# Boosting – Pojačano obučavanje

Pojačavanje pretpostavlja dostupnost bazičnog ili slabog algoritma učenja koji, s obzirom na označene primere za obuku, daje bazični ili slab model. Cilj pojačavanja je poboljšanje performansi slabog algoritma učenja, dok se on tretira kao „crna kutija“ koja se može pozivati više puta, poput potprograma, ali čija se unutrašnjost ne može posmatrati ili menjati. O ovom algoritmu učenja želimo da napravimo samo minimalne pretpostavke. Možda je najmanje što možemo pretpostaviti da slabi klasifikatori nisu sasvim trivijalni u smislu da su njihove stope grešaka barem malo bolje od klasifikatora čije je svako predviđanje nasumično nagađanje. Stoga, slabi klasifikatori mogu biti grubi i umereno netačni, ali ne u potpunosti trivijalni i neinformativni. Pretpostavka da osnovni algoritam produkuje slabu hipotezu koja je bar malo bolja od slučajnog pogađanja na primerima na kojima je trenirna naziva se slabom pretpostavkom učenja (*weak learning assumption*) i ona je od centralnog značaja za proučavanje pojačavanja.

Kao i bilo koji algoritam za obučavanje, algoritam pojačavanja kao ulaz uzima trening primere $(x_1, y_1), \dots, (x_m, y_m)$ gde je svako $x_i$ instanca sa oznakom $y_i$. Za sada ćemo pretpostaviti najjednostavniju varijantu klasifikacije kod koje je svaki primer označen sa +1 ili sa -1.

Jedino sredstvo učenja iz podataka koje algoritam pojačavanja koristi je pozivanje osnovnog algoritma učenja. Međutim, ako se osnovni algoritam jednostavno koristi više puta, uvek sa istim skupom podataka za obuku, ne možemo očekivati da se dogodi nešto zanimljivo; umesto toga, očekujemo da će isti ili gotovo isti osnovni klasifikator biti proizveden iznova i iznova. Ako želi da poboljša osnovni algoritam, algoritam pojačavanja mora na neki način manipulisati podacima koje mu šalje.

Ključna ideja koja stoji iza pojačavanja je odabir skupova za obuku za osnovni algoritam na takav način da on bude primoran da zaključi nešto novo u vezi sa podacima svaki put kada se pozove. To se može postići adaptivnom promenom distribucije trening podataka u zavisnosti od prethodnih grešaka klasifikacije. Izbor skupova za obuku se vrši tako da su performanse prethodnih bazičnih klasifikatora na njima bile vrlo loše - čak i slabije od njihovih redovnih slabih performansi. Ako se ovakav izbor može postići, onda možemo očekivati da osnovni algoritam stori novi model koji je značajno drugačiji od svojih prethodnika.

U nastavku će biti opisan konkretan algoritam pojačavanja AdaBoost, koji uključuje prethodno iznetu ideju i čiji je pseudokod dat ispod. AdaBoost se izvršava kroz ture ili iterativne pozive osnovnog algoritma. Za odabir trening skupova koje će pružiti osnovnom algoritmu, AdaBoost koristi raspodelu trening primera. Raspodela koja se koristi u turi $t$ se obeležava sa $D_t$, a težina koju dodeljuje $i$-tom trening primeru u turi $t$ je $w_t(i)$. Intuitivno, ova težina je mera značaja pravilne klasifikacije primera $i$ u trenutnoj turi. Na početku su sve težine jednake, ali u svakoj turi se povećavaju težine pogrešno klasifikovanih primera, tako da teški primeri uzastopno dobijaju veću težinu, primoravajući osnovni algoritam da usredsredi svoju pažnju na njih.

Dati su trening primeri: $(\mathrm{x}_1, y_1), \dots, (\mathrm{x}_n, y_n)$ gde je $y_i \in \{-1, +1\}$
$H = 0$
$\forall i, w_0(i) = 1/n$
Za $t = 1, \dots, T$
    Obučiti slabi model $h_t$
    $\epsilon_t = \sum_{i:h_t(x_i) \neq y_i} w_t(i)$
    IF $\epsilon_t < \frac{1}{2}$
        $\alpha_t = \frac{1}{2} ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

$H = H + \alpha_t h_t$

$\forall i: w_{t+1}(i) = \frac{1}{Z_t} w_t(i) e^{-\alpha_t y_i h_t(x_i)}$

ELSE return $H$

return $H$

end.

Zadatak osnovnog algoritma je da odredi osnovnu hipotezu $h_t: X \to \{-1, +1\}$ u skladu sa raspodelom težina $D_t$. U skladu sa prethodnom pričom, kvalitet osnovne hipoteze se meri kao suma težina pogrešno klasifikovanih primera iz raspodele $D_t$ (težinska greška):

$$\epsilon_t = P_{x_i \in D_t}(h_t(x_i) \neq y_i) = \sum_{i: h_t(x_i) \neq y_i} w_t(i)$$

Ovde $P_{x_i \in D_t}$ označava verovatnoću slučajnog odabira primera $i$ u skladu sa raspodelom $D_t$. Otuda je težinska greška $\epsilon_t$ verovatnoća da hipoteza $h_t$ pogrešno klasifikuje slučajno (po raspodeli $D_t$) odabrani primer $x_i$. Greška se meri nad istim delom trening skupa nad kojim se hipoteza obučava.

Slabi algoritam učenja teži da proizvede slabi model sa što manjom težinskom greškom. Ne očekujemo da će ova greška biti posebno mala u apsolutnom smislu, već malo bolja od slučajnog odabira i obično daleko od nule.

Ako klasifikator izvršava svako svoje predviđanje potpuno nasumičnim odabirom jedne od oznaka –1 ili +1 sa jednakom verovatnoćom, tada će verovatnoća da on pogrešno klasifikuje bilo koji primer biti tačno 1/2. Zbog toga će greška ovog klasifikatora uvek biti 1/2, bez obzira na podatke na kojima se greška meri. Pretpostavka je otuda da model dobijen slabim algoritmom treba da ima grešku koja je najviše $\frac{1}{2} - \gamma$, gde je $\gamma$ jako mala pozitivna vrednost.

U praksi postoji više načina na koje bazični algoritmi učenja koriste težine $w_t(i)$ koje AdaBoost algoritam dodeljuje primerima za učenje. U nekim slučajevima, ove težine se koriste direktno u bazičnom algoritmu, dok se drugim model obučava na skupu kome nisu dodeljene težine, ali je verovatnoća da primer postane deo skupa za učenje proporcionalna težini primera.

Kada bazični algoritam proizvede bazični klasifikator $h_t$, AdaBoost bira parametar $\alpha_t$ koji određuje značaj hipoteze $h_t$ (veličinu koraka, stopu učenja koju ćemo napraviti pod uticajem ove hipoteze):

$$\alpha_t = \frac{1}{2} ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right)$$

Vrednost parametra $\alpha_t$ raste sa opadanjem greške $\epsilon_t$. Dakle, što se precizniji klasifikator, to je njegov značaj veći. Svaki bazični model nas pomalo pomera ka ciljnom modelu, ali je taj put kojim idemo „slažući“ bazične modele, krivudav ili u obliku izlomljene linije.

Težine primera $w_t(i)$ se dalje menjaju u skladu sa pravilom:

$$w_{t+1}(i) = \frac{w_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t}, & h_t(x_i) = y_i \\ e^{\alpha_t}, & h_t(x_i) \neq y_i \end{cases} = \frac{w_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$$

Gde je $Z_t = \sum_{i=1}^{n} w_t(i) e^{-\alpha_t y_i h_t(x_i)}$ faktor normalizacije.

Nakon mnogih poziva bazičnog algoritma, AdaBoost kombinuje osnovne klasifikatore u jedan kombinovani ili konačni klasifikator H. To se postiže jednostavnim ponderisanim glasanjem osnovnih klasifikatora. Za dati novi primer $x$, kombinovani klasifikator uzima “glasove” (predviđanja) bazičnih

klasifikatora i vraća ponderisanu većinu njihovih predviđanja. Ovde je glas $t$-tog bazičnog klasifikatora $h_t$ ponderisan prethodno odabranim parametrom značajnosti$\alpha_t$. Dobijena formula za model H je:

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

**Primer**

U svrhu ilustracije rada AdaBoost algoritma, u nastavku će biti predstavljen primer sa $m = 10$ trening instanci od kojih je 5 obeleženo sa +1, a 5 sa -1, kao na slici. Pretpostavimo da bazični algoritmi vrše klasifikaciju definisanu vertikalnim ili horizontalnim linijama. Na primer, bazični klasifikator definisan vertikalnom linijom može klasifikovati sve tačke desno od linije kao pozitivne, a sve tačke levo kao negativne.

Može se proveriti da nijedan bazični klasifikator ne klasifikuje tačno više od sedam od deset trening primera, što znači da nijedan nema neponderisanu grešku treninga ispod 30%. U ovom primeru videćemo kako, koristeći takve bazične klasifikatore, AdaBoost je pravi kombinovani klasifikator koji ispravno klasifikuje sve trening primere u samo T = 3 runde.

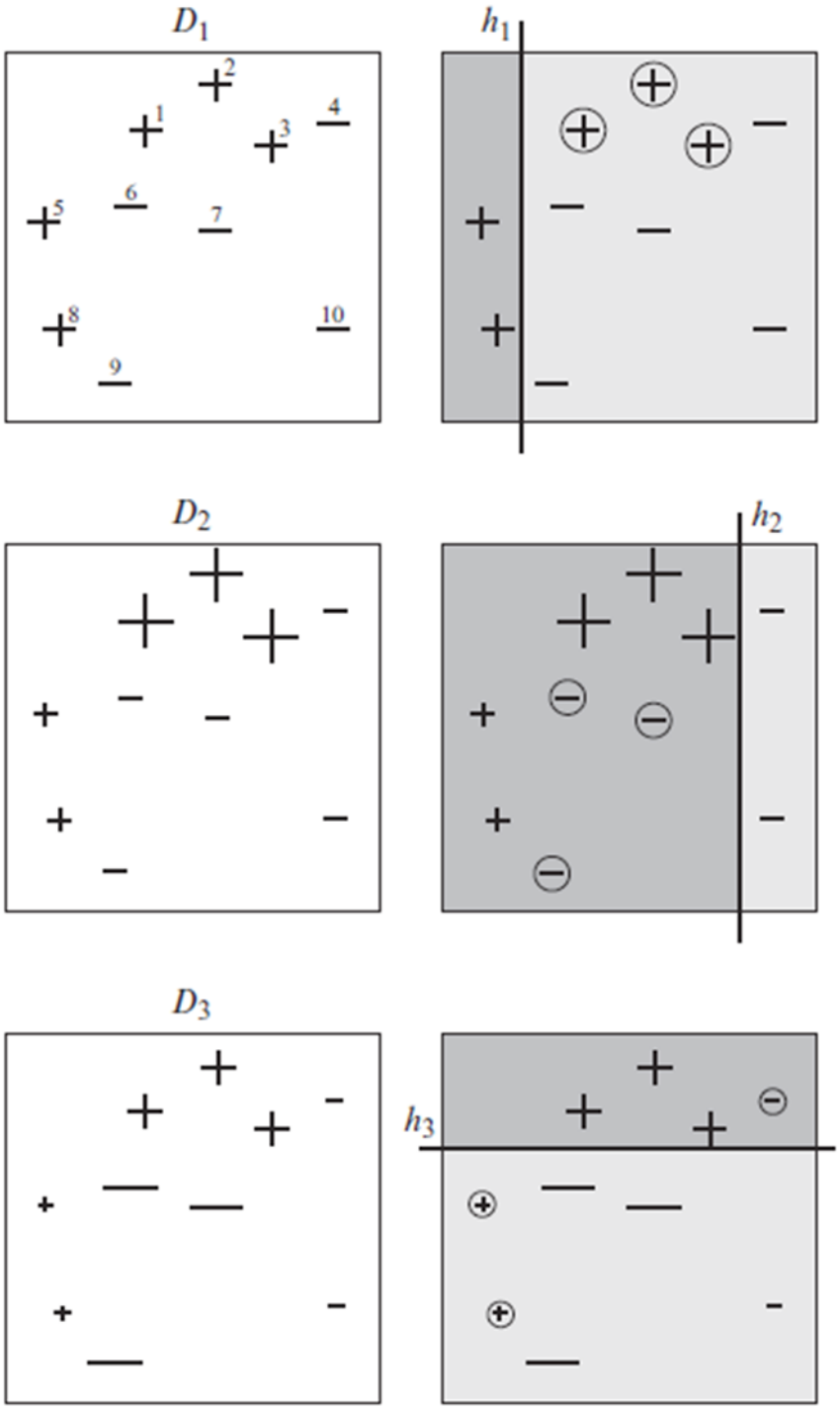
U prvom krugu AdaBoost dodeljuje jednaku težinu svim primerima, kao što je naznačeno na slici crtanjem svih primera u polju označenom D1 u istoj veličini. Bazični algoritam u prvoj rundi produkuje hipotezu označenu sa $h_1$ na slici, koja tačke klasifikuje kao pozitivne ako i samo ako leže levo od ove linije. Ova hipoteza pogrešno klasifikuje tri tačke - tri zaokružene pozitivne tačke - pa je njena greška $\epsilon_1$ jednaka 0.3. Ako značaj hipoteze računamo po formuli:

$$\alpha_t = \frac{1}{2} ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

Onda će hipoteza $h_1$ imati značaj $\alpha_1 \approx 0.42$.

Prilikom formiranja skupa za obučavanje $D_2$ težine primera koji su u prethodnoj rundi pogrešno klasifikovani će biti veća, dok će se težine ostalih primera smanjiti (kao što je prikazano na slici i u tabeli). U drugoj turi obučen je model $h_2$ koji korektno klasifikuje ona tri primera koje je $h_1$ pogrešno klasifikovao, ali zato maši druga tri primera koja imaju težinu od oko 0.07, tako da je greška koju pravi $h_2$ jednaka $\epsilon_2$ ≈0.21, što daje $\alpha_2 \approx 0.65$.

U skupa za obučavanje $D_3$ težine primera koji su u prethodnoj rundi pogrešno klasifikovani će biti veća, dok će se težine ostalih primera smanjiti (kao što je prikazano na slici i u tabeli). Hipoteza $h_3$ dobro klasifikuje sve primere kod kojih su prethodni klasifikatori grešili, ali pogrešno klasifikuje tri primera koji imaju jako malu težinu, pošto su njih prethodni modeli dobro klasifikovali. U trećoj rundi $\epsilon_3$ ≈0.14, a $\alpha_3 \approx 0.92$.




Kombinovani model $H$ je težinska kombinacija hipoteza $h_1, h_2$ i $h_3$, pri čemu svaka od njih ima udeo $\alpha_1, \alpha_2$ odnosno $\alpha_3$. Iako svaki od slabih klasifikatora pogrešno klasifikuje tri od deset primera, kombinovani klasifikator, kao što je prikazano na slici, tačno klasifikuje sve primere treninga. Na primer, klasifikacija negativnog primera u gornjem desnom uglu (instanca #4), koji je klasifikovan kao negativan hipotezama $h_1$ i $h_2$ , ali je po hipotezi $h_3$ pozitivan, je:

$$sgn(-\alpha_1 - \alpha_2 + \alpha_3) = sgn(-0.42 - 0.65 + 0.92) = sgn(-0.15) = -1$$

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $D_1(i)$ | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | $\epsilon_1 = 0.30, \alpha_1 \approx 0.42$ |
| $e^{-\alpha_1 y_i h_1(x_i)}$ | 1.53 | 1.53 | 1.53 | 0.65 | 0.65 | 0.65 | 0.65 | 0.65 | 0.65 | 0.65 | |
| $D_1(i)\ e^{-\alpha_1 y_i h_1(x_i)}$ | 0.15 | 0.15 | 0.15 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | $Z_1 \approx 0.92$ |
| $D_2(i)$ | 0.17 | 0.17 | 0.17 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | $\epsilon_2 \approx 0.21, \alpha_2 \approx 0.65$ |
| $e^{-\alpha_2 y_i h_2(x_i)}$ | 0.52 | 0.52 | 0.52 | 0.52 | 0.52 | 1.91 | 1.91 | 0.52 | 1.91 | 0.52 | |
| $D_2(i)\ e^{-\alpha_2 y_i h_2(x_i)}$ | 0.09 | 0.09 | 0.09 | 0.04 | 0.04 | 0.14 | 0.14 | 0.04 | 0.14 | 0.04 | $Z_2 \approx 0.82$ |
| $D_3(i)$ | 0.11 | 0.11 | 0.11 | 0.05 | 0.05 | 0.17 | 0.17 | 0.05 | 0.17 | 0.05 | $\epsilon_3 \approx 0.14, \alpha_3 \approx 0.92$ |
| $e^{-\alpha_3 y_i h_3(x_i)}$ | 0.40 | 0.40 | 0.40 | 2.52 | 2.52 | 0.40 | 0.40 | 2.52 | 0.40 | 0.40 | |
| $D_3(i)\ e^{-\alpha_3 y_i h_3(x_i)}$ | 0.04 | 0.04 | 0.04 | 0.11 | 0.11 | 0.07 | 0.07 | 0.11 | 0.07 | 0.02 | $Z_3 \approx 0.69$ |

Calculations are shown for the ten examples as numbered in the figure. Examples on which hypothesis $h_t$ makes a mistake are indicated by underlined figures in the rows marked $D_t$.

Vratimo se na $\alpha_t$. Vidimo da se ono određuje po nekakvoj formuli od $\epsilon_t$. Postavlja se pitanje kako se došlo do ove formule. $\alpha$ je određeno tako da minimizuje gubitak koji ćemo napraviti ako na naš kombinovani klasifikator $H$ dodamo bazični klasifikator $h_t$ sa korakom $\alpha_t$. Funkcija gubitka (loss function) je eksponencijalna

$$l(H) = \sum_{i=1}^{n} e^{-y_i[H(x_i)+\alpha_t h_t(x_i)]}$$

Tako da je

$$\alpha_t = argmin_{\alpha_t} \sum_{i=1}^{n} e^{-y_i[H(x_i)+\alpha_t h_t(x_i)]}$$

Kako tražimo $\alpha_t$ za koje je greška minimalna, to je potrebno da nađemo izvod greške po $\alpha_t$ i izjednačimo ga sa nulom:

$$\frac{\partial l(H)}{\partial \alpha_t} = \frac{\partial}{\partial \alpha_t} \sum_{i=1}^{n} e^{-y_i[H(x_i)+\alpha_t h_t(x_i)]} = \frac{\partial}{\partial \alpha_t} \sum_{i=1}^{n} e^{-y_i H(x_i) - y_i \alpha_t h_t(x_i)} =$$

$$\frac{\partial}{\partial \alpha_t} \sum_{i=1}^{n} e^{-y_i H(x_i)}\, e^{-y_i \alpha_t h_t(x_i)} = \sum_{i=1}^{n} e^{-y_i H(x_i)} \frac{\partial}{\partial \alpha_t} \left(e^{-y_i \alpha_t h_t(x_i)}\right) =$$

$$\sum_{i=1}^{n} e^{-y_i H(x_i)} e^{-y_i \alpha_t h_t(x_i)} \frac{\partial}{\partial \alpha_t} \left(-y_i \alpha_t h_t(x_i)\right) =$$

$$\sum_{i=1}^{n} e^{-y_i H(x_i) - y_i \alpha_t h_t(x_i)} \left(-y_i h_t(x_i)\right) = \sum_{i=1}^{n} -y_i h_t(x_i) e^{-y_i[H(x_i)+\alpha_t h_t(x_i)]} = 0$$

Ako hipoteza ima vrednost -1 ili +1, u zavisnosti od toga da li je primer klasifikovan kao pozitivan ili kao negativan, onda će proizvod $y_i h_t(x_i)$ u slučaju kada je hipoteza izvršila tačnu (netačnu) klasifikaciju imati vrednost +1 (-1):

$$-\sum_{i:y_i h_t(x_i)=1} e^{-y_i[H(x_i)+\alpha_t h_t(x_i)]} + \sum_{i:y_i h_t(x_i)=-1} e^{-y_i[H(x_i)+\alpha_t h_t(x_i)]} = 0$$

$$-\sum_{i:y_i h_t(x_i)=1} e^{-y_i H(x_i)-\alpha_t y_i h_t(x_i)} + \sum_{i:y_i h_t(x_i)=-1} e^{-y_i H(x_i)-\alpha_t y_i h_t(x_i)} = 0$$

$$-\sum_{i:y_i h_t(x_i)=1} e^{-y_i H(x_i)}\, e^{-\alpha_t} + \sum_{i:y_i h_t(x_i)=-1} e^{-y_i H(x_i)} e^{\alpha_t} = 0 \qquad (*)$$

Tokom AdaBoost algoritma, težine se trening primerima dodeljuju tako da održavaju uticaj klasifikacije konkretnog trening primera u rundi $t$ na grešku kombinovanog klasifikatora $H$. Formalno:

$$w_t(i) = \frac{1}{Z} e^{-y_i H(x_i)}, Z = \sum_{i=1}^{n} e^{-y_i H(x_i)}$$

Otuda, (*) postaje:

$$-\sum_{i:y_i h_t(x_i)=1} w_t(i)\, e^{-\alpha_t} + \sum_{i:y_i h_t(x_i)=-1} w_t(i) e^{\alpha_t} = 0$$

Ranije smo definisali $\epsilon_t$ kao:

$$\epsilon_t = \sum_{i:h_t(x_i)\neq y_i} w_t(i)$$

U slučajevima kada je predikcija tačna $\sum_{i:h_t(x_i)=y_i} w_t(i)$ će biti jednako $1-\epsilon_t$. Otuda imamo:

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

$$-(1-\epsilon_t)\frac{1}{e^{\alpha_t}} + \epsilon_t e^{\alpha_t} = 0 / \cdot e^{\alpha_t}$$

$$-(1-\epsilon_t) + \epsilon_t e^{2\alpha_t} = 0$$

$$e^{2\alpha_t} = \frac{1-\epsilon_t}{\epsilon_t}$$

$$\alpha_t = \frac{1}{2} ln \frac{1-\epsilon_t}{\epsilon_t}$$