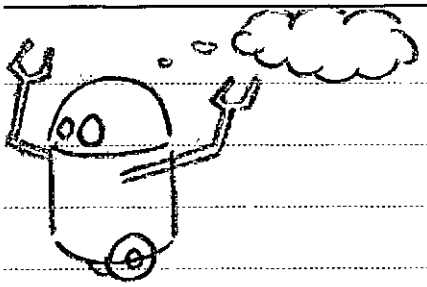


یادگیری ماشین

* مراجع:

Machine learning : Tom Mitchell

Reinforcement learning : S. Sutton

* اساتید درس:

shiry@aut.ac.ir

ce.it.aut.ac.ir/~shiry

* مدرس:

Kourosh Meshgi

0912-2129404

2nd Semester 1387-8

Subject :

Year . Month . Date . ( )

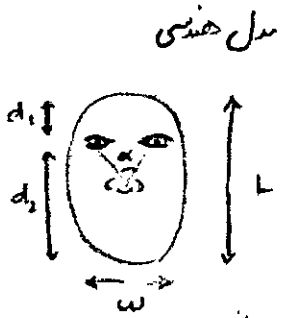
* یادگیری
  - کسب دانش ← بهبود عملکرد ☑ دانستن چی
  - تجربه ← مهارت ☑ دانستن

- یادگیری عبارتست از بهبود عملکرد از طریق تجربه
- یادگیری عبارتست از بدست آوردن دانش یا فهم آن از طریق مطالعه، آموزش و یا تجربه

* یادگیری ماشین
  - بینایی مصنوعی ← توسعه برنامه (عامل) ← رفتار عاقلانه، ...
  - یادگیری ماشین ← برنامه
    - کسب تجربه
    - تغییر در ساختار برنامه

* ابزار
  - Matlab
  - Weka

* آلیسه
☑ برنامه‌ای نوشتید که صورت انسان را در داخل تصویر تشخیص دهد.



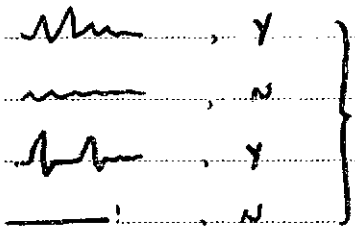
دنیای واقعی (تصویر گرفته) ← دنیای کامپیوتر (مدل)

دنیای واقعی ← در آوردن مدل ← ابزار ← تشخیص مدل در تصاویر
برنامه نویسی

Subject:
Year. Month. Date. 2

☑ یادگیری قلبی

± مثال



⇒ الگوریتم یادگیری → برنامه ای که از طریق مثال یادگیری می کند که + و − را جدا کند.

الگوریتم یادگیری ← چطور آن را به برنامه داد ← جمع آوری مثال

Feature extraction

* کاربرد
  - فیلتر رو باتا
  - داده کاوی ← معمولاً برمبنای داده های آماری
  - تشخیص گفتار
  - شناسایی متن
  - بیوانفورماتیک
  - بازی های کامپیوتری

* طبقه بندی یادگیری ماشین
  - دسته بندی
  - حل مسئله و طراحی دخل

* مبنای ارزیابی الگوریتم های یادگیری
  - دقت دسته بندی
  - صحت راه حل و کیفیت آن
  - سرعت عملکرد

Subject:
Year. Month. Date. ( )

* تکنیک های مختلف یادگیری

- Inductive ← استنتاج
- Connectionist ← مدل مغز
- Bayesian ← آمارو احتمالات
- Reinforcement ← تجربه در محیط
- Evolutionary ← مدلی تکامل

* انواع یادگیری از لحاظ ناظر

- Supervised
- Unsupervised

☑ این از مهمترین کاربردهای ML ← function approximation

* هدف اصلی یادگیری باید تعمیم داشته باشد ← Generalization

* اجزای طراحی یادگیری

- Task : وظیفه‌ای که ماشین باید انجام بدهد.
- Performance : معیاری که کارایی برنامه را با آن می‌سنجند.
- Training Exp : مراحل آموزشی برنامه برای کسب تجربه

☑ رانندگی در بزرگراه

- ورودی : دوربین روی سقف اتومبیل
- تجربه آموزشی : شب رانده ، رانندگی می‌کند و کامپیوتر رفتار را نظاره می‌کند

☑ بازی Checkers

- تجربه آموزشی ①
چند تن بازی می‌کنند و بازی آنها نظارت می‌شود. ← هزینه بر
- تجربه آموزشی ①
با سیستم بر علیه خودش بازی کند

- نتیجه یادگیری: در صفحه B را به M ببریم، بهترین حرکت M را به ما بدهد.
move: B → M
ایجاد بسیار دشوار است.
حال سعی می‌کنیم برای هر صفحه یک مقدار عددی را یاد بگیریم که ارزش صفحه را
$\hat{V}$: B → ℝ
به ما بدهد

- معیار امتیازدهی: آرایش نهایی ← به‌راحتی مقدار دهی می‌شود
وضعیت‌های میانی ← چطور به آرایش نهایی می‌رسد.

- فیدبک: مستقیم ← حالت آرایش صفحه و حرکت صحیح هر حالت توسط مربی مشخص می‌شود.
غیرمستقیم ← مجموعه‌ای از حرکات پشت سر هم و در بازی و نتیجه آن موجود باشد.
که به ازای هر حرکت پاداش نمی‌گیریم.

- یادگیری بازی: Function Approximation ($V \to \hat{V}$)
  - مدل بزرگ
  - مجموعه آموزش
  - چند جمله‌ای از ویژگی‌ها
  - شبکه عصبی

- Trade-off: دقت ↔ تعداد مثال‌ها

Sample ↑ ⇒ $\hat{V}$ → V

Subject :
Year . Month . Date . ( )

• تقریب زنی

$\hat{V} \longrightarrow w_i$

$f(B) = w_0 + w_1 x_1 + \cdots + w_n x_n$

↳ feature 1

روش‌های تنظیم وزن‌ها در شبکه‌های استفاده می‌شود . مثلاً Least Mean Square

• احتمالاً این روش تا در نخواهد بود که انسان را کاملاً دقیق چون در مدل‌سازی آن ، طرز فکر ساده‌ای مثلاً خطی استفاده می‌شده است .

• مراحل کار

۱- تعیین خصیصه آموزشی ← ابعاد

۲- تعیین تابع هدف

$B \longrightarrow M$

$B \longrightarrow V(B)$

$B \longrightarrow \hat{V} \longrightarrow w_i$

۳- نحوه نمایش تابع یادگرفته شده ← خطی / چند جمله‌ای / شبکه عصبی

۴- تعیین الگوریتم یادگیری

* زمینه‌های مناسب الگوریتم یادگیری

Real Time

• سرعت
• سادگی ← حجم محاسبات + حافظه
• دقت ← قابلیت اطمینان

* سؤالهای مطرح درباره مسائل یادگیری

• کدام الگوریتم
• چه مقدار داده آموزشی مورد نیاز است ← نسبه به پیچیدگی تابع یادگیری

- مثال آموزش بعدی ← آموزش از اول
  آموزش تدریجی
- تبدیل مسئله یادگیری به مسئله تقریب تابع
- بازنمایی خودکار ← سازه‌ای بیشتر

* یادگیری ذهنی مسئله جستجو است.
این جستجو به دنبال یک فرضیه است. ← یک تابع
این فرضیه باید با داده‌های آموزشی و دانش قبلی سازگار باشد.
اینکار Bias کردن نام دارد. ← تمایل به چه تابع انتخاب شده
یادگیری نیازمند تحلیل است.

Subject:
Year. Month. Date. ( )

یادگیری مفهوم — Concept Learning

* مفهوم
  - تعریف: دسته‌بندی مغز برای اشیاء، حوادث، یا ایده‌ها که دارای مجموعه‌ای مشترکی هستند.
  - هدف: دسته‌بندی
  - یادگیری مفهوم در انسان

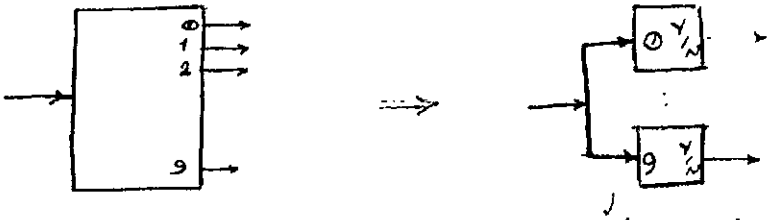
- مکتب Hull ← Stimulus - Response Association
- مکتب Rosh ← مثالهای متعدد و نه قوانین

- یادگیری در کامپیوتر
  - درست آوردن تعریف مفهوم طی ارزیابی مثالهای مثبت و منفی
  - یادگیری تابع بولی

☑ آیا این شئ کامپیوتر است؟ ✓
این شئ چیست؟ X

- روشهای یادگیری چند مفهوم، هیچ سیستم را آموزش می‌دهیم.

☑ یادگیری ارقام



- مثال آموزشی ← مجموعه از وکتورها
  ↙ دسته‌بندی مثبت یا منفی هر مثال

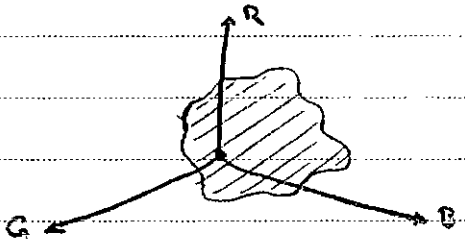
Subject:
Year. Month. Date. (5)

• نمایشها ← مثال
فرضیه ← بایس ← تابع بین ویژگیها و جواب + یا -
الگوریتم و انواع تابع نشان دادن

☑ تشخیص لبخند

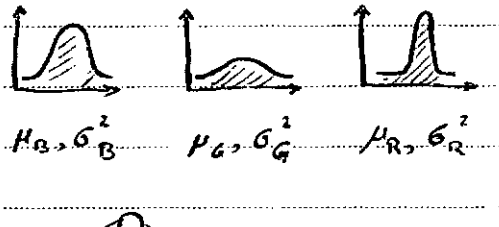
لبخند ← چهره ← رنگ پوست ← RGB
نمایش شاها برای تشخیص صورت

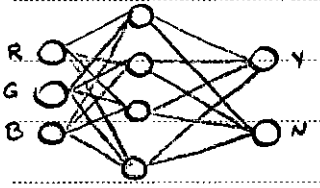
R	G	B	face
0	0	0	N
⋮			⋮
100	73	24	Y



در جدول جایی نشود (256^3) ← این تابع یاد می‌گیرد که این رنگ صورت است یا خیر؟

شبکه عصبی یا تابع لوسی ← Bias





• مهندسی در دنیای واقعی ← ویژگیها در دنیای کامپیوتر

☑ چهره ← ویژگیهای ساده: رنگ، اندازه
ویژگیهای مصنوعی: تبدیل فوریه، تبدیل موجک

Subject :
Year . Month . Date . ( )

• تحلیل فرضیه

مقدار ویژگی ← هر مقداری قابل قبول باشد : ?
هیچ مقداری قابل قبول نباشد : ∅
مقدار خاصی در غیر این صورت :

تابع فرضیه (h) ← مجموعه تمام فرضیه ها (H)

می خواهیم در مثال ورزش آبی به دنبال معلم بگردیم که جواب را بدهد.

< ?, ?, cold, high, ?, ? > → No

• عمومی ترین فرضیه ← همه ویژگیها "?" باشند
اختصاصی ترین فرضیه ← همه ویژگیها "∅" باشند

• نمونه : مجموعه ای از ویژگیها که مفهوم را توصیف می کنند.

• تابع هدف ← برای مثالهای مثبت = 1
برای مثالهای منفی = 0

• فرض استقرایی یادگیری استقرایی
هر تابعی که بتواند تقریب خوبی از تابع هدف برای یک مجموعه آموزشی با اندازه کافی بزرگ باشد، تقریب خوبی برای تابع هدف را در مورد مثالهای دیده نشده هم تقریب میزند.

• تعداد کل فرضیه ها :

کلی : π (تعداد مقادیر مجاز ویژگی + ∅ + ?)
معادل : $1 + \pi$ (تعداد مقادیر مجاز ویژگی + ?)

↳ یعنی هیچ مثال مثبت نسبت ← حالت تماماً نهی

Subject:
Year. Month. Date. 6

• روش List-Then-Eliminate
- تمام فرضیه‌ها را ردیف می‌کند.
- با هر مثال آموزشی، لیست فرضیه‌ها را هرس می‌کند.
- در پایان تعدادی فرضیه باقی می‌ماند.
- لیست کردن تمام فرضیه‌ها غیر ممکن است (فضای بزرگ)
- یادگیری: جستجو در میان تابعی که تمام مثال‌ها را درست می‌کند.
- انتخاب بهتر: کلیه عمومی‌تر ← دسته‌بندی مثال‌های + بالاتر می‌رود.

• جستجو در یادگیری مفهومی
اجازه می‌دهد همه جا را ببرد.
دارای یک heuristic است.

Neural Network ⟶ Gradient Descend
Genetic Algorithm ⟶ Survival of Fittest
Particle Swarm ⟶ Local / Global Experiment

• مقایسه فرضیه‌ها
- کدام فرضیه بهتر است؟ ← heuristic
- معمولاً عمومی‌تر بودن فرضیه مطلوب است.
- قابلیت تقسیم

$h_j \geq_g h_k$ (عمومی‌تر) : $\forall x \in X : h_k(x) = 1 \longrightarrow h_j(x) = 1$

- انعطاف‌پذیر تقسیم

Subject :
Year . Month . Date . ( )

• روش Find-S

- اختصاصی ترین فرضیه که با مثالهای آموزشی تطبیق دارد را پیدا می کند.
- الگوریتم

$h \leftarrow \phi$ همه

$h \leftarrow$ با مثالهای + تعمیم می دهد.

☑

شروع	$< \phi , \phi , \phi , \phi , \phi , \phi >$
مثال 1	$< S , W , N , _ , W , S >$
مثال 2	$< S , W , ? , S , W , S >$
مثال 3	مثال منفی را در نظر نمی گیرد.
مثال 4	$< S , W , ? , ? , ? , ? >$

- الگوریتم فرضیه های منفی را در نظر نمی گیرد ← با فرض درست بودن مثالهای آموزشی C

$h \leq_g C$

- سوال ← آیا فرضیه سازگار دیگری وجود دارد؟

آیا داده خالی از نویز است؟ ← کاملا" حساس به نویز

آیا اختصاصی ترین فرضیه خوب است؟

اگر چندین فرضیه اختصاصی حداکثر وجود داشت چه می شد؟

اگر چندین فرضیه اختصاصی حداکثر وجود نداشت چه می شد؟

جمع بندی

یادگیری مفهوم { فرضیه ← ماپس به یک تابع
نمایش دیگر کلیها
روش جستجو

• روش Candidate Eliminate

- برای رفع مشکل مثال‌های منفی Find_S
- دید کلی به جزئی
- سازگاری

$Consistent(h, D) \Leftrightarrow [\forall (x, c(x)) \in D]\ h(x) = c(x)$

فرضیه‌ای که با مفهوم هدف سازگار است (برای مثال آموزشی)

- Version Space

مجموعه تمام فرضیه‌های سازگار با مثال آموزشی (VS)

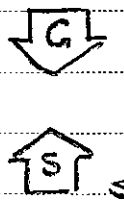
- پیاده‌کردن VS به‌طوری که مقدور نیست
- مفهوم

$G_0 \leftarrow$ تمام ؟

$S_0 \leftarrow$ تمام $\emptyset$

$G \leftarrow$ تعمیم G_0 که شامل مثال منفی نباشد

$S \leftarrow$ تعمیم S که شامل مثال مثبت باشد



- الگوریتم

1. G_0, S_0
2. + مثال ⟶ مثال‌های ناسازگار S حذف
   مثال‌های ناسازگار G حذف
   حداقل تعمیم G ⟶ از S رد نشود
3. − مثال ⟶ ...
   حداکثر اختصاصی S ⟶ از G رد نشود
4. اگر S و G از هم رد شوند ⟶ جواب ندارد
5. G حداکثر و S حداکثر

- اگر متغیر بیش از دو مقداری باشد، برای ترتیب آن چه باید کرد؟ تعمیم داد که عمومیت بیشتر دارد

PAPCO

زیادی است راین ضعف بزرگی است (چون مانعای ترکیب عطفی OR ندارد)

فضای فرضیه ترکیب عطفی ⟶ ضعف فضای فرضیه

Subject:

Year. Month. Date. ( )

G : $\langle S, ?, ?, ?, ?, ? \rangle \quad \langle ?, \omega, ?, ?, ?, ? \rangle$ ☑

S : $\langle S, \omega, ?, S, ?, ? \rangle$

سوال : ① چرا لست VS تهیه شود؟

② نحوه ارزیابی فرضیه جدید

جواب : $\langle S, \omega, ?, ?, ?, ? \rangle$

$\langle S, ?, ?, S, ?, ? \rangle$

$\langle ?, \omega, ?, S, ?, ? \rangle$

- شرط خاتمه ← ① جواب ← G و S از هم رد می شوند

همگرایی ← $G = S$ ← 1 جواب

$S <_J S$ ← بالا 2

Japanese Economy Car ☑

G_0 : $\langle ? \; ? \; ? \; ? \; ? \; ? \rangle$

G_1 : G_0

G_2 : $\langle ? \; H \; ? \; ? \; ? \rangle \langle ? \; ? \; B \; ? \; ? \rangle \langle ? \; ? \; ? \; BD \; ? \rangle \langle ? \; ? \; ? \; ? \; E \rangle$

G_3 : $\langle ? \; ? \; B \; ? \; ? \rangle \langle ? \; ? \; ? \; ? \; E \rangle$

G_4 : $\langle J \; ? \; B \; ? \; ? \rangle \langle ? \; ? \; B \; ? \; E \rangle \langle J \; ? \; ? \; ? \; E \rangle$

G_5 : $\langle J \; ? \; ? \; ? \; E \rangle$ ⎫

S_5 : $\langle J \; ? \; ? \; ? \; E \rangle$ ⎭ → همگرایی

S_4 : S_3

S_3 : $\langle J \; ? \; B \; ? \; E \rangle$

S_2 : S_1

S_1 : $\langle J \; H \; B \; BD \; E \rangle$

S_0 : $\langle \emptyset \; \emptyset \; \emptyset \; \emptyset \; \emptyset \rangle$

جواب : $\langle J \; ? \; ? \; ? \; E \rangle$

Subject:
Year. Month. Date. (8)

— فرض این الگوریتم این است که مثال غلط وجود نداشته باشد

☑ ادامه مثال

G_5 : < J ? ? ? E >

G_6 : G_5

G_7 : algorithm collapses ✗

S_7 : S_6

S_6 : < J ? ? ? E >

S_5 : < J ? ? ? E >

— Collapse ← یک مثال نویزی در کل داده‌ها وجود دارد.
↙ نحوه نمایش برای مفهوم هدف درست نیست. ← ترکیب عطفی
دیزایهای کافی نیست.

— راه حلها ← تعریض فضای فرضیه ← ☑ تعریض تابع ارزیابی علمی
↙ جمع آوری مثال درست و جامع

— در واقع الگوریتم زمانی collapse می‌کند که مفهوم هدف در فضای فرضیه نباشد.

— برای اینکه محدود به ترکیب عطفی نباشیم ← بازنویسی بدون بایس ← گسترش فضا
☑ به فضای فرضیه NOT و OR را هم اضافه کنیم.

S → OR (مثالهای مثبت)

G → OR (NOT (مثالهای منفی))

overfitting ← S تنها برای مثالهایی که تاکنون دیده جواب می‌دهد.
↙ مثالهای آموزشی غلط می‌شود.
↙ چاره: دادن عمدی داده نویزی، محدود کردن فضای فرضیه

Subject :

Year . Month . Date . ( )

— یک سیستم یادگیری که هیچ پیش فرضی در مورد ماهیت تابع هدف نداشته باشد.

← قادر به دسته بندی داده های نادیده نخواهد بود.

← با دلیل صدرت مسئله باید آنرا تایین کرد.

☑ دسته بندی داده جدید در شال Enjoy Sport

1. مقایسه با هر 5 تا شال ← رای گیری

✎ تمارین شماره 3، 4، 5 و 9 از فصل دوم + خلاصه مقاله

مشکلات یادگیری ماشین

- نویز در مثالهای آموزشی
- محدودیت فضای فرضیه ← عدول با یک یادگیری امکانپذیر است
- مدت تعلیم مثال ها

Subject:
Year. Month. Date. 9

◎ یادگیری درخت تصمیم — Decision Tree

* مسئله یادگیری با درخت تصمیم
  - نحوه نمایش فرضیه ها ← درخت تصمیم به صورت Sum of Products
  - روش یادگیری ← ID3

* درخت تصمیم
  - تعدادی گره ها ← گره میانی
    - برگ
    - ریشه
  - تعدادی شاخه
  - بازسازی فرایند تصمیم گیری انسان

- گره ها = ویژگی
- شاخه ها = حالات مختلفی که ویژگی می پذیرد
- برگ ها = تصمیم

- یادگیری ← مطالعه
  - مثال های آموزشی

این سیستم های جزء برای تشخیص پزشکی کاربردی بودند چون هم دانش کتابها را داشتند هم مثال های آموزشی. ولی مشکلاتی داشتند.

- مثال های جدید ← مثال های خیلی زیاد
  - قدرت تعمیم زیاد
- اختلاف قوانین

Subject:
Year. Month. Date. ( )

- نقش مدل ... هر مدلی غلط است (البته تا حدی)
  - ساده سازی دنیای واقعی
  - استفاده از داده‌هایی که دقیقتر از مدل مغزی پزشک است.
  - دسته‌بندی ← قابل مدل کردن است

- سمن ششم !

* نمایش فرضیه
  - درخت + و -
  - بازنمایی درخت

* کاربرد درخت تصمیم
  - برای مسائلی مناسب است که شاهای آموزشی به صورت زوج مشخص باشند.

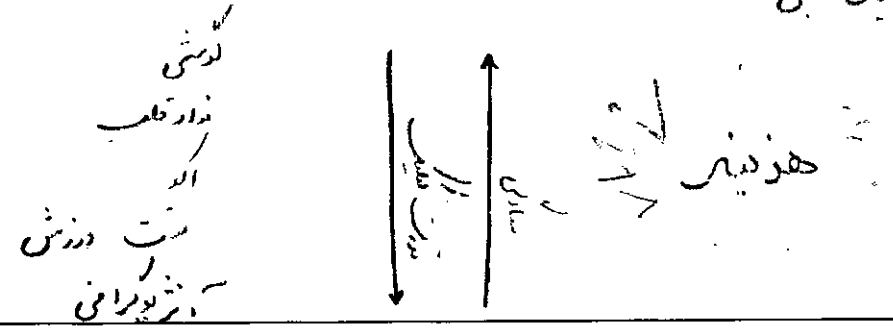
⟨ Feature , Value ⟩ → Class

- تابع هدف دارای خروجی گسسته است و نیاز به توصیف فصلی داشته باشد.

* سؤالهای اساسی
  ? اول کدام ویژگی آزمایش می‌شود؟ ← تفکیک کنندگی بیشتر
  - اهمیت
  - الگوریتم سادگی

☑ در بیمارستان، اول تب را می‌گیرند.

☑ بیماری طبی



Subject:
Year. Month. Date. 10

• درخت تصمیم برای انتخاب ویژگی بر مبنای مقدار تفکیک پذیری است و راه ریاضی دارد. این روش ساده‌ای را ارائه می‌کند.

✓ هنر حل یک مسئله، تبدیل یک مسئله به ویژگی‌هاست.

- بیماری لثه‌ش داغ، شاعر یک شعر، قیمت سهام، نوع سند
- کتاب، مصاحبه، جستجو، ...
- در یادگیری ماشین فضای نرخ‌ها مهم است ← بایاس

* ویژگی‌های درخت تصمیم

- ترتیب ندارند گسسته
- مقادیر در برابر ویژگی دودویی
- برای داده‌های با حجم بالا ← Data Mining
- قابل نمایش به صورت توابع if-then ← قابل فهم
- ترکیب عطفی و فصلی
- کار با داده‌هایی که بردارشان ناقص است Missing Features ←

✓ گفتیم غلط نیست که آخرا دوباره بیاندازیم

* درخت تصمیم ≡ Sum of Products (SoP)

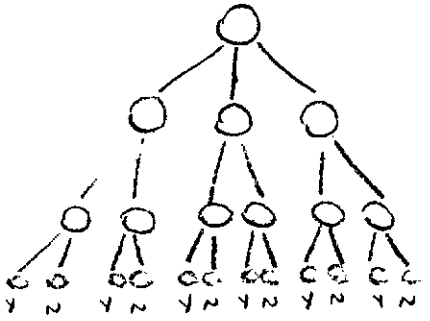
مسیر از ریشه به برگ ≡ AND

خود درخت ≡ OR

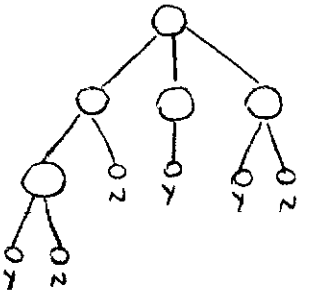
Subject:
Year. Month. Date. ( )

* درخت تصمیم مثالها را حفظ می‌کند ← درخت ساده تر

ID3 بایاس الگوریتم



ساده سازی



درخت کامل

پیچیده ترین درخت ممکن

تابع دولی پیچیده

تعداد سطحی برابر تعداد ویژگیها

درخت تصمیم

☑ علت استفاده از فقط ترکیب عطفی در الگوریتم حذف مانده قدرت تفهیم است.

* چقدر درخت تصمیم وجود دارد که با داده ها دقتی سازگاری دارد؟ «زیاد»

- کشیدن همه درختها ← آزمایش همه
- پیچیده ترین درخت ← هرس کردن
- ساخت درخت با خط ویژگی خاص ← قدرت تفهیم

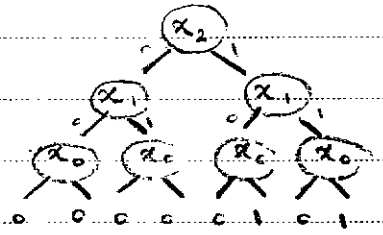
Subject :

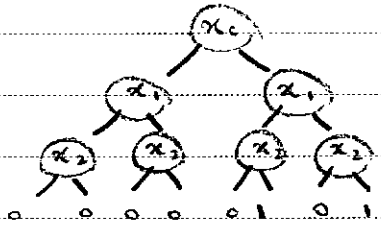
Year Month Date 11

☑ ریشه؟

تابع هدف : $f(x_2, x_1, x_0)$

x_2	x_1	x_0	class
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1





* به دنبال چه درختی هستیم ؟

• درخت کامل ← هزینه ساخت بالای فضای بزرگ

✓ مثال را حفظ می‌کند ← تعمیم ×

• درخت ID3 ← Inducing Dicision Trees

ساخت Quinlan

برمبنای الگوریتم پایه CLS

نسخه بهبود یافته C4.5

برمبنای ایده : دنیا ذاتاً ساده است

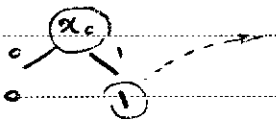
ساخت کوچکترین درخت تصمیم که با داده‌ها سازگار باشد.

☑

سازگاری : مثال‌های ۰، ۲، ۳، ۶

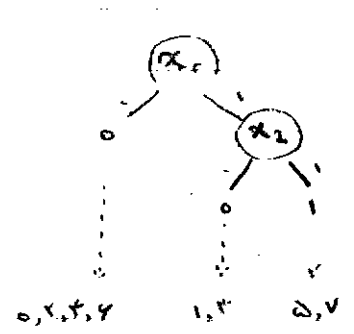
دارای تعدادی خطا در داده آموزشی ⟷ قابلیت تعمیم

:) :(

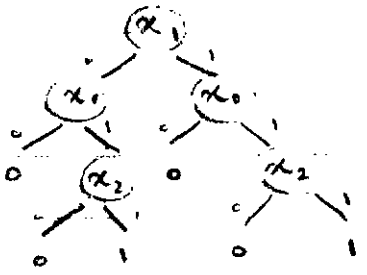


Subject :

Year . Month . Date . ( )



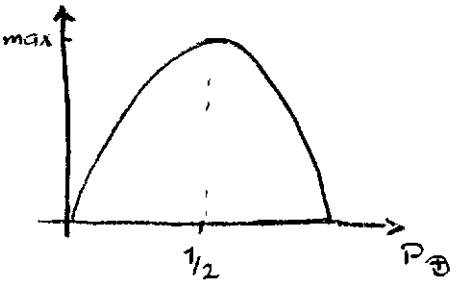
اضافه کردن ویژگی x_2
سازگار با داده های آموزشی



x_1 در ریشه ← حداکثر کننده نسبت
x_1 می تواند با یک سوال تطبیق تعدادی از
داده ها را مشخص کند.
پس گزینه مناسبی برای ریشه نیست.

* ID3
- روش بازگشتی
۱- یک Feature را در ریشه بگذاریم.
۲- اگر به ۷ یا ۸ رسیدیم، متوقف شود.
۳- در غیر در صفت دوباره الگوریتم را اجرا می کنیم
- کدام ویژگی در ریشه
- نیاز به معیار کمی ← آنتروپی

{ ۷ , ۷ , ۷ , ۷ } ⟶ S : ۰
{ ۸ , ۸ , ۸ , ۸ } ⟶ S : ۰
{ ۷ , ۷ , ۷ , ۸ } ⟶ S : $\frac{1}{4}$
{ ۷ , ۷ , ۸ , ۸ } ⟶ S : $\frac{1}{2}$



• آنتروپی
- میزان بی نظمی
- جدول

$$S = - P_{\oplus} \log_2 P_{\oplus} - P_{\ominus} \log_2 P_{\ominus}$$

$P_{\oplus}$: درصد مثال‌های مثبت
$P_{\ominus}$: درصد مثال‌های منفی

- قرارداد

$$0 \times \log 0 = 0$$

• الگوریتم سعی می کند در هر مرحله آنتروپی را تا جای ممکن کاهش دهد (ΔS_{max})

☑

$x_6 \xrightarrow{0}$ ۰, ۲, ۴, ۶
$\searrow^{1}$?

• مجموعه نمونه‌های S
ویژگی A
بهره اطلاعات G (information gain)

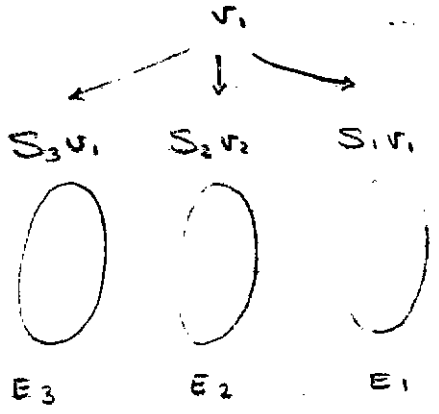
$$Gain(S, A) = Entropy(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Subject:
Year. Month. Date. ( )

• کاربرد در درخت

۱- ساده ترین درخت

مجموعه تمام مثالهای آموزشی (S)



$$G = E - \frac{S_1}{S} E_1 - \frac{S_2}{S} E_2 - \frac{S_3}{S} E_3$$

۲- حال یک ویژگی انتخاب شده است، آنرا از مجموعه ویژگیها حذف می کنیم و برای هر زیر درخت دوباره روند بالا را ادامه می دهیم.

Enjoy Water Sport ☑

تابع ویژگی

مثالهای نمونه

ویژگی ٤ < بله ، خیر >

درخت تقسیم مناسب است.

$$P_{\oplus} = \frac{9}{14} \qquad P_{\ominus} = \frac{5}{14}$$

$$E(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$$

با فرض قرار دادن هر ویژگی در ریشه، مانند ویژگی ٤ صدق می کند.

Subject:

Year. Month. Date. 13

outlook

S O R

N: D_1 D_2 D_8 — D_6 D_{14}

Y: D_9 D_{11} D_3 D_7 D_{12} D_{13} D_4 D_5 D_{10}

$w_1 = \frac{5}{14}$ $w_2 = \frac{4}{14}$ $w_3 = \frac{5}{14}$

$E_1 = -\frac{2}{5}\lg\frac{2}{5} - \frac{3}{5}\lg\frac{3}{5}$ $E_2 = 0$ $E_3 = E_1$

$G(S, outlook) = E(S) - w_1 E_1 - w_2 E_2 - w_3 E_3 = 0.246$

humidity

H N

N: D_1 D_2 D_8 D_{14} D_6

Y: D_9 D_3 D_{12} D_5 D_7 D_9 D_{10} D_{11} D_{13}

$G(S, humidity) = 0.15$

$G(S, wind) = 0.05$

$G(S, temperature) = 0.03$

پس outlook در ریشه قرار می‌گیرد.

چون با دیدن رضیت «O» تکلیف تا حدی مشخص است.

- نرم افزار پیاده سازی C4.5 ← Weka
- تمرین سری دوم: درختهای تقسیم

Subject:
Year. Month. Date. ( )

Overfitting *

• کلیه داده های آموزشی علت آن است.

☑ ترتیب مهم

۱- نیز ۲- نیز ۳- نیز ← پس از مرحله ترتیب شود نیز می آید.

$Err(train(h_1)) < Err(train(h_0))$

یعنی داده های آموزشی را خوب یاد گرفته است.

$Err(D(h_1)) > Err(D(h_0))$

یعنی به طور کلی بدتر عمل می کند.

• علت دیگر آن وجود نویز یا خطا در داده های آموزشی است.

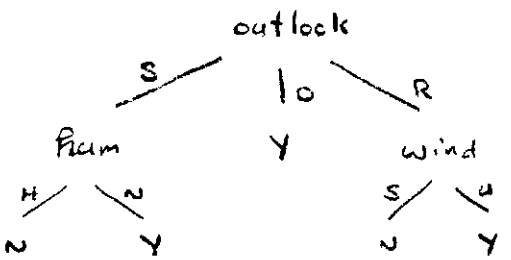
☑ درخت تصمیم

اندازه درخت ↑ ← دقت در داده آموزشی ↑
↓ ↙ دقت در داده تست ↓

علت: overfitting

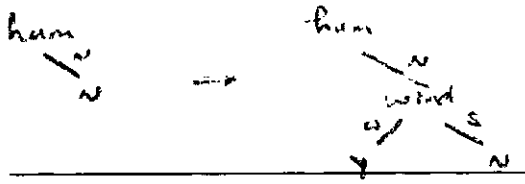
☑ داده ناقص در درخت تصمیم

درخت های شامل شدن آن
مثال غلط، بزرگتر می شود.
فضای فرضیه پیچیده تر می شود.



مثال غلط

<outlook = S, tem = M, hum = N, W = S> ∈



Subject:
Year. Month. Date. 14

• راه‌های جلوگیری

۱- جلوگیری از رشد درخت قبل از رسیدن به مرحله یادگیری کامل داده‌ای آموزشی

۲- هرس کردن درخت پس از رشد

• هرس کردن

- نیاز به تقسیم داده‌ها ← ۷۰٪ آموزشی + ۳۰٪ تست

✓ Cross-validation ← کمبود داده‌ها

لزوم داده‌ها تست باشد.

- الگوریتم

1. Train with ID3 over train set
2. Prune a branch
3. Replace branch with Y, N
   choose Y, N regarding maximum vote of pruned leaves
4. Test Tree
5. If it performs better, keep it otherwise restore original tree
6. Go to 2 for next branch

• علت‌های دیگر overfitting

- فضای فرضیه بزرگ
- فضای بدون بایاس

← حفظ کردن مثال‌ها

• نشانه‌های overfitting

- اختلاف زیاد بین داده‌های آموزشی و تست از لحاظ دقت
- رسیدن به فضای فرضیه پیچیده و ناخواسته

Subject:

Year. Month. Date. ( )

• ممکن است ویژگی خاصی دارای Info Gain بالایی باشد به طوری که یک مسأله تقریباً تطبیق داشته باشد ولی قدرت تعمیم را از بین می‌برد.

☑ اضافه کردن تاریخ ← Enjoy Sport

برای پیدا کردن چنین ویژگی می‌توان از سه خطوط زیر استفاده کرد:

- آزمایش با داده‌های Validation
- آستانه گیری از تعیین آنتروپی
- تعداد زیاد فرضیه‌ها

$$\text{Gain Ration}(S,A) = \frac{\text{Gain}(S,A)}{\text{Split Information}(S,A)}$$

$$= -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log \frac{|S_i|}{|S|}$$

☑

Split = n ⟶ SI = $\log_2^n$

Split = 2 ⟶ SI = $\log_2^2 = 1$

Split = 1 ⟶ SI = $\log_2^1 = 0$! ⟶ تطبیق معلوم است اصلاً ویژگی نخواهد خواست مخرج ۰ می‌شود

* ویژگی‌های پیوسته

• باید گسسته سازی شود.

• یکی از راه‌ها sort کردن داده‌ها و گسسته‌بندی آنهاست.

Subject:

Year. Month. Date. 15

Tem	40	48	60	72	80	90
	N	N	Y	Y	Y	N

A | B | C

$\frac{48+60}{2}$ $\frac{80+90}{2}$

* Missing Values

- این راه در نظر گرفتن حالت اضافه برای کل ویژگی تحت عنوان «تعریف نشده»
- قرار دادن هر مثال و قرار دادن مقادیر مختلف ویژگی برای آن‌ها
- استفاده از دانش پیشین

☑ بیماری بدن بیمار بررسی شده

۱- بیماری ← Y ، N ، ① ← وجود ندارد

۲- < —— ، —— ، Y ، —— >

< —— ، —— ، ? ، —— > < —— ، —— ، N ، —— >

۳- اگر حامله بیماری هستند / در این سن بیماری کشنده (نوزاد) / ...

Rule Post Pruning چیست؟

Subject:
Year. Month. Date. ( )

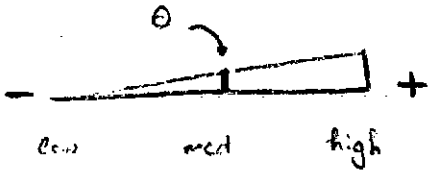
یادگیری بیزین

Spam Filtering

یک سری کلمات خاص ← موضوع
لیست سیاه Email
تعداد زیادی از یک فرستنده مشخص
بازخورد کاربر ← همه موارد مجدد به تحلیل می‌شوند.
مثلاً ← HoneyPot

alaki @ aut.ac.ir

مراحل:
۱- تمیز کردن Email مزاحم
۲- تجزیه و تحلیل
۳- فرستن مثال جدید و آستانه گیری



* استدلال بیزین

- براساس احتمالات
- اساس این روش بر این است که برای هر قطعیت یک درجه احتمال وجود دارد.
- مشاهده داده جدید ← استدلال بر مبنای تغییر احتمال آن ← تصمیم بگیرد

Spam

$P(x)$ = Email یک ← spam بودن ← احتمال اولیه = 50 %.

نامه جدید رسیده است، حال با بازخورد می‌دهد.

$P(x)$ = ↑ یا ↓ = احتمال تابعیت

Subject:
Year. Month. Date. 16

• مزیت : کاملاً معلوم است که چه می کند. ← پایه‌ی ریاضی قوی

* یادگیری با بیزین
• دانش موجود را به صورت احتمالاتی مدل می کند. ← توزیع احتمال
• داده‌ها را جمع آوری می نماید. ← روش آماری ← احتمال ادله
• مشاهده داده‌ها ← احتمال ثانویه
• احتمال ثانویه
  ↙ اندازه گیری عدم قطعیت
  ↓ پیش بینی
  ↘ تصمیم گیری

* تفاوت با درخت تصمیم
• پیش بینی احتمالاتی در برابر جواب قاطع
Yes / No ← 93٪ ←

* مشکلات عملی
• نیاز به دانش ادله در مورد تعداد زیادی متغیر ادله
☑ احتمال دیدن هر کلمه در Email های سالم و Spam
☑ ترم کامپیوتری I Love U

• وجود نداشتن اطلاعات قبلی

• تخمین زدن احتمال
☑ معاینه‌ی روبات

• محاسبات پرهزینه

Subject :

Year . Month . Date . ( )

- انتخاب توزیع احتمال مناسب برای بایس کردن
- ☑ فرض توزیع نرمال برای استفاده از قیلتر کالمن

* محاسبات

- h : فرضیه ای از میان فضای فرضیه H
- D : داده های آموزشی
-

$$P(h|D) = \frac{P(D|h) \times P(h)}{P(D)}$$

☑ Spam Filtering

h_1 = Spam h_2 = not spam

D = ۴ email مجموعه

☑ تشخیص شاعر شعر

h_1 = فردوسی h_2 = حافظ ... h_5 = بیدل

D = مجموعه اشعار

- $P(h)$: احتمال اولیه ، سئل از مثال مورد بررسی Prior prb.
- $P(h|D)$: احتمال ثانویه ، پس از دیدن مثال Posterior prb.
- $P(D|h)$: احتمال اینکه داده خاصی به فرضیه خاص متعلق باشد . likelihood
- $P(D)$: احتمال دیدن داده evidance

☑

$P(h|D) \longrightarrow P(\text{spam} \mid \text{email 1})$

Subject:
Year. Month. Date. 17

$P(D) \uparrow \Rightarrow P(h|D) \downarrow$

☑ احتمال اینکه از روی کلمه Salam بشود حدس زد که spam است کم است چون این داده زیاد دیده می‌شود.

* تصمیم‌گیری از روی احتمال ثانویه

• ماکزیمم پستین Maximum Aposteriori Hypothesis (MAP) ☑

P(Spam) = 70 %
P(~Spam) = 30 % ⟹ Spam

$h_{MAP} = \underset{h \in H}{\arg\max}\ P(D|h) \times P(h)$

— می‌توان از مخرج صرفنظر کرد ← Likelihood
— مجموع آنها «1» نمی‌شود.
• با هم اگر محاسبه $P(h)$ دشوار باشد یا مهم نباشد می‌توان آنرا حذف کرد.

$h_{MAP} = \underset{h \in H}{\arg\max}\ P(D|h)$

(ML) Maximum Likelihood Hypothesis —

Subject:
Year. Month. Date. ( )

☑ آزمایش سرطان

h_1: cancer $\longrightarrow P(\text{cancer}) = 0.008$

h_2: $\sim$ cancer $\longrightarrow P(\sim \text{cancer}) = 0.992$

$P(+ \mid \text{cancer}) = 0.98$

$P(- \mid \sim \text{cancer}) = 0.97$

شخصی به آزمایشگاه می‌رود، نتیجه آن مثبت می‌شود، آیا سرطان دارد؟

$P(\text{cancer} \mid +) = ?$

محاسبات

$$P(C \mid +) = \frac{P(+ \mid C) \times P(C)}{P(+)} = \frac{0.0078}{P(+)}$$

$$P(\sim C \mid +) = \frac{P(+ \mid \sim C) \times P(\sim C)}{P(+)} = \frac{0.0298}{P(+)}$$

$\Rightarrow h_{MAP} = \sim \text{cancer}$

کلا هر کسی به بیمارستان می‌آید می‌گوییم سالم است.

سالم باشد ← از نتیجه آزمایشگاه شکفتی می‌کند

سرطانی باشد ← سرطان ادرای کند

☑

$P(h_1 \mid D) = 0.4$

$P(h_2 \mid D) = 0.3$

$P(h_3 \mid D) = 0.3$

انتخاب MAP فرضیه h_1 است.

حال اگر مدرکی مانند x موجود باشد که:

$P(h_1) = +$, $P(h_2) = -$, $P(h_3) = -$

استفاده از MAP بهینه نیست.

Subject:

Year. Month. Date. 18

* ترکیب بهینه bayes optimal classifier

$$h = \arg\max_{v_j \in H} \sum_{h_i \in H} p(v_j | h_i) \cdot p(h_i | D)$$

- v_j : دسته بندی
- $p(v_j | h_i)$: درستی پیشنهاد از فرضیه
- Optimal : تاثیر وزن دهنده در احتمال تاثیر دادن

☑

h_1 = Spam due to email subject +

h_2 = Spam ... context −

h_3 = Spam ... sender −

h_4 = Spam ... subject + context +

$$\text{Spam} = \sum p(+ | h_i) \times p(h_i | D)$$

- نیز عملی برای فضای فرضیه بزرگ

* دسته بندی کننده بیزین ساده Naive bayes classifier

☑ دسته بندی متن ← روزنامه

- شرایط کاربرد ← ترکیب با ترکیب منطقی ← مثل یادگیری مفهوم
  - استقلال شرطی
  - تابع هدف دارای متغیر محدود
  - مجموعه مثالهای آموزشی بزرگ ← محاسبه احتمال

Subject:

Year. Month. Date. ( )

تابع هدف $f : X \longrightarrow V$

$x \in X : a_1, a_2 \dots a_n$

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j \mid a_1 \dots a_n)$$

$$= \arg\max_{v_j \in V} \frac{P(a_1 \dots a_n \mid v_j)\, P(v_j)}{P(a_1 \dots a_n)}$$

$$= \arg\max_{v_j \in V} P(a_1 \dots a_n \mid v_j)\, P(v_j)$$

$P(v_j) =$ تعداد دفعات رخ دادن v_j در مثالهای آموزشی

- محاسبه $P(a_1 \dots a_n \mid v_j)$
  - داده آموزشی زیاد
  احتمالاً چند مثال موجود است.

- فرض استقلال شرطی
  حاصلضرب احتمالات تک ویژگی‌ها
  در هم مستقل هستند.

$$P(a_1 \dots a_n \mid v_j) = P(a_1 \mid v_j) \times \dots \times P(a_n \mid v_j)$$
$$= \prod_{i=1}^{n} P(a_i \mid v_j)$$

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_{i=1}^{n} P(a_i \mid v_j)$$

Subject:
Year. Month. Date. 19

Play Tenni's ☑

query: $x = (O=s, T=c, H=h, W=s)$

$$V_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_{i=1}^{n} P(a_i | v_j)$$

یادگیری ≡ محاسبه مقادیر احتمال به طور Offline

$V = \{yes, no\}$

$P(yes) = 9/14$

$P(no) = 5/14$

Outlook = Sunny	Temp = Cool	Hum = Normal	Wind = Strong
$P(O=s \mid yes) = 2/9$	$P(T=c \mid yes) = 3/9$	$P(H=h \mid yes) = 3/9$	$P(W=s \mid yes) = 3/9$
$P(O=s \mid no) = 3/5$	$P(T=c \mid no) = 1/5$	$P(H=h \mid no) = 4/5$	$P(W=s \mid no) = 3/5$

$P(yes \mid x) = 9/14 \times (2/9 \times 3/9 \times 3/9 \times 3/9) = 0.0053$

$P(no \mid x) = 5/14 \times (3/5 \times 1/5 \times 4/5 \times 3/5) = 0.0203$ ✓

- مقاوم در برابر Noise
- ویژگی‌های مستقل نباشند ← جواب معنی ندارد.
- در جامعه آماری بزرگ بی‌تأثیر نیست ← تخمین دارد.
- اعداد بدست آمده هم لگاریتمی هستند

• گاهی یک ویژگی در نمونه‌های آموزشی دیده نمی‌شود ($P(a_i|v_j)=0$)
از m-estimate of prob. استفاده می‌شود.

Naïve ⟶ $P(\cdot)=\frac{n_c}{n}$

m-estimate ⟶ $P(\cdot)=\frac{n_c + m.p}{n+m}$

معمولاً اگر k دسته وجود داشته باشد $p=1/k$ که مثلاً مثل اینست که به تمام دسته‌ها m مثال اضافه کرده باشند

☑ تعیین موضوع وبلاگ

۱. ترکیب عطفی ویژگی‌ها ⟶ کلمات ⟶ کلیدی / عنوان پست / لینک

از روی هر پست، ۱۰۰ کلمه اول را بررسی داریم و آنرا به عنوان عنوان در نظر می‌گیرند

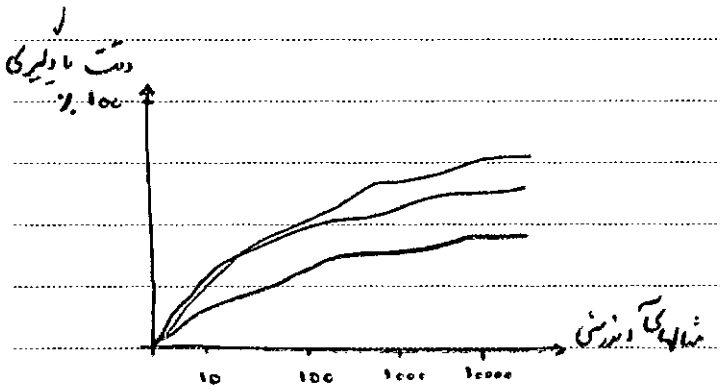
۲. استقلال شرطی
تقریباً قبول می‌کنیم.

۳. تابع هدف محدود
دوستی، داستان، خاطرات، ... ⟶ محدود

۴. مجموعه آموزشی
پست‌های وبلاگ‌های مختلف با دسته‌بندی معلوم

Subject:

Year. Month. Date. 20



سختی باد لیزری

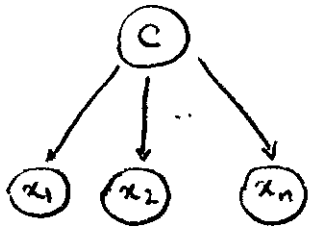
Subject :

Year . Month . Date . ( )

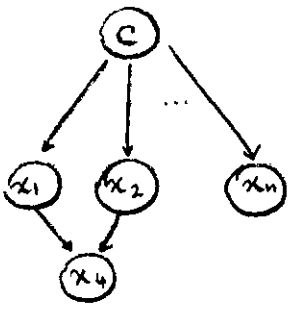
Bayesian Belief Net

شبکه باور بیزی

شبکه باور بیزین



دسته بندی کننده باور بیزی



Node : Concept

Arc : Depedancy

ادامه دارد ...

Subject:
Year. Month. Date. 21

ارزیابی فرضیه ها Hypothesis Evaluation

* بایاس

Data
↳ Learning Algorithm
↳ Hypothesis
↳ Performance Assessment

* کمبود داده های آموزشی
  - تمایل به توزیع طی داده ها
  - اندازه داده ها مناسب
  - استفاده از داده های مرجع
  - توزیع داده ها لزوما مناسب نباشد.
  - مشکلات کمبود داده

— بایاس : اختلاف بین دقت واقعی و دقت تخمین زده شده
— واریانس : پراکندگی نتایج در آزمایشهای مختلف

* مدل کمی

$p(x)$ = احتمال رخداد هر نمونه x (نامعلوم)

- خطای نمونه

خطای فرضیه روی نمونه های موجود

$error_s(h) = \frac{e}{n}$

- خطای واقعی

خطای فرضیه روی کل داده های مرجود = احتمال اینکه یک نمونه به غلط دسته بندی شود.

بدلیل در دسترس نبودن خطای واقعی از روی خطای نمونه است.

Subject:
Year. Month. Date. ( )

در واقع صورت مسئله عبارتست از تخمین خطای واقعی از روی خطای نمونه

• خطا

$$\delta(f(x), h(x)) = \begin{cases} 0, & f(x) = h(x) \\ 1, & f(x) \neq h(x) \end{cases}$$

$$error_D(x) = \sum_{x \in D} p(x) \cdot \delta(x) \qquad S \subseteq D$$

$$error_S(x) = \sum_{x \in S} p(x) \cdot \delta(x)$$

• اگر اطلاعات بیشتری موجود نباشد، بهترین مقدار برای e_D همان e_S است

• با احتمال 95٪ خطای واقعی در بازه زیر است:

$$error_S \pm 1.96 \times \sqrt{\frac{error_S \times (1 - error_S)}{n}}$$

فرض ← مثالهای مستقل از یکدیگر

$n \geq 5$

زمانی تخمین بهتر برای تقریب بدست می‌آید که:

$n \cdot error_S \times (1 - error_S) \geq 5$

N٪ $\iff$ Z_n

95٪ $\iff$ 1.96 , 99٪ $\iff$ 2.58

Subject:

Year. Month. Date. 22

• این رابطه از توزیع دو جمله‌ای ناشی می‌شود

p = احتمال موفقیت یک آزمایش ⟵ احتمال ارزیابی غلط یک نمونه (مجهول)

r = تعداد موفقیت‌ها ⟵ تعداد دفعات ارزیابی‌های غلط از بین n نمونه

N = تعداد کل نمونه‌ها

$$P(r) = \frac{N!}{r!\,(N-r)!}\, p^r \times (1-p)^{N-r}$$

$$E\{p(r)\} = N \times p$$

$$Var\{p(r)\} = N \times p \times (1-p)$$

$$Bias = error_D - error_S$$

$$= E\{p(r)\} - p(r)$$

$$= (N-1) \times p$$

$$\sigma_{error_S} = \frac{\sigma_r}{S} = \sqrt{\frac{p(1-p)}{n}}$$

(انحراف معیار خطای نمونه ⟵ خطای واقعی ⟵ مجهول

برای تقریب زدن این مقدار از رابطهٔ زیر استفاده می‌شود

$$p \approx error_S \implies \sigma_{error_S} \approx \sqrt{\frac{error_S \times (1-error_S)}{n}}$$

• فاصله اطمینان برای توزیع دو جمله‌ای سخت است.

با فرض n بزرگ توزیع دو جمله‌ای به سمت توزیع نرمال میل می‌کند.

N٪ فاصله اطمینان $= Bias \pm z_N \times Variance^{1/2}$

Subject :

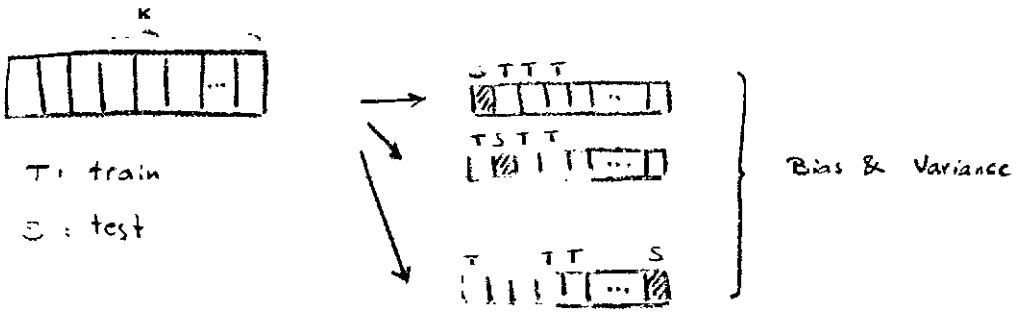
Year . Month . Date . ( )

* مقایسه الگوریتم ها ی یادگیری

1 روش ⟶ E_1 ٪ خطا ⟶ $E_1 \pm \sigma_1$ خطای واقعی

2 روش ⟶ E_2 ٪ خطا ⟶ $E_2 \pm \sigma_2$ خطای واقعی

* K-Fold Cross - Validation



Subject:
Year Month Date 23

یادگیری بر پایه نمونه — Instant Based learning

Concept learning

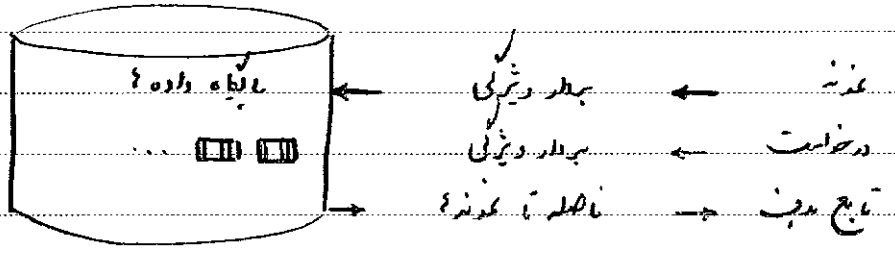
Decision Tree

Bayesian learning

در این روش‌ها پس از استفاده از نمونه برای ساختن مدل، دیگر به نمونه‌ها کاری نداریم.
در یادگیری بر پایه نمونه از خود نمونه‌ها استفاده می‌شود.

IBL

- مثال‌های آموزشی ذخیره می‌شوند. ← Memory Based
- با مشاهده مثال جدید، با فاصله‌گیری از مثال‌های ذخیره شده بررسی می‌شود و یک مقدار برای تابع هدف پیش‌بینی می‌شود.
- پردازشی در مرحله آموزش ندارد ← lazy
- فرضیه‌ای ایجاد نمی‌شود.
- تعمیم دادن تا مشاهده مثال جدید به تعویق می‌افتد.



- ترتیب حداکثرهای از تابع هدف را ایجاد می‌کند. این ترتیب فقط به همسایگی نمونه جدید قابل اعمال بوده و پرس‌وجوی مداوم بر روی فضای نمونه‌ها عمل می‌کند ← ترتیب محلی

☑

تقریب سراسری { درخت ← تمام داده ها
DT: تمام داده ها
BN: توزیع احتمال ← تمام داده ها

- قدرت یادگیری تابعهای هدف پیچیده ← ترکیب توابع ساده‌تر محلی

* چه زمانی استفاده می‌شود؟
  - مثالها به شکل زوجهای صفت-مقدار باشند
  - تعداد دسته‌های محدود
  - تعداد داده آموزشی زیاد ← بررسی فضای داده ها

* مزایا
  - آموزش سریع ← ذخیره در حافظه داده
  - تابع هدف پیچیده را یادگیری کند
  - اطلاعات از دست نمی‌دهد

* معایب
  - زمان جستجوی زیاد
  - حساسیت در برابر ویژگیهای نامربوط

☑ راه ارتقا ← خوشه‌بندی درونی ها و مقایسه با برچسب دسته ها

* توابع شباهت
  - توابع فاصله
  - عمل مقایسه ویژگیها امتیاز پذیر باشند

Subject:
Year. Month. Date. 24

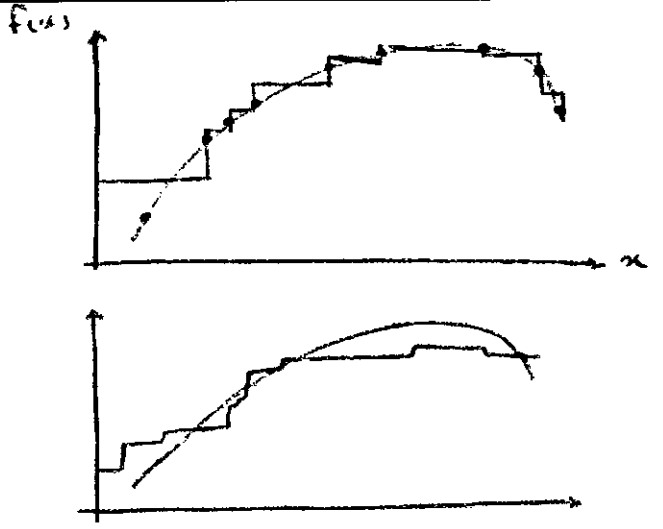
* انتخاب نمونه ها
  - تعداد زیاد
  - پوشش کل فضا
  - حتی الامکان نویزی نباشد.
  - تفکیک پذیر خطی

* تابع دسته بندی
  - با مشاهده یک مثال دسته بندی آن را تعیین می کند.

* روش های مختلف
  - K - Nearest Neighbor
    تابع هدف گسسته، مقادیر بردار
  - Locally Weighted Regression
  - Radial Basis Function Network
  - Case-Based Reasoning
  - General Regression Neural Networks

* KNN
  - ساده ترین و متداول ترین روش
  - فضای n بعدی اقلیدسی
  - فاصله ی اقلیدسی
  - k همسایه نزدیک ← انتخاب مناسب این مقدار دشوار است.
  - تابع هدف ← گسسته ← رأی گیری بین دسته بندی همسایه ها
    ← پیوسته ← میانگین گیری بین دسته بندی همسایه ها

Subject:
Year. Month. Date.



K=1

K=3

- $k=1$ ← حساس به نویز
  ↙ ترتیب بهتر برای فضای چهار

- فضای نزدیکترین k-NN = voronoi diagram
  فضا را به چند جهه‌ای تقسیم می‌کند.
  تمام نقاط این چند وجهی دسته بندی مشابهی دارد.

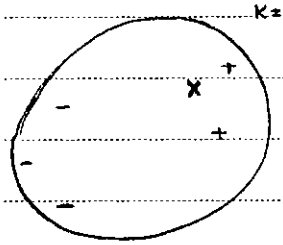
- مانهتن اسپرد
  دسته بندی نقطه مشابه دسته بندی نقطه‌ای دیگری خواهد بود که در نزدیکی قرار دارد.

- برای محاسبه فاصله اقلیدسی چون دیتاها را به عدد نرمال می‌کنیم.

$$x' = \frac{x - \bar{x}}{\sigma}$$

• نسخه بهبود یافته kNN

در حالت عادی اگر k بزرگ باشد، اثر نامطلوبی از داده‌های دوری گیرد



می‌توان تاثیر داده‌های دورتر را کم کرد. اینار با وزن دادن به آنها امکانپذیر است. این وزن عکس مجذور فاصله است.

$w_i = d^{-2}$

• Curse of Dimensionality

فاصله تا تمام ویژگیها اندازه‌گیری می‌شود. در خلاف روشهایی مانند درخت تصمیم که در آن سعی می‌شد تا فقط از ویژگی‌های مرتبط استفاده شود.

در اینحالت وجود ویژگیهای نامربوط مخل است. به طور گسترده‌ای روی فاصله تاثیر گذار باشد

یکی از راه حلها وزن دادن ویژگی‌ها مرتبط است.

برای تعیین وزن ویژگی‌ها می‌توان از روش cross-validation استفاده کرد.

روش به گونه‌ای تغییر می‌کند که خطای دسته‌بندی در نمونه‌های آموزشی کاهش یابد.

اگر ضریب یکسان صفر شود می‌توان یک ویژگی را حذف کرد.

این عمل در واقع وزن دادن باینری به ویژگیهاست.

• چگونه داده‌ها در DB را جستجو کنیم که جستجو سریعتر باشد برای همسایگان نزدیکتر همسایه Indexing

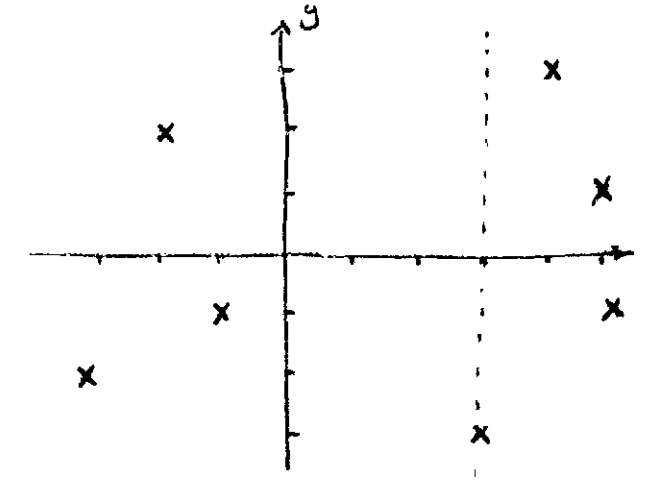
استفاده از روشهایی مانند kd-tree که در آن نمونه‌ها بر روی یک درخت ذخیره شده و نمونه‌های نزدیک به هم در همان گره یا گره‌های نزدیک به هم ذخیره شوند

سپس ابتدا به همین گره‌ها بررسی شود.

Subject :

Year . Month . Date . ( )

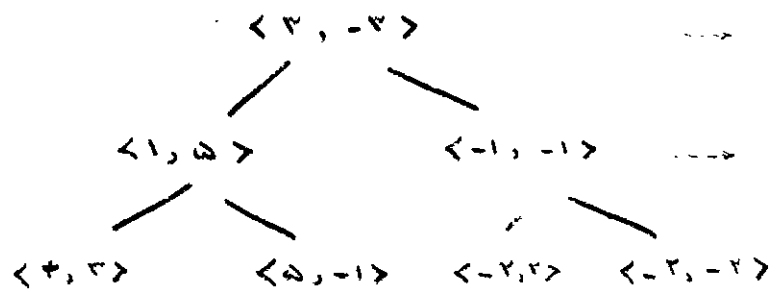
kd-tree



داده ها را انتخاب می‌کنیم که نقطه‌ای
دوتوی را نصف کند. برای هر محور همین
کار را تکرار می‌کنیم.

نصف محور x ⟵ $x_0 = 3$

داده مرتبط ⟨۳, -۳⟩



- معایب
  - داده اولویتم هنگام انجام دسته‌بندی کم است
  - تعیین یک تابع فاصله مناسب
  - تاثیر منفی داده‌های نامرتبط
  - حافظه بزرگ برای نگهداری همه داده‌ها

Regression : تقریب داده با منحنی خطی

Residual : خطای حاصل از تقریب

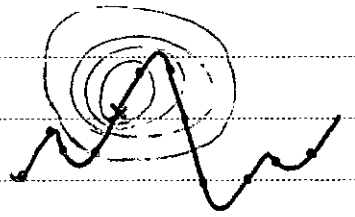
Kernel : تابع تقریب درون‌یابی → $1/d^2$, e^{-d}, $1/(1+d)$
رابطه خاص با فاصله داده

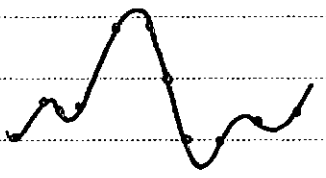
Subject:
Year. Month. Date. 26

Locally Weighted Regression *

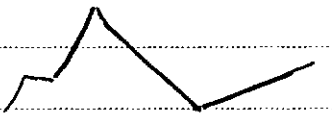
- تعمیم k-NN
- تقریب صریحی از تابع $\hat{f}$ حول ناحیه محلی در پیرامون نقطه مورد بررسی
- کل شاها را بادنی گیرد بلکه برای هر ناحیه یک تابع ساده یاد می گیرد.

LWR و k-NN





LWR



k-NN

- حالت خطی

از یک تابع خطی برای تقریب تابع هدف در نزدیکی شال مورد بررسی استفاده می شود.
برای تنظیم وزنها از شیب نزولی استفاده می شود.

$$\hat{f}(x) = w_1 a_1 + w_2 a_2 + \dots + w_n a_n$$

$$Err = \frac{1}{2}(f - \hat{f})^2$$

$$\Delta w = \eta \sum (f - \hat{f})\, a_i$$

$$w' = w - \Delta w$$

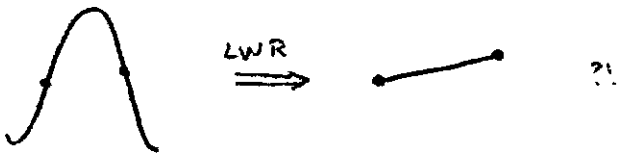
انتخاب تابع هدف برای تقریب خطی

- k همسایه
- تمام داده ها با اعمال کرنل وزنها دری آنها
- تمام داده ها بالاخص k همسایه

Subject :

Year . Month . Date . ( )

- مسئله آن تقریب زننده است



Radial Basis Functions *

- تابع در هر نقطه با یک تابع گوسین ، نظیر می‌گیرد - سطوحی تابع کرنل (K_u)
  میانگین روی داده آموزشی
  واریانس روی K همسایگی

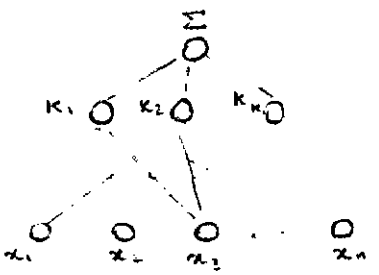
- تعداد توابع = تعداد نمونه‌های آموزشی

- شباهت با LWR ← تقریب تابعه

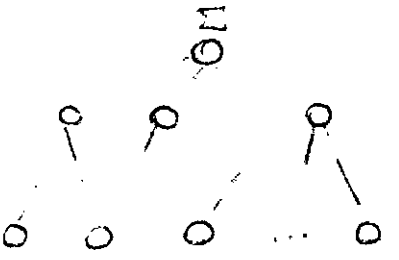
- تقریب نهایی برابر است با مجموع وزن‌دار توابع همسایگی تک تک نمونه‌ها

$$\hat{f} = \sum w_u \cdot K_u$$

- برای تابع گوسی سه چیز باید مشخص شود : میانگین ، واریانس ، وزن



استفاده از تمام نمونه‌ها



2 - NN

- همیشه تقریب درستی ارائه می‌دهد (اثبات شود)

Subject:
Year Month Date 27

Case Based Reasoning *

- مسئله روز
- یک مسئله یادگیری نمونه ← نمونه‌ها نقاطی در فضای n بعدی
  دسته‌بندی بر حسب به نمونه‌های مشابه
  یادگیری تنبل
- CBR ← ارائه سمبلیک
- اجزای CBR ← DB از مثال‌های قبلی
  یادگیری نمونه‌های مشابه
  Indexing ← تطبیق با شبیه‌ترین مثال
  بدست آوردن راه حل با کمک شبیه‌ترین مثال
  تطبیق راه حل ← تغییر دادن راه حل به طوری که اختلاف بین مثال و داده‌ی شبیه پیدا شده را در نظر بگیرد.
- چرخه

New Problem → Retrive Similar → Adapt → Solution

DB ← Retain ← Review ← CBR وارد

مانند سیستم انسانی

Subject :

Year . Month . Date . ( )

- در CBR ، مثالهای مشابه هم است را پیدا می‌کند و از روی آنها قانون استخراج می‌کند. استخراج به صورت خودکار انجام می‌شود یا توسط Expert مدیریت می‌گیرد (در حالت CBR ثبت)
- پس از پیدا کردن قوانین ، مشابهت بین مثال و داده‌های DB پیدا می‌شود و شبیه‌ترین داده استخراج می‌شود
- پس از استخراج داده ، تفاوتهای احتمالی با مثال پیدا کرده و با قانون مورد نظر اعمال می‌شود

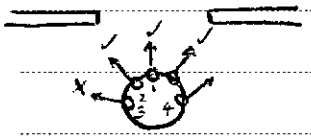
Subject:
Year Month Date 28

شبکه‌های باور بیزین Bayesian Belief Network

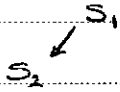
* مدل Naive Baysian : واحت شدن از دست شرط استقلال در بیزین

☑ ربات ـ تعدادی سنسور در اطراف خود دارد. ـ مدل: خارج شدن از در
سئوال: آیا در خارج شدن از در، مقادیر سنسورها از هم مستقلند؟

<S1, S2, S3, ..., Sn, h> h: door / door



سه سنسور بالا وابسته‌اند
دو سنسور سمت چپ وابسته نیستند



* مدلهای گرافیکی نام دیگر شبکه‌های بیزی است چون احتمالها را با گراف به هم مربوط می‌کنند
امکان ترکیب دانش قبلی (حیزه) در حل مسئله با این گراف میسر است.

- نود = متغیر، ویژگی
- یال = وابستگی
- جهت یال = رابطه علیت

* مزایای BN نسبت NN
- استفاده از دانش حیزه
- سادگی و قابل فهم
- کار با داده‌های ناقص

* استقلال شرطی

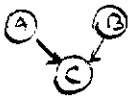
$P(X \mid Y, Z) = P(X \mid Z)$

Subject :

Year . Month . Date . ( )

$P(A)$

$P(B|A)$



$P(C|A,B)$ ←

A B	C=0	C=1
0 0		
0 1		
1 0		
1 1		

استخراج دانش اولیه ← ساخت شبکه ← با استفاده از دانش موجود

احتمالات ← توزیع شواهد داده ؟

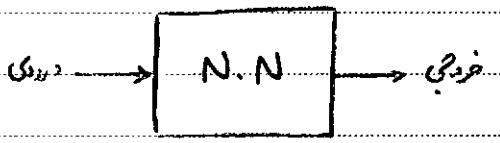
* بیزین BN

- ساختار معلوم ← مقدار احتمال شرطی بیزین
- ساختار مجهول ← معیاری برای مقدار بیزین
- ساختار بیزین

Subject:
Year. Month. Date. 29

شبکه‌های عصبی Neural Networks

* مزایا
  - یادگیری توابع خیلی پیچیده و یا برداری
  - مقاوم در برابر خطای داده‌های آموزشی
  - پرکاربرد و مدرن



رابطه ورودی و خروجی به سادگی معلوم نیست و
راحت است که تقریب زده شود. به خصوص
در توابع فیزیکی که رابطه باید کشف شود و مدل
ساخته شود.

* دیدگاه
  - بر پایه محاسبه
  - بر پایه اتصال واحدهای پردازشی (زیرمجموعه Connectionism)
  - هر واحد : نرون
  - ساختار : ورودی ← واحد پردازشی ← واحد پردازشی دیگر ← ... ← خروجی

لایه ورودی — لایه مخفی — لایه خروجی

- مدل استاندارد : هر واحد پردازشی به تمام خروجیهای لایه قبل متصل است.
- الهام از طبیعت : از مغز موجودات زنده

* قابلیت‌های شبکه‌های عصبی
  - محاسبه‌ی توابع معلوم

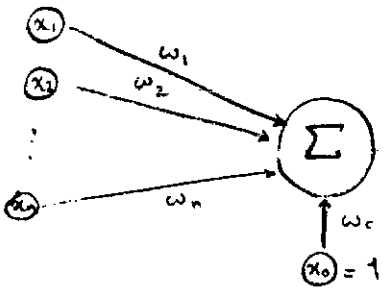
Subject :

Year . Month . Date . ( )

- تقریب یک تابع ناشناخته
- شناسایی الگو
- پردازش سیگنال
- یادگیری

* Perceptron

- نوعی از شبکه عصبی بر مبنای یک واحد محاسباتی
- عمل آن : جمع وزندار ورودیها (رابطه خطی)
  آستانه گیری از خروجی



- در واقع یک صفحه جداکننده را مشخص می‌کند.
- قابلیت تقریب توابع خطی
- نمی‌تواند توابعی که خطی جداپذیر نیستند را مانند تابع XOR یاد بگیرد
- ورودی ثابت : بایاس
- $\hat{f}(x_i) = \text{Sign}\left(\sum \omega_i x_i\right)$

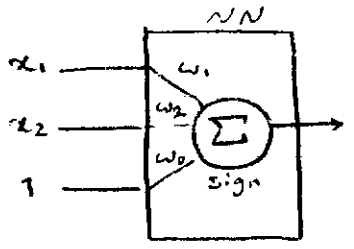
تابع AND

دو ورودی

یک خروجی

یک واحد پردازشی

یک بایاس



Subject:
Year . . . Month . Date . 30

- پادگیری پرسپترون : بدست آوردن مقادیر درستی برای w_i
- فضای فرضیه : مجموعه تمام مقادیر ممکن برای بردار وزن

فضای جستجو بسیار بزرگ

نیاز به روشی برای جستجو

- روش آموزش پرسپترون

۱- مقادیر اولیه وزنها ← تصادفی

۲- پرسپترون را به تک تک مثالهای آموزشی اعمال می کنیم

ارزیابی خطا ← تصحیح وزنها

۳- آیا تمام مثالها درست ارزیابی شده اند؟

بله ← پایان

خیر ← برو به مرحله ۲

x_2	x_1	y
۰	۰	۰
۰	۱	۰
۱	۰	۰
۱	۱	۱

w_0 w_1 w_2

۰ ۰ ۰ S۴ : $\Delta = 1$

۰.۱ ۰.۱ ۰.۱

⋮

- قانون آموزش پرسپترون

- استفاده از Δ
- اعمال ضریب (نرخ یادگیری)
- اعمال ورودی (برای مشخص کردن سهم آن از خطا)

$$w_i' = w_i + \eta (o - t) x_i$$

- همگرایی : اگر خطی جدایذیر باشند

PqPCO

Subject:
Year. Month. Date.

- قانون آموزش دلتا
  - استفاده از Gradient Descent
  - در بسته Least Square Error
  - اول خطا را تعریف می‌کند
  - ☑ متداول‌ترین خطا

$$E = \frac{1}{2} \sum_i (t_i - o_i)^2$$

روی تمام نمونه‌های آموزشی

$$o_i = \sum \omega_k x_k$$

به وزن ω وابسته است

- چالش اینجاست که خطا برابر صفر باشد، پس باید به سمت آن رفت.
- گرادیان نزولی شیب را به دست می‌دهد. پس از خطا منفی گرفته می‌شود.
- قانون اصلاح وزن‌ها

$$\Delta \omega_i = -\eta \frac{\partial E}{\partial \omega_i}$$

حرکت در جهت منفی گرادیان

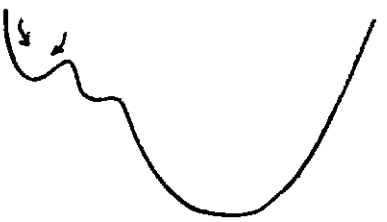
☑ برای پرسپترون

$$E = \frac{1}{2} \sum_i (t_i - o_i)^2$$

$$\frac{\partial E}{\partial \omega_i} = \sum_i (t_i - o_i)(-x_i)$$

$$\Delta \omega_i = \eta \sum_i (t_i - o_i)(x_i)$$

- مشکلات شیب نزولی



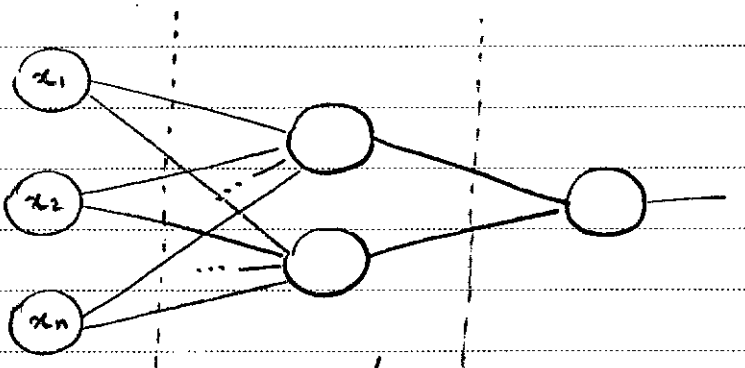
مینیمم محلی (تضمینی برای رسیدن به جواب محلی وجود ندارد)

Subject:
Year: Month: Date: 31

• راه اندازی (online)
- استفاده از Stocastic gradient desc.
- محاسبه خطا برای تک تک ورودی ها
- ماهی ۱۰ تا ۱۰۰ با فراز و میمیم محلی بخورد عمل می کند.

* پیچیدگی
- تعداد نرون ها
- تعداد لایه ها
- فضای جستجو
- پایداری سخت

* کاربرد
• مشخص کردن ورودی ها ← بردار ویژگی ها
• مثال های آموزشی
• تعیین تعداد لایه ها
• تعیین واحدهای پردازشی هر لایه
• تعیین توابع هر واحد پردازشی
- انتخاب تابع خروجی (کاربرد دسته بندی، اندازه ها، ...)
- پیچیدگی



تابع فعال ساز مثلا tanh برای شناخت روابط پیچیده
تابع دانسته به نوع تابع خروجی

Subject :

Year . Month . Date . ( )

Subject:
Year. Month. Date. ( )

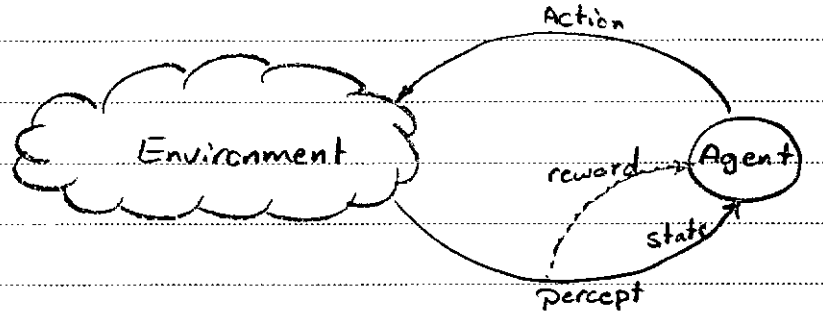
Reinforcement learning — یادگیری تقویتی

* انواع یادگیری
  - یادگیری با ناظر : ناظری داریم که کار او جمع آوری نمونه و دسته بندی آن است.

<دسته بندی ، نمونه> : ورودی
تقریب تابع یا رابطه : خروجی

  - یادگیری بدون ناظر : یادگیری خودکار دسته بندی
  - یادگیری تقویتی

* یادگیری تقویتی

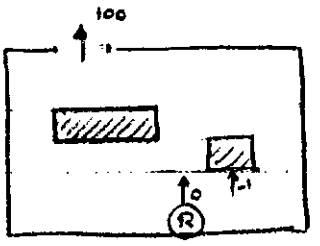


* استراتژی های اصلی
  - الگوریتم های زنتیک
  - استفاده از روشهای آماری و برنامه نویسی پویا
  - الگوهای یادگیری

* مدل سازی
  - محیط مجموعه‌ای از S حالت
  - مجموعه عملهای ممکن برای عامل A
  - پاداش محیط در برابر عملی که عامل انجام داده است ، r

Subject:
Year. Month. Date.

☑ مخارج شدن روبات از اتاق



* عامل : انتخاب عمل
محیط : پاداش + حالت بعدی

* عامل سعی می کند طوری رفتار کند که تابع پاداش ماکزیمم شود.

* انواع پاداش
• تجمعی

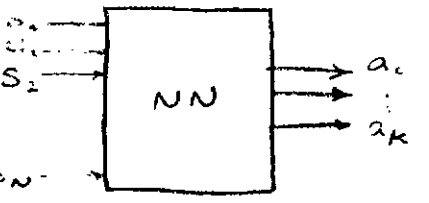
$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots$$
$$= \sum (\gamma)^k r_{t+k+1}$$

$0 \leq \gamma < 1$

$\gamma = 0$ ⟵ تنها حرکت بعدی را می بیند
$\gamma = 1$ ⟵ جواب همیشه بدست نمی آید.

* اعمال عامل از خط مشی π تبعیت می کند.
هدف RL است که خط مشی بهینه π^* پیدا کند - طوری که مقدار امید ریاضی دون را برای تمام حالات ماکزیمم کند

☑ راه حل عامل مبتنی بر NN



ورودی $< s_1, s_2, \ldots, s_N >$
وقتی در حالت S_i هستیم، بهترین عمل a_i است.

Subject:
Year. Month. Date. ( )

ماشینهای بردار پشتیبان Support Vector Machine

* ایده اصلی: اگر دسته‌ها به صورت خطی جداپذیر باشند، ابرصفحه‌ای با حداکثر حاشیه بدست می‌آورد. اگر هم جداپذیر نباشند، بعد داده‌ها را بالا می‌برد خطی آنها جدا می‌کند.

* مزایا نسبت به شبکه عصبی
  - پایداری ماشین نسبت به نویز
  - overFit نشدن

* خط بهینه

$$w^T x + b = 0$$

ابتدا از هر صنف، دو صفحه موازی جدا می‌شود و آنقدر از آن دور می‌شود تا به داده‌ها برسد. طبق تئوری آماری این کار قدرت تعمیم را بالا می‌برد.

Subject :

Year . Month . Date . ( )

Subject:
Year. Month. Date. ( )

* اصل بهینگی Bellman

یک خط مشی بهینه دارای این خاصیت است که بدون توجه به حالات و تصمیمات قبل، از این به بعد با در نظر گرفتن حالات ناشی از خط مشی بهینه برود.

☑ برنامه نویسی پویا، فرض می‌کند که هر حرکت فضای خط مشی بهینه خود است و بنابراین بهترین حرکت خود را انتخاب می‌کند و کاری به پس از آن ندارد.

Subject :

Year . Month . Date . ( )

Subject:
Year. Month. Date. ( )

* روشهای بدون مدل

در حالت عادی بدنبال رسیدن به π^* بودیم و از روی <state , reward> سعی کردیم به آن برسیم.
چون ابتدا به طور تصادفی رفتار بود، ابتدا به هر state یک مقدار تخصیص داده می‌شد و سپس policy از آن استخراج می‌شد.
این راه حلها نیاز به دانستن مدل محیط داشت تا بداند از هر حالت به چه حالت دیگری منتقل می‌شود. ($S \rightarrow V \rightarrow \pi$)
حالت دیگری این است که تنها نیاز داریم تا بدانیم کدام action بهتر است و آنرا انجام می‌دهیم و نیاز به مدل هم نداریم. محیط تنها از طریق تجربه شناخته می‌شود.
روش‌های Monte Carlo روشهای stochastic هستند که داده‌ها به صورت تصادفی تولید می‌شوند.

* روش تخمین خط مشی Monte Carlo
- هدف: یادگیری $V^\pi(s)$ برای یک سیاست مشخص
- مشاهدات: هر بار که وارد اپیزود به حالت خاصی رسیدیم آنرا مشاهدات می‌گیریم
- داده‌ها: تعداد اپیزود تحت π که شامل S باشند.

* روشهای MC
- Every-Visit MC: میانگین پاداشها برای هر بار که s وارد اپیزود مشاهدات می‌شود.
- First-Visit MC: " " اولین " " " " " "

* پایه‌های RL
- تابع ارزش: <state,action> , <state>
- MDP: Markov Decision Process: عملی که از الان انجام می‌دهیم مستقل از سابقه تصمیم‌هاست
- TD learning: چند عمل بعد را در تصمیم دخالت می‌دهیم.

PqPCO

Subject:

Year. Month. Date. ( )

Policy Iteration ← off policy : سیاست یادگیری با سیاست تعیین شده است.
on policy : سیاست اجرایی همان سیاستی که یاد گرفته شده است

* جنبه‌های مختلف RL در الگوریتم‌های مختلف

- Function Approximation ← تقریب تابع نگاشت فضای حالت - فضای واکنش‌ها
- Trace ← جنبه‌ای استفاده از سیر قبلی
- Backup ← جنبه‌ای استفاده از پاداش دریافت شده

* MDP

S : states

A : actions possible in each state

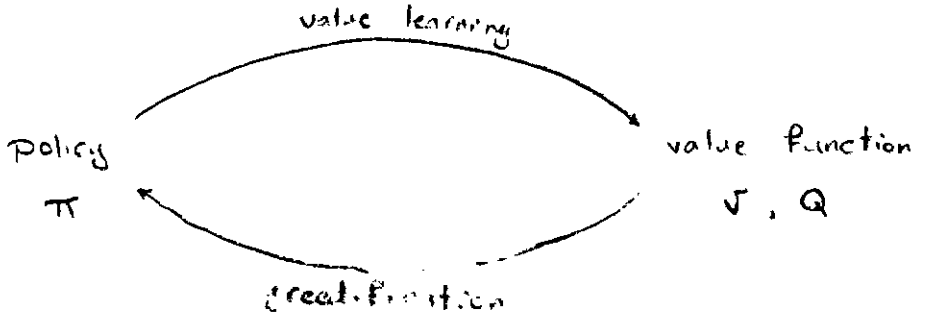
P : prob. of transition from s to s'

R : expected reward

γ : discount rate for delayed reward

* هدف RL

- بیشینه کردن پاداش دریافتی در درازمدت
  - بیشینه: حریصانه
  - تخمین: استفاده ریاضی



Subject:
Year. _ Month. Date. ( )

On-policy Monte Carlo Control *

- اکتشاف ← پیدا کردن تمام راه‌ی ممکن میره ← ε-Greedy
- on policy ← سعی می‌کند سیاست در حال اجرا را ارزیابی کند و بهبود بخشد

- راهی ε-Greedy

$\frac{\varepsilon}{|A|}$ یا $1+\varepsilon-\frac{\varepsilon}{|A|}$

Off-policy Monte Carlo Control *

- سیاست رفتاری ← رفتاری که عامل طبق آن عمل می‌کند تا یاد بگیرد.
- ... ، Evolutionary ...
- سیاست تخمینی ← سیاستی که عامل آنرا یاد می‌گیرد.

فرض Exploring Start *

- هر حالتی احتمال این را دارد که عامل از آن کار خود را آغاز کند.
- مثل: unvisited <s,a> pairs

SARSA *

State - Action - Reward - State - Action

- استفاده از یک مرحله بعدی ← TD(0)
- پاداش نسبت به Qh تدریجی در جدول بخش می‌شود ← Backup
- سرعت شرابی کمی تندتر از Qh به دلیل اکتشاف بیشتر و آینده‌نگری
- on-policy
- Q ← Value Function
- استفاده از Q یعنی، حالت یادگیری تدریجی به الگوریتم می‌دهد.

PAPCO

Subject:
Year. Month. Date. ( )

* Simulation
- بهتر از زمانی دنیای واقعی عمل هزینه بر است.
- برای مدلسازی نیاز به مدل محیط داریم که با مفهوم الگوریتم‌های Model Free در تناقض است.

* Action Selection
- Greedy ← تصمیم حریصانه در تمام مراحل
- ε-Greedy ← تصمیم حریصانه با یک احتمال تصادفی

* Function Approx
- Table
- NN
- CMAC
- Decision Tree
- REF

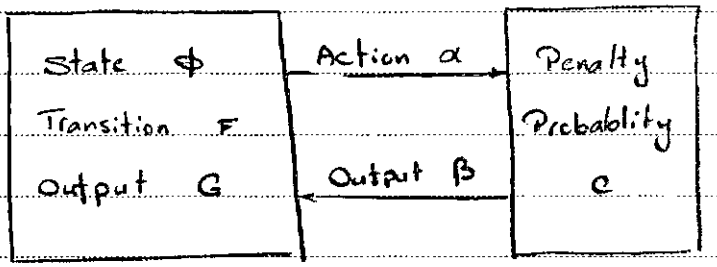
* Eligibility Traces
- مهدی رویایی (s, a)
- سریعتر همگرا شدن
- به حالتهای پرتکرار بیشتر توجه می‌کند.

Subject:
Year. Month. Date. ( )

* اتوماتای یادگیری

* اتوماتای یادگیری تطبیقی
  - سیستم قطعی ← مدل آن معلوم است
    مدل دارای عدم قطعیت زیادی است.
  - سیستم یادگیر تقویتی

* اتوماتا
  - ماشین قطعی
  - برای تعقیب دنباله‌ای از عملیات یا پاسخ به دستور العمل تهیه شده



- تقسیم‌بندی سیستم بر مبنای پاسخ محیط و دانشهای گذشته
  یا مقایسه با MDP

- تفاوت LA : RL و LA نسبت به MDP.

* تعریف

environment : $< \alpha, c, \beta >$

α : finite action / output set

β : input / response set

PAPCO c : set of penalty probability of each action

Subject:
Year. Month. Date. ( )

* انواع مدل LA

- P - Model ⟶ بسته : مرکزی
- Q - Model ⟶ بسته : زمان
- S - Model ⟶ بسته : خروجی

* مدل LA

- State Φ ⟶ حالتهای درونی LA (نمایانگر حافظه)
- Action α
- Input β
- Next State F ⟶ $F: \Phi \times \beta \longrightarrow \Phi'$
- Next Action H ⟶ $H: \Phi \times \beta \longrightarrow \alpha$

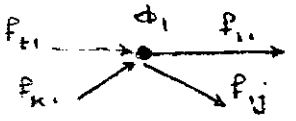
$$LA: < \Phi, \alpha, \beta, F(\Phi, \beta), H(\Phi, \beta) >$$

$\alpha = G(\Phi)$ حالت خاص: state-output automaton

* ارتباط با محیط تصادفی

- معمولاً بین از دو نتایج F ، G تصادفی نیست.
- ما فقط روی این احتمالات صحبت می کنیم.

$$F(\beta_i) = \begin{matrix} \phi_1 \\ \phi_n \end{matrix} \begin{bmatrix} f_{11} & & \phi_n \\ & & \\ f_{n1} & & f_{nn} \end{bmatrix}$$



* تعریف مدلهای تصادفی

- تصادفی محض

$P_i(n) = \frac{1}{r}$; $i = 1, 2, \ldots, r$

Subject:
Year. Month. Date. ( )

$M(n) = E[\beta(n) \mid p(n)] = \sum c_i \cdot p_i(n)$

p_i: بردار احتمال پاسخ‌های ممکن
در حالت تصادفی محض

$M_0 = \frac{1}{r} \sum c_i$

$E[M(n)] = E[\beta(n)]$

Expedient ← اگر حداقل از حالت متوسط کمتر باشد.

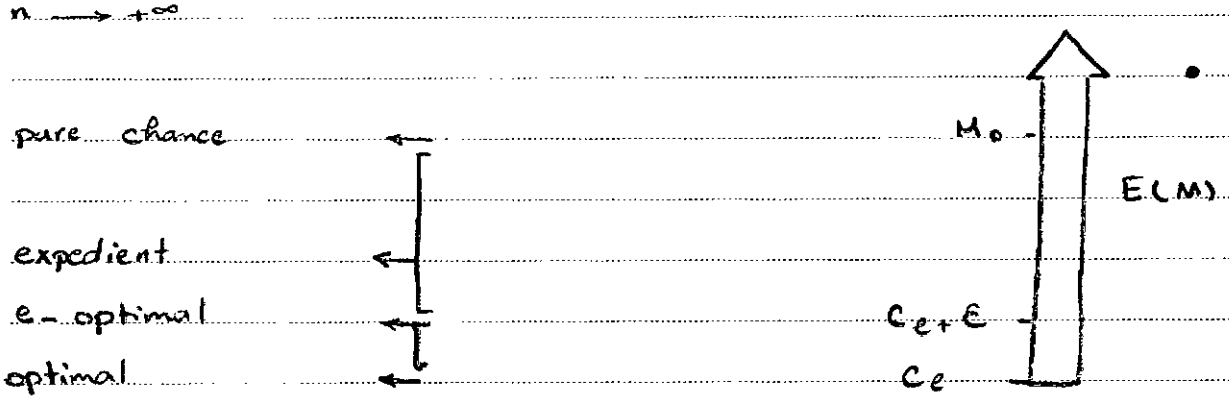
$\lim_{n \to +\infty} E[M(n)] < M_0$

optimal ← اگر حداقل بهینه باشد.

$\lim_{n \to +\infty} E[M(n)] = c_e$

e-optimal ← اگر به تقریب حداقل بهینه باشد

$\lim_{n \to +\infty} E[M(n)] = c_e + \varepsilon$



Subject:

Year. Month. Date. ( )

* آشنایی با دلیر سلولی

• شبکه‌ای از سلول‌ها که برنامه یک سلول هستند.

• عمل هر سلول علاوه بر رشد و جذب، به رفتار دیگران هم بستگی دارد.