یادگیری ماشین

نیم‌سال اول ۱۴۰۱ ـ ۱۴۰۰

مدرس: دکتر بیگی

دانشکده‌ی مهندسی کامپیوتر

تمرین سری چهارم

موعد تحویل: ۱۳ خرداد

- مهلت ارسال پاسخ تا ساعت ۲۳:۵۹ روز مشخص‌شده است.
- هم‌کاری و هم‌فکری شما در انجام تمرین مانعی ندارد اما پاسخ ارسالی هر کس حتما باید توسط خود او نوشته شده باشد.
- در صورت هم‌فکری و یا استفاده از هر منابع خارج درسی، نام هم‌فکران و آدرس منابع مورد استفاده برای حل سوال مورد نظر را ذکر کنید.
- لطفا تصویری واضح از پاسخ سوالات نظری بارگذاری کنید. در غیر این صورت پاسخ شما تصحیح نخواهد شد.

سوالات نظری (۵۰ نمره)

مسئله‌ی ۱. (۱۵ نمره)

در این مسأله میخواهیم الگوریتم‌هایی بر مبنای Boosting ولی با تابع هدف‌هایی متفاوت از تابع هدف Adaboost تعریف کنیم. فرض کنید که داده‌های آموزش به صورت m نمونه‌ی برچسب‌دار به صورت $\{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$ داده شده‌اند و به ازای هر i می‌دانیم که $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \{-1, +1\}$. همچنین فرض کنید که $\Phi : \mathbb{R} \to \mathbb{R}$ یک تابع اکید صعودی، محدب و مشتق پذیر است و می‌دانیم که به ازای هر $x \geqslant 0$، $\Phi(x) \geqslant 1$ و به ازای هر $x < 0$، $\Phi(x) > 0$ است.

فرض کنید که کلاس فرض $\mathcal{H}$ داده شده است.

آ. نشان دهید که اگر تعداد اعضای مجموعه‌ی $\mathcal{H}$ نامحدود باشد هم می‌توان با انتخاب یک N مناسب و اعضایی مناسب از $\mathcal{H}$ همه‌ی ترکیب خطی‌های $\mathcal{H}$ روی داده‌های آموزش را به صورت محدود نشان داد. به عبارت دیگر اگر یک ترکیب خطی دلخواه از همه‌ی اعضای $\mathcal{H}$ را با f نشان دهیم، نشان دهید که می‌توان یک N مناسب و توابع $\left\{\tilde{h}_1, \cdots, \tilde{h}_N\right\}$ را به گونه‌ای از $\mathcal{H}$ برداشت که به ازای هر $i \in [m]$ ضرایب $\tilde{\beta}_1, \cdots, \tilde{\beta}_N$ وجود داشته باشند به گونه‌ای که

$$\sum_{j=1}^{N} \tilde{\beta}_j \tilde{h}_j(\mathbf{x}_i) = f(\mathbf{x}_i)$$

ب. تابع هزینه‌ی $L(\boldsymbol{\beta}) = \sum_{i=1}^{m} \Phi(-y_i f(\mathbf{x}_i))$ را درنظر بگیرید که در آن f یک ترکیب خطی از دسته‌بندهای اولیه است، یعنی $f(\mathbf{x}) = \sum_{j=1}^{N} \beta_j h_j(\mathbf{x})$. همچنین $\boldsymbol{\beta}$ یک بردار N بعدی است. می‌خواهیم برمبنای تابع هزینه‌ی L یک الگوریتم Boosting جدید طراحی کنیم.

الگوریتم‌های Boosting بر مبنای الگوریتم Coordinate Descent طراحی می‌شوند و عملا باید مقدار بهینه ضرایب $\boldsymbol{\beta}_j$ را با استفاده از این الگوریتم بیابیم. الگوریتم Coordinate Descent یک الگوریتم تکرارشونده است که در آن مقدار بردار $\boldsymbol{\beta}$ در هر مرحله به صورت زیر بروزرسانی می‌شود

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t + \eta_t \mathbf{e}_{k_t}$$

به عبارت دقیقتر، در هر مرحله یک مولفه‌ی مناسب از بردار $\boldsymbol{\beta}$ (که در بالا با k_t نشان داده‌شده) را انتخاب کرده و سپس مقدار آن مولفه را به اندازه‌ی η_t (که در هر تکرار متغیر است) تغییر می‌دهیم. حال به دست آورید که در هر مرحله حرکت در کدام جهت بهتر است؟ یعنی حرکت در کدام جهت سبب کاهش بیشتر تابع هزینه می‌شود؟

پ. توابع زیر را در نظر بگیرید.

- ۱) $\Phi_1(-u) = \mathbf{1}_{u<0}$
- ۲) $\Phi_2(-u) = (1-u)^2$
- ۳) $\Phi_3(-u) = \max\{0, 1-u\}$
- ۴) $\Phi_4(-u) = \log\left(1+e^{-u}\right)$

کدام تابع (ها) شرایط ذکر شده در صورت سوال برای ϕ را برآورده می‌کنند؟

ت. برای تابع (توابعی) که در بخش قبل شناسایی کردید و با توجه به معیاری که در قسمت‌های قبلی به دست آوردید اندازه بهینه گام حرکتی در الگوریتم Coordinate descent را بیابید. (لازم نیست به یک عبارت بسته برای η_t برسید، همینکه به معادله‌ای برسید که جواب آن، مقدار بهینه‌ی η_t باشد کافیست)

ت. سپس شبه کد کلی الگوریتم را بنویسد.

مسئله‌ی ۲. (۷ نمره)

الگوریتم Adaboost را درنظر بگیرید. اگر دقت کنید این الگوریتم هر بار یک دسته‌بند به صورتی که طبق توزیع آن مرحله کمترین خطا را داشته باشد انتخاب می‌شود. اثبات کنید این الگوریتم هیچگاه دو تابع یکسان در دو مرحله متوالی انتخاب نمی‌کند $(h_t \neq h_{t+1})$.

مسئله‌ی ۳. (۸ نمره)

الگوریتم Winnow را برای مساله یادگیری برخط درنظر بگیرید.

$\textsc{Winnow}(\eta)$

$w_1 \leftarrow \mathbf{1}/N$
for $t \leftarrow 1$ **to** T **do**
  $\textsc{Receive}(\mathbf{x}_t)$
  $\widehat{y}_t \leftarrow \operatorname{sgn}(\mathbf{w}_t \cdot \mathbf{x}_t)$
  $\textsc{Receive}(y_t)$
  if $(\widehat{y}_t \neq y_t)$ **then**
    $Z_t \leftarrow \sum_{i=1}^{N} w_{t,i} \exp(\eta y_t x_{t,i})$
    for $i \leftarrow 1$ **to** N **do**
      $w_{t+1,i} \leftarrow \frac{w_{t,i}\exp(\eta y_t x_{t,i})}{Z_t}$
  else $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t$
return $\mathbf{w}_{T+1}$

فرض کنید که $\mathbf{x}_1, \dots, \mathbf{x}_T \in \mathbb{R}^N$ و $\forall t: \|x_t\|_\infty \leqslant 1$ است. همچنین فرض کنید $u \in \mathbb{R}^N$ وجود دارد به طوری که $\|u\|_1 \leqslant 1$ و $u_i \geqslant 0$ برقرار است. اثبات کنید که اگر شرط $\forall t: \ \langle u, x_t \rangle\, y_t \geqslant \delta$ برقرار باشد، حداکثر تعداد خطاهای انجام شده توسط این الگوریتم $\frac{2\ln N}{\delta^2}$ است.

مسئله‌ی ۴. (۱۰ نمره)

فرض کنید $\mathcal{H}$ یک کلاس فرضیات باشد. تعریف کنید

$$S=\{(\mathbf{x}_1, h^*(\mathbf{x}_1)), (\mathbf{x}_r, h^*(\mathbf{x}_r)), \ldots, (\mathbf{x}_T, h^*(\mathbf{x}_T))\}$$

که T یک عدد طبیعی و $h^* \in \mathcal{H}$ است. خطاهای الگوریتم A روی مجموعه‌ی S را $M_A(S)$ و سوپریمم $M_A(S)$ روی تمام دنباله‌ها را $M_A(\mathcal{H})$ می‌نامیم.

می‌خواهیم مفهوم شقه‌شدن (Shattering) را در چارچوب درخت تعریف کنیم. یک درخت دودویی شقه شده به عمق d توسط $\mathcal{H}$ دنباله‌ای از نمونه‌های $\mathbf{v}_1, \ldots, \mathbf{v}_{2^d-1} \in \mathcal{X}$ است به طوری که برای هر برچسب‌گذاری $y_1, \ldots, y_d \in \{0,1\}^d$ تابعی مثل $h \in \mathcal{H}$ وجود داشته باشد که برای هر t داشته باشیم $h(\mathbf{v}_{i_t}) = y_t$ و i_tها به صورت $i_{t+1} = 2i_t + y_t$ تعریف می‌شوند (به معنا و شهود درختی این تعریف فکر کنید).

بعد لیتل‌استون (Ldim) مجموعه‌ی توابع $\mathcal{H}$ برابر عمق عمیق‌ترین درخت شقه شده توسط $\mathcal{H}$ است.

آ. نشان دهید که برای هر الگوریتم برخط A داریم $M_A(\mathcal{H}) \geqslant \mathrm{Ldim}(\mathcal{H})$.

ب. نشان دهید که $\mathrm{VCdim}(\mathcal{H}) \leqslant \mathrm{Lim}(\mathcal{H})$.

پ. نشان دهید که الگوریتمی وجود دارد که در آن $M_A(\mathcal{H}) \leqslant \mathrm{Ldim}(\mathcal{H})$ است.

مسئله‌ی ۵. (۱۰ نمره)

فرض کنید یک مجموعه‌ی $\mathcal{X}$ از نمونه‌های ممکن داریم. یک تابع رده‌بندی به شکل زیر در نظر بگیرید

$$f: \mathcal{X} \times \mathcal{X} \to \{-1, +1\}$$

اگر $f(x, x') = +1$ یعنی x بر x' ارجحیت دارد و اگر $f(x, x') = -1$ یعنی x' بر x ارجحیت دارد. فرض کنید (x, x') از توزیع مشترک $D_{\mathcal{X}\times\mathcal{X}}$ تولید می‌شوند. همچنین فرض کنید که تعداد m نمونه به صورت

$$S=\{(x_1, x'_1, y_1), (x_2, x'_2, y_2), \ldots, (x_m.x'_m, y_m)\}$$

داشته باشیم که به صورت iid نمونه برداری شده‌اند.

هدف پیدا کردن یک تابع $h \in H$ است به طوری که اگر $h(x) > h(x')$ یعنی x بر x' ارجحیت دارد!

آ. خیلی ساده توضیح دهید که چرا خطای تجربی را می‌توان به صورت زیر نوشت.

$$\hat{\mathrm{Err}}(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\left(f(x_i, x'_i)\left(h(x_i) - h(x'_i)\right) < 0\right)$$

و سپس برای یک تابع $h \in H$ خطای عمومی $\mathrm{Err}(h)$ را برحسب D و f بنویسید.

ب. برای یک تابع $h \in H$ و یک عدد $\rho > 0$ تعریف می‌کنیم

$$\hat{\mathrm{Err}}_\rho(h) \triangleq \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\left(f(x_i, x'_i)\left(h(x_i) - h(x'_i)\right) \leqslant \rho\right)$$

برای یک عدد $\delta \in (0,1)$ خطای عمومی $\mathrm{Err}(h)$ را با تابعی از $\hat{\mathrm{Err}}_\rho(h)$، $\mathfrak{R}_{D_1}(H)$، $\mathfrak{R}_{D_2}(H)$، تعداد نمونه‌ها و δ و با احتمال $1-\delta$ باند بزنید که در آن $\mathfrak{R}_{D_1}(H)$ و $\mathfrak{R}_{D_2}(H)$ پیچیدگی متوسط رادماخر H روی توزیع حاشیه‌ای D روی x و x' است.

موفق باشید :)