

# یادگیری با نظارت

## یادگیری جمعی

المپیاد هوش مصنوعی - یادگیری ماشین



۱ مقدمه

۲ Bagging

۳ Boosting

## یادگیری جمعی چیست؟

- یادگیری جمعی یا گروهی، یک روش است که در آن چندین یادگیرنده‌ی ضعیف را با هم ترکیب می‌کنیم تا سعی کنیم یک یادگیرنده‌ی قوی بسازیم.
- گروه‌ها چندین فرضیه را با هم ترکیب می‌کنند تا شاید به یک فرضیه‌ی بهتر برسند.
- اگر تمام یادگیرنده‌های ضعیفی که استفاده می‌کنیم از یک نوع باشند، از عبارت Ensemble Learning استفاده می‌کنیم و اگر از انواع مختلف باشند، از عبارت Multiple Classifier استفاده می‌کنیم.
- در ادامه، در مورد دو تکنیک Bagging و Boosting صحبت می‌کنیم.

## معامله‌ی بایاس و واریانس

- یادگیرنده‌های ضعیف عموماً:
  - واریانس کمی دارند و overfit نمی‌کنند.
  - بایاس بالایی دارند و نمی‌توانند توابع پیچیده را یاد بگیرند.
- تکنیک Boosting برای کاهش بایاس استفاده می‌شود.
- تکنیک Bagging برای کاهش واریانس استفاده می‌شود.

مقدمه ۱

Bagging ۲

Boosting ۳

- نام Bagging از عبارت AGGREGATING Bootstrap ساخته شده. کلمه‌ی Bootstrap به "راه‌اندازی اولیه" اشاره دارد و کلمه‌ی Aggregating به معنی گروه‌بندی و تجميع است. یعنی چندین یادگیرنده داریم که هر یک را به روشی مختلف راه‌اندازی می‌کنیم و در نهایت پیش‌بینی‌هایشان را تجميع می‌کنیم!
- از داده‌های تمرین، به طور یکنواخت چندین نمونه می‌گیریم (با جایگزینی. یعنی نمونه‌ها می‌توانند عنصر مشترک داشته باشند). سپس هر یک از یادگیرنده‌ها را روی یکی از این نمونه‌ها تمرین می‌دهیم (به جای این که کل داده‌های تمرین را به آنها بدهیم).
- پیش‌بینی‌های یادگیرنده‌ها را با هم ترکیب می‌کنیم:
  - برای رگرسیون، میانگین پیش‌بینی‌ها را به عنوان پیش‌بینی نهایی بیان می‌کنیم.
  - برای کلاس‌بندی، کلاسی که رای بیشتری بیاورد را به عنوان پیش‌بینی نهایی بیان می‌کنیم.

## الگوریتم Bagging

- فرض کنید دیتاست آموزشی را  $D$  می‌نامیم و  $n$  داده‌ی آموزشی داریم و می‌خواهیم  $T$  یادگیرنده آموزش بدهیم.
- به ازای  $t$  از  $1$  تا  $T$  انجام بده:
  - از  $D$  به تعداد  $N$  بار نمونه می‌گیریم. دقت کنید که این نمونه‌ها ممکن است تکراری باشند (یعنی نمونه‌گیری را با جای‌گذاری انجام می‌دهیم). این  $N$  نمونه را دیتاست جدید  $D_t$  می‌نامیم.
  - مدل  $h_t$  را با دیتاست  $D_t$  آموزش می‌دهیم.
- مرحله‌ی پیش‌بینی (Inference یا همان استنباط):
  - ورودی را به تمام مدل‌ها بده. سپس تمام پیش‌بینی‌ها را تجمیع کن (مثلاً با میانگین‌گیری و یا با رای‌گیری) و نتیجه را به عنوان پیش‌بینی کلی گزارش کن.



## استفاده از Bagging روی درخت‌های تصمیم

- درخت تصمیم یک مدل محبوب برای کلاس‌بندی است زیرا:
  - تفسیرپذیر است. یعنی انسان می‌تواند بفهمد که در آن، یک ورودی چرا و با چه منطقی منجر به یک خروجی می‌شود.
  - در آن، فیچرها (صفات) هم می‌توانند گسسته و هم پیوسته باشند.
  - نسبت به داده‌های پرت مقاوم است.
  - بایاس آن کم است.
- اما مشکل درختان تصمیم این است که واریانس بالایی دارند.
- برای همین کاندید مناسبی برای Bagging هستند!
  - با میانگین‌گیری از تعدادی درخت تصمیم که بایاس کمی دارند، به نتیجه‌ای می‌رسیم که در آن بایاس همچنان کم است اما واریانس هم کاهش یافته.

## جنگل تصادفی

- با کنار هم قرار دادن تعدادی درخت، به جنگل می‌رسیم!
- این مدل حاصل اعمال Bagging روی درختان تصمیم است.
- دوست داریم درختانمان با همدیگر فرق کنند تا جنگلمان متنوع و رنگارنگ باشد! زیرا اگر درختان شبیه هم باشند، دیگر یک درخت با چند درخت چه فرقی دارد؟
- برای کاهش همبستگی بین درختان، دو کار می‌کنیم:
  - همانطور که گفتیم، داده‌های آموزشی آنها با هم متفاوت است.
  - اگر داده‌هایمان  $d$  فیچر داشته باشند، در هر درخت  $m$  فیچر را به طور تصادفی از بین این  $d$  فیچر انتخاب می‌کنیم و درخت را فقط طبق آن  $m$  فیچر آموزش می‌دهیم. معمولاً انتخاب  $m \leq \sqrt{d}$  انتخاب مناسبی است.

مقدمه ۱

Bagging ۲

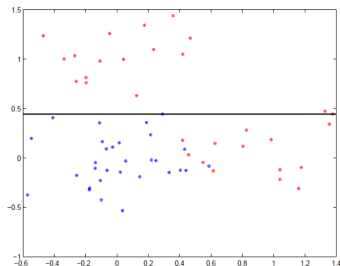
Boosting ۳

## ایده‌ی Boosting

- ما می‌توانیم روش‌های "ضعیف" برای کلاس‌بندی یا رگرسیون را با هم ترکیب کنیم تا به یک روش "قوی" برسیم.
- چند کلاس‌بندی "ضعیف": بیز ساده‌لوحانه، رگرسیون لجیستیک، و درختان تصمیمی که عمق کمی دارند. این مدل‌ها بایاس بالایی دارند و برای همین کاندید مناسبی برای Boosting هستند.
- فضای ورودی برای مدل‌های ضعیف پیچیده است. برای همین فضای ورودی را به چند بخش کوچک‌تر افراز می‌کنیم و هر یک از مدل‌های ضعیف را مسئول یاد گرفتن یکی از بخش‌های کوچک می‌کنیم.
- نتیجه‌ی نهایی حاصل رای‌گیری وزن‌دار بین نتایج مدل‌های کوچک است. بین مدل‌ها، به آنهایی که بهترند وزن بیشتری می‌دهیم.

## کننده‌های تصمیم

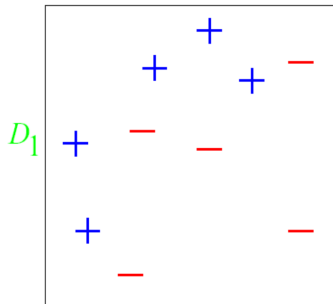
- به یک درخت تصمیم با عمق یک، یک یادگیرنده ضعیف می‌گوییم.
  - این کلاس‌بند صرفاً از یک فیچر برای کلاس‌بندی استفاده می‌کند:
- $$h(\mathbf{x}; \theta) = \text{sign}(w_1 x_k - w_0)$$
- $$\theta = \{k, w_1, w_0\}$$



شکل ۱: یک مثال.

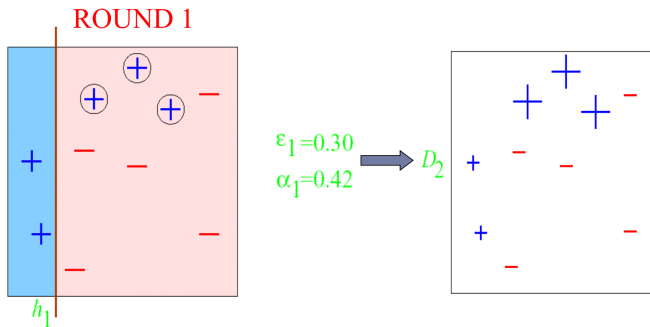
- یکی از محبوب‌ترین روش‌های Boosting، آدا‌بوست است.
- کلاس‌بندهای ضعیف را به طوری متوالی می‌سازیم و هر کلاس‌بند را با توجه به کلاس‌بندهای قبلی، طوری می‌سازیم که بیشترین کمک را به ما بکند.
- به هر نمونه‌ای که توسط کلاس‌بندهای قبلی به غلط کلاس‌بندی شده باشد، وزن بیشتری می‌دهیم تا در کلاس‌بند جدید با احتمال بیشتری درست کلاس‌بندی شود.

## مثال آدابوست (برگرفته از Boosting)



شکل ۲: ابتدا تمام داده‌های آموزشی وزن یکسان دارند.

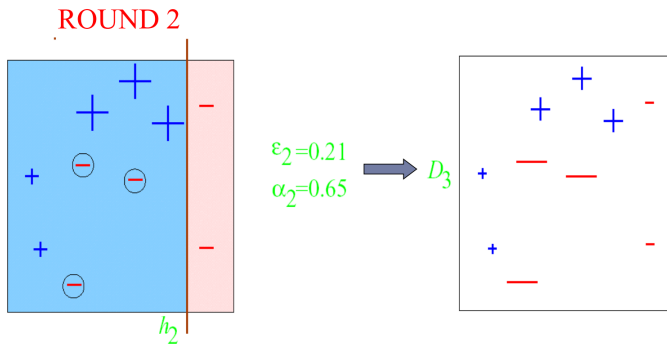
## مثال آدابوست



شکل ۳: پس از آموزش یک یادگیرنده ضعیف روی داده‌ها، وزن داده‌ها را طبق این که آیا درست کلاس‌بندی شده‌اند یا خیر عوض می‌کنیم.



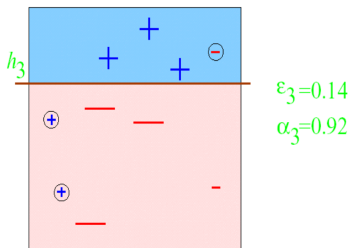
## مثال آدابوست



شکل ۴: مرحله‌ی بعدی.

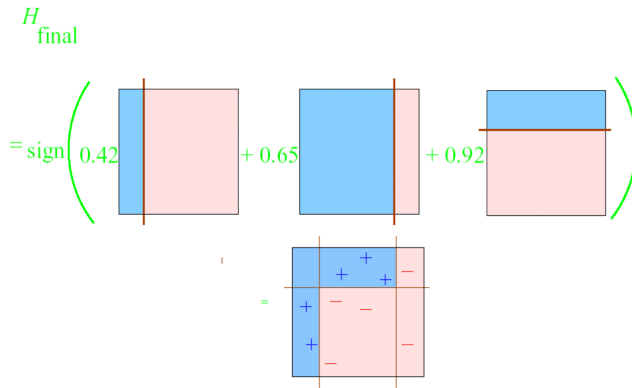
## مثال آدابوست

## ROUND 3



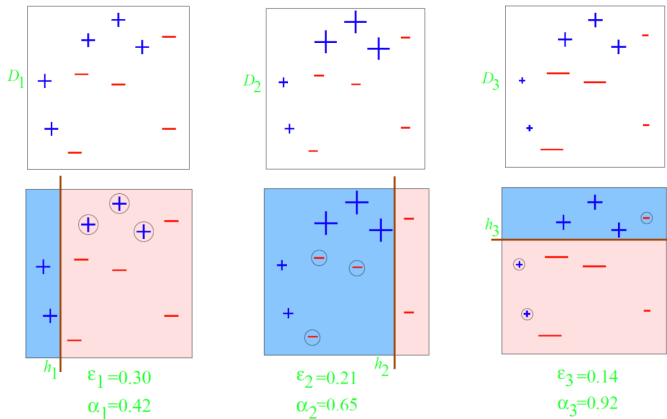
شکل ۵: مرحله‌ی بعدی.

## مثال آدابوست



شکل ۶: در نهایت، کلاس‌بندهای کوچکمان را ترکیب می‌کنیم.

## مثال آدابوست



شکل ۷: کلاس‌بندهای کوچک در یک نگاه.

# الگوریتم AdaBoost، مرحله‌ی آموزش

- ابتدا به تمام داده‌های آموزشی، وزن اولیّی  $\frac{1}{N}$  بده. یعنی  $w_1^{(i)} = \frac{1}{N}$  که در آن  $w_1^{(i)}$  نشان‌دهنده‌ی وزن اولیّی داده‌ی  $i$ ام می‌باشد و  $N$  داده داریم.
- می‌خواهیم  $T$  عدد کلاس‌بند به طور متوالی بسازیم. از  $t = 1$  شروع کن و تا  $t = T$  ادامه بده:
  - طبقه‌بند  $h_t(x)$  را طوری آموزش بده که خطای وزن‌دار را کمینه کند. خطای وزن‌دار کل، حاصل جمع خطای وزن‌دار داده‌هاست. اگر یک داده درست طبقه‌بندی شود، خطای آن ۰ است و اگر اشتباه طبقه‌بندی شود، خطای آن ۱ است که سپس باید ضربدر وزنش شود. به زبان ریاضی:

$$J_t = \sum_{i=1}^N w_t^{(i)} \times I(y^{(i)} \neq h_t(x^{(i)}))$$

- به طبقه‌بند جدیدی که ساختیم، مطابق خطایش یک وزن اختصاص بده (می‌خواهیم طبقه‌بندهای بهتر، وزن بیشتری در تصمیم‌گیری نهایی داشته باشند).
- وزن هر داده‌ای که با  $h_t$  درست طبقه‌بندی می‌شود را کاهش بده و وزن هر داده‌ای که با  $h_t$  اشتباه طبقه‌بندی می‌شود را افزایش بده. با این کار، طبقه‌بند بعدی بیشتر روی داده‌هایی که این طبقه‌بند نتوانسته طبقه‌بندی کند تمرکز می‌کند.

## الگوریتم AdaBoost، مرحله‌ی استنباط

- به هر یک از یادگیرنده‌های ضعیف یا همان طبقه‌بندها یک وزن یادگیری مطابق با این رابطه اختصاص می‌دهیم:

$$\epsilon_m = \frac{\sum_{i=1}^N w_m^{(i)} \times (y^{(i)} \neq h_m(i))}{\sum_{i=1}^N w_m^{(i)}}$$

- به هر یادگیرنده مطابق با رابطه زیر یک وزن اختصاص می‌دهیم

$$\alpha_m = \ln \left( \frac{1 - \epsilon_m}{\epsilon_m} \right)$$

- در نهایت مدل نهایی اینگونه تعریف میشود:

$$H_M(x) = \frac{1}{2} \sum_{m=1}^M \alpha_m h_m(x)$$

$$\hat{y} = \text{sign}(H_M(x))$$