Date: Sub:

انواع داده
- کمی (Quantitative): قابل شمارش و اندازه‌گیری به صورت عدد بیان می‌شود
- کیفی (Qualitative):
  - Categorical دسته‌های مختلف قرارگیرند معمولا فاصله و ترتیب ندارند
  - اگر ترتیب داشته باشند (Ordinal)
  - اگر ترتیب نداشته باشند (صوری-اسمی) (Nominal)

آمار: برای خلاصه سازی داده ها و یا بدست آوردن یک الگوی خاص از روابط آماری استفاده می‌شود. دو نوع شاخص آماری وجود دارد: مرکزی، پراکندگی

شاخص مرکزی
- میانگین
  - حسابی: $\mu_A = \frac{1}{n}\sum_{i=1}^{n} x_i$
  - هندسی: $\mu_G = \left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$
  - هارمونیک: $\mu_H = \frac{n}{\sum_{i=1}^{n}\frac{1}{x_i}}$

$\mu_H \leq \mu_G \leq \mu_A$

$$\mu_W = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

Date: $P(A|B) = \frac{P(A \cap B)}{P(B)}$

میانه را بدون مرتب کردن بدست آورید.

پراکندگی — دامنه تغییرات — Max - Min

واریانس و انحراف معیار

$$s^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_A)^2$$

$\sigma = \sqrt{s^2}$ standard deveation

کوواریانس: ارتباط میان دو ویژگی را بیان می کند

$$Cov(x,y) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_{Ax})(y_i - \mu_{By})$$

$$Cov(x,y) = E(x \cdot y) - \bar{x}\,\bar{y}$$

همبستگی

$$Corr(x,y) = \frac{Cov(x,y)}{\sqrt{Cov(x,x) + Cov(y,y)}}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

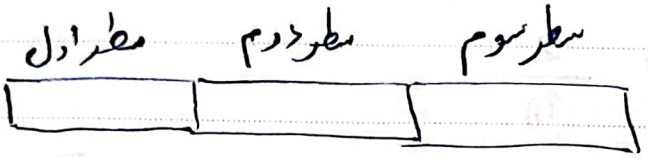
Date: Sub:

ماتریس کوواریانس Coumat (i, j) =

Cov(Feature$_i$, Feature$_j$)

Classification

① مشتقم سطرها را کنار تصاویر هم بچینیم تا آرایه شود بدهیم به



Nearest Neighborhood

② کوواریانس سطرهای تصویر با خودش را محاسبه کنیم سپس Nearest-Neighbor می‌دهیم

③ Color histogram => Nearest-Neighbor

N سطر - ۲۵۵ ویژگی

---

test hold out - Cross-validation - K-fold

train

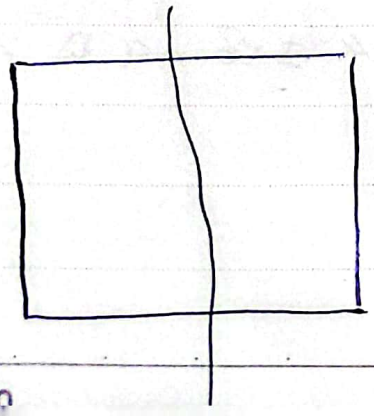
leave one out - hold out

test جداش کند. ← برای داده‌های کم هر بار یکی را برای

metrics for Evaluating Classifier ← Performance

$$\text{Accuracy} = \frac{\text{num true classified}}{\text{num all data}} = \frac{TP+TN}{P+N}$$

TN
FP
FN



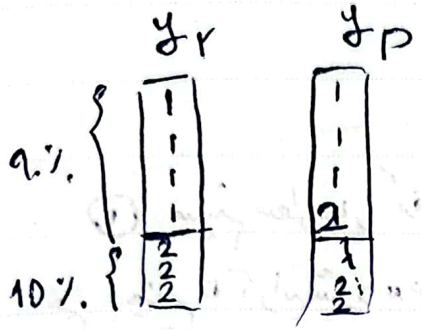
$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{recall} = \frac{TP}{TP+FN}$$

Date: Sub:

F1-score

برای مسائل چند کلاسه مهم را + در نظر می‌گیرند بقیه داده‌ها همه را - در نظر می‌گیرند.

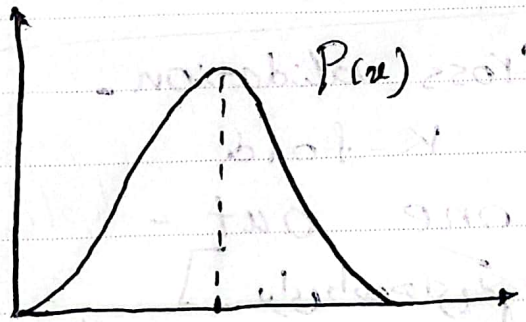


$$\text{precision} = \frac{2}{3}$$

$$\text{recall} = \frac{2}{10}$$

معیار $\bar{F}$

معیار F_1 score



$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

$\mu \pm \sigma$ ۶۸٪

$\mu \pm 2\sigma$ ۹۵٪

Date: Sub:

توزیع استاندارد $\begin{cases} \mu = 0 \\ \sigma = 1 \end{cases}$

توزیع نرمال ← استاندارد داده‌ها منهای میانگین تقسیم بر انحراف معیار

کنیم انحراف معیار نیز برابر ۱ شود.

رگرسیون خطی

$$y = \alpha + \beta x$$

$$\alpha = \bar{y} - \beta \bar{x}$$

$\bar{y}$ میانگین y هایی که داریم (مثلا ۱۰ نقطه)

$$\beta = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})}$$

---

principal Component Analysis

PCA : یک روش کاهش ابعاد داده است.

مثلا X داریم که شامل n داده هست که هر کدام D بعد (مثلا ۱۰۰ بعد) دارند

$$X_{n \times D} \; W_{D \times d} = X_{new\; n \times d}$$

Date: Sub:

الگوریتم :

داده هارا ارسال می‌کنیم چگونه؟

Linear Scaling:

① $\frac{x_i - \min(X)}{\max(X) - \min(X)} \times 2 - 1$

---

② $Z_{\text{normalized}}$ $x'_i = \frac{x_i - \mu}{6}$

ماتریس کوواریانس ویژگی هارا حساب می‌کنیم C

بردار ویژه و مقدار ویژه C را بدست می‌آوریم

$C \times E_{vector} = \lambda \times E_{vector}$

بردار ویژه هارا براساس مقدار ویژه λ مرتب می‌کنیم

$\lambda_1 > \lambda_2 > \lambda_3$

$|\lambda I - C| = 0$ ← دترمینان

چند جمله‌ای درجه D - D ریشه دارد

هر مقدار ویژه λ_1 را در رابطه $*$ قرار می‌دهیم

بدست می‌آید E_{vector} $D \times 1$

بردارهای ویژه بر هم عمودند (متعامد هستند)

Elipon

Date: Sub:

برای هر λ یک Eigenvector داریم که $D \times 1$ هستند

$W_{d \times D}$

d تا از آنها را انتخاب می‌کنیم و W را قرار می‌دهیم ستون اول λ با بیشترین مقدار را قرار می‌دهیم.

d مناسب برای هر مسئله چقدر باشد خوب است؟

$$\frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{D} \lambda_i} \geq 0.95$$

کوچکترین K که در معادله صدق کند

توزیع در ابعاد جدید 95 درصد اطلاعات حفظ می‌شود

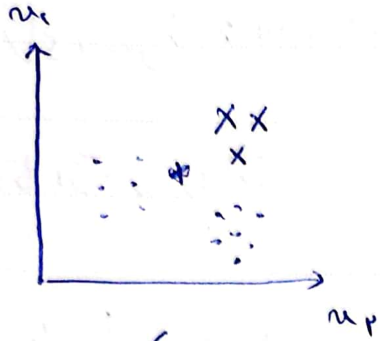
PCA یک روش کاهش بعد خطی است.

روش‌های دسته بندی

1. K_Nearest_Neighbor _ ε_nearest_ neighbor
2. support vector machine (SVM)

Date: Sub:

روش های دسته بندی:



o کلاس ۱

X کلاس ۲

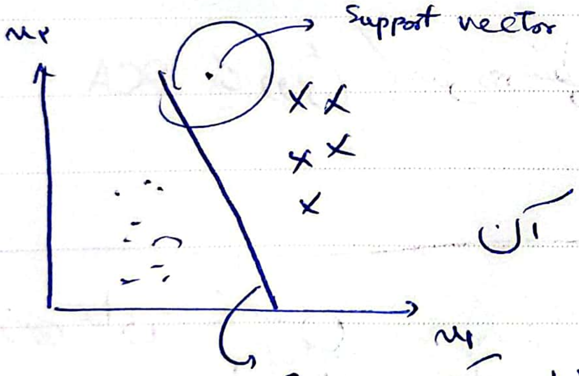
* تست

برچسب گذاری ← k تا نزدیک‌ترین همسایه را پیدا کن ← داده تست

در فاصله k همسایگی را اندازه گیری می‌کنند

* برای دسته‌بندی های Binary ← دسته های زوج k انتخاب نمی‌کنند چرا؟ مشکل است
۵۰ ۵۰ می‌شود

Support vector machine:

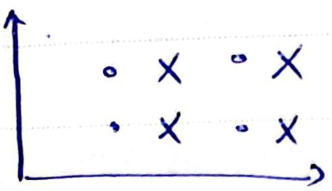


در SVM خطی را پیدا می‌کنی که margin آن
بیشترین باشد!

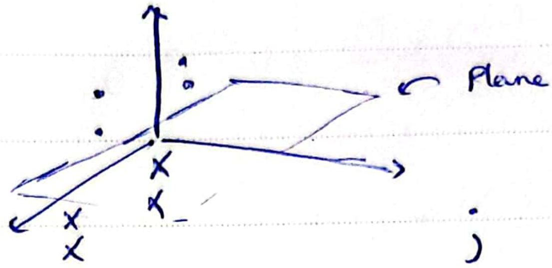
svm خط رسم می‌کند
و معادله خط را پیدا می‌کند

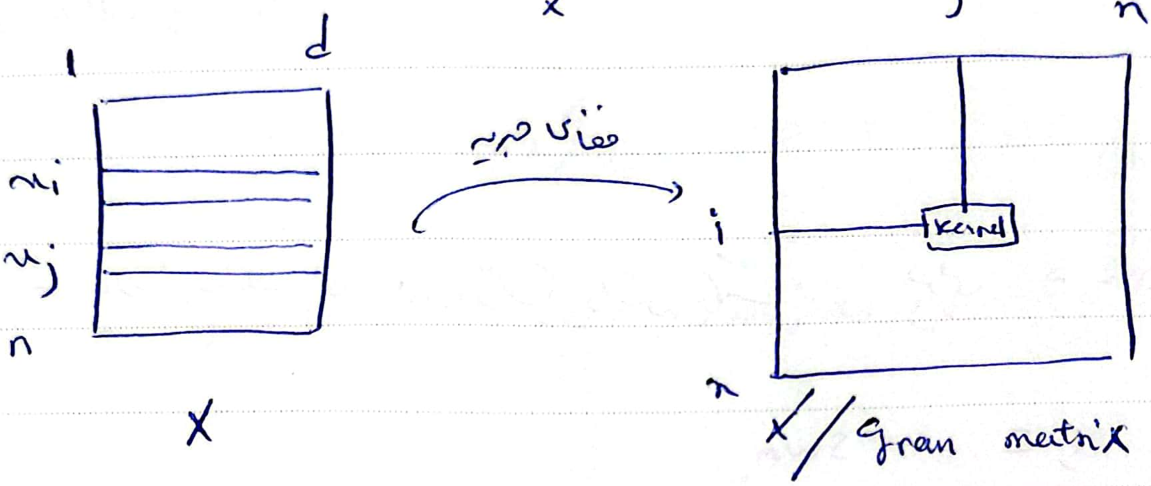
فضای n بعدی که Hyper plane نام دارد

با استفاده از جبر خطی معادلات آن را پیدا می‌کنند تاکید بر روی این داده‌های در مرز margin
هست پیدا کرده و فاصله را بیشینه می‌کند



برای جداسازی آن ببری آن kernel نیازمند به ϕ
آن را نمایش می‌دهیم





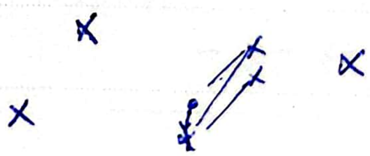
$\text{kernel}(\vec{x_i}, \vec{x_j}) \Rightarrow X'$ در

شباهت ۲ بردار را به ما می‌دهد

ممکن است همواره فضا داشته باشیم در آن که یک margin ε
را می‌دهیم که این فضا الکی اشغال نکرد

شباهت ۲ بردار چیست؟ / kernel شباهت ۲ بردار را به ما می‌دهد)

kernel
1. linear → $\vec{a} \cdot \vec{b} = |a| \cdot |b| \cos\phi$
   - 0° → شباهت بالا
   - 90° → کمی شباهت
   - 180° → مخالف هم
2. Poly → $(a \times b + r)^d$ → d: معمولا ۲ ، r: معمولا صفر
3. RBF → $\exp\left(\frac{\|a-b\|^2}{2\sigma^2}\right)$ → $2\sigma^2$: معمولا ۱
   - معمولا جواب های بهتری می‌دهد



Weighted knn

Date: Sub:

٤

برای دسته بندی داده با بیشتر از ۲ کلاس ، ۳ کلاس و بیشتر

One vs One :

One vs other :

فرض کنید مسئله ۵ کلاس داشته باشیم ، برای $m = ۵$

در روش اول

1, 2 → svm

1, 3 → svm   ۲, 3 → svm

1, 4 → svm   ۲, 4 → svm   3, 4 → svm

1, 5 → svm   ۲, 5 → svm   3, 5 → svm   4, 5 → svm

۱۰ svm برای ۵ کلاس

$m(m-1)$ → svm داریم

روش دوم یک svm برای یک کلاس می سازیم

یکی ۱ Svm

۱ → +

۲, ۳, ٤, ۵ → -

---

۲ → +

۱, ۳, ٤, ۵ → -

---

۳ → +

۱, ۲, ٤, ۵ → -

به همین ترتیب پیش می رویم

m تا svm داریم

Date: Sub:

روش دسته بندی بیز Bayesian

$$P(C=c_i) \times \prod_{j=1}^{n} P(A_j = a_j \mid C = c_i)$$

(تعداد ویژگی‌ها یا ستون)

مقدار احتمال رابطه مقابل را محاسبه می‌کنیم.

این رابطه را برای داده test روی هر کلاس c_i بدست می‌آید، درنهایت

برای کلاس "ی" که بیشترین مقدار را دارد، داده test به آن

می‌شود

$A_1, A_2, \ldots, A_n$ ← ویژگی‌ها

داده تست $d < a_1, a_2, \ldots, a_n >$ ← مقادیر ویژگی‌های یک داده نمونه

شماره ردیف — ویژگی‌ها — برچسب کلاس

ID	A	B	C
1	a	d	y
2	a	e	y
3	b	f	y
4	c	e	y
5	b	f	y
6	b	f	n
7	b	e	n
8	c	d	n
9	c	f	n

مثال:

A B

test < a, f >

① $P(C=y) \times \prod_{j=1}^{2} P(A_j = a_j \mid C=y) =$

② $P(C=n) \times \prod_{j=1}^{2} P(A_j = a_j \mid C=n) =$

Date: Sub:

① $P(C=y)\,P(A=a \mid C=y) \times P(B=f \mid C=y) =$

$\frac{1}{2} \times \frac{2}{5} \times \frac{2}{5} = \frac{2}{25} = \frac{8}{100}$

② $P(C=n)\,P(A=a \mid C=n) \times P(B=f \mid C=n) =$

$\frac{1}{2} \times \frac{1}{5} \times \frac{2}{5} = \frac{1}{25} = \frac{4}{100}$

$\frac{P(A \cap C)}{P(C=n)} = \frac{\frac{2}{10}}{\frac{1}{2}} = \frac{2}{10} \times \frac{2}{1} = \frac{4}{5}$

چون مخرج را حذف کردیم احتمال دقیق در نیامد

ستون ۵ و ۶ ویژگی یکسان برچسب متفاوت دارند

⇐ ویژگی‌ها کافی نیست

اثبات رابطه نیز در کلاس گفته شد.

$P(A \mid B) = \frac{P(B \mid A)\,P(A)}{P(B)}$

$P(C=C_i \mid A_1=a_1 \text{ و } \dots \text{ و } A_n=a_n)$

$\frac{P(A_1=a_1, \dots A_n=a_n \mid C=C_i)\,P(C=C_i)}{P(A_1=a_1 \text{ و } \dots \text{ و } A_n=a_n)}$

← برای یک کلاس C_i این مقادیر یکسان هستند آنرا از مخرج حذف می‌کنیم

Date: Sub:

$$\Rightarrow P(A_1=a_1, \dots A_n=a_n \mid C=C_i) =$$

افزار زنجیر

$$P(A_1=C_1 \mid A_2=a_2, \dots A_n=a_n, C=C_i) \times$$

$$P(A=a_2, \dots A_n=a_n \mid C_i)$$

چون A_n از هم مستقلند

$$P(A_1=a_1 \mid A_2=a_2, \dots A_n=a_n, C=C_i) =$$

$$P(A_1=a_1 \mid C=C_i)$$

~~$P(A_1=a_1$~~ $\prod_{j=1}^{n} P(A_j=a_j \mid C=C_i)$ $\Rightarrow$

$$P(C=C_i) \times \prod_{j=1}^{n} P_{Aj}$$

gram matrix - sum - normalization

دوستی آماری

درخت تصمیم Decision Tree

برای ساختن درخت چگونه عمل کنیم؟ فرض می‌کنیم داده‌ها Category مقدار داده شده‌اند یا گسسته هستند؟ feature انتخاب شده

① در هر مرحله چه ویژگی

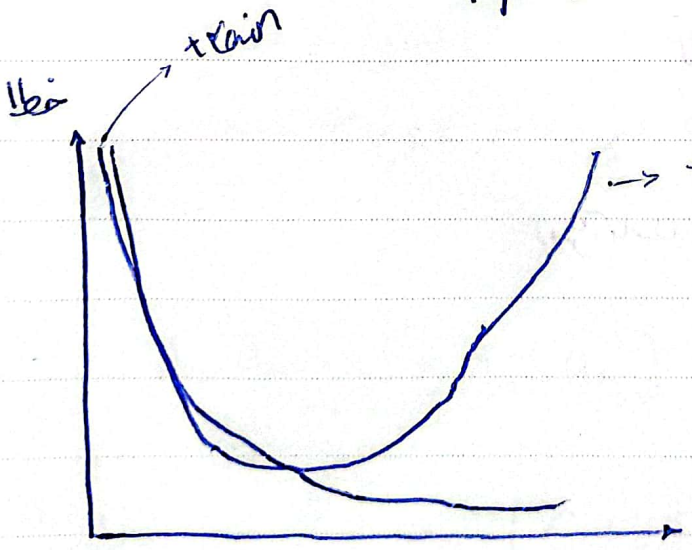
② عمق درخت چقدر باید باشد؟

② ۱- زمانی که داده‌های برگ درخت برچسب یکسان داشته باشند

۲- برای عمق یک حد بالا تعیین کنیم ۳- حداقل برای تعداد گره‌ها تعیین کنیم

۴- تعداد داده‌های موجود در گره خیلی کم شود (۲ یا ۳ داده) گسترش این گره را متوقف می‌کنیم (این مقدار را از قبل مشخص می‌کنیم).

۵- دقت بر روی Validation



صورت مسئله overfitting

Date: Sub:

referral => get user_id input referrals with
~~show user_id~~ with depth show them
+ num

add depth + => admin panel

~~show~~ watch users

Ex:- $\tau = RC$ $R = \frac{C}{\tau}$ s

$R = \frac{\tau}{C}$ s

Date: Sub:

برای رفع overfitting خود داده های train را به دو قسمت تقسیم می کنیم

train 70% } 10% validation } test → 20%

هر بار قبل از افزایش پیچیدگی مدل دقت مدل را بر روی validation امتحان می کنیم اگر کاهش یافت آموزش و افزایش پیچیدگی محاسباتی مدل را متوقف می کنیم

برای مثال شروطی که برای عمق درخت می گذاریم می توانند از overfitting جلوگیری کند.

---

① 1- information gain

$$IG(H) = Entropy(D) - Entropy_A(D)$$

↳ ویژگی H یا

$$Entropy(D) = -\sum_{i=1}^{C} P_i \times Log_2^{P_i}$$

$$Entropy_A(D) = \sum_{j=1}^{V} \frac{|D_j|}{|D|} \times Entropy(D_j)$$

C ← تعداد کلاس $\quad$ $\bar{V}$ → تعداد سطح های ویژگی A

P_i ← احتمال نمونه ای از داده متعلق به کلاس i باشد

D_j → قسمتی از داده ها که ویژگی A آنها مقدار V_j داشته باشد.

$|D|$ → تعداد، اندازه

Date: Sub:

معیارهای انتخاب ویژگی برای تقسیم عنوان گره ریشه درخت:

1- Information gain
2- Gini index
3- Gram ratio
4- Likelihood ratio

$$Gini(D) = 1 - \sum_{i=1}^{L} P_i^2$$

$$Gini_A(D) = \frac{|D_1|}{|D|} \times Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

A (ویژگی/ستون)

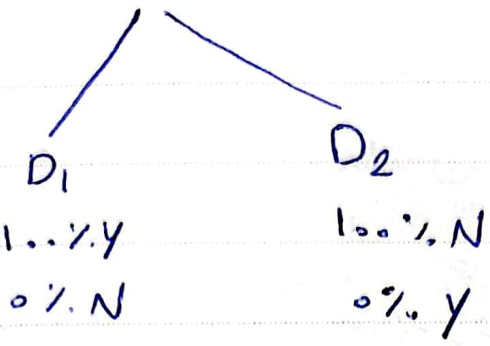
C تعداد کلاس‌ها P_i احتمال وجود داده در کلاس i

$A \rightarrow$ به دو مجموعه D_1 و D_2 تقسیم می‌شود $\rightarrow$ در D_1 و D_2 مجددا $Gini(D_1)$ را محاسبه می‌کنیم برای همین تعداد

$$Gini(A) = Gini(D) - Gini_A(D)$$

ویژگی که این مقدار را max کند انتخاب می‌کنیم

Date: Sub:

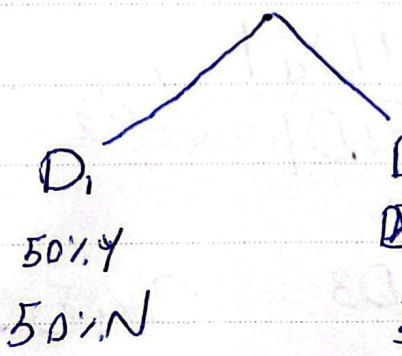


$$\text{---} \propto Gini(D_1) + (\quad) \propto Gini(D_2) = 0$$

$Gini(D_1) = 1-(1^2+0^2) = 0$

$Gini(D_2) = 1-(0+1^2) = 0$

اگر داده ها کاملا از هم تفکیک شوند Gini = 0 می شود.



$$\text{---} \propto Gini(D_1) + (\quad) \propto Gini(D_2) = \text{عدد دلخواه } 1/2$$

$Gini(D_1) = 1-((\frac{1}{2})^2+(\frac{1}{2})^2) = 0/5$

$Gini(D_2) = 0/5$

$$\frac{|D_1|}{|D_2|} \propto \frac{1}{2} + \frac{|D_2|}{|D|} \propto 1/2 =$$

$$\frac{1}{2}\left(\frac{|D_1|+|D_2|}{|D|}\right)^{|D|} = 1/2$$

D_1 و D_2 برچسب های ویژگی هستند مثلا یکی از برچسب های ستون A را D_1 و بقیه برچسب ها را D_2 در نظر می گیریم این کار را برای تمامی حالت ها انجام می دهیم

Date: Sub:

gain ratio → info gain

دانشمال ی کند

زیرا I در بعضی موارد اگر تعداد سطح های ویژگی زیاد باشد آنرا انتخاب ی کند

$$\text{gain Ratio}(A) = \frac{\text{Info Gain}(A)}{\text{splitInfo}(A)}$$

A ← ویژگی

$$\text{splitInfo}(A) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \log_2 \frac{|D_j|}{|D|}$$

likelihood ratio ID3 C45 → نمونه برای نقاش

CART

Date: Sub:

جنگل تصادفی Random Forest

روش‌های مختلف ترکیب دسته بندها
Combination of Classifiers

- Boosting
- Bagging
- stacking

براساس وزن ← معمولاً overfit می‌شه

Bagging : دسته‌بندی داده‌ها به تعداد کیسه مجزا تقسیم می‌کنیم - همه داده‌های Train

به Classfier ها داده نشود - (حذف وزنی در Bag) - حذف داده‌ها در Bag

Random Forest نوعی Bagging است. برای ترکیب از میانگین استفاده می‌کنیم

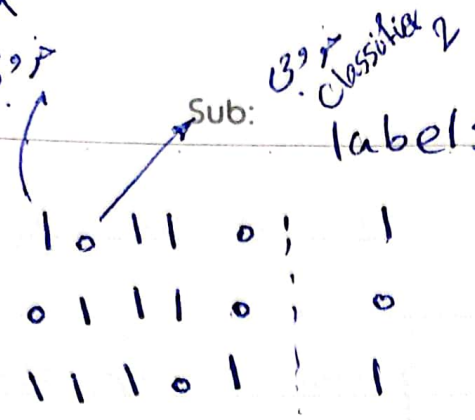
stacking : نوع دسته‌بند را عوض می‌کند مثلا DT با عمق‌های مختلف یا SVM با Kernel های مختلف استفاده می‌کنیم - pattern label

دسته بندها را بدست می‌آوریم

predict(D_i)

Classfier1	1	0	1
Classifier2	0	1	1
Classifier3	1	1	1
	1	1	0
100	0	0	1

Date: Sub:



X'

دوباره X' را به عنوان ویژگی های بدست آمده به یک classifier دیگر می دهیم تا آموزش ببیند pattern داده ها را به صفر یا یک نگاشت کند. به این classfier اصطلاحا "meta-learner" می گوییم. می توانیم کنار ستون های خروجی n دسته بند و D ویژگی اولیه x را نیز قرار دهیم و بعد meta learner را آموزش دهیم.

شبکه عصبی:

Date: Sub:

چرا در صورت مسئله‌های چند کلاسه استفاده از precision ، recall ، F1

بهتر از accuracy است چ زیرا هربار precision و recall هرکلاس را

در نظر بگیریم (هربار کلاس هدف مثلا 1 یا 2 یا 3 و ...) را به عنوان

Positive در نظر گرفتیم و سایر کلاس ها را - در نظر قرار می گیریم

استفاده Confusion matrix برای درک بهتر labelهایی که به اشتباه تشخیص

داده می شوند .

صورت مسئله Clustering به دو مدل است => ۱- تعداد cluster از

قبل مشخص باشد ۲- روش clustering خودش به تعداد مناسب clusterها

پی ببرد .

داده ها بدون برچسب هستند می خواهیم آنها را در چند طبقه تقسیم کنیم

Date: Sub:

روش های خوشه بندی
- تعداد خوشه ها از قبل داده شده اند
  - K-means
  - K-medoid
- الگوریتم تعداد خوشه های مناسب را بیابد
  - سلسله مراتبی
    - Top-down
    - bottom-up
  - DB scan

الگوریتم

۱- به تعداد K مرکز خوشه تقاضی درباره ویژگی های داده ها تولید می کنیم (تصادفی)

۲- داده را بر اساس نزدیکی به این K مرکز خوشه برچسب گذاری می کنیم

۳- مراکز خوشه جدید را بر اساس میانگین برچسب های مشابه محاسبه می کنیم

چون ممکن است نتیجه نسبت به K تعداد خوشه تقاضی متفاوت باشد بعضی اوقات مرحله ۱ را با شرط K تا از بهترین نقاط از یکدیگر اجرا می کنیم

Top down : همه داده ها را یک خوشه در نظر می گیرد سپس split می کنیم

Bottom up : هر داده را یک خوشه می گیرد سپس دو خوشه با کمترین فاصله را با هم merge می کند

Date: Sub:

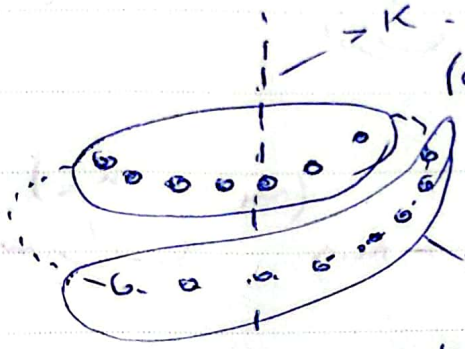
topdown: Complete LinKage max $d_{B}(a,b)$ $a \in A, b \in B$

bottomup: single LinKage min $d_{B}(a,b)$ $a \in A, b \in B$

dendrogram → bottomap
tree → top-down

درخت‌نگه
در این دو روش می‌سازیم و
K را با استفاده از آن تعیین می‌کنیم

این روش‌ها توزیع داده‌ها را در نظر می‌گیرند اما روش DBscan این کار را انجام



می‌دهد. (توزیع هلالی را در نظر گرفت)

ورودی n: حداقل تعداد داده در cluster

DBscan 1- انتخاب داده‌ها در فاصله ε از هم خوشه هستند

2- از داده‌هایی که خوشه بندی نشده اند یکی را انتخاب می‌کنیم

DBscan برای پیدا کردن نویز نیز استفاده می‌شود. مثلا مقدار این دو از n کمتر است آنها حذف می‌کنیم