AdaBoost

Karla Brkić

17. prosinca 2008.

[pretpostavlja se da je čitatelj upoznat s osnovnim pojmovima iz strojnog učenja i raspoznavanja uzoraka: (binarni) klasifikator, uzorak, skup za učenje, perceptron]

Općenito o AdaBoostu

AdaBoost (*Ada*ptive *Boost*ing) je meta-algoritam iz područja strojnog učenja kojim se konstruira "jaki" klasifikator pomoću većeg broja "slabih" klasifikatora. "Slabi" klasifikator je bilo koji klasifikator koji je malo bolji od slučajnog pogađanja, tj. klasifikator koji razvrstava uzorke s točnošću većom od 50 % (tipično perceptron, decizijsko stablo i sl.).

AdaBoost je

- ◇ *meta-algoritam* jer koristi proizvoljne algoritme za učenje slabih klasifikatora (perceptron, decizijsko stablo...)
- ◇ *adaptivan* jer dinamički prilagođava skup za učenje kako bi konačni klasifikator bio što bolji

Osnovna zamisao

Pretpostavimo da je zadatak stvoriti klasifikator koji razvrstava uzorke u jedan od dva razreda (binarni klasifikator). Zadan je skup označenih uzoraka za učenje.

Algoritam AdaBoost najprije pridjeljuje težine elementima skupa za učenje. Na tako otežanom skupu uči se slabi klasifikator s klasifikacijskom funkcijom $h_t(x)$. Elementima koje dobiveni klasifikator pogrešno klasificira težine se povećavaju, dok se težine ispravno klasificiranih elemenata smanjuju. Na promijenjenom skupu uči se novi slabi klasifikator s funkcijom $h_{t+1}(x)$. Postupak se ponavlja proizvoljan broj puta T, dok se ne dobije T slabih klasifikatora. Svaki dobiveni slabi klasifikator mora ispravno klasificirati barem 50 % težine skupa za učenje! Rezultantni jaki klasifikator je funkcija predznaka linearne kombinacije T slabih klasifikatora.

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{1}$$

$$H(x) = \mathrm{sgn}(f(x)) \tag{2}$$

Parametar α_t označava kvalitetu klasifikatora h_t i proporcionalan je njegovoj točnosti. Što je klasifikator točniji, to će njegova težina u ukupnom klasifikatoru biti veća.

Formalizacija

Neka je zadan skup za učenje$\{(x_1, y_1), ..., (x_m, y_m)\}$, pri čemu je x_i element nekog prostora uzoraka X, a y_i pripada skupu oznaka $Y = \{-1, +1\}$. Algoritam AdaBoost izvodi se na sljedeći način:

1. Inicijaliziraju se težine uzoraka iz skupa za učenje, $D_1(i) = \frac{1}{m}$ (i-ta težina odgovara i-tom uzorku)

2. Za $t = 1, ..., T$

◇ Pokreće se postupak učenja slabog klasifikatora na uzorcima za učenje uz korištenje težina D_t

◇ Učenje slabog klasifikatora vraća klasifikacijsku funkciju $h_t : X \to \{-1, +1\}$. Pogreška slabog klasifikatora ε_t bit će suma težina svih uzoraka iz skupa za učenje koje je pogrešno razvrstao,

$$\varepsilon_t = \sum_j D_t(j), \text{ gdje je } y_j \neq h_t(x_j) \tag{3}$$

Ukoliko je moguće, u postupku učenja odabrat ćemo onaj slabi klasifikator za koji je ε_t minimalan. Ako je pogreška najboljeg klasifikatora veća od 0.5, postupak se zaustavlja.

◇ Odabire se parametar α_t (vidi poglavlje o odabiru i izraz 11). Parametar α_t intuitivno možemo razumjeti kao mjeru dobrote klasifikatora h_t. Njegova je namjena dvojaka: koristi se za podešavanje težina uzoraka u pojedinom koraku, ali i kao faktor uz h_t u konačnom jakom klasifikatoru. U poglavlju o odabiru pokazano je da tako primijenjeni α_t minimizira ukupnu pogrešku klasifikatora.

◇ Podešavaju se težine uzoraka za sljedeći korak

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & h_t(x_i) = y_i \\ e^{\alpha_t} & h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \times e^{-\alpha_t h_t(x_i) y_i}}{Z_t} \end{aligned} \tag{4}$$

◇ Z_t je normalizacijski faktor koji se bira tako da suma težina svih uzoraka D_{t+1} bude jednaka jedan[1], $\sum_i D_{t+1}(i) = 1$

$$Z_t = \sum_{i=1}^{m} D_t(i) e^{-\alpha_t h_t(x_i) y_i} \tag{5}$$

Konačni jaki klasifikator je

$$H(x) = \text{sgn}(\sum_{t=1}^{T} \alpha_t h_t(x))$$

[1] Kažemo da je u tom slučaju D_{t+1} *distribucija vjerojatnosti.*

Podešavanje težina

Kao što je navedeno u prethodnom odlomku, težine elemenata skupa za učenje podešavaju se u svakom koraku prema jednadžbi 4:

$$D_{t+1} = \frac{D_t(i) \times e^{-\alpha_t h_t(x_i) y_i}}{Z_t}$$

Zašto je odabrana baš funkcija $e^{-\alpha_t h_t(x_i) y_i}$? Osnovna zamisao algoritma je u svakoj rundi povećati težine onih uzoraka koji su bili pogrešno klasificirani, a smanjiti težine dobro klasificiranih. Povećanje pojedine težine očito se lako ostvaruje tako da se ta težina pomnoži s brojem većim od 1, a smanjenje težine može se dobiti množenjem s brojem manjim od jedan.

Za bilo koji $\beta \in \mathbb{R}$ vrijedi

$$\begin{aligned} e^{\beta} &< 1, \quad \beta < 0 \\ e^{\beta} &= 1, \quad \beta = 0 \\ e^{\beta} &< 1, \quad \beta > 0 \end{aligned}$$

Stoga vrijedi i

$$e^{-\alpha_t h_t(x_i) y_i} \begin{cases} < 1, & h_t(x_i) = y_i \\ > 1, & h_t(x_i) \neq y_i \end{cases}$$

Parametar α_t omogućuje fino podešavanje postupka učenja. Na koji način odabrati optimalni α_t?

Definicija pogreške klasifikatora i gornja granica pogreške

Željeli bismo odabrati α_t tako da minimiziramo ukupnu pogrešku jakog klasifikatora. Definirajmo najprije ukupnu pogrešku.

Uzet ćemo vrlo jednostavnu definiciju: ukupna pogreška jakog klasifikatora je postotak primjera u skupu za učenje koje jaki klasifikator pogrešno klasificira. Zapisano matematički:

$$\varepsilon_H = \frac{1}{m} |\{x_i : (H(x_i) \neq y_i)\}| \tag{6}$$

Dakle, postotak pogrešno klasificiranih primjera jednak je kardinalitetu skupa tih primjera podijeljenim s ukupnim brojem primjera u skupu za učenje.

Pokazat ćemo da je ε_H odozgora ograničen funkcijom normalizacijskih faktora Z_t, $t \in [1, ..., T]$.

Teorem. Za pogrešku ε_H klasifikatora $H(x)$ vrijedi

$$\varepsilon_H \leq \prod_{t=1}^{T} Z_t$$

$$\frac{1}{m} |\{x_i : (H(x_i) \neq y_i)\}| \leq \prod_{t=1}^{T} Z_t \tag{7}$$

Dokaz.

◇ *Dio prvi*

Raspišimo izraz za težine u pojedinim koracima:

$$
\begin{aligned}
D_1(i) &= \frac{1}{m} \\
D_2(i) &= \frac{D_1(i)\cdot e^{-\alpha_1 h_1(x_i)y_i}}{Z_1} = \frac{e^{-\alpha_1 h_1(x_i)y_i}}{m\cdot Z_1} \\
D_3(i) &= \frac{D_2(i)\cdot e^{-\alpha_2 h_2(x_i)y_i}}{Z_2} = \frac{e^{-\alpha_1 h_1(x_i)y_i}\cdot e^{-\alpha_2 h_2(x_i)y_i}}{m\cdot Z_1\cdot Z_2} \\
&\vdots \\
D_{k+1}(i) &= \frac{e^{-\alpha_1 h_t(x_i)y_1}\cdot e^{-\alpha_2 h_2(x_i)y_2}\cdot\ldots\cdot e^{-\alpha_k h_k(i)y_k}}{m\cdot Z_1\cdot Z_2\cdot\ldots\cdot Z_k} \qquad (8)
\end{aligned}
$$

Napišimo čemu bi bile jednake težine u $T+1$ koraku (uoči: algoritam traje jedan korak manje, T koraka). Jednostavnim uvrštavanjem u izraz 8 dobiva se:

$$
\begin{aligned}
D_{T+1}(i) &= \frac{D_T(i)e^{-\alpha_T h_T(x_i)y_T}}{Z_T} \\
&= \frac{e^{-\sum_t \alpha_t h_t(x_i)y_i}}{m\prod_t Z_t}
\end{aligned}
$$

Prema izrazu 1 je

$$f(x) = \sum_{t=1}^{T}\alpha_t h_t(x)$$

pa je tada

$$D_{T+1}(i) = \frac{e^{-f(x_i)y_i}}{m\prod_t Z_t} \qquad (9)$$

◇ *Dio drugi*

Pretpostavimo da je uzorak i pogrešno klasificiran. Tada je $H(x_i) \neq y_i$.Definirajmo pomoćnu notaciju

$$[[l(x)]] = \begin{cases} 0 & \text{ako ne vrijedi l(x)} \\ 1 & \text{ako vrijedi l(x)} \end{cases}$$

Uz pretpostavku da je uzorak pogrešno klasificiran sigurno vrijedi i

$$[[H(x_i) \neq y_i]] \leq e^{-y_i f(x_i)} \qquad (10)$$

s obzirom da lijeva strana nejednadžbe može poprimiti isključivo vrijednosti 0 ili 1, dok je desna strana nejednadžbe sigurno veća ili jednaka 1.

◇ *Dokaz*

Uvedimo sumu po čitavom skupu za učenje u izraz 10, te upotrijebimo izraz 9:

$$\begin{aligned}
\frac{1}{m}\sum_{i=1}^{m}[[H(x_i) \neq y_i]] &\leq \frac{1}{m}\sum_{i=1}^{m} e^{-y_i f(x_i)} \\
&= \sum_{i=1}^{m}(D_{T+1}(i) \cdot \prod_t Z_t) \\
&= \prod_t Z_t \cdot \sum_{i=1}^{m}(D_{T+1}(i) \\
&= \prod_t Z_t \cdot 1 = \prod_t Z_t
\end{aligned}$$

Izraz lijevo odgovara kardinalitetu skupa svih pogrešno klasificiranih primjera. Time smo dokazali tvrdnju 7:

$$\frac{1}{m}|\{x_i : (H(x_i) \neq y_i)\}| \leq \prod_{t=1}^{T} Z_t$$

QED.

Vrlo važna posljedica ovog teorema: *ukupnu pogrešku jakog klasifikatora možemo minimizirati tako da minimiziramo Z_t u svakom koraku algoritma*!

Odabir α_t

Pokazali smo da je ukupna pogreška jakog klasifikatora ograničena produktom normalizacijskih faktora Z_t u pojedinim koracima algoritma. Napišimo još jednom izraz za Z_t:

$$Z_t = \sum_{i=1}^{m} D_t(i) e^{-\alpha_t h_t(x_i) y_i}$$

Vidimo da je Z_t funkcija parametra α_t. Kako bi pojedini Z_t bio minimalan, valja odabrati onaj α_t koji ga minimizira. Odredimo taj α_t tako da deriviramo izraz za Z_t i dobiveno izjednačimo s nulom:

$$\begin{aligned}
\frac{d}{d\alpha_t} Z_t &= \sum_{i=1}^{m} D_t(i)(-h_t(x_i) y_i) e^{-\alpha_t h_t(x_i) y_i} \\
&= -\sum_{i:h_t(x_i)=y_i} D_t(i) e^{-\alpha_t} + \sum_{i:h_t(x_i)\neq y_i} D_t(i) e^{\alpha_t} \\
&= -e^{-\alpha_t}\left(\sum_{i:h_t(x_i)=y_i} D_t(i)\right) + e^{\alpha_t}\left(\sum_{i:h_t(x_i)\neq y_i} D_t(i)\right)
\end{aligned}$$

Uočimo da gornje sume odgovaraju vrijednostima $1 - \varepsilon_t$ i ε_t (vidi izraz 3), pa dalje slijedi:

$$\frac{d}{d\alpha_t} Z_t = -e^{-\alpha_t}(1-\varepsilon_t) + e^{\alpha_t}\varepsilon_t$$

U točki minimuma derivacija mora biti 0:

$$-e^{-\alpha_t}(1-\varepsilon_t) + e^{\alpha_t}\varepsilon_t = 0$$

Uvedimo supstituciju $e^{\alpha_t} = x$.

$$\begin{aligned} -\frac{1}{x}(1-\varepsilon_t) + x\varepsilon_t = 0 \\ \frac{\varepsilon_t - 1 + x^2\varepsilon_t}{x} = 0 \\ \varepsilon_t - 1 + x^2\varepsilon_t = 0 \\ x^2 = \frac{1-\varepsilon_t}{\varepsilon_t} \\ x = \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}} \end{aligned}$$

Slijedi

$$\begin{aligned} e^{\alpha_t} &= \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}} \\ \alpha_t &= \ln\sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}} \\ \alpha_t &= \frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t} \end{aligned} \tag{11}$$

Odabir optimalnog slabog klasifikatora

Slabi klasifikator može biti bilo što - npr. perceptron, decizijsko stablo i sl. Nameću se dva pitanja

1. Kako algoritmu za učenje slabog klasifikatora "objasniti" da neki uzorci više vrijede? Kako bi algoritam za učenje slabog klasifikatora trebao upotrijebiti težine koje čuva algoritam AdaBoost?

2. Kako odabrati optimalan slabi klasifikator ako je skup mogućih slabih klasifikatora beskonačan?

Pojasnimo što znači da je skup klasifikatora beskonačan. Neka je zadan određen broj točaka u ravnini, te neka svaka točka pripada jednom od dva razreda. Želimo naučiti razdvajati točke. Upotrijebimo li postupak učenja perceptrona, dobit ćemo decizijsku ravninu koja će u našem slučaju biti svedena na pravac $y = kx + l$. Štoviše, postojat će beskonačno mnogo pravaca koji će dobro odvajati točke. Sa stanovišta algoritma za učenje perceptrona, svi će pravci biti jednako dobri. Dakle, skup mogućih klasifikatora (pravaca) je beskonačan.

Pokušajmo sada drugi pristup: neka naš skup klasifikatora više nije skup svih mogućih perceptrona, nego neka je to skup pravaca $y = i$, $i \in [1, ..., 10]$. Optimalan klasifikator naći ćemo tako da isprobamo sve moguće klasifikatore i odaberemo koji najbolje odvaja točke.

Uz takvo pojašnjenje, evo i odgovora na pitanja:

1. Neki algoritmi za učenje slabih klasifikatora podržavaju korištenje težina uzoraka i u tom im se slučaju težine uzoraka mogu predati kao parametar. Ukoliko ne raspolažemo s takvim algoritmom, možemo na temelju težina stvoriti novi skup za učenje iz kojeg će biti izuzeti uzorci s najmanjim težinama te ponovno pokrenuti postupak učenja. Ukoliko je skup mogućih slabih klasifikatora beskonačan, morat ćemo se poslužiti jednom od spomenutih metoda i nadati se da će algoritam za učenje obaviti dobar posao.

2. Ukoliko je skup mogućih slabih klasifikatora konačan, često je moguće naprosto izračunati pogrešku svakog od mogućih slabih klasifikatora te odabrati onaj koji tu pogrešku minimizira.