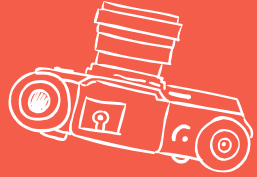
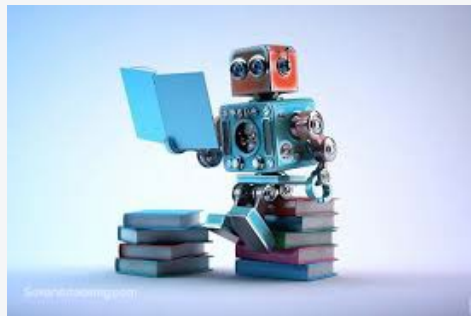


به نام خدا



یادگیری ماشین





یادگیری ماشین

آرش عبدی هجراندوست

arash.abdi.hejrandoost@gmail.com

دانشگاه علم و صنعت

دانشکده مهندسی کامپیوتر

نیم سال اول ۱۴۰۱-۱۴۰۲

یادگیری برخط – Online Learning

✗ فرض iid (independent and identically distributed)

○ اگر آینده بی ربط به گذشته باشد، چه می‌توان یادگرفت؟

○ اما فرض استقلال آینده از گذشته هم فرض سفتی است!

✗ مثال:

○ سیستم شناسایی چهره

■ استقلال؟

■ توزیع یکسان؟

○ سیستم تشخیص حملات تحت شبکه

■ استقلال؟

■ توزیع یکسان؟

✗ راه حل: عنوان اسلاید!

یادگیری برخط

✗ استراتژی کلی:

- نمونه ای را ببین
- فروچی تولید کن
- جواب درست را ببین
- و متنبه شو !! (اصلاح کن)

✗ اگر محیط رقابتی باشد:

- دائما فریب می‌خوریم؟

✗ نزول در راستای گرادیان تصادفی (Stochastic Gradient Descent)

- تداوم مرحله آموزش به زمان آزمایش محصول

الگوریتم اکثریت وزن دار تصادفی (Randomized Weighted Majority Algorithm)

- ✗ چند فرد خبره داریم که باید بین نظرات آنان تصمیم بگیریم.
- ✗ می‌توان به هر یک بر اساس تاریخچه عملکردش، (در یادگیری برخط) وزن و اهمیت داد.
- ✗ به جای افراد خبره، می‌توان از روش‌های یادگیری ماشین مجزا که قبلاً به نوعی آموزش دیده‌اند استفاده کرد.

Initialize a set of weights $\{w_1, \dots, w_K\}$ all to 1.

for each problem to be solved do

1. Receive the predictions $\{\hat{y}_1, \dots, \hat{y}_K\}$ from the experts.
2. Randomly choose an expert k^* in proportion to its weight: $P(k) = w_k$
3. yield \hat{y}_{k^*} as the answer to this problem.
4. Receive the correct answer y .
5. For each expert k such that $\hat{y}_k \neq y$, update $w_k \leftarrow \beta w_k$
6. Normalize the weights so that $\sum_k w_k = 1$.

B بین صفر تا یک است

✗ متناسب با میزان دقت افراد، شانس انتخاب داده می‌شود.
✗ چرا در هر لحظه فقط بهترین فرد انتخاب نشود و بقیه هم شانس داشته باشند؟

ارزیابی

- ✗ ارزیابی بر حسب میزان ضرر (regret)
- تعداد خطای بیشتر نسبت به بهترین خبره
- بهترین خبره در پایان چرخه و در همه نمونه ها (و نه بهترین تا این لحظه)
- ✗ میتوان حد بالای خطای الگوریتم را تعیین کرد:

$$M < \frac{M^* \ln(1/\beta) + \ln K}{1 - \beta}.$$

- M : تعداد خطای الگوریتم
- M^* : تعداد خطای بهترین خبره (در پایان کار)

$$M < \frac{M^* \ln(1/\beta) + \ln K}{1 - \beta}.$$

$M < 1.39M^* + 4.6 \leftarrow B=0.5$ و $K=10$ ✗

$M < 1.15M^* + 9.2 \leftarrow B=0.75$ و $K=10$ ✗

✗ اگر B نزدیک به صفر انتخاب شود:

○ نوسان زیاد بین (مجلس) خبرگان

○ مگر آنکه یکی، خیلی بهتر از بقیه باشد

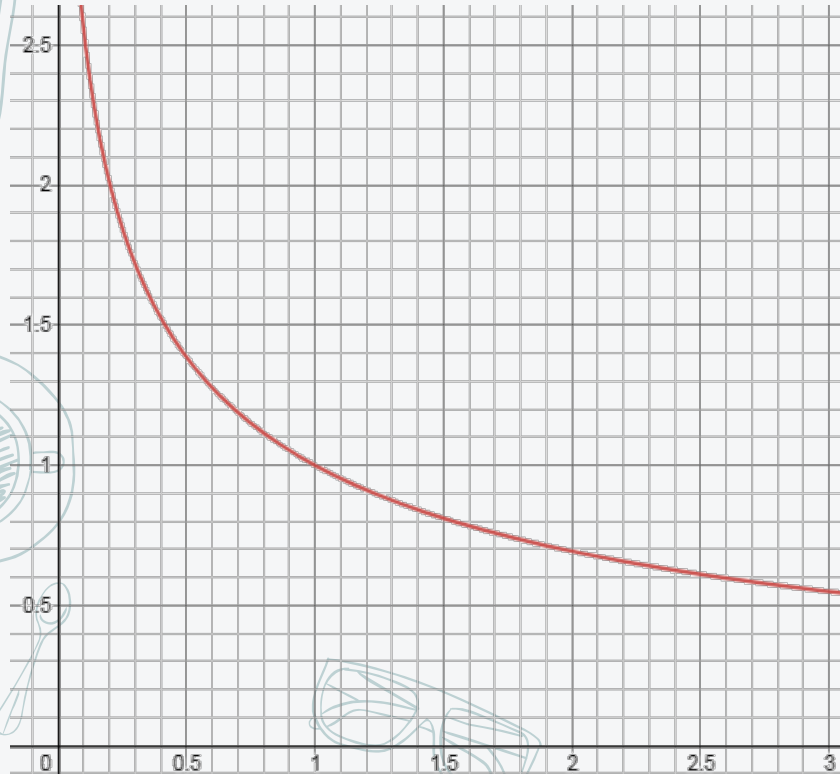
■ که بعید است

✗ اگر B نزدیک به ۱ باشد:

○ تخیرات آرام است

○ هزینه زیادی در اوایل الگوریتم پرداخت میشود (با اعتماد به خبره های ناخوب)

$$M < \frac{M^* \ln(1/\beta) + \ln K}{1 - \beta}$$



✗ اگر تعداد خطا به سمت بی نهایت میل کند و فقط ضریب M^* منظور شود:

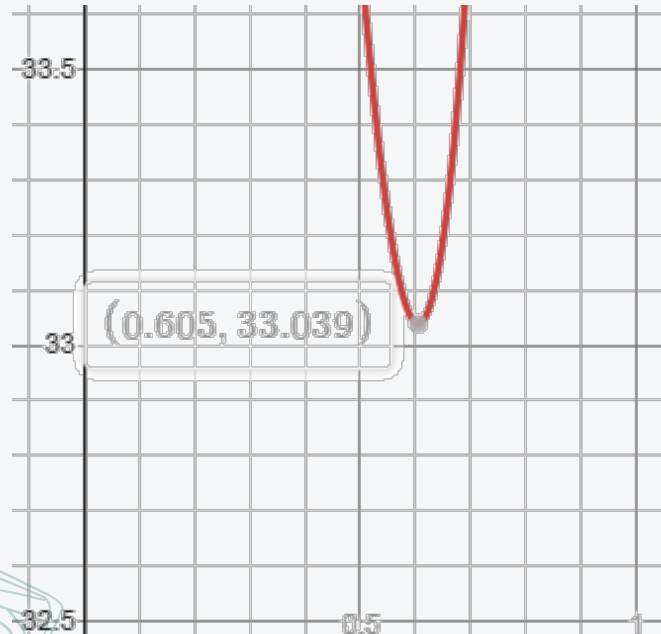
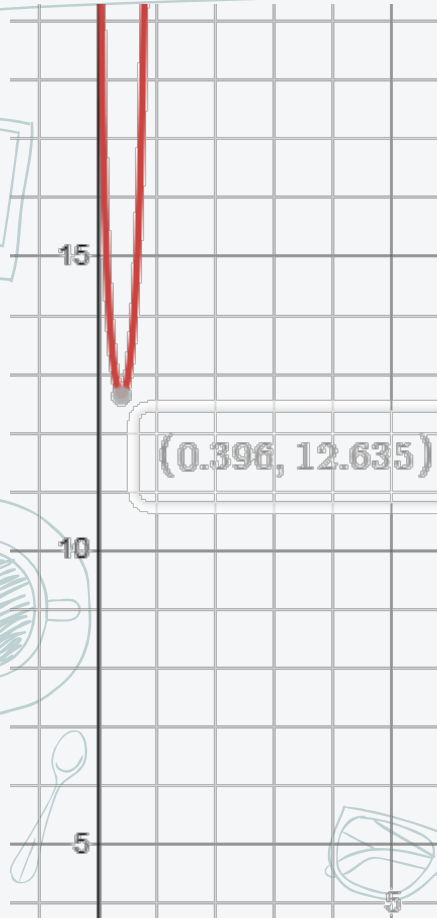
- B بین صفر تا یک است
- B کمتر ← مدبالای خطای کوچکتر
- مطلوب
- بهترین مقدار $B = 1$
- اما مخرج کسر صفر شده و
- ترم دوم $(\ln k / 1 - B)$ بی نهایت می شود!
- ✗ به گونه دیگری باید تحلیل کرد ...

$$M < \frac{M^* \ln(1/\beta) + \ln K}{1 - \beta}.$$

برای هر M^* و K می‌توان مقدار مناسبی برای B یافت:

$M^*=5$ و $K=20$ ○

$M^*=20$ و $K=20$ ○



✗ یادگیری برخط وقتی تخییرات داده ها در طول زمان زیاد است توصیه می شود

○ اگر تخییرات شدید نباشد، امکان یادگیری آفلاین وجود دارد.

✗ اما اگر حجم داده ها خیلی زیاد باشد، حتی با تخییرات آرام و تدریجی در داده ها هم یادگیری آنلاین توصیه می شود:

○ آموزش مجدد مدل زمان بر است.

✗ اغلب مدل های یادگیری ماشین که مبتنی بر تابع خطا هستند، نسخه یادگیری برخط نیز دارند.

○ به روز رسانی وزن/پارامترهای مدل به صورت تدریجی و افزایشی

متودولوژی یا مهندسی توسعه سیستم یادگیری ماشین

✗ نسبت برنامه نویسی با مهندسی نرم افزار

○ متودولوژی توسعه نرم افزار

○ تسهیل و بالابردن نرخ موفقیت پروژه ها

○ توصیه های عمومی

✗ در یادگیری ماشین، در ابتدای راه هستیم

✗ هنوز متودولوژی ها چندان پخته و مجرب نیستند.

مساله چیست؟

✗ همه چیز از صورت مساله آغاز می‌شود.

✗ لازم نیست همه جا یادگیری ماشین در پروژه ها چپانده شود !!

○ سیستم پیشنهاد دهنده در وب سایت بر اساس لایک یا یادگیری ماشین؟

○ چراغ راهنمایی هوشمند یا چراغ راهنمایی زماندار؟

✗ سوال اول: چه مساله ای قرار است برای کاربر نهایی حل شود؟

✗ سوال دوم: کدام بخش از حل مساله با یادگیری ماشین حل می‌شود؟

○ بعد می‌توان برای هدف منظور (بر وزن مریض منظور!)، تابع هدف تعریف کرد و الخ!

✗ صورت مساله از صنعت آغاز می‌شود و نه از علم

○ و نه از چیزهایی که بلدیم، یا علاقه داریم، یا درسش را خوانده‌ایم، یا بقیه

استفاده کرده‌اند، یا کلاس دارد(!) یا ...

✗ مساله به شما می‌گوید که:

- یادگیری بانظارت داریم یا نیمه نظارتی یا تقویتی
- نیاز به دیتاست بیشتر داریم یا ویژگی جدید یا روش جدید یادگیری
- کدام روش یادگیری احتمالا مفیدتر است
- چه دقتی مورد نیاز است
- برای رسیدن به چه دقتی چه میزان هزینه محقول است
- و ...

درباره دیتا (دادگان)

برچسب! ✗

○ نویزی یا دروغین

○ فقدان برچسب

Weakly Supervised Learning ✗

○ برچسب هایی با نویز، نادقیق یا نامفهوم، غیر قابل اعتماد

مدیریت دیتا (دادگان)

✗ جمع آوری و تولید دادگان

✗ جمع‌سپاری تولید دادگان

○ کپچا

✗ جمع آوری توسط مشتریان

○ مسیریاب‌ها

✗ استفاده از دادگان سایر زمینه‌ها در مساله خود

○ یادگیری انتقالی - Transfer Learning

○ استفاده از دادگان‌های عمومی برای مساله‌ای که دادگان کمی دارد

■ یا استفاده از مدل‌های آموزش دیده شده روی سایر دادگان‌ها

○ اضافه کردن دادگان اندک خود به دادگان قرضی و آموزش مجدد

○ تنظیم وزن‌های لایه آخر شبکه عمیق

✗ توجه به منشأ و هدف تولید دادگان (که شاید با نیاز ما منطبق نباشد)

اهمیت دیتا

✗ داشتن **چرخه تامین** و **نگهداری** دادگانی قابل **اعتماد**، **کافی**، **امن** و **درست**، بسیار **هیاتی تر** است از جزئیات دقیق روش یادگیری ماشین

✗ درباره دیتا بپرسید:

- برای هدف ما درست است؟
- کافی است؟
- حالت های مورد نیاز را پوشش می دهد؟
- داده بی ربط/قابل حذف در آن نیست؟
- دقیقاً هدف ما را پوشش می دهد یا برای نیازی عمومی تر تولید شده؟
- آیا مقادیر خالی (Missing Values) در دادگان داریم؟ چه کنیم؟
- آیا داده پرت (Outlier) داریم؟ چقدر موثر است؟ چه کنیم؟

تعداد و اندازه دادگان

✗ منمنی یادگیری

✗ ایده های موردی و خلاقانه برای تعداد داده مورد نیاز:

- چند میلیون برای مسائل پیچیده
- چند هزار برای مسائل ساده تر
- برای هر دسته، چندصد یا چند هزار
- ۱۰ برابر تعداد پارامترهای مدل
- ۱۰ برابر ابعاد مساله
- نمونه های بیشتر برای یادگیری غیر خطی نسبت یا یادگیری خطی
- نمونه های بیشتر اگر دقت بالاتر لازم است
- و ...

تقویت دادگان – Data Augmentation

✗ در تصویر:

○ چرخاندن، انتقال، برش، تغییر اندازه، تغییر شدت روشنایی، افزودن نویز و ...

✗ در غیر تصویر (سیگنال):

○ نویز، شیفت، ترکیب و ...

■ متناسب با مساله

دسته های نامتوازن – Unbalanced Classes

✗ ممکن است الگوریتمی که همیشه جواب ثابت می‌دهد، دقتش ۹۹٪ باشد!

✗ UnderSampling:

○ همه نمونه‌های دسته اکثریت را انتخاب نکن

✗ OverSampling:

○ نمونه‌های دسته اقلیت را چند بار انتخاب کن

✗ تابع خطای وزن‌دار:

○ جریمه بیشتری برای خطاهای یک دسته در نظر بگیر

داده‌های پرت – outlier

✗ نمونه تشخیص

✗ میزان تاثیر در مدل یادگیری

○ مثلا رگرسیون خطی

○ درخت تصمیم

■ Random forest

■ Gradient boosting

○ سایر دوستان

مهندسی ویژگی (Feature)

✗ گسسته سازی (Quantization)

✗ نرمال سازی

○ طول واحد

○ میانگین و واریانس

○ ...

✗ تبدیل داده‌های چندمالتی به چند ویژگی بولین

○ One-hot encoding

✗ ویژگی‌های خاص مساله

✗ “At the end of the day, some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used.”, Pedro Domingos

آنالیز، شهود و تصویرسازی داده

✗ مشاهده داده ها

✗ اطلاعات آماری از داده ها

○ هیستوگرام

✗ درک و دریافتی از نحوه توزیع داده ها، داده های پرت، تعداد خوشه ها و ...

✗ رسیدن به پیش فرض‌هایی از مدل یادگیری مناسب برای داده ها

✗ خوشه بندی داده ها

○ بررسی مراکز خوشه‌ها

○ بررسی داده‌های پرت

○ معیار فاصله برای خوشه بندی؟

○ نحوه انتخاب معیار فاصله

○ تکرار چرخه

T-distributed stochastic neighbor embedding (t-SNE)

✗ کاهش ابعاد داده (برای نمایش تصویری)

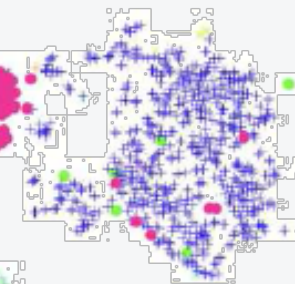
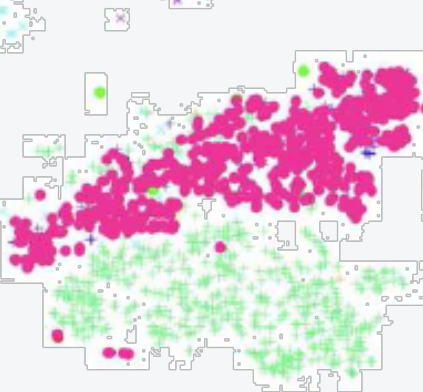
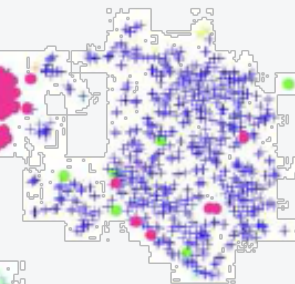
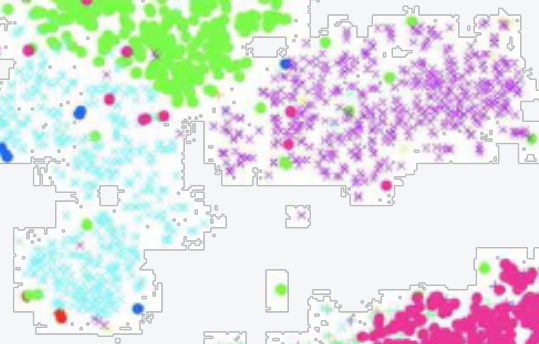
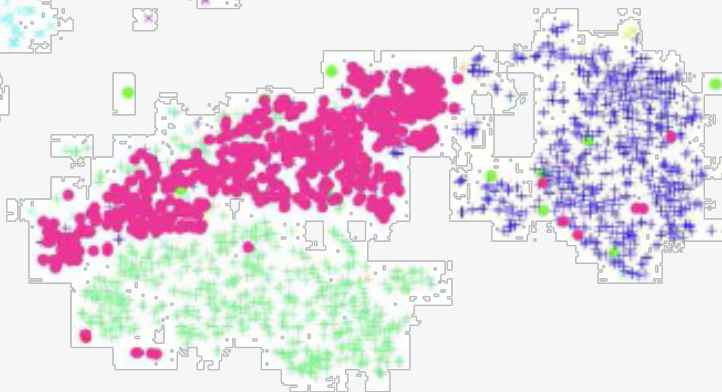
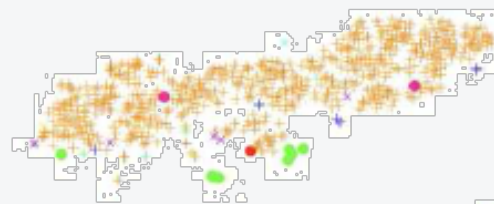
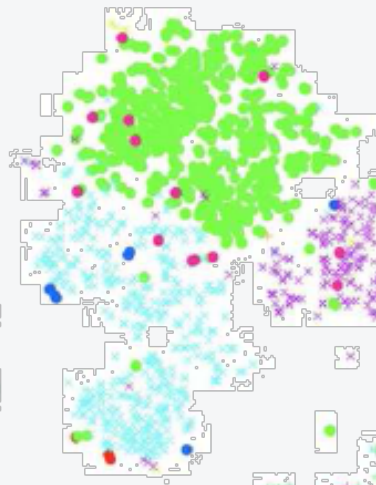
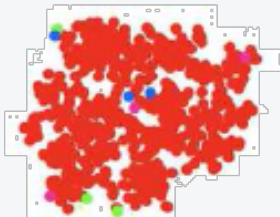
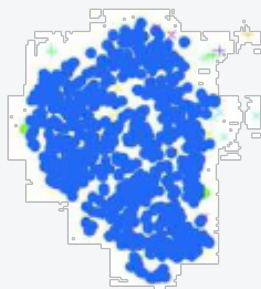
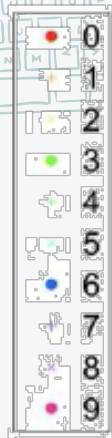
○ PCA و ...

✗ نگاشت داده‌های به فضای با ابعاد کم به طوری که شباهت و تفاوت
جفت نمونه‌ها حفظ شود

○ t-SNE

t-SNE

MNIST ✗



مروری بر چند مفهوم اطلاعاتی

× آنتروپی

- پراکندگی
- میزان اطلاعات موجود در یک متغیر تصادفی
- که با کشف مقدار آن، به دست می‌آید

× همبستگی اطلاعات – mutual information

- بین دو متغیر تصادفی
- توزیع توأم
- استقلال

$$I(X; Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} P_{(X,Y)}(x, y) \log \left(\frac{P_{(X,Y)}(x, y)}{P_X(x) P_Y(y)} \right)$$

$$I(X; Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} P_{(X,Y)}(x, y) \log \left(\frac{P_{(X,Y)}(x, y)}{P_X(x) P_Y(y)} \right)$$

✗ اگر دو متغیر مستقل باشند:

$$MI=0 \quad \circ$$

✗ اگر وابستگی کامل باشد (در حالت خاص هر دو متغیر یکی باشند)

$$MI=H(x) \quad \circ$$

Entropy ■

✗ MI: میزان کاهش عدم قطعیت در یک متغیر با مشاهده متغیر دیگر

○ میزان اطلاعات مشترک

○ مثلاً اگر مستقل باشند یا یکی باشند؟

❌ روابط معادل:

$$\begin{aligned}
 I(X; Y) &\equiv H(X) - H(X | Y) \\
 &\equiv H(Y) - H(Y | X) \\
 &\equiv H(X) + H(Y) - H(X, Y) \\
 &\equiv H(X, Y) - H(X | Y) - H(Y | X)
 \end{aligned}$$

❌ سطر اول : تعریف دستاورد اطلاعات یا information gain

❌ در درخت تصمیم:

○ X: دسته فروجی

○ Y: ویژگی

Gain: میزان اطلاعات به دست آمده در باره فروجی دسته با مشاهده یک ویژگی
 ■ میزان کاهش عدم قطعیت در مورد وضعیت دسته با مشاهده یک فروجی (MI)

Kullback-Leibler divergence

✗ $D_{KL}(P||Q)$: آنتروپی نسبی

✗ میزان تفاوت توزیع احتمال Q نسبت به توزیع احتمال P

✗ متقارن نیست (از Q به P)

$$D_{KL}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right)$$

✗ مقدار $D_{KL}(P||Q)$ نامنفی است.

✗ اگر $p=q$ باشد: $D_{KL}(P||Q)=0$

✗ در حالت کلی مقدار بیشینه (حد بالا) ندارد.



ادامه بحث t-SNE

✗ انتقال داده‌های با ابعاد بالا به فضایی با ۲-۳ بعد
○ حفظ شباهت و تفاوت بین داده‌ها

✗ سه گام کلی:

- تعریف یک توزیع احتمال بین جفت نمونه‌ها در فضای اولیه
- جفت‌های شبیه/نزدیک به هم، احتمال بیشتر می‌گیرند و برعکس
- تعریف یک توزیع احتمال مشابه بین جفت نمونه‌ها در فضای ثانویه
- کمینه کردن D_{KL} بین دو توزیع فوق (متغیر: مختصات نقاط در فضای ثانویه)

- ✗ t-SNE معمولاً نمایشی خوشه‌بندی شده در فضای ثانویه ارائه می‌دهد
- حتی اگر داده‌ها چندان هم خوشه خوشه نباشند!
 - نقش پارامترها در t-SNE مهم است.
 - ممکن است نیاز باشد بین نمایش خروجی (درک از فضای داده‌ها) و تنظیم پارامترهای t-SNE چند بار رفت و برگشت داشت.

با داشتن N نمونه x_1, \dots, x_N  توزیع احتمالاتی در فضای اولیه: 

For $i \neq j$, define


$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

set $p_{i|i} = 0$.

$$\sum_j p_{j|i} = 1 \text{ for all } i.$$

x_i : نمونه مرکزی 

تعریف شباهت داده x_j با x_i : 

احتمال آنکه x_i داده x_j را به عنوان همسایه انتخاب کند 

با فرض آنکه همسایه‌ها با احتمالی متناسب با توزیع گوسی با مرکزیت x_i انتخاب شوند 

For $i \neq j$, define

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

set $p_{i|i} = 0$.

$$\sum_j p_{j|i} = 1 \text{ for all } i.$$

مقادیر σ_i متناسب با چگالی
توزیع داده ها انتخاب می شوند.

○ در نوامی چگال تر، مقادیر کوچکتر انتخاب میشود و برعکس
چرا؟ ■

می توان به جای فاصله اقلیدسی در رابطه فوق معیار فاصله دیگری جاگزین کرد.

فضای ثانویه: $(y_i \in R^d, d = 2, \text{or } 3) y_1, \dots, y_N$ ✗

For $i \neq j$, define

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}$$

set $q_{ii} = 0$

برای یافتن y ها: کمینه کردن رابطه زیر با روش gradient descent ✗

$$\text{KL}(P \parallel Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

ادامه متودولوژی یادگیری ماشین: انتخاب مدل

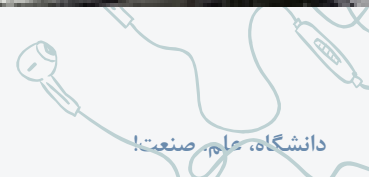
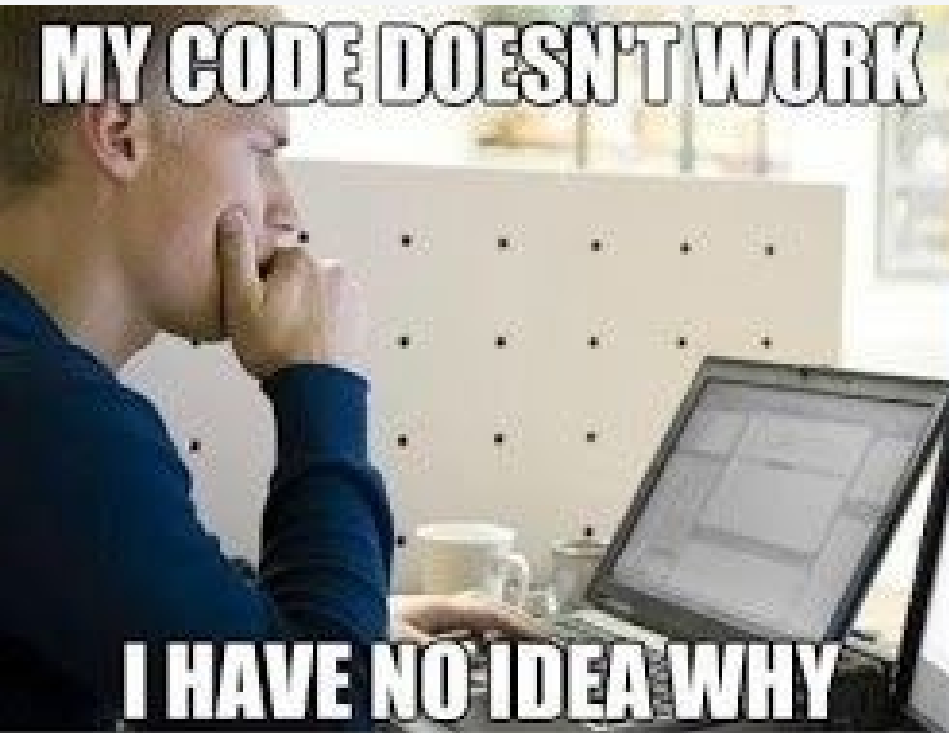
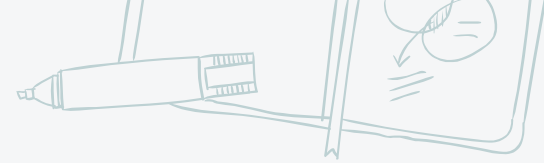
✗ فرآیند یادگیری ماشین

- جمع آوری داده‌ها
- انتخاب مدل
- آموزش و اعتبارسنجی
- تنظیم پارامترها
- ... و

✗ نیاز به debugging دارد!

چرا جواب نداد !!؟

✗



انتخاب مدل

Random Forest ✗

وقتی ویژگی‌های متعدد طراحی شده و احتمالا خیلی از آنها بی ربط هستند ○

روشهای بدون پارامتر: ✗

وقتی داده‌های زیاد داریم و دانش اولیه‌ای نداریم ○

هزینه اجرای بالاتر ○

ماشین بردار پشتیبان: ✗

اگر حجم داده‌ها خیلی زیاد نباشد ○

شبکه‌های عمیق: ✗

مسائل شناسایی الگو (تصویر، صوت و...) ○

اهمیت خطا



✗ خطای دسته‌بندی:

○ تعداد دسته بندی غلط به کل نمونه‌ها؟

✗ FP, FN

✗ دسته بندی غلط ایمیل (اسپم) : FP

✗ تشخیص غلط نداشتن سرطان : FN

○ در مقایسه با تشخیص غلط داشتن سرطان

✗ نمودار ROC (receiver operating characteristic)

○ نمایش TP برای هر FP (ناشی از تغییر پارامترها)

○ (area under ROC) AUC

✗ Confusion Matrix

فقط تابع خطا اهمیت ندارد

- × هزینه آموزش
- × هزینه آزمایش
- × مصرف باتری (!!)
- × سرعت اجرا
- × سادگی آموزش مجدد
- × و ...

× موارد فوق نیز در تعیین شکست/پیروزی یک پروژه یادگیری ماشین نقش دارند

مراقبت و نگهداری از سیستم زیر بار

✗ تست سیستم در محیط واقعی

○ محیط واقعی

■ ورودی نویزی (نویز واقعی، نه شبیه سازی شده)

■ Outlier

■ ورودی غیر قابل پیش بینی

○ هندل کردن خطا

■ سریع، کم هزینه، امکان پذیر!!

○ امکان گزارش خطا (بازخورد)

✗ مانیتورینگ عملکرد

✗ پاسخگویی:

○ مسئولیت خطا با چه کسی است؟

■ اگر ایمیل مهمی اسپم شناخته شد؟

■ اگر ابزار هویت خودکاری باعث ورود افراد غریبه به سازمان شد؟

تفسیرپذیری عملکرد سامانه

- ✗ فروچی مدل برای یک ورودی مشخص به چه علت است؟
- ✗ اگر ورودی چه تغییری بکند، فروچی چه تغییری خواهد داشت؟

- درخت تصمیم
- شبکه عمیق
- سایر مدل‌ها

- ✗ هواپیمای هوشمندی که نحوه عملکرد سامانه هوشمندش قابل تفسیر است قابل اعتمادتر است از ...

ایجاد توازن بین توجه به داده جدید و قدیم

- ✗ مدلی مجرب (امتحان پس داده) با داده‌های کمی قدیمی تر
- ✗ مدلی fresh با داده‌های جدید ولی در محیط واقعی تست نشده
- ✗ کدام بهتر است؟
 - ترکیب؟
 - انتشار تدریجی
 - انتشار نسخه آزمایشی (آلفا و بتا)



با تشکر

