do podziału węzła, korzystamy jedynie z obserwacji z pełną informacją
    - wybieramy najlepszą zmienną i optymalny punkt podziału
    - pierwszą zmienną zastępczą (i odpowiadającym jej punktem podziału) jest ta, która najdokładniej przybliża podział uzyskany dzięki zmiennej pierwszego wyboru, potem druga itd.
    - jeśli przepuszczając obserwacje przez drzewo brakuje wartości zmiennej pierwszego wyboru, korzystamy kolejno ze zmiennych zastępczych
    - powyższa metoda wykorzystuje korelacje między zmiennymi objaśniającymi; im wyższa korelacja, tym mniejsza strata informacji w wyniku braku danych

Dlaczego podziały binarne?

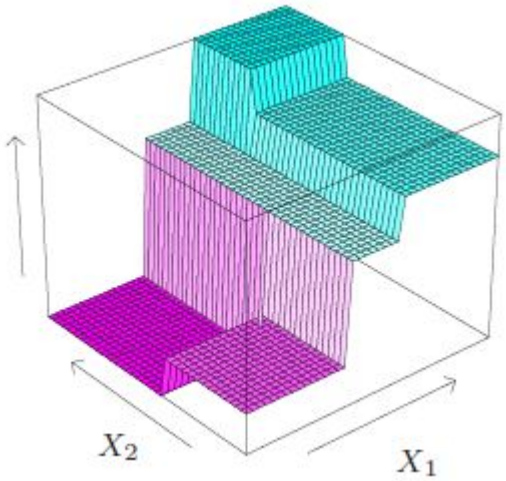
- Zamiast dzielić obserwacje na dwie grupy w każdym węźle można rozważyć podział na więcej podzbiorów
- Czasami takie podejście może być użyteczne, ale nie jest dobrą strategią w ogólności
  - podział na wiele grup partycjonuje dane zbyt szybko zostawiając niedostateczną ich liczbę na potrzeby niższych poziomów drzewa.
- Podział na wiele grup można uzyskać ciągiem podziałów binarnych

Wady drzew: niestabilność

- Drzewa są obarczone dużą wariancją
  - mała zmiana w danych może skutkować istotnie różnym ciągiem podziałów, co przekłada się na niepewność w interpretacji
- Główną przyczyną tej niestabilności jest hierarchiczna natura procesu
  - błąd popełniony na szczycie drzewa ciągnie się do samego dołu
- Skutki można łagodzić dobierając stabilniejsze kryteria podziału
- Lepszym rozwiązaniem jest uśrednianie wielu drzew poprzez bagging (bootstrap aggregating) i lasy losowe
- Bagging dzieli zbiór treningowy rozmiaru *n* na *m* zbiorów treningowych rozmiaru *n'* poprzez losowanie z rozkładu jednostajnego ze zwracaniem
  - przy dużym zbiorze treningowym i *n*=*n'*, średnio w każdym z *m* zbiorów dostaniemy 63.2% niepowtarzających się obserwacji

Wady drzew: brak wygładzenia

- Brak wygładzenia powierzchni predykcyjnej (prediction surface)
- W przypadku klasyfikacji ze stratą 0/1 nie jest to duży problem, ponieważ obciążenie estymatora prawdopodobieństwa przynależności do klasy ma ograniczony efekt
- Może mieć silnie negatywny wpływ w przypadku regresji, gdzie oczekujemy wygładzonych funkcji
- Procedura MARS częściowo rozwiązuje ten problem



Wady drzew: określenie struktury addytywnej

- Rozważmy model regresji: $Y=c_1 I(X_1<t_1)+c_2 I(X_2<t_2)+\varepsilon$, gdzie ε jest szumem o średniej zero
- W drzewie binarnym pierwszy węzeł będzie odpowiadał zmiennej X_1 i punktowi t_1
- Aby uchwycić addytywną strukturę oba nowo powstałe liście należy podzielić według zmiennej X_2 w punkcie t_2
- To może się faktycznie zdarzyć, ale nie ma czynników, które by ukierunkowywały model w tę stronę
- Przy większej liczbie zmiennych odtworzenie takiej struktury jest mało prawdopodobne
- Rozpoznanie struktury addytywnej w rozbudowanym drzewie jest bardzo trudne
- Procedura MARS porzuca strukturę drzewa na rzecz struktury addytywnej