

# A review on Data Mining/Science

امیر تیمور پاینده نجف آبادی

# سرفصلها

- مقدمه: علم داده چیست؟ چالش های انگیزشی خاستگاه آن و ظایف علم داده
- داده: انواع داده ها، ویژگی ها و اندازه گیری، انواع داده ها
- کیفیت داده ها: مسائل مربوط به اندازه گیری و جمع آوری داده ها و غیره
- پیش پردازش داده ها:
- معیارهای تشابه و عدم تشابه:
- آشنایی با چهار وظیفه اصلی علم داده:
- طبقه بندی: مفاهیم و تکنیک های اساسی، چارچوب کلی برای طبقه بندی، طبقه بندی درخت تصمیم، یک الگوریتم اساسی برای ساخت درخت تصمیم، تحلیل پیوند
- تحلیل خوشه ای
- تحلیل پیوند
- تشخیص ناهنجاری

# تعریف داده کاوی و علم داده

- داده کاوی فرآیند کشف خودکار اطلاعات مفید از انبار داده های بزرگ است.
- به فرآیندهای
- به دست آوردن دانش از انبار داده ها،
- جمع آوری داده ها،
- تحلیل و بصری سازی داده ها
- با استفاده از ابزارها و روش های مختلف علم داده گویند.
- علم داده، شاخه ای از علوم است که روش های سنتی تحلیل داده ها را با الگوریتم های پیچیده برای پردازش داده ها ترکیب می کند.
- داده کاوی بیشتر **اهداف تجاری** دارد اما علم داده بیشتر **اهداف علمی** دارد.
- داده کاوی یک تکنیک است. علم داده یک رشته است.
- **بهتر است:** داده کاوی را زیرمجموعه ای از علم داده دانست، که به فرآیند کشف الگوهای و سایر اطلاعات کلیدی از مجموعه داده های عظیم آشاره دارد.

# برخی از کاربردهای آن

- **کسب و کار و صنعت:** جمع آوری داده های فروش (اسکنر بار کد و فناوری کارت هوشمند) به خرده فروشان این امکان را داده است که داده های لحظه به لحظه در مورد خرید مشتریان را در باجه های صندوق فروشگاه های خود جمع آوری کنند. خرده فروشان می توانند از این اطلاعات، همراه با سایر داده های حیاتی تجاری، مانند گزارش های وب سرور از وب سایت های تجارت الکترونیک و سوابق خدمات مشتری از مراکز تماس، استفاده کنند تا به آنها کمک کند نیازهای مشتریان خود را بهتر درک کنند و تصمیمات تجاری آگاهانه تری بگیرند.
- این علم همچنین می تواند به خرده فروشان کمک کند تا به سؤالات مهم تجاری مانند:
  - سودآورترین مشتریان چه کسانی هستند؟
  - چه محصولاتی را می توان به صورت متقطع فروخته کرد؟
  - چشم انداز درآمد شرکت برای سال آینده چیست؟
- این سؤالات الهام بخش باعث توسعه بیشتر تکنیک های این علم شده اند.

# کاربردهای دیگر

- استفاده از داده های وب: فیلتر کردن پیام های Spam، پاسخ به پرسش های جستجو، و پیشنهاد به روزرسانی ها و تحلیل شبکه های اجتماعی.
  - استفاده از داده های حسگرها و دستگاه های متحرک: تلفن های هوشمند و دستگاه های محاسباتی پوشیدنی
  - توسعه مدل های پیشگو در پزشکی
  - توسعه مدل های کشف تقلب
  - مبلغی که مشتری در یک فروشگاه آنلاین خرج خواهد کرد.
  - احتمال وقوع سکته در چند ساعت آینده
  - طبقه ریسکی افراد
  - مثال های بیشتر؟؟؟

# تذکر

- به تمامی روش‌های کشف اطلاعات نمی‌توان، عنوان علم داده یا داده کاوی داد.
- مثلاً:
  - جستجوی سوابق فردی در یک پایگاه داده
  - یافتن صفحات وب حاوی مجموعه خاصی از کلمات کلیدی
- زیرا چنین وظایفی را می‌توان از طریق تعاملات ساده با یک سیستم مدیریت پایگاه داده یا یک سیستم بازیابی اطلاعات انجام داد.
- این سیستم‌ها برای سازماندهی و بازیابی مؤثر اطلاعات از مخازن داده‌های بزرگ، بر تکنیک‌های سنتی علوم کامپیوتر (که شامل ساختارهای نمایه‌سازی پیچیده و الگوریتم‌های پردازش هستند) استفاده می‌کنند.
- با این وجود، تکنیک‌های علم داده و داده کاوی نیز برای بهبود عملکرد چنین سیستم‌هایی یا بهبود کیفیت نتایج جستجو (مثلاً بررسی ارتباط آنها با دستورهای ورودی) کمک می‌کند

## چالش های روش‌های سنتی آماری که برای پاسخ به آنها علم داده و داده کاوی ابداع شدند

- مقیاس پذیری: به دلیل پیشرفت در تولید و جمع آوری داده‌ها، مجموعه داده‌هایی با اندازه‌های بسیار بزرگ (مثلاً تراپایت، پتابایت یا حتی اگزابایت) تولید و جمع آوری می‌شوند.
- بسیاری از روش‌های آمار سنتی نمی‌توانند این حجم عظیم داده را تحلیل کنند.

• روش برخورد علم داده یا داده کاوی: ؟؟؟:

## ادامه چالشها

- **ابعاد بسیار بالای داده ها:** در حال حاضر با مجموعه داده ها مواجه هستیم که صدها یا هزاران ویژگی (بعد) دارند.
- به عنوان مثال، مجموعه داده ای را در نظر بگیرید که شامل اندازه گیری دما در مکان های مختلف است. اگر اندازه گیری دما به طور مکرر برای مدت طولانی انجام شود، تعداد ابعاد (ویژگی ها) متناسب با تعداد اندازه گیری های انجام شده، افزایش می یابد.
- تکنیک های سنتی تحلیل داده ها که برای داده های با ابعاد پایین ایجاد شده اند، اغلب برای چنین داده هایی با ابعاد بالا به خوبی کار نمی کنند. همچنین، برای برخی از الگوریتم های تحلیل داده ها، با افزایش ابعاد (تعداد ویژگی ها) پیچیدگی محاسباتی به سرعت افزایش می یابد.
- روش برخورد علم داده یا داده کاوی: ؟؟؟

## ادامه چالشها

- **داده‌های ناهمگن و پیچیده:** روش‌های سنتی تحلیل داده‌ها اغلب با مجموعه داده‌های سروکار دارند که دارای ویژگی‌هایی از یک نوع، پیوسته یا طبقه‌ای هستند.
- در دنیا این جدید داده‌های بسیار متنوع تر و ناهمگن‌تر تولید می‌شوند نیاز به تکنیک‌هایی که بتوانند ویژگی‌های ناهمگن را مدیریت کنند، به شدت افزایش یافته است.
- مثال‌هایی از این نوع داده‌های غیرسنتی:
  - شامل داده‌های وب و شبکه‌های اجتماعی، شامل متن، لینک‌ها، تصاویر، صدا و ویدئو است.
  - داده‌های DNA با ساختار متوالی و چند بعدی.
  - داده‌های آب و هوایی که شامل اندازه‌گیری (دما، فشار و غیره) در زمان‌ها و مکان‌های مختلف در سطح زمین است.
  - تکنیک‌های توسعه یافته برای استخراج اطلاعات از چنین داده‌های پیچیده‌ای باید روابط موجود در داده‌ها را در نظر بگیرند، مانند خود همبستگی زمانی و مکانی، اتصال نمودار و روابط والد-فرزنده بین عناصر موجود
  - روش برخورد علم داده یا داده کاوی: ؟؟؟

## ادامه چالشها

- مالکیت و توزیع داده ها: گاهی اوقات، داده های مورد نیاز برای تحلیل در یک مکان ذخیره نمی شود یا متعلق به یک سازمان نیستند. در عوض، داده ها از نظر جغرافیایی بین منابع متعلق به چندین نهاد توزیع شده اند.

برای استخراج دانش از این داده ها باید تکنیک های تحلیل داده های توزیع شده، توسعه پیدا کنند. برخی از چالش های کلیدی که الگوریتم های تحلیل داده های توزیع شده، با آن مواجه هستند عبارتنداز:

- نحوه کاهش میزان ارتباط مورد نیاز برای انجام محاسبات توزیع شده،
- نحوه ادغام موثر نتایج به دست آمده از منابع متعدد،
- نحوه رسیدگی به مسائل امنیتی و حریم خصوصی داده ها
- روش برخورد علم داده یا داده کاوی:؟؟؟

## ادامه چالشها

- **تحلیل غیر سنتی:** رویکرد آماری سنتی مبتنی بر پارادایم فرضیه و آزمون است. به عبارت دیگر، یک فرضیه ارائه می شود، یک آزمایش برای جمع آوری داده ها طراحی می شود و سپس داده ها با توجه به فرضیه مورد تحلیل قرار می گیرند.
- متأسفانه، این فرآیند بسیار پر زحمت است. وظایف تحلیل داده های فعلی اغلب به تولید و ارزیابی هزاران فرضیه نیاز دارند و در نتیجه، توسعه برخی از تکنیک های داده کاوی با تمایل به خودکارسازی فرآیند تولید و ارزیابی فرضیه ها انجام شده است.
- علاوه بر این، مجموعه داده های تحلیل شده در داده کاوی معمولاً نتیجه یک آزمایش با دقت طراحی شده نیستند و اغلب نمونه های مشاهده شده از داده ها را به جای نمونه های تصادفی نشان می دهند.

## تاریخچه داده کاوی

- علی رغم آنکه داده کاوی به طور سنتی به عنوان یک فرآیند میانی در چارچوب کشف اطلاعات از پایگاه داده ها (knowledge discovery in databases) یا KDD در نظر گرفته می شود
- برای سال ها به عنوان یک زمینه آکادمیک در علوم کامپیوتر ظاهر شده بود
- در اوخر دهه ۱۹۸۰، مجموعه ای از کارگاه های آموزشی با موضوع «کشف دانش در پایگاه های داده» سازماندهی شد. این کارگاهها محققانی از رشته های مختلف گرد هم آمدند تا چالش ها و فرصت ها در استفاده از تکنیک های محاسباتی برای استخراج دانش عملی از پایگاه های داده بزرگ را مورد بحث و بررسی قرار دهند.
- این کارگاه ها به سرعت تبدیل به کنفرانس های بسیار محبوبی شدند که با حضور محققان و دست اندر کاران از دانشگاه و صنعت برگزار شد.
- موفقیت این کنفرانس ها به همراه علاقه ای که کسب و کارها و صنعت به جذب نیروهای جدید با پیشنه داده کاوی نشان داده اند، به رشد فوق العاده این حوزه دامن زده است.

# تاریخچه علم داده

- واژه علم داده اولین بار توسط ویلیام کلیولند (سال ۲۰۰۱) مورد استفاده قرار گرفت.
- او در مقاله با عنوان «علم داده: برنامه‌ای برای گسترش جنبه‌های فنی در رشته آمار» پیشنهاد کرد که علم داده به عنوان یک رشته مستقل شناخته و به صورت دانشگاهی ارائه شود.
- کلیولند این رشته جدید را مرتبط با علوم کامپیوتر و آمار می‌دانست. از نظر او:
  - مهندسین کامپیوتر شناخت کمی از روش‌های تحلیل و استباط به کمک داده دارند
  - همچنین دانش محاسباتی متخصصین آمار، تنها به تحلیل داده‌ها محدود است
- بنابراین تلفیق این دو گروه می‌تواند منجر به نوآوری‌های زیادی شود. دپارتمانهای علم داده باید اساتیدی داشته باشد که بتوانند دانش تحلیل داده‌ها را با دانش استخراج داده و محاسباتی تلفیق کنند.

- داده کاوی فرآیند کشف بینش ها، الگوها و اطلاعات ارزشمند از مجموعه داده های گسترده با استفاده از تکنیک ها، الگوریتم ها و ابزارهای مختلف است. این شامل استخراج دانش پنهان، روندها و روابط است که می تواند به تصمیم گیری ها و پیش بینی های آگاهانه کمک کند.
- علم داده رشته‌ای است که شامل استخراج بینش و دانش از داده‌ها از طریق تکنیک‌های مختلف، شامل تحلیل آماری، یادگیری ماشین و تخصص حوزه، برای اطلاع‌رسانی در تصمیم‌گیری و حل مشکلات پیچیده است.

- ویژگی های کلیدی داده کاوی عبارتند از:
  - کشف الگو
  - پیشگویی (پیش بینی)
  - اتوماسیون (اتومات کردن روندها)
  - داده های در مقیاس بزرگ
  - بین رشته ای
  - رویکرد اکتشافی
  - غیر بدیهی بودن
  - مقیاس پذیری
  - داده محور بودن
  - تعامل پذیر بودن
- تحلیل داده ها
- بین رشته ای
- مدل سازی روند پیشگویی
- تحلیل داده های بزرگ (Big Data)
- مصوروسازی داده ها
- انجام آزمون فرض
- داده محور بودن
- حل مسئله
- رویکرد اکتشافی
- پشتیبانی تصمیم بر اساس داده ها (تعامل پذیر بودن)

# کشف الگو (Pattern discovery)

- داده کاوی حول شناسایی الگوهای روندها و روابط معنی دار درون داده ها متمرکز است.
- این فرآیند شامل غربال کردن مجموعه داده های گسترده برای کشف بینش هایی است که ممکن است از طریق روش های تحلیل سنتی آشکار نباشند.
- این الگوهای می توانند اشکال مختلفی مانند پیوندها، توالی ها، خوشها یا ناهنجاری ها داشته باشند و نقش مهمی در فرآیندهای تصمیم گیری در حوزه های مختلف، از بازاریابی تا مراقبت های بهداشتی، ایفا می کنند.

# پیشگویی (Prediction)

- داده کاوی به سازمان ها قدرت می دهد تا با بررسی داده های تاریخی، مدل های پیشگو را توسعه دهند.
- این مدل ها می توانند برای پیش بینی رویدادها، رفتارها یا روندهای آتی مورد استفاده قرار گیرند و در نتیجه به کسب و کارها در تصمیم گیری فعالانه کمک کنند
- به عنوان مثال:
  - موسسات مالی را قادر می سازد تا ریسک اعتباری را پیش بینی کند،
  - متخصصان مراقبت های بهداشتی را برای پیش بینی شیوع بیماری
  - خرده فروشان را قادر می سازد تا تقاضای مشتری را پیش بینی کند.

# اتوماسیون (خودکارسازی روندها) (Automation)

- یکی از مزایای کلیدی داده کاوی قابلیت اتماسیون آن است.
- با استفاده از الگوریتم های پیچیده و تکنیک های یادگیری ماشین، فرآیندهای داده کاوی می توانند تا حد زیادی خودکار شوند و نیاز به تحلیل دستی داده ها را کاهش دهند.
- این نه تنها کارایی را افزایش می دهد، بلکه تضمین می کند که حجم زیادی از داده ها می توانند به موقع پردازش شوند.

# داده های در مقیاس بزرگ (Large-scale data)

- داده کاوی ایده آل برای مدیریت مجموعه داده های در مقیاس بزرگ، است
- می تواند بینش های ارزشمندی را از مجموعه های داده های عظیم، (داده های ساختار یافته و بدون ساختار) مدیریت و استخراج کند
- کاربر اطمینان حاصل می کند که هیچ اطلاعات ارزشمندی نادیده گرفته نمی شود.

# میان رشته ای (Multidisciplinary)

- داده کاوی از طیف گسترده ای از علوم مانند آمار، یادگیری ماشین، هوش مصنوعی و مدیریت پایگاه داده استخراج می شود.
- این رویکرد میان رشته ای، داده کاوی ها را قادر می سازد تا از تکنیک ها و ابزارهای مختلفی برای رسیدگی به چالش های مختلف تحلیل داده ها به طور موثر استفاده کنند.

# رویکردهای اکتشافی (Exploratory)

- برخلاف روش‌های سنتی تحلیل داده‌ها، که ممکن است بر تأیید فرضیه‌های از پیش تعیین شده تمرکز کنند،
- داده‌کاوی یک رویکرد اکتشافی را تشویق می‌کند.
- تحلیلگران می‌توانند با داده‌ها تعامل داشته باشند، سؤالات باز مطرح کنند و الگوهای روابط پیش‌بینی نشده را کشف کنند.
- این کاوش با ذهن باز، اغلب به آشکار شدن بینش‌هایی منجر می‌شود که در غیر این صورت ممکن بود پنهان بماند.

# غیربدیهی بودن (Non-trivial)

- داده کاوی به خلاصه سازی ساده داده ها محدود نمی شود.
- با روابط پیچیده و غیر آشکار در داده ها سروکار دارد.
- با چالش های دنیای واقعی مقابله می کند که در آن بینش های به دست آمده دقیقاً به این دلیل ارزشمند هستند که بلافاصله آشکار نمی شوند و آن را به ابزاری قدرتمند برای پشتیبانی تصمیم گیری تبدیل می کند.

# مقیاس پذیری (Scalability)

- تکنیک های داده کاوی به گونه ای طراحی شده اند که با افزایش اندازه و پیچیدگی داده ها مقیاس شوند.
- همانطور که حجم داده ها همچنان در حال رشد است، داده کاوی سازگار باقی می ماند
- و آن را به یک دارایی ارزشمند برای سازمان هایی تبدیل می کند که با اندازه ها و پیچیدگی های مختلف داده سرو کار دارند.

# داده محور (Data-driven)

- در هسته خود، داده کاوی یک رویکرد داده محور (data-centric approach) است.
- به داده ها به عنوان منبع اولیه دانش و بینش متکی است.
- این دیدگاه مبتنی بر داده تضمین می کند که تصمیمات ریشه در شواهد تجربی و به روزترین اطلاعات دارند و به فرآیندهای تصمیم گیری آگاهانه تر و دقیق تر کمک می کند.

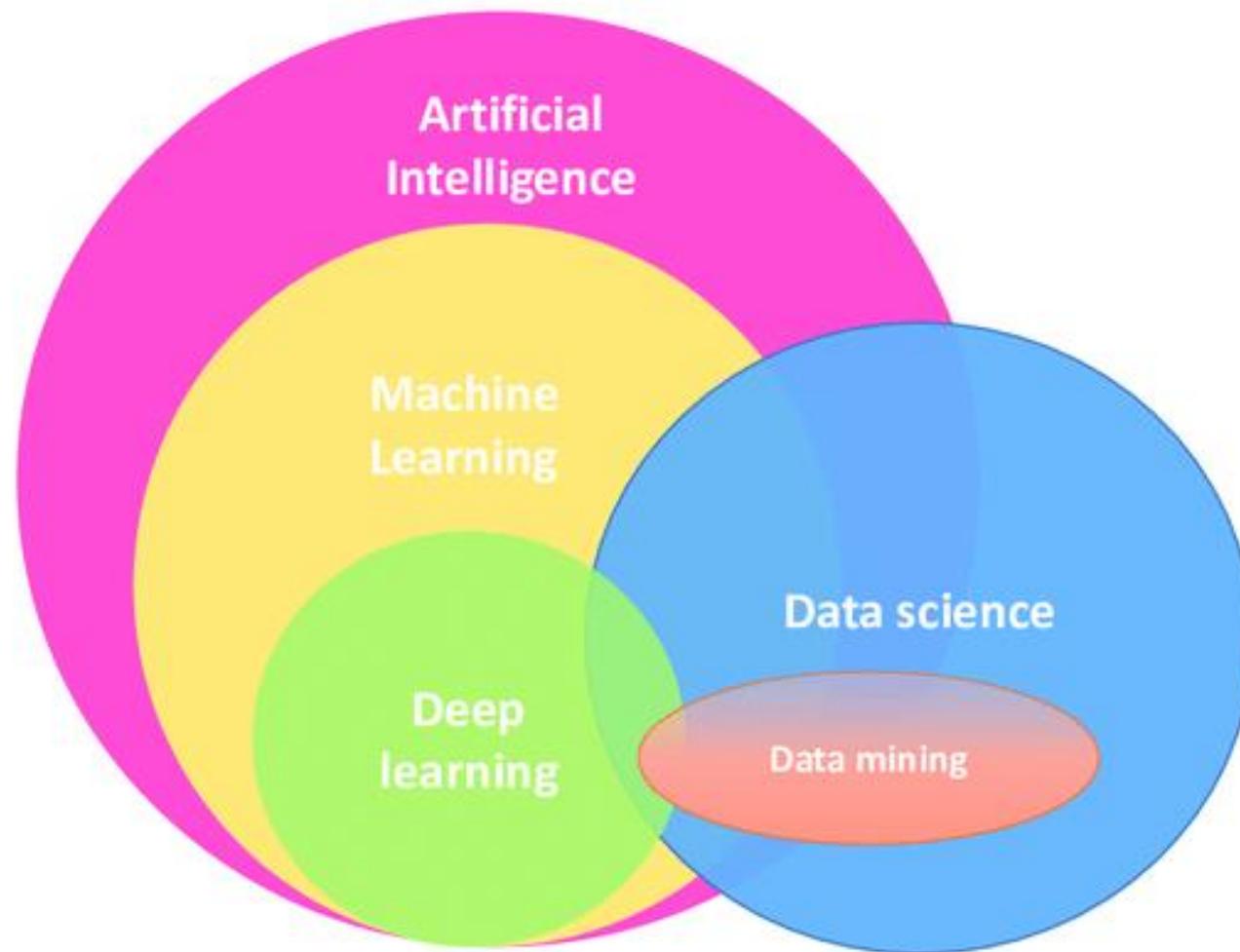
# تعامل (Interactivity)

- داده کاوی اغلب شامل تعامل و بازخورد کاربر است.
- تحلیلگران می توانند جستجوهای خود را به دقت تنظیم کنند، مدل ها را تنظیم کنند و فرآیند داده کاوی را تکرار کنند.
- این تعامل امکان پالایش نتایج را فراهم می کند و به بینش های دقیق تر و عملی تر منجر می شود زیرا تحلیلگران در کم عمق تری از داده ها به دست می آورند.

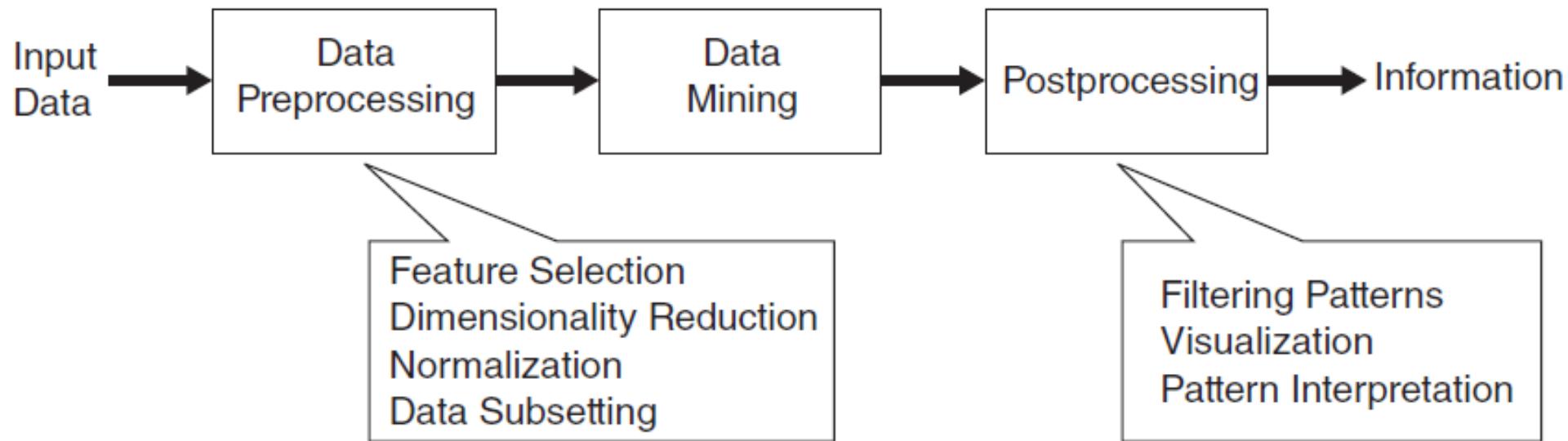
# علم داده

- بر اساس روش‌شناسی و الگوریتم‌هایی ساخته شد که محققان قبلًا از آنها استفاده کرده‌اند.
- به طور خاص، محققان علم داده از ایده‌هایی مانند (۱) نمونه‌گیری، برآورد و آزمایش فرضیه‌ها از آمار و (۲) الگوریتم‌های جستجو، تکنیک‌های مدل‌سازی و تئوری‌های یادگیری از هوش مصنوعی، تشخیص الگو و یادگیری ماشین استفاده می‌کنند.
- علم داده همچنین به سرعت ایده‌هایی را از حوزه‌های دیگر، از جمله بهینه‌سازی، محاسبات تکاملی، نظریه اطلاعات، پردازش سیگنال، تجسم و بازیابی اطلاعات اتخاذ کرده و آنها را برای حل چالش‌های استخراج داده‌های بزرگ گسترش داده است.

context of wireless networks.



- داده کاوی (علم داده) بخشی جدایی ناپذیر از **کشف دانش در پایگاه های داده** (knowledge discovery in databases) است که فرآیند کلی تبدیل داده های خام به اطلاعات مفید است.



**Figure 1.1.** The process of knowledge discovery in databases (KDD).

- داده های ورودی را می توان در قالب های مختلف (صفحات گسترده یا جداول مرتبط) ذخیره کرد
- هدف از پیش پردازش (preprocessing) تبدیل داده های ورودی خام به فرمت مناسب برای تحلیل بعدی است.
- مراحل مختلف در پیش پردازش داده ها عبارتنداز:
  - ترکیب داده ها از منابع متعدد،
  - پاک کردن داده ها برای حذف نویز و مشاهدات تکراری
  - انتخاب سوابق و ویژگی های مرتبط با وظیفه داده کاوی در دست اقدام است.
- چون داده ها به روش های مختلف جمع آوری و ذخیره می شوند، پیش پردازش داده ها شاید پر زحمت ترین و زمان بر ترین مرحله در فرآیند کلی کشف دانش باشد.

# پس پردازش postprocessing

- ادغام نتایج داده کاوی در سیستم های پشتیبانی تصمیم یکی از مراحل فرآیند کلی کشف دانش است. به عنوان مثال، در برنامه های کاربردی تجاری، بینش ارائه شده توسط نتایج داده کاوی را می توان با ابزارهای مدیریتی ادغام کرد تا تبلیغات بازاریابی موثر انجام و آزمایش شوند.
- چنین ادغامی نیاز به مرحله **پس پردازش** دارد تا اطمینان حاصل شود که فقط نتایج معترض و مفید در سیستم پشتیبانی تصمیم گنجانده می شود.
- نمونه ای از **پس پردازش** عبارتند از:
  - مصورسازی است که به تحلیلگران اجازه می دهد تا داده ها و نتایج داده کاوی را از دیدگاه های مختلف بررسی کنند.
  - روش های آزمون فرض که به تحلیلگر اجازه می دهد در طول پس پردازش نتایج جعلی داده کاوی را مشخص و حذف کند.

# چهار وظیفه اصلی علم داده

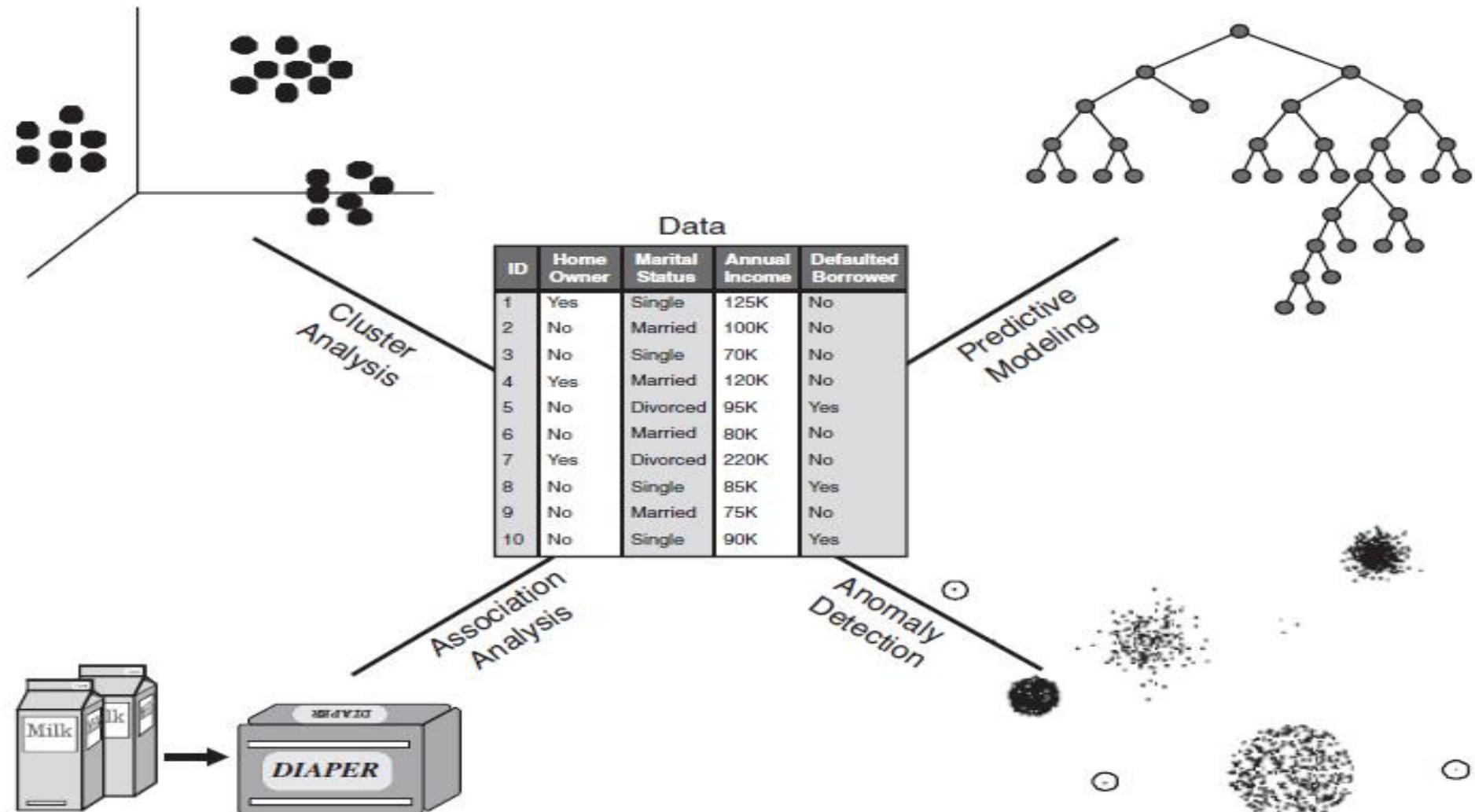


Figure 1.3. Four of the core data mining tasks.

# وظایف اصلی علم داده

- **پیشگویی:** هدف این کارها پیشگویی ارزش یک ویژگی (صفت، متغیر) خاص بر اساس مقادیر سایر ویژگی‌ها است. مشخصه‌ای که باید پیشگویی شود معمولاً به عنوان متغیر هدف یا وابسته شناخته می‌شود، در حالی که ویژگی‌هایی که برای پیشگویی استفاده می‌شوند به عنوان متغیرهای توضیحی یا مستقل شناخته می‌شوند.
- **استخراج الگوهای پیوند، خوشه بندی:** هدف این الگوها (پیوندها، روندها، خوشه‌ها، مسیرها و ناهنجاری‌ها) کشف روابط زیربنایی را در داده‌ها است. وظایف داده کاوی توصیفی اغلب ماهیت اکتشافی دارند و اغلب به تکنیک‌های پس پردازش برای اعتبارسنجی و توضیح نتایج نیاز دارند.

# ۱- مدل های پیشگو

- مدل های پیشگو به دنبال ساخت مدل برای متغیر هدف به عنوان تابعی از متغیرهای توضیحی (مستقل) است. دو نوع کار مدل پیشگو وجود دارد:
- طبقه بندی که برای متغیرهای هدف گستته استفاده می شود
- رگرسیون که برای متغیرهای هدف پیوسته استفاده می شود.
- به عنوان مثال:
- پیش بینی اینکه آیا یک کاربر وب در یک کتابفروشی آنلайн خرید می کند یا خیر، یک کار طبقه بندی است زیرا متغیر هدف دارای ارزش دودویی است.
- پیش بینی قیمت آتی یک سهم یک کار رگرسیونی است زیرا قیمت یک ویژگی با ارزش پیوسته است.
- هدف هر دو کار یادگیری مدلی است که خطای بین مقادیر پیش بینی شده و واقعی متغیر هدف را به حداقل برساند.

## ۲- تحلیل پیوند (Association)

- از تحلیل همبستگی برای کشف الگوهایی استفاده می شود که ویژگی های مرتبط قوی در داده ها را توصیف می کند.
- الگوهای کشف شده معمولاً در قالب قوانین ضمنی یا زیر مجموعه ویژگی ها نشان داده می شوند.
- به دلیل اندازه نمایی فضای جستجو، هدف از این تحلیل بیشتر استخراج جالب ترین و کارآمدترین الگوها است.
- مثال های از این روش تحلیلی، عبارتنداز:
  - یافتن گروههایی از ژن هایی است که دارای عملکرد مرتبط هستند،
  - شناسایی صفحات وب که با هم قابل دسترسی هستند
  - درک روابط بین عناصر مختلف سیستم آب و هوای زمین.

## ارائه مثال

Table 1.1. Market basket data.

Transaction ID	Items
1	{Bread, Butter, Diapers, Milk}
2	{Coffee, Sugar, Cookies, Salmon}
3	{Bread, Butter, Coffee, Diapers, Milk, Eggs}
4	{Bread, Butter, Salmon, Chicken}
5	{Eggs, Bread, Butter}
6	{Salmon, Diapers, Milk}
7	{Bread, Tea, Sugar, Eggs}
8	{Coffee, Sugar, Chicken, Eggs}
9	{Bread, Diapers, Milk, Salt}
10	{Tea, Eggs, Cookies, Diapers, Milk}

## ادامه مثال

- جدول بالا داده های فروش جمع آوری شده در باجه های صندوق یک فروشگاه مواد غذایی را نشان می دهد.
- تحلیل پیوند را می توان برای یافتن اقلامی که اغلب با هم توسط مشتریان خریداری می شوند، به کار برد.
- برای مثال، ممکن است الگوی {پوشک} → {شیر} را کشف کنیم، که نشان می دهد مشتریانی که پوشک می خرند تمایل به خرید شیر نیز دارند.
- از این نوع الگوها می توان برای شناسایی فرصت های بالقوه فروش متقابل در میان اقلام مرتبط استفاده کرد.

### ۳- تحلیل خوشه ای

- تحلیل خوشه ای به دنبال یافتن گروه هایی از مشاهدات نزدیک به هم است
- مشاهداتی که به یک خوشه تعلق دارند بیشتر از مشاهداتی که متعلق به خوشه های دیگر هستند به یکدیگر شباهت دارند.
- خوشه بندی برای گروه بندی مجموعه ای از مشتریان مرتبط، یافتن مناطقی از اقیانوس که تأثیر قابل توجهی بر آب و هوای زمین دارند و فشرده سازی داده ها استفاده شده است.

# مثال خوشه بندی اسناد (متن کاوی یا text mining)

Table 1.2. Collection of news articles.

Article	Word-frequency pairs
1	dollar: 1, industry: 4, country: 2, loan: 3, deal: 2, government: 2
2	machinery: 2, labor: 3, market: 4, industry: 2, work: 3, country: 1
3	job: 5, inflation: 3, rise: 2, jobless: 2, market: 3, country: 2, index: 3
4	domestic: 3, forecast: 2, gain: 1, market: 2, sale: 3, price: 2
5	patient: 4, symptom: 2, drug: 3, health: 2, clinic: 2, doctor: 2
6	pharmaceutical: 2, company: 3, drug: 2, vaccine: 1, flu: 3
7	death: 2, cancer: 4, drug: 3, public: 4, health: 3, director: 2
8	medical: 2, cost: 3, increase: 2, patient: 2, health: 3, care: 1

## ادامه مثال

- جدول بالا کلمات کلیدی ۸ مقاله خبری نشان می دهد.
- هر مقاله به صورت مجموعه‌ای از زوج کلمه کلیدی و تعداد دفعاتی که کلمه در مقاله ظاهر شده، گزارش شده است.
- دو خوش طبیعی در مجموعه داده وجود دارد:
  - خوش اول شامل چهار مقاله اول است که مربوط به اخبار مربوط به اقتصاد است،
  - خوش دوم شامل چهار مقاله آخر است که مربوط به اخبار مربوط به مراقبت‌های بهداشتی است.
- یک الگوریتم خوش بندی خوب باید بتواند این دو خوش را بر اساس شباهت بین کلماتی که در مقالات ظاهر می شود شناسایی کند.

## ۴- تشخیص ناهنجاری

- در تشخیص ناهنجاری به دنبال کشف و شناسایی مشاهداتی هستیم که ویژگی های آنها به طور قابل توجهی با بقیه داده ها متفاوت است.
- چنین مشاهداتی به عنوان ناهنجاری (outlier) یا پرت (Anomaly) شناخته می شوند.
- هدف یک الگوریتم تشخیص ناهنجاری، کشف ناهنجاری های واقعی و اجتناب از برچسب زدن نادرست به داده های عادی است.
- به عبارت دیگر، یک آشکارساز ناهنجاری خوب باید دارای نرخ تشخیص بالا و نرخ هشدار کاذب پایین باشد.
- کاربردهای تشخیص ناهنجاری شامل تشخیص تقلب، نفوذ به شبکه، الگوهای غیرمعمول بیماری ها و اختلالات اکوسیستم مانند خشکسالی، سیل، آتش سوزی، طوفان و غیره است.

# مثال کشف تقلب

- یک شرکت کارت اعتباری، تراکنش های انجام شده توسط هر دارنده کارت اعتباری را همراه با اطلاعات شخصی مانند محدودیت اعتبار، سن، درآمد سالانه و آدرس ثبت می کند.
- از آنجایی که تعداد تقلب ها در مقایسه با تعداد تراکنش های قانونی نسبتاً کم است،
- می توان از تکنیک های تشخیص ناهنجاری برای ایجاد پروفایلی از تراکنش های قانونی برای کاربران استفاده کرد.
- هنگامی که یک تراکنش جدید وارد می شود، با مشخصات کاربر مقایسه می شود.
- اگر ویژگی های تراکنش بسیار متفاوت از نمایه ایجاد شده قبلی باشد،
- تراکنش به عنوان بالقوه تقلبی علامت گذاری می شود.

## داده ها

- نوع و دسته بندی داده ها را می توان با رویکردهای متفاوت انجام داد.
- به عنوان مثال، صفات مورد استفاده برای توصیف می تواند «كمی» یا «كیفی» باشند.
- داده ها اغلب دارای ویژگی های خاص هستند. به عنوان مثال، برخی از مجموعه های داده حاوی سری های زمانی یا روابط صریح با یکدیگر هستند.
- جای تعجب نیست که نوع داده تعیین می کند ابزار و تکنیک ها آماری برای تحلیل داده ها هستند.
- در واقع، تحقیقات جدید در علم داده اغلب به دلیل نیاز به تطبیق با حوزه های کاربردی جدید و انواع جدید داده های آنها انجام می شود.

## کیفیت داده ها

- داده ها اغلب از کیفیت مناسبی برخوردار نیستند.
- در حالی که اکثر تکنیک های علم داده می توانند سطحی از نقص در داده ها را تحمل کنند،
- تمرکز بر درک و بهبود کیفیت داده ها، معمولاً باعث بهبود کیفیت تحلیل می شود
- مسائل مربوط به کیفیت داده که اغلب نیاز به رسیدگی دارند، شامل:
  - وجود نویز و نقاط پرت است.
  - داده های گمشده،
  - متناقض یا تکراری؛
  - داده هایی که مغرضانه می باشند.

## پیش پردازش برای مناسب تر کردن داده ها برای علم داده

- اغلب، داده های خام باید پردازش شوند تا برای تحلیل مناسب شوند.
- در حالی که یک هدف ممکن است بهبود کیفیت داده باشد،
- اهداف دیگر بر اصلاح داده ها به گونه ای تمرکز می کنند که بهتر با یک تکنیک یا ابزار داده کاوی مشخص شده مطابقت داشته باشد.
- مثلاً ویژگی پیوسته، نرمال بودن و غیره
- کاهش بُعد (تعداد صفات): بسیاری از تکنیک ها زمانی موثرتر هستند که داده ها دارای تعداد نسبتاً کمی از ویژگی ها باشند.

# تحلیل داده ها از نظر روابط آن

- یک رویکرد برای تحلیل داده ها، یافتن روابط بین داده ها و سپس انجام تحلیل باقی مانده با استفاده از این روابط به جای خود اشیاء داده است.
- به عنوان مثال، ما می توانیم شباهت یا فاصله بین جفت اشیاء را محاسبه کنیم و سپس تحلیل (خوش بندی، طبقه بندی، یا تشخیص ناهنجاری) را بر اساس این شباهت ها یا فواصل انجام دهیم.
- تعداد زیادی از این شباهت ها یا اندازه گیری های فاصله وجود دارد و انتخاب مناسب به نوع داده و کاربرد خاص بستگی دارد.

## مثال برای نشان دادن اهمیت دانستن داده ها و در نظر روابط قبل از انجام تحلیل

- وضعیت فرضی زیر را در نظر بگیرید.
- فرض کنید ایمیلی از یک محقق پزشکی در مورد پروژه ای که مشتاق کار روی آن هستید، به شرح زیر دریافت کرده اید.
- داده های زیر اطلاعات ۱۰۰۰ بیمار است.
- هر سطر حاوی اطلاعات یک بیمار است و از پنج فیلد تشکیل شده است.
- ما می خواهیم آخرین فیلد را با استفاده از فیلدهای دیگر پیش بینی کنیم.
- از آنجایی که برای چند روز به خارج از شهر می روم، وقت ندارم اطلاعات بیشتری در مورد داده ها ارائه کنم، اما امیدوارم این کار شما را خیلی کند نکند. و اگر مشکلی ندارید، آیا می توانیم وقتی برگشتم برای بحث درباره نتایج اولیه تان ملاقات کنیم؟ ممکن است چند نفر دیگر از تیم را دعوت کنم.

## ادامه مثال

012	232	33.5	0	10.7
020	121	16.9	2	210.1
027	165	24.0	0	427.6
⋮	⋮	⋮	⋮	⋮

- نگاهی کوتاه به داده ها چیز عجیبی را نشان نمی دهد. شما شک خود را کنار بگذارید و تحلیل را شروع کنید.
- اما دو روز بعد، احساس می کنید که پیشرفت کرده اید. شما به جلسه می آید
- در حالی که منتظر رسیدن دیگران هستید، با یک آماردان که روی همین پروژه کار می کند صحبت می کنید.
- وقتی متوجه می شود که شما نیز داده های پروژه کار کرده اید، از او می پرسد که آیا می توانید یک مروف مختصر از نتایج خود به او بدهید.

## ادامه مثال

- آماردان: شما داده های همه بیماران را تحلیل کردید؟
- متخصص علم داده: بله. علی رغم آنکه زمان زیادی برای تحلیل نداشتم، ولی همه داده ها را تحلیل کردم و چند نتیجه جالب بدست آوردم.
- آماردان: عالی است! مشکلات زیادی روی این داده ها وجود داشت بنابراین من نتوانستم تمامی داده ها را تحلیل کنم.
- متخصص علم داده: اوه! من در هیچ مشکل خاصی ندیدم.
- آماردان: خب، ابتدا فیلد ۵ وجود دارد، متغیری که می خواهیم پیش بینی کنیم. در میان افرادی که این نوع داده ها را تحلیل می کنند، رایج است که اگر بالگاریتم مقادیر کار کنید، نتایج بهتری بدست می آورند. من این را من بعداً متوجه شدم، آیا این موضوع را به شما گفتند؟
- متخصص علم داده: خیر.

- آماردان: اما مطمئناً در مورد فیلد ۴ شنیده اید؟ قرار است در مقیاسی از ۱ تا ۱۰ اندازه گیری شود که نشان دهنده مقدار گمشده است، اما به دلیل خطای ورود داده ها، همه ۱۰ ها به تبدیل شدند. متأسفانه، از آنجایی که برخی از بیماران مقادیر گمشده ای برای این فیلد دارند، نمی توان گفت که در این فیلد واقعی است یا نه. تعداد کمی از رکوردها این مشکل را دارند.
- داده کاو: جالب است. آیا مشکلات دیگری وجود داشت؟
- آماردان: بله، فیلد های ۲ و ۳ اساساً یکسان هستند، اما من فرض می کنم که شما احتمالاً متوجه این موضوع شده اید.
- متخصص علم داده: بله، اما این فیلد ها تنها پیش بینی کننده های ضعیف برای فیلد ۵ بودند.
- آماردان: به هر حال، با توجه به همه آن مشکلات، من تعجب می کنم که شما توانستید هر کاری را انجام دهید.

- متخصص علم داده: درست است، اما نتایج من واقعاً خوب است. فیلد ۱ یک پیش بینی کننده بسیار قوی برای فیلد ۵ است. من متعجبم که قبلاً به این موضوع توجه نشده بود.
- آماردان: چی؟ فیلد ۱ فقط یک شماره شناسه است.
- متخصص علم داده: با این وجود، نتایج من برای خود صحبت می کند.
- آماردان: اوه، نه! تازه یادم اومد ما پس از مرتب سازی رکوردها بر اساس فیلد ۵، شماره های شناسه را اختصاص دادیم. ارتباط قوی وجود دارد، اما بی معنی است. متأسفم
- این مثال بر اهمیت "شناخت ماهیت داده ها" تأکید می کند.

# مروی بر مفاهیم اولیه

## 5.1 Learning Algorithms

A machine learning algorithm is an algorithm that is able to learn from data. But what do we mean by learning? Mitchell (1997) provides a succinct definition: “A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ .” One can imagine a wide variety of experiences  $E$ , tasks  $T$ , and performance measures  $P$ , and we do not attempt in this book to formally define what may be used for each of these entities. Instead, in the following sections, we provide intuitive descriptions and examples of the different kinds of tasks, performance measures, and experiences that can be used to construct machine learning algorithms.

### 5.1.1 The Task, $T$

Machine learning enables us to tackle tasks that are too difficult to solve with fixed programs written and designed by human beings. From a scientific and philosophical point of view, machine learning is interesting because developing our understanding of it entails developing our understanding of the principles that underlie intelligence.

In this relatively formal definition of the word “task,” the process of learning itself is not the task. Learning is our means of attaining the ability to perform the task. For example, if we want a robot to be able to walk, then walking is the task. We could program the robot to learn to walk, or we could attempt to directly write a program that specifies how to walk manually.

Many kinds of tasks can be solved with machine learning. Some of the most common machine learning tasks include the following:

**Classification:** In this type of task, the computer program is asked to specify which of  $k$  categories some input belongs to. To solve this task, the learning algorithm is usually asked to produce a function  $f : \mathbb{R}^n \rightarrow \{1, \dots, k\}$ . When  $y = f(\mathbf{x})$ , the model assigns an input described by vector  $\mathbf{x}$  to a category identified by numeric code  $y$ . There are other variants of the classification task, for example, where  $f$  outputs a probability distribution over classes. An example of a classification task is object recognition, where the input is an image (usually described as a set of pixel brightness values), and the output is a numeric code identifying the object in the image. For example, the Willow Garage PR2 robot is able to act as a waiter that can recognize different kinds of drinks and deliver them to people on command (Good-

**Classification with missing inputs:** Classification becomes more challenging if the computer program is not guaranteed that every measurement in its input vector will always be provided. To solve the classification task, the learning algorithm only has to define a *single* function mapping from a vector input to a categorical output. When some of the inputs may be missing, rather than providing a single classification function, the learning algorithm must learn **a set of functions**. Each function corresponds to classifying  $\mathbf{x}$  with a different subset of its inputs missing. This kind of situation arises frequently in medical diagnosis, because many kinds of medical tests are expensive or invasive. One way to efficiently define such a large set of functions is to learn a probability distribution over all the relevant variables, then solve the classification task by marginalizing out the missing variables. With  $n$  input variables, we can now obtain all  $2^n$  different classification functions needed for each possible set of missing inputs, but the computer program needs to learn only a single function describing the joint probability distribution. See Goodfellow et al. (2013b) for an example of a deep probabilistic model applied to such a task in this way. Many of the other tasks described in this section can also be generalized to work with missing inputs; classification with missing inputs is just one example of what machine learning can do.

**Regression:** In this type of task, the computer program is asked to predict a numerical value given some input. To solve this task, the learning algorithm is asked to output a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ . This type of task is similar to classification, except that the format of output is different. An example of a regression task is the prediction of the expected claim amount that an insured person will make (used to set insurance premiums), or the prediction of future prices of securities. These kinds of predictions are also used for algorithmic trading.

**Transcription:** In this type of task, the machine learning system is asked to observe a relatively unstructured representation of some kind of data and transcribe the information into discrete textual form. For example, in optical character recognition, the computer program is shown a photograph containing an image of text and is asked to return this text in the form of a sequence of characters (e.g., in ASCII or Unicode format). Google Street View uses deep learning to process address numbers in this way (Goodfellow et al., 2014d). Another example is speech recognition, where the computer program is provided an audio waveform and emits a sequence of characters or word ID codes describing the words that were spoken in the audio recording. Deep learning is a crucial component of modern speech recognition systems used at major companies, including Microsoft, IBM and Google (Hinton et al., 2012b).

**Machine translation:** In a machine translation task, the input already consists of a sequence of symbols in some language, and the computer program must convert this into a sequence of symbols in another language. This is commonly applied to natural languages, such as translating from English to French. Deep learning has recently begun to have an important impact on this kind of task (Sutskever et al., 2014; Bahdanau et al., 2015).

**Structured output:** Structured output tasks involve any task where the output is a vector (or other data structure containing multiple values) with important relationships between the different elements. This is a broad category and subsumes the transcription and translation tasks described above, as well as many other tasks. One example is parsing—mapping a natural language sentence into a tree that describes its grammatical structure by tagging nodes of the trees as being verbs, nouns, adverbs, and so on. See Collobert (2011) for an example of deep learning applied to a parsing task. Another example is pixel-wise segmentation of images, where the computer program assigns every pixel in an image to a specific category.

**Anomaly detection:** In this type of task, the computer program sifts through a set of events or objects and flags some of them as being unusual or atypical. An example of an anomaly detection task is credit card fraud detection. By modeling your purchasing habits, a credit card company can detect misuse of your cards. If a thief steals your credit card or credit card information, the thief’s purchases will often come from a different probability distribution over purchase types than your own. The credit card company can prevent fraud by placing a hold on an account as soon as that card has been used for an uncharacteristic purchase. See Chandola et al. (2009) for a survey of anomaly detection methods.

**Synthesis and sampling:** In this type of task, the machine learning algorithm is asked to generate new examples that are similar to those in the training data. Synthesis and sampling via machine learning can be useful for media applications when generating large volumes of content by hand would be expensive, boring, or require too much time. For example, video games can automatically generate textures for large objects or landscapes, rather than requiring an artist to manually label each pixel (Luo et al., 2013). In some cases, we want the sampling or synthesis procedure to generate a specific kind of output given the input. For example, in a speech synthesis task, we provide a written sentence and ask the program to emit an audio waveform containing a spoken version of that sentence. This is a kind of structured output task, but with the added qualification that there is no single correct output for each input, and we explicitly desire a large amount of variation in the output, in order for the output to seem more natural and realistic.

**Imputation of missing values:** In this type of task, the machine learning algorithm is given a new example  $\mathbf{x} \in \mathbb{R}^n$ , but with some entries  $x_i$  of  $\mathbf{x}$  missing. The algorithm must provide a prediction of the values of the missing entries.

**Denoising:** In this type of task, the machine learning algorithm is given in input a *corrupted example*  $\tilde{\mathbf{x}} \in \mathbb{R}^n$  obtained by an unknown corruption process from a *clean example*  $\mathbf{x} \in \mathbb{R}^n$ . The learner must predict the clean example  $\mathbf{x}$  from its corrupted version  $\tilde{\mathbf{x}}$ , or more generally predict the conditional probability distribution  $p(\mathbf{x} | \tilde{\mathbf{x}})$ .

**Density estimation or probability mass function estimation:** In the density estimation problem, the machine learning algorithm is asked to learn a function  $p_{\text{model}} : \mathbb{R}^n \rightarrow \mathbb{R}$ , where  $p_{\text{model}}(\mathbf{x})$  can be interpreted as a probability density function (if  $\mathbf{x}$  is continuous) or a probability mass function (if  $\mathbf{x}$  is discrete) on the space that the examples were drawn from. To do such a task well (we will specify exactly what that means when we discuss performance measures  $P$ ), the algorithm needs to learn the structure of the data it has seen. It must know where examples cluster tightly and where they are unlikely to occur. Most of the tasks described above require the learning algorithm to at least implicitly capture the structure of the probability distribution. Density estimation enables us to explicitly capture that distribution. In principle, we can then perform computations on that distribution to solve the other tasks as well. For example, if we have performed density estimation to obtain a probability distribution  $p(\mathbf{x})$ , we can use that distribution to solve the missing value imputation task. If a value  $x_i$  is missing, and all the other values, denoted  $\mathbf{x}_{-i}$ , are given, then we know the distribution over it is given by  $p(x_i | \mathbf{x}_{-i})$ . In practice, density estimation does not always enable us to solve all these related tasks, because in many cases the required operations on  $p(\mathbf{x})$  are computationally intractable.

## مروری بر مفاهیم اولیه

- یک مجموعه داده را اغلب می توان به عنوان مجموعه ای از اشیاء تعریف نمود.
- نام های دیگر یک داده شامل رکورد، نقطه، بردار، الگو، رویداد، مورد، نمونه، مشاهده یا موجودیت است.
- به نوبه خود، هر داده با تعدادی ویژگی (attribute) توصیف می شوند که ویژگی های یک شی، مانند جرم یک جسم فیزیکی یا زمانی که یک رویداد در آن رخ داده است، توصیف می شوند.
- نام های دیگر یک ویژگی متغیر، مشخصه، فیلد، ویژگی (feature) یا بعد است.

# مثال اطلاعات دانشجویان

Table 2.1. A sample data set containing student information.

Student ID	Year	Grade Point Average (GPA)	...
	:		
1034262	Senior	3.24	...
1052663	Sophomore	3.51	...
1082246	Freshman	3.62	...
	:		

- تعریف صفت یا ویژگی: ویژگی یک شی است که می‌تواند از یک شی به شیء دیگر، یا از زمانی به زمان دیگر متفاوت باشد.
- به عنوان مثال، رنگ چشم از فردی به فرد دیگر متفاوت است،
- در حالی که دمای یک شی در طول زمان متفاوت است.
- لازم به ذکر است رنگ چشم یک ویژگی نمادین با تعداد کمی مقدار ممکن (قهوه‌ای، سیاه، آبی، سبز، فندقی و غیره) است،
- در حالی که دما یک ویژگی عددی با تعداد بالقوه نامحدود مقدار است.

- مقیاس اندازه گیری یک قانون (تابع) است که یک مقدار عددی یا یک نماد را با یک ویژگی یک شی مرتبط می کند.
- معمولاً نوع یک ویژگی بر اساس مقیاس اندازه گیری آن ویژگی تعیین می شود.
- یک ویژگی را می توان با استفاده از مقیاس های اندازه گیری مختلف توصیف کرد
- یک راه ساده و مفید برای تعیین نوع یک ویژگی، شناسایی ویژگی های اعدادی است که با آن ویژگی مرتبط است.
- به عنوان مثال، یک ویژگی مانند طول دارای بسیاری از خواص های اعداد است. می توان آنرا از نظر منطقی مقایسه و مترتب نمود.

- عملگرهای چهارگانه زیر برای تعیین نوع یک ویژگی استفاده می شود.

1. Distinctness = and  $\neq$
2. Order  $<$ ,  $\leq$ ,  $>$ , and  $\geq$
3. Addition + and -
4. Multiplication  $\times$  and /

- با توجه به این عملگرها می توانیم چهار نوع صفت زیر را تعریف کنیم:
- اسمی (nominal)، ترتیبی (ordinal)، فاصله (interval) و نسبتی (ratio).
- جدول زیر تعاریف این انواع را به همراه اطلاعاتی در مورد عملیات آماری معتبر برای هر نوع ارائه می دهد.

- هر نوع ویژگی دارای تمام ویژگی ها و عملیات انواع ویژگی های بالای خود است.
- در نتیجه، هر ویژگی یا عملیاتی که برای صفات اسمی، ترتیبی و فاصله ای معتبر باشد، برای ویژگی های نیز معتبر است.
- به عبارت دیگر، تعریف انواع صفت به صورت تجمعی است.

**Table 2.2.** Different attribute types.

Attribute Type	Description	Examples	Operations
Categorical (Qualitative)	Nominal  The values of a nominal attribute are just different names; i.e., nominal values provide only enough information to distinguish one object from another. $(=, \neq)$	zip codes, employee ID numbers, eye color, gender	mode, entropy, contingency correlation, $\chi^2$ test
	Ordinal  The values of an ordinal attribute provide enough information to order objects. $(<, >)$	hardness of minerals, $\{good, better, best\}$ , grades, street numbers	median, percentiles, rank correlation, run tests, sign tests

Numeric (Quantitative)	Interval	For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. $(+, -)$	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, $t$ and $F$ tests
	Ratio	For ratio variables, both differences and ratios are meaningful. $(\times, /)$	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

# تعیین نوع صفت بر اساس مقداری که می تواند قبول کند

- گستته: یک ویژگی گستته دارای مجموعه مقادیر متناهی یا نامتناهی (شمارا) است.
- چنین ویژگی هایی می توانند دسته بندی شوند،
- مانند کد پستی یا شماره شناسه
- ویژگی های گستته اغلب با استفاده از اعداد صحیح نمایش داده می شوند.
- ویژگی های باینری (دوتایی) یک مورد خاص از ویژگی های گستته است و فقط دو مقدار را قبول می کند، به عنوان مثال درست/نادرست، بله/خیر، مرد/زن
- پیوسته: یک صفت پیوسته، صفتی است که مقادیر آن اعداد حقیقی باشد.
- به عنوان مثال می توان به ویژگی هایی مانند دما، قد یا وزن اشاره کرد.

## ویژگی‌های کلی مجموعه‌های داده

- مجموعه‌های داده، حداقل سه ویژگی زیر را دارند: این ویژگیها تأثیر قابل توجهی بر تکنیک‌های داده‌کاوی مورد استفاده دارد.
- ابعاد: ابعاد یک مجموعه داده، تعداد ویژگی‌هایی است که اشیاء در آن مجموعه داده دارند. تحلیل داده‌ها با ابعاد کم از نظر کیفی با تحلیل داده‌های متوسط یا با ابعاد بالا متفاوت است.
- توزیع: توزیع یک مجموعه داده عبارت است از فراوانی وقوع مقادیر مختلف در نظر گرفت. به طور معادل، توزیع یک مجموعه داده را می‌توان به عنوان توصیفی از غلظت اشیاء در مناطق مختلف فضای داده در نظر گرفت. آماردانان انواع مختلفی از توزیع‌ها را بر شمرده‌اند، به عنوان مثال، گاوی (عادی) و خواص آنها را شرح داده‌اند.
- وضوح: اغلب امکان به دست آوردن داده‌ها در سطوح مختلف وضوح وجود دارد و اغلب خواص داده‌ها در وضوح‌های مختلف متفاوت است. به عنوان مثال، سطح زمین در وضوح چند متری بسیار ناهموار، اما در وضوح ده‌ها کیلومتری نسبتاً صاف است. الگوهای موجود در داده‌ها نیز به سطح وضوح بستگی دارد. اگر وضوح خیلی خوب باشد، ممکن است یک الگو قابل مشاهده نباشد یا ممکن است در نویز پنهان شود. اگر وضوح بیش از حد درشت باشد، الگو می‌تواند ناپدید شود. به عنوان مثال، تغییرات فشار اتمسفر در مقیاس ساعت، حرکت طوفان‌ها و سایر سیستم‌های آب و هوایی را منعکس می‌کند. در مقیاس ماه‌ها، چنین پدیده‌هایی قابل تشخیص نیستند.

## انواع مجموعه داده ها

- انواع مختلفی از مجموعه داده ها وجود دارد، و با توسعه و بلوغ حوزه داده کاوی، مجموعه های داده ای متنوع تری برای تحلیل در دسترس قرار می گیرند.
- در ادامه، برخی از رایج ترین انواع مجموعه داده ها را شرح می دهیم.
- برای سهولت، انواع مجموعه داده ها را به سه گروه دسته بندی کرده ایم:
- **داده های رکورد**,
- **داده های مبتنی بر نمودار**
- **داده های مرتب شده**.
- این دسته بندی ها همه احتمالات را پوشش نمی دهند و گروه بندی های دیگر قطعاً امکان پذیر است.

# داده‌های رکورد

- در بسیاری از روش‌های داده کاوی فرض بر آن است که مجموعه داده مجموعه‌ای از رکوردها (اشیاء داده) است.
- که هر یک از مجموعه ثابتی از فیلد‌های داده (ویژگی‌ها) تشکیل شده است. شکل ۲.۲ (a) را بینید.
- برای ابتدایی ترین شکل داده رکورد، هیچ رابطه صریحی بین رکوردها یا فیلد‌های داده وجود ندارد و هر رکورد (شیء) دارای مجموعه‌ای از ویژگی‌ها است.
- داده‌های رکورد معمولاً یا در فایل‌های مسطح یا در پایگاه‌های داده مرتبط ذخیره می‌شوند.
- پایگاه داده‌های رابطه‌ای مطمئناً بیش از مجموعه‌ای از رکوردها هستند، اما داده کاوی اغلب از هیچ یک از اطلاعات اضافی موجود در پایگاه داده مرتبط استفاده نمی‌کند.
- در عوض، پایگاه داده به عنوان مکانی مناسب برای یافتن رکوردها عمل می‌کند. انواع مختلف داده‌های رکورد در شکل زیر توضیح داده شده است.

Tid	Refund	Marital Status	Taxable Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

(a) Record data.

TID	ITEMS
1	Bread, Soda, Milk
2	Beer, Bread
3	Beer, Soda, Diapers, Milk
4	Beer, Bread, Diapers, Milk
5	Soda, Diapers, Milk

(b) Transaction data.

Projection of x Load	Projection of y Load	Distance	Load	Thickness
10.23	5.27	15.22	27	1.2
12.65	6.25	16.22	22	1.1
13.54	7.23	17.34	23	1.2
14.27	8.43	18.45	25	0.9

(c) Data matrix.

	team	coach	play	g	score	game	wh	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

(d) Document-term matrix.

Figure 2.2. Different variations of record data.

- داده های تراکنش یا سبد بازار: داده های تراکنش نوع خاصی از داده های ثبت شده هستند که هر رکورد (معامله) شامل مجموعه ای از اقلام خریداری شده است.
- یک فروشگاه مواد غذایی را در نظر بگیرید. مجموعه محصولات خریداری شده توسط مشتری در طی یک خرید، یک معامله را تشکیل می دهد، در حالی که تک تک محصولات خریداری شده، اقلام هستند.
- به این نوع داده ها داده های سبد بازار می گویند، زیرا اقلام موجود در هر رکورد، محصولات موجود در "سبد بازار" یک فرد است.
- داده های تراکنش اغلب، ویژگی ها باینری دارند و نشان می دهند که آیا یک کالا خریداری شده است یا خیر،
- اما به طور کلی، ویژگی ها می توانند گستته یا پیوسته باشند، مانند تعداد اقلام خریداری شده یا مبلغی که برای آن اقلام هزینه شده است. شکل ۲.۲ (b) یک مجموعه داده تراکنش نمونه را نشان می دهد. هر ردیف نشان دهنده خریدهای یک مشتری خاص در یک زمان خاص است.

- ماتریس داده: اگر همه اشیاء داده در مجموعه داده ها دارای مجموعه ثابتی از ویژگی های عددی باشند، آنگاه اشیاء داده را می توان به عنوان نقاط (بردار) در یک فضای چند بعدی در نظر گرفت،
- مجموعه ای از چنین اشیاء داده ای را می توان به عنوان یک ماتریس  $m \times n$  تفسیر کرد، که در آن  $m$  ردیف، یک برای هر شی، و  $n$  ستون، برای هر ویژگی وجود دارد
- شکل ۲.۲ (c) یک ماتریس داده نمونه را نشان می دهد.

- ماتریس داده های پراکنده: ماتریس داده های پراکنده یک حالت خاص از یک ماتریس داده است

- که در آن ویژگی ها از یک نوع هستند و نامتقارن هستند. یعنی فقط مقادیر غیر صفر مهم هستند.
- اگر ترتیب عبارات (کلمات) در یک سند نادیده گرفته شود، آنگاه یک سند را می توان به عنوان یک بردار اصطلاح نشان داد، که در آن هر عبارت جزء (ویژگی) بردار است و مقدار هر جزء تعداد دفعاتی است که واژه مربوطه در سند مشاهده شده است. این نمایش مجموعه ای از اسناد اغلب ماتریس سند-واژه نامیده می شود. شکل ۲.۲ (d)

## داده های مبتنی بر نمودار

- یک نمودار گاهی اوقات می تواند یک نمایش راحت و قدرتمند برای داده ها باشد.
- ما دو مورد خاص را در نظر می گیریم:
  - (۱) نمودار روابط بین اشیاء داده را ثبت می کند
  - (۲) خود اشیاء داده به عنوان نمودار نمایش داده می شوند.

# داده ها با روابط بین اشیاء

- روابط بین اشیا اغلب اطلاعات مهمی را منتقل می کند.
- در چنین مواردی، داده ها اغلب به صورت نمودار نمایش داده می شوند.
- به طور خاص، اشیاء داده به گره های گراف نگاشت می شوند، در حالی که روابط بین اشیاء توسط پیوندهای بین اشیا و ویژگی های پیوند، مانند جهت و وزن، ثبت می شوند.
- صفحات وب را در شبکه جهانی وب در نظر بگیرید که حاوی متن و پیوندهایی به صفحات دیگر هستند.
- به منظور پردازش جستجو، موتورهای جستجوی وب صفحات وب را جمع آوری و پردازش می کنند تا محتوای آنها را استخراج کنند. با این حال، به خوبی شناخته شده است که پیوندها **به واژه هر** صفحه اطلاعات زیادی در مورد ارتباط یک صفحه وب با یک جستجو را ارائه می دهند، و بنابراین، باید در نظر گرفته شوند (شکل زیر).
- مثال دیگر از این داده های نموداری، شبکه های اجتماعی هستند که در آن اشیاء داده، افراد و روابط بین آنها، تعامل آنها از طریق رسانه های اجتماعی است.

## Useful Links:

- [Bibliography](#)
- Other Useful Web sites
  - [ACM SIGKDD](#)
  - [KDnuggets](#)
  - [The Data Mine](#)

## Knowledge Discovery and Data Mining Bibliography

(Gets updated frequently, so visit often!)

- [Books](#)
- [General Data Mining](#)

### Book References in Data Mining and Knowledge Discovery

Usama Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Ithurusamy, "Advances in Knowledge Discovery and Data Mining", AAAI Press/the MIT Press, 1996.

J. Ross Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers, 1993. Michael Berry and Gordon Linoff, "Data Mining Techniques (For Marketing, Sales, and Customer Support)", John Wiley & Sons, 1997.

### General Data Mining

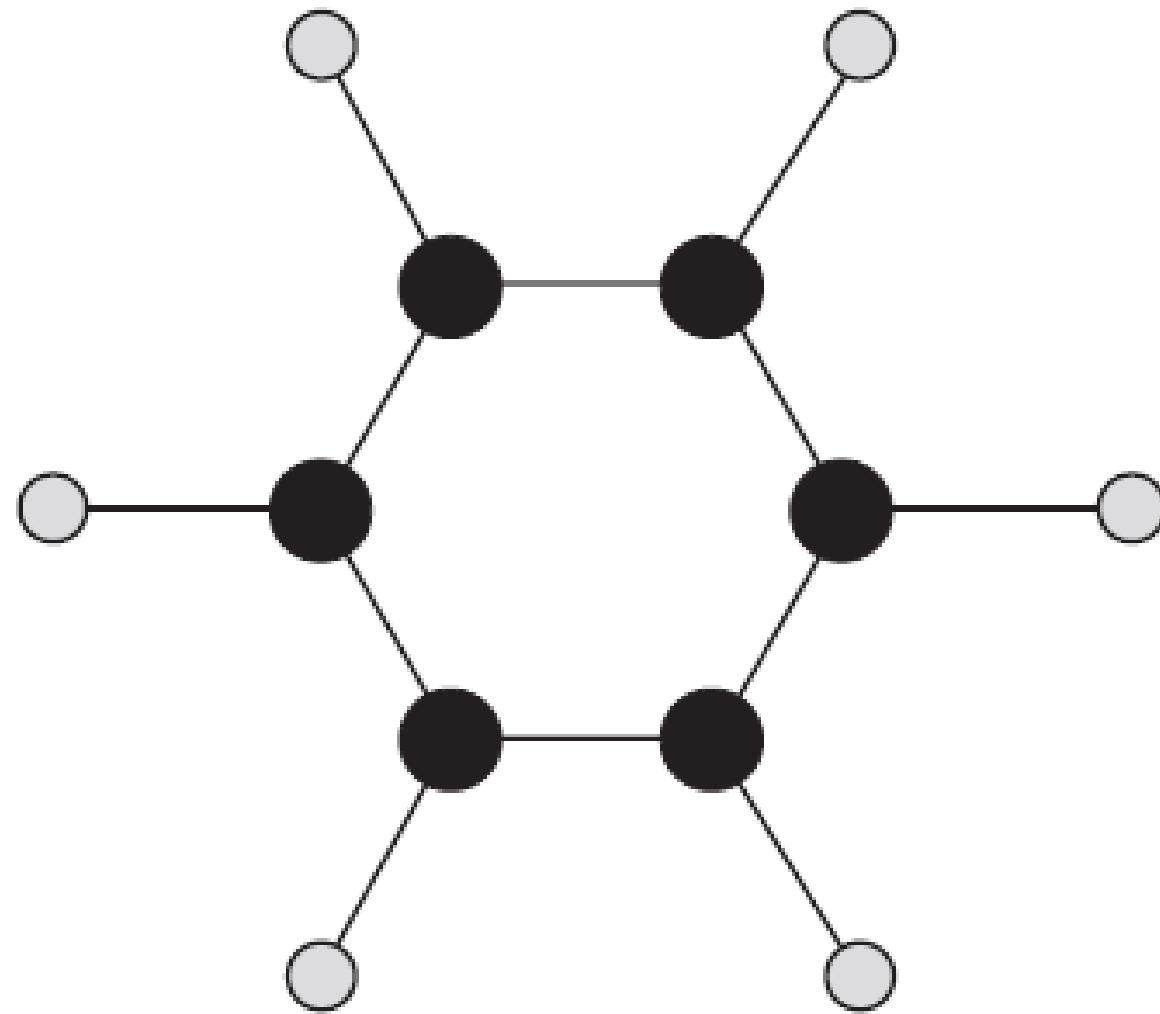
Usama Fayyad, "Mining Databases: Towards Algorithms for Knowledge Discovery", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, vol. 21, no. 1, March 1998.

Christopher Matheus, Philip Chan, and Gregory Piatetsky-Shapiro, "Systems for knowledge Discovery in databases", IEEE Transactions on Knowledge and Data Engineering, 5(6):903-913, December 1993.

## داده ها با اشیایی که نمودار هستند

- اگر اشیاء ساختاری داشته باشند، یعنی اشیاء حاوی اشیاء فرعی هستند که دارای روابط هستند، آنگاه چنین اشیایی اغلب به صورت نمودار نمایش داده می شوند.
- به عنوان مثال، ساختار ترکیبات شیمیایی را می توان با یک نمودار نشان داد، که در آن گره ها اتم هستند و پیوندهای بین گره ها پیوندهای شیمیایی هستند.
- شکل زیر ترکیب شیمیایی بنزن را نشان می دهد که حاوی اتم های کربن (سیاه) و هیدروژن (خاکستری) است.
- نمایش گراف این امکان را فراهم می کند که مشخص شود کدام زیرساختها اغلب در مجموعه ای از ترکیبات رخ می دهند و تعیین اینکه آیا وجود هر یک از این زیرساختها با وجود یا عدم وجود خواص شیمیایی خاص، مانند نقطه ذوب یا گرمای تشکیل مرتبط است یا خیر.
- نمودار کاوی مکرر، که شاخه ای از داده کاوی است که چنین داده هایی را تحلیل می کند.

# ترکیب شیمیایی بنزن



## داده های مرتب شده

- برای برخی از انواع داده ها، ویژگی ها دارای روابطی هستند که شامل نظم در زمان یا مکان است.
- انواع مختلف داده های مرتب شده عبارتند از:
- **داده های تراکنش متوالی:** داده های تراکنش متوالی را می توان به عنوان بسط داده های تراکنش در نظر گرفت، که در آن هر تراکنش دارای زمانی مرتبط با آن است. شکل ۲.۴ (a)
- **سری زمانی:** داده های سری زمانی نوع خاصی از داده های مرتب شده است که در آن هر رکورد یک سری زمانی است، یعنی یک سری اندازه گیری در طول زمان. شکل ۲.۴ (c)

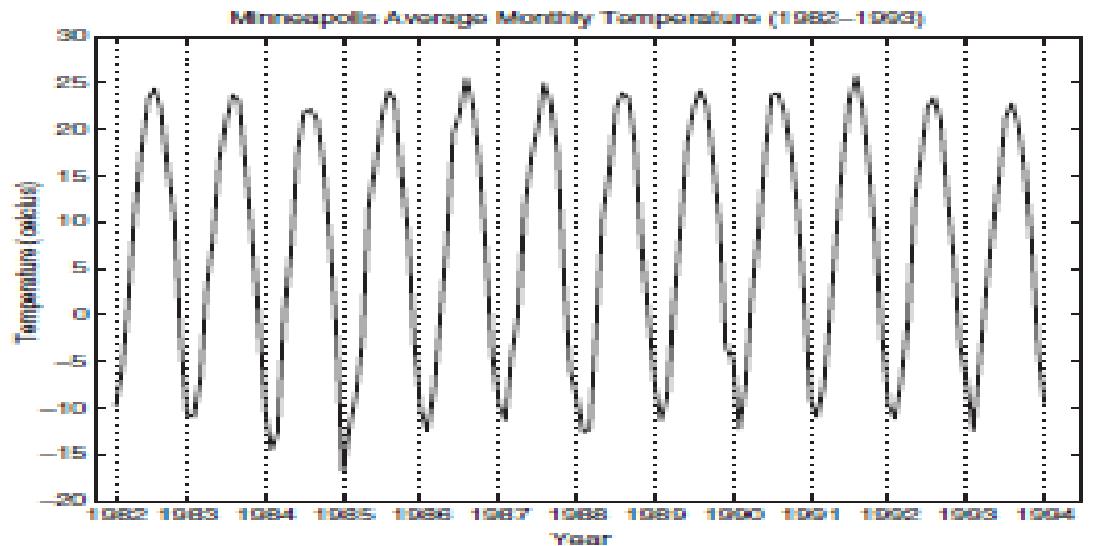
## داده های مرتب شده

- **داده های توالی:** داده های توالی شامل مجموعه داده ای است که دنباله ای از موجودیت های منفرد هستند، مانند دنباله ای از کلمات یا حروف. این کاملاً شبیه به داده های متوالی است، با این تفاوت که هیچ برچسب زمانی وجود ندارد. در عوض، موقعیت ها در یک دنباله مرتب وجود دارد. به عنوان مثال، اطلاعات ژنتیکی گیاهان و جانوران را می توان در قالب توالی هایی از نوکلئوتیدها که به عنوان ژن شناخته می شوند، نشان داد. شکل ۲.۴ (b).
- **داده های مکانی و مکانی - زمانی:** برخی از اشیاء علاوه بر انواع دیگر ویژگی ها دارای ویژگی های مکانی مانند موقعیت ها هستند. نمونه ای از داده های مکانی، داده های آب و هوا (بارش، دما، فشار) است که برای مکان های جغرافیایی مختلف جمع آوری می شود.
- اغلب چنین اندازه گیری هایی در طول زمان جمع آوری می شوند و بنابراین، داده ها از سری های زمانی در مکان های مختلف تشکیل می شوند. در آن صورت به داده ها به عنوان داده های مکانی - زمانی اشاره می کنیم. اگرچه تحلیل می تواند به طور جداگانه برای هر زمان یا مکان خاص انجام شود، تحلیل کامل تر داده های مکانی - زمانی نیاز به در نظر گرفتن هر دو جنبه مکانی و زمانی داده ها دارد. شکل ۲.۴ (d).

Time	Customer	Items Purchased
t1	C1	A, B
t2	C3	A, C
t2	C1	C, D
t3	C2	A, D
t4	C2	E
t5	C1	A, E

Customer	Time and Items Purchased
C1	(t1: A,B) (t2:C,D) (t5:A,E)
C2	(t3: A, D) (t4: E)
C3	(t2: A, C)

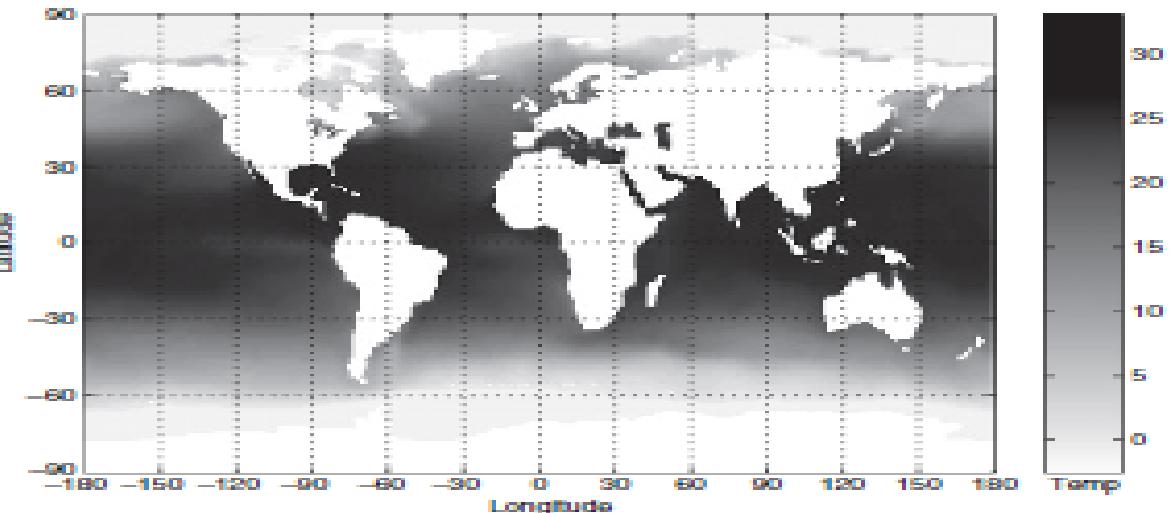
(a) Sequential transaction data.



(c) Temperature time series.

GGTTCCGCCTTCAGCCCCGCGCC  
CGCAGGGCCCAGCCCCGCGCCGTC  
GAGAAGGGCCCGCCCTGGCGGGCG  
GGGGGAGGCAGGGCCGCCGAGC  
CCAACCGAGTCCGACCAGGTGCC  
CCCTCTGCTCGGCCTAGACCTGA  
GCTCATTAGGCAGCAGCGGACAG  
GCCAAAGTAGAACACCGCGAAAGCGC  
TGGGGCTGCCTGCTGCGACCAGGG

(b) Genomic sequence data.



(d) Spatial temperature data.

Flaure 2.4. Different variations of ordered data.

## کیفیت داده

- الگوریتم‌های داده کاوی اغلب بر روی داده‌هایی که برای هدف دیگری یا برای کاربردهای آینده، اما نامشخص جمع آوری شده‌اند، اعمال می‌شوند.
- به همین دلیل، داده کاوی معمولاً نمی‌تواند از مزایای قابل توجه «پرداختن به کیفیت جمع آوری از منبع» استفاده کند.
- در مقابل، در بسیاری از روش‌های آماری، کاربر به طراحی آزمایش‌ها یا بررسی‌هایی می‌پردازند که به سطح از پیش تعیین شده‌ای از کیفیت داده‌ها دست می‌یابند.
- داده کاوی بر (۱) تشخیص و تصحیح مشکلات کیفیت داده‌ها و (۲) استفاده از الگوریتم‌هایی که می‌توانند کیفیت داده‌های ضعیف را تحمل کنند، متمرکز است.
- اولین مرحله، تشخیص و تصحیح، اغلب پاکسازی داده نامیده می‌شود.

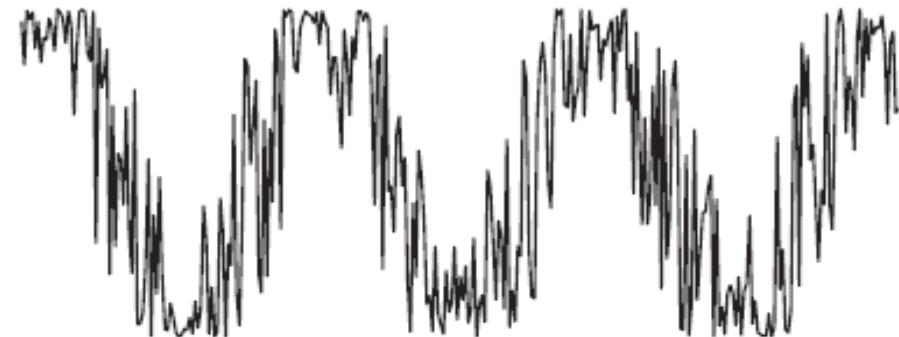
- اصطلاح خطای اندازه گیری به هر مشکلی اطلاق می شود که در نتیجه فرآیند اندازه گیری ایجاد شود.
- یک مشکل رایج این است که مقدار ثبت شده تا حدی با مقدار واقعی متفاوت باشد.
- برای صفات پیوسته، تفاوت عددی مقدار اندازه گیری شده و واقعی، را خطا می نامند.
- اصطلاح خطای جمع آوری داده ها به خطاهایی مانند حذف داده یا مقادیر ویژگی یا گنجاندن نامناسب یک داده اشاره دارد.
- به عنوان مثال، مطالعه حیوانات یک گونه خاص ممکن است شامل حیواناتی از گونه های مرتبط باشد که از نظر ظاهری مشابه گونه های مورد علاقه هستند.
- هم خطاهای اندازه گیری و هم خطاهای جمع آوری داده ها می توانند سیستماتیک یا تصادفی باشند.

## نویز و خطاهای ساختگی

- نویز جزء تصادفی یک خطای اندازه گیری است.
- معمولاً شامل تحریف یا انحراف یک مقدار یا اضافه کردن اجزاء جعلی است.
- شکل ۲.۵ یک سری زمانی قبل و بعد از اینکه اضافه شدن نویز تصادفی، را نشان می دهد.
- اگر کمی نویز بیشتر به سری زمانی اضافه می شد، شکل کاملاً از بین می رفت.



(a) Time series.



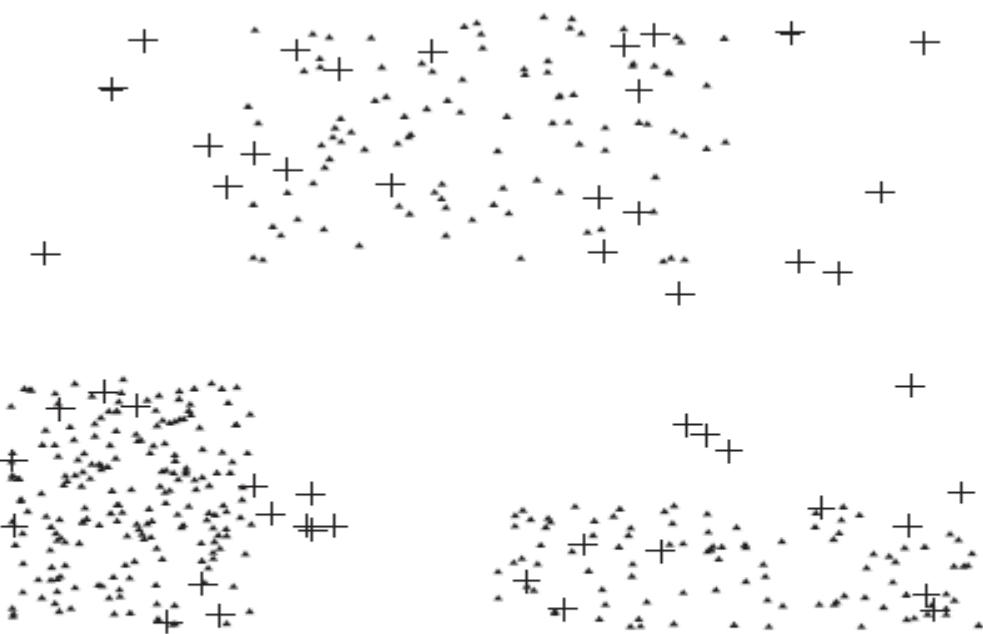
(b) Time series with noise.

Figure 2.5. Noise in a time series context.

- شکل ۲.۶ مجموعه ای از نقاط داده را قبل و بعد از اضافه شدن برخی نقاط نویز (که با علامت + مشخص می شود) نشان می دهد.
- توجه داشته باشید که برخی از نقاط نویز با نقاط غیر نویز آمیخته شده اند.



(a) Three groups of points.



(b) With noise points (+) added.

**Figure 2.6.** Noise in a spatial context.

- حذف نویز اغلب دشوار است،
- بسیاری از کارها در داده کاوی بر روی ابداع الگوریتم های پایدار (robust algorithm) متمرکز است که نتایج قابل قبولی را حتی در صورت وجود نویز ایجاد می کند.
- خطاهای داده می توانند نتیجه یک پدیده قطعی تر باشد، به آنها خطاهای ساختگی می گویند.

# دقت (Accuracy)، اریبی (Precision) و درستی (Bias)

- تعریف دقت: نزدیکی اندازه گیری های مکرر (با همان کمیت) به یکدیگر را دقت گویند.
- تعریف اریبی: تغییر سیستماتیک اندازه گیری ها از کمیت اندازه گیری شده، را اریبی گویند.
- دقت اغلب با انحراف استاندارد اندازه گیری می شود، در حالی که اریبی با گرفتن تفاوت بین میانگین مجموعه مقادیر و مقدار شناخته شده کمیت اندازه گیری شده اندازه گیری می شود.
- تعریف درستی: نزدیکی مقدار حاصل از مدل به مقدار واقعی را میزان درستی گویند.
- درستی به دقت و اریبی بستگی دارد، اما فرمول خاصی برای درستی از نظر این دو کمیت وجود ندارد.

### 5.1.2 The Performance Measure, $P$

To evaluate the abilities of a machine learning algorithm, we must design a quantitative measure of its performance. Usually this performance measure  $P$  is specific to the task  $T$  being carried out by the system.

For tasks such as classification, classification with missing inputs, and transcription, we often measure the **accuracy** of the model. Accuracy is just the proportion of examples for which the model produces the correct output. We can also obtain equivalent information by measuring the **error rate**, the proportion of examples for which the model produces an incorrect output. We often refer to the error rate as the expected 0-1 loss. The 0-1 loss on a particular example is 0 if it is correctly classified and 1 if it is not. For tasks such as density estimation, it does not make sense to measure accuracy, error rate, or any other kind of 0-1 loss. Instead, we must use a different performance metric that gives the model a continuous-valued score for each example. The most common approach is to report the average log-probability the model assigns to some examples.

Usually we are interested in how well the machine learning algorithm performs on data that it has not seen before, since this determines how well it will work when deployed in the real world. We therefore evaluate these performance measures using a **test set** of data that is separate from the data used for training the machine learning system.

# الگوریتم های یادگیری در علم داده

- الگوریتم های با نظارت
- الگوریتم های بدون نظارت
- الگوریتم های نیمه نظارتی

### 5.1.3 The Experience, $E$

Machine learning algorithms can be broadly categorized as **unsupervised** or **supervised** by what kind of experience they are allowed to have during the learning process.

Most of the learning algorithms in this book can be understood as being allowed to experience an entire **dataset**. A dataset is a collection of many examples, as defined in section 5.1.1. Sometimes we call examples **data points**.

**Unsupervised learning algorithms** experience a dataset containing many features, then learn useful properties of the structure of this dataset. In the context of deep learning, we usually want to learn the entire probability distribution that generated a dataset, whether explicitly, as in density estimation, or implicitly, for tasks like synthesis or denoising. Some other unsupervised learning algorithms perform other roles, like clustering, which consists of dividing the dataset into clusters of similar examples.

**Supervised learning algorithms** experience a dataset containing features, but each example is also associated with a **label** or **target**. For example, the Iris dataset is annotated with the species of each iris plant. A supervised learning algorithm can study the Iris dataset and learn to classify iris plants into three different species based on their measurements.

**Supervised learning algorithms** experience a dataset containing features, but each example is also associated with a **label** or **target**. For example, the Iris dataset is annotated with the species of each iris plant. A supervised learning algorithm can study the Iris dataset and learn to classify iris plants into three different species based on their measurements.

Roughly speaking, unsupervised learning involves observing several examples of a random vector  $\mathbf{x}$  and attempting to implicitly or explicitly learn the probability distribution  $p(\mathbf{x})$ , or some interesting properties of that distribution; while supervised learning involves observing several examples of a random vector  $\mathbf{x}$  and an associated value or vector  $\mathbf{y}$ , then learning to predict  $\mathbf{y}$  from  $\mathbf{x}$ , usually by estimating  $p(\mathbf{y} \mid \mathbf{x})$ . The term **supervised learning** originates from the view of the target  $\mathbf{y}$  being provided by an instructor or teacher who shows the machine learning system what to do. In unsupervised learning, there is no instructor or teacher, and the algorithm must learn to make sense of the data without this guide.

Unsupervised learning and supervised learning are not formally defined terms. The lines between them are often blurred. Many machine learning technologies can be used to perform both tasks. For example, the chain rule of probability states that for a vector  $\mathbf{x} \in \mathbb{R}^n$ , the joint distribution can be decomposed as

$$p(\mathbf{x}) = \prod_{i=1}^n p(x_i | x_1, \dots, x_{i-1}). \quad (5.1)$$

This decomposition means that we can solve the ostensibly unsupervised problem of modeling  $p(\mathbf{x})$  by splitting it into  $n$  supervised learning problems. Alternatively, we can solve the supervised learning problem of learning  $p(y | \mathbf{x})$  by using traditional unsupervised learning technologies to learn the joint distribution  $p(\mathbf{x}, y)$ , then inferring

$$p(y | \mathbf{x}) = \frac{p(\mathbf{x}, y)}{\sum_{y'} p(\mathbf{x}, y')}. \quad (5.2)$$

Though unsupervised learning and supervised learning are not completely formal or distinct concepts, they do help roughly categorize some of the things we do with machine learning algorithms. Traditionally, people refer to regression, classification and structured output problems as supervised learning. Density estimation in support of other tasks is usually considered unsupervised learning.

Other variants of the learning paradigm are possible. For example, in semi-supervised learning, some examples include a supervision target but others do not. In multi-instance learning, an entire collection of examples is labeled as containing or not containing an example of a class, but the individual members of the collection are not labeled. For a recent example of multi-instance learning with deep models, see Kotzias et al. (2015).

Some machine learning algorithms do not just experience a fixed dataset. For example, **reinforcement learning** algorithms interact with an environment, so there is a feedback loop between the learning system and its experiences. Such algorithms are beyond the scope of this book. Please see Sutton and Barto (1998) or Bertsekas and Tsitsiklis (1996) for information about reinforcement learning, and Mnih et al. (2013) for the deep learning approach to reinforcement learning.

• داده های پرت: (۱) داده ای هستند که به نوعی دارای ویژگی هایی هستند که با بسیاری از دیگر داده در مجموعه داده متفاوت است، یا (۲) مقادیر یک ویژگی که با توجه به مقادیر معمول برای آن ویژگی غیرعادی است.

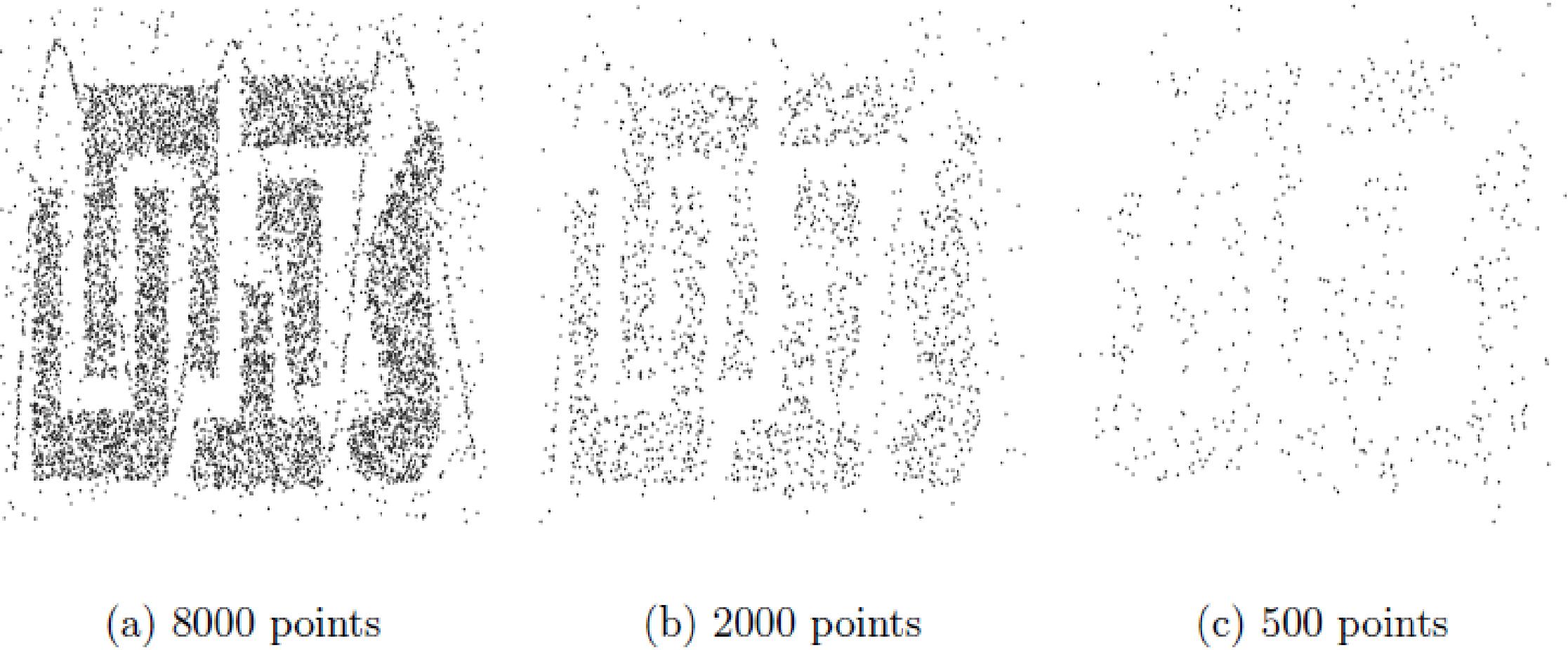
- از طرف دیگر، می توان آنها را به عنوان مقادیر غیرعادی نامید.
- آزادی عمل قابل توجهی در تعریف یک نقطه پرت وجود دارد و تعاریف مختلف زیادی توسط جوامع آمار و داده کاوی ارائه شده است.
- علاوه بر این، تمایز بین مفاهیم نویز و نقاط پرت مهم است.
- بر خلاف نویز، نقاط پرت می توانند اشیاء داده یا مقادیر قانونی باشند که ما علاقه مند به شناسایی آنها هستیم.
- به عنوان مثال، در تقلب و تشخیص نفوذ شبکه، هدف یافتن اشیا یا رویدادهای غیرعادی از میان تعداد زیادی از موارد عادی است.

## دانش در مورد داده ها

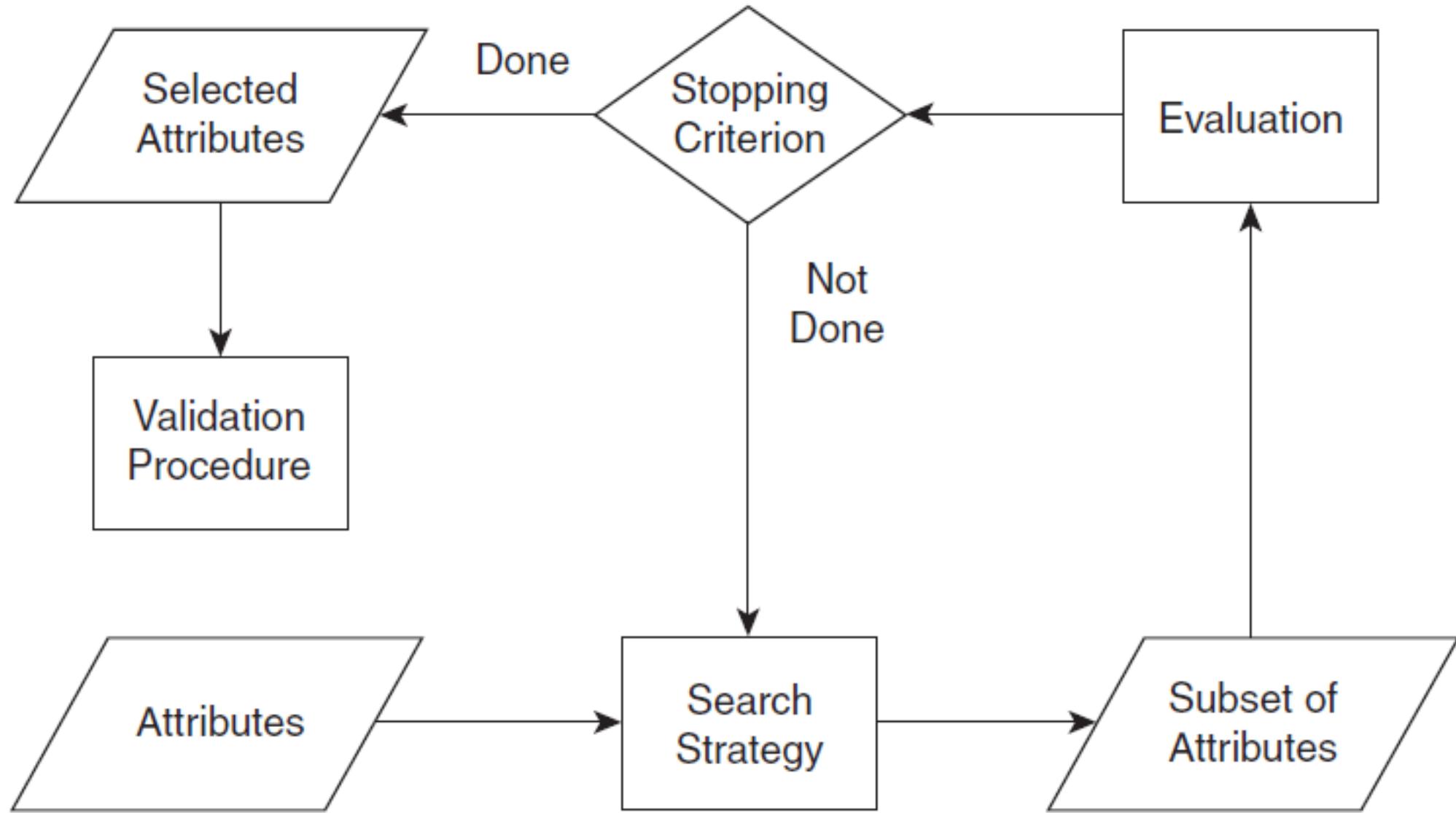
- در حالت ایده آل، مجموعه داده ها با اسنادی همراه هستند که جنبه های مختلف داده ها را توصیف می کند.
- کیفیت این مستندات می تواند به تحلیل بعدی کمک کند یا مانع از آن شود.
- به عنوان مثال، اگر اسناد چندین ویژگی را به عنوان ارتباط قوی شناسایی کند، این ویژگی ها احتمالاً اطلاعات بسیار اضافی را ارائه می کنند
- سایر ویژگی های مهم عبارتند از دقت داده ها، نوع ویژگی ها (اسمی، ترتیبی، فاصله، نسبت)، مقیاس اندازه گیری (مثلاً متر یا فوت برای طول)، و مبدأ داده ها.

## پیش پردازش داده ها

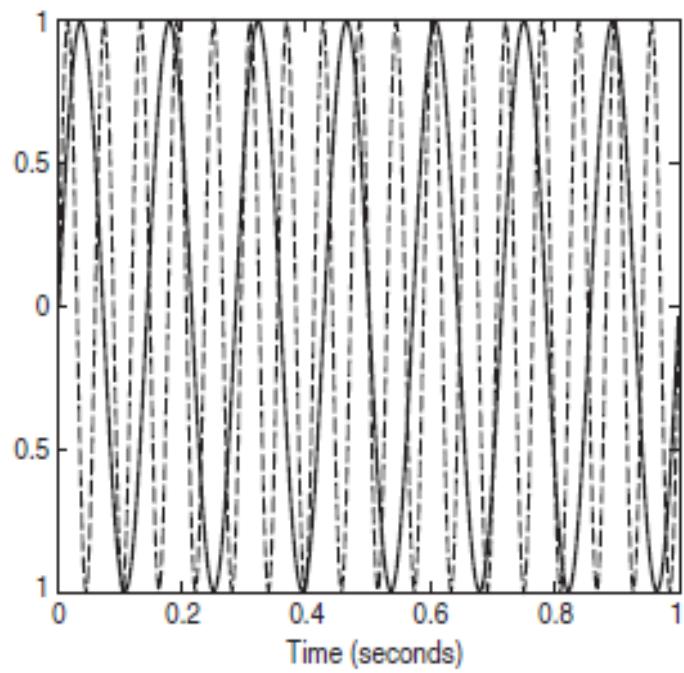
- پیش پردازش داده ها حوزه وسیعی است و شامل تعدادی استراتژی و تکنیک های مختلف می شود که به روش های پیچیده ای به هم مرتبط هستند.
- در ادامه برخی از مهم ترین ایده ها و رویکردها را ارائه می کنیم و سعی می کنیم به روابط متقابل آنها اشاره کنیم. به طور خاص در مورد موضوعات زیر بحث خواهیم کرد:
  - تجمعی
  - نمونه گیری
  - کاهش ابعاد
  - انتخاب زیر مجموعه ای از ویژگی،
  - ایجاد یک ویژگی جدید
  - گسته سازی و یا باینری سازی
  - تبدیل متغیربر روی متغیرها



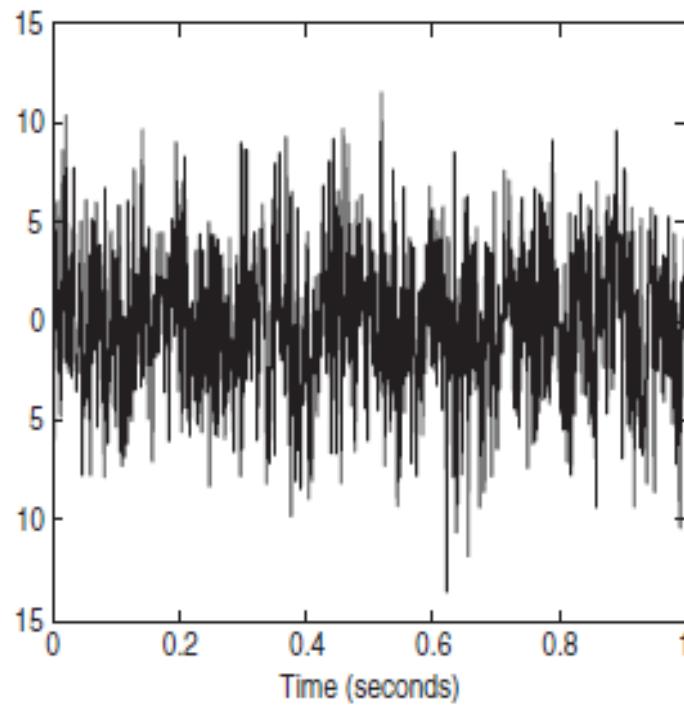
**Figure 2.9.** Example of the loss of structure with sampling.



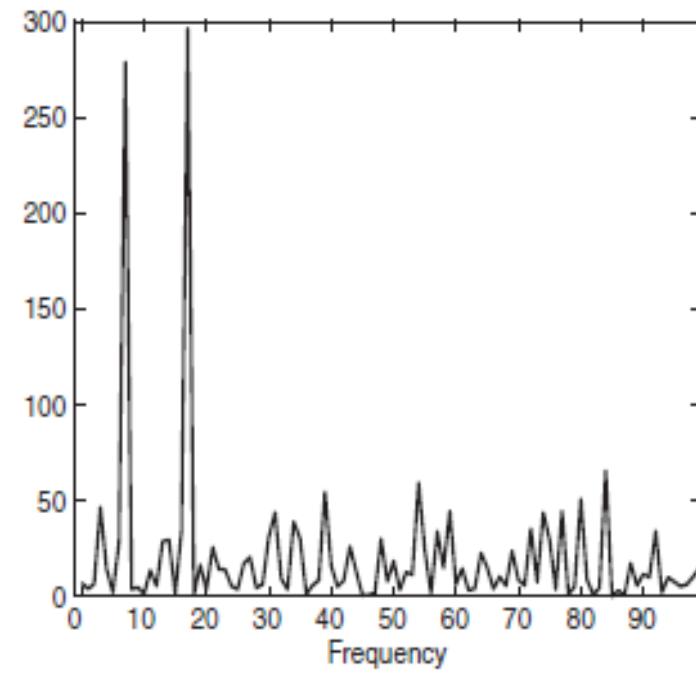
**Figure 2.11.** Flowchart of a feature subset selection process.



(a) Two time series.



(b) Noisy time series.



(c) Power spectrum.

**Figure 2.12.** Application of the Fourier transform to identify the underlying frequencies in time series data.

## سنجهش شباهت و عدم تشابه

- شباهت و عدم تشابه مهم هستند زیرا توسط تعدادی از تکنیک های داده کاوی مانند خوش بندی، طبقه بندی نزدیکترین همسایه و تشخیص ناهنجاری استفاده می شوند.
- در بسیاری از موارد، پس از محاسبه این شباهت ها یا تفاوت ها، به مجموعه داده های اولیه نیازی نیست.
- چنین رویکردهایی را می توان به عنوان تبدیل داده ها به فضای مشابه (عدم شباهت) و سپس انجام تحلیل مشاهده کرد.
- ابتدا با تعاریف سطح بالا از شباهت و عدم تشابه، و چگونگی ارتباط آنها، شروع می کنیم.
- برای راحتی، اصطلاح **نزدیکی** (proximity) برای اشاره به شباهت یا عدم تشابه استفاده می شود.
- از آنجایی که نزدیکی بین دو شی تابعی از نزدیکی بین ویژگی های متناظر دو شی است، ابتدا نحوه اندازه گیری مجاورت بین اشیایی که فقط یک ویژگی دارند را شرح می دهیم.

- این شامل معیارهایی مانند:
- اندازه‌های شباهت ژاکارد و کسینوس است که برای داده‌های پراکنده، مانند اسناد،
- همبستگی و فاصله اقلیدسی مفید هستند، که برای داده‌های غیر پراکنده (چگال) مانند سری‌های زمانی یا چندگانه مفید هستند.
- نقاط بعدی ما همچنین اطلاعات متقابل را در نظر می‌گیریم، که می‌تواند برای بسیاری از انواع داده‌ها اعمال شود و برای تشخیص روابط غیرخطی خوب است.
- در این بحث، ما خود را به اشیایی با انواع ویژگی‌های نسبتاً همگن، معمولاً با اینتری یا پیوسته محدود می‌کنیم.

- **شباخت** بین دو شی یک معیار عددی برای درجه یکسانی دو جسم است.
- در نتیجه، شباخت ها برای جفت اشیایی که بیشتر شبیه هم هستند، بیشتر است.
- شباخت ها معمولاً<sup>ا</sup> غیر منفی هستند و اغلب بین ۰ (بدون شباخت) و ۱ (شباخت کامل) هستند.

- **عدم تشابه** بین دو شی یک معیار عددی برای درجه متفاوت بودن دو جسم است.
- عدم تشابه برای جفت اشیاء مشابه کمتر است.
- اغلب، اصطلاح فاصله به عنوان مترادف برای عدم تشابه استفاده می شود،
- گاهی اوقات تفاوت ها در بازه  $[0, 1]$  قرار می گیرند، اما معمولاً<sup>ا</sup> بین ۰ تا  $\infty$  نیز متغیر است.

## شباخت و عدم تشابه بین صفات ساده

- نزدیکی اشیاء با چند ویژگی معمولاً با ترکیب نزدیکی ویژگی‌های منفرد تعریف می‌شود
- بنابراین، ابتدا نزدیکی بین اشیاء دارای یک ویژگی واحد را مورد بحث قرار می‌دهیم.
- اشیایی را در نظر بگیرید که با یک ویژگی اسمی توصیف شده‌اند. شباخت دو شیء به چه معناست؟
- از آنجایی که صفات اسمی فقط اطلاعاتی را در مورد متمایز بودن اشیاء می‌رسانند، تنها چیزی که می‌توانیم بگوییم این است که دو شی یا ارزش یکسانی دارند یا ندارند.
- از این رو، در این مورد شباخت به طور سنتی به عنوان ۱ در صورتی که مقادیر ویژگی مطابقت داشته باشند، و در غیر این صورت ۰ تعریف می‌شود.
- یک عدم تشابه به صورت معکوس تعریف می‌شود: ۰ اگر مقادیر مشخصه مطابقت داشته باشند و ۱ اگر با هم مطابقت ندارند.

- برای اشیاء با یک ویژگی ترتیبی، وضعیت پیچیده تر است
- زیرا اطلاعات مربوط به ترتیب باید در نظر گرفته شود.
- ویژگی‌ای را در نظر بگیرید که کیفیت یک محصول را اندازه‌گیری می‌کند،
- مثلاً یک آب نبات، در مقیاس {ضعیف، منصفانه، قابل قبول، خوب، فوق العاده}.
- منطقی به نظر می‌رسد که محصول P1، که دارای رتبه فوق العاده است، به محصول P2 که دارای رتبه خوب است، نزدیک تر باشد تا محصول P3 که دارای رتبه قابل قبول است.
- برای کمی کردن این مشاهدات، مقادیر صفت ترتیبی اغلب به اعداد صحیح متوالی نگاشت می‌شوند که از ۰ یا ۱ شروع می‌شوند،
- به عنوان مثال، {ضعیف = ۰، منصفانه = ۱، قابل قبول = ۲، خوب = ۳، فوق العاده = ۴}.
- سپس  $d(P1, P2) = 3-2 = 1$
- $d(P1, P2) = (3-2)/4 = 0.25$ . و اقرار گیرد.

**Table 2.7. Similarity and dissimilarity for simple attributes**

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$	$s = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$
Ordinal	$d =  x - y /(n - 1)$ (values mapped to integers 0 to $n - 1$ , where $n$ is the number of values)	$s = 1 - d$
Interval or Ratio	$d =  x - y $	$s = -d, s = \frac{1}{1+d}, s = e^{-d},$ $s = 1 - \frac{d - \min_d}{\max_d - \min_d}$

# Dissimilarities between Data Objects

• فوائل

**Euclidean distance**     $d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2},$

**Minkowski distance**     $d(\mathbf{x}, \mathbf{y}) = \left( \sum_{k=1}^n |x_k - y_k|^r \right)^{1/r},$

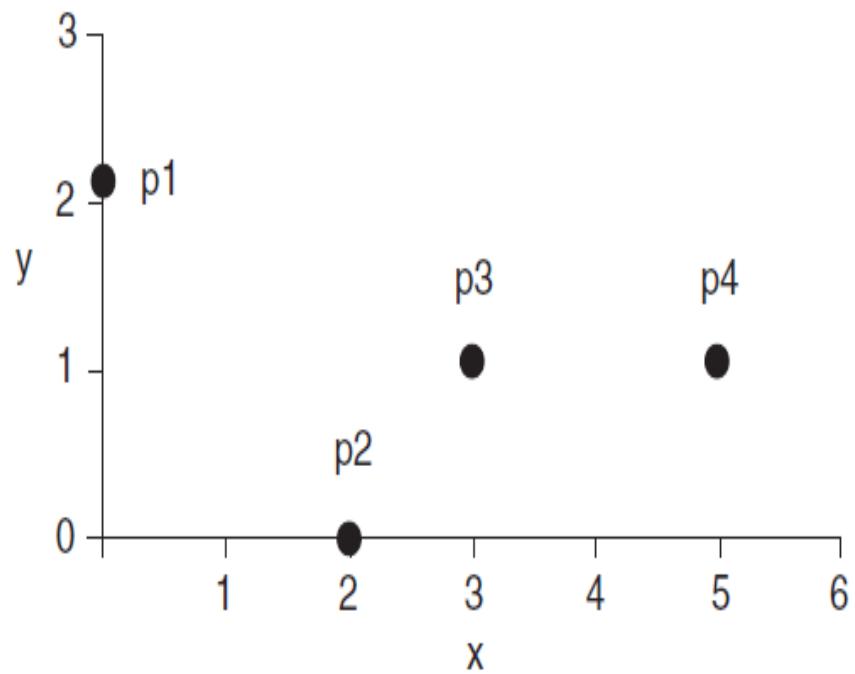


Figure 2.15. Four two-dimensional points.

Table 2.8.  $x$  and  $y$  coordinates of four points.

point	$x$ coordinate	$y$ coordinate
p1	0	2
p2	2	0
p3	3	1
p4	5	1

**Table 2.9.** Euclidean distance matrix for Table 2.8. **Table 2.10.**  $L_1$  distance matrix for Table 2.8.

	p1	p2	p3	p4
p1	0.0	2.8	3.2	5.1
p2	2.8	0.0	1.4	3.2
p3	3.2	1.4	0.0	2.0
p4	5.1	3.2	2.0	0.0

$L_1$	p1	p2	p3	p4
p1	0.0	4.0	4.0	6.0
p2	4.0	0.0	2.0	4.0
p3	4.0	2.0	0.0	2.0
p4	6.0	4.0	2.0	0.0

**Table 2.11.**  $L_\infty$  distance matrix for Table 2.8.

$L_\infty$	p1	p2	p3	p4
p1	0.0	2.0	3.0	5.0
p2	2.0	0.0	1.0	3.0
p3	3.0	1.0	0.0	2.0
p4	5.0	3.0	2.0	0.0

# Similarity Measures for Binary Data

$f_{00}$  = the number of attributes where  $x$  is 0 and  $y$  is 0

$f_{01}$  = the number of attributes where  $x$  is 0 and  $y$  is 1

$f_{10}$  = the number of attributes where  $x$  is 1 and  $y$  is 0

$f_{11}$  = the number of attributes where  $x$  is 1 and  $y$  is 1

## Simple Matching Coefficient (*SMC*)

$$SMC = \frac{\text{number of matching attribute values}}{\text{number of attributes}} = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}}.$$

## Jaccard coefficient

$$J = \frac{\text{number of matching presences}}{\text{number of attributes not involved in 00 matches}} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}.$$

**Example**       $x = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$   
                       $y = (0, 0, 0, 0, 0, 0, 1, 0, 0, 1)$

$f_{01} = 2$     the number of attributes where  $x$  was 0 and  $y$  was 1

$f_{10} = 1$     the number of attributes where  $x$  was 1 and  $y$  was 0

$f_{00} = 7$     the number of attributes where  $x$  was 0 and  $y$  was 0

$f_{11} = 0$     the number of attributes where  $x$  was 1 and  $y$  was 1

$$SMC = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}} = \frac{0 + 7}{2 + 1 + 0 + 7} = 0.7$$

$$J = \frac{f_{11}}{f_{01} + f_{10} + f_{11}} = \frac{0}{2 + 1 + 0} = 0$$

# Cosine Similarity

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{\mathbf{x}' \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|},$$

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{k=1}^n x_k y_k = \mathbf{x}' \mathbf{y}, \quad \|\mathbf{x}\| = \sqrt{\sum_{k=1}^n x_k^2} = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle} = \sqrt{\mathbf{x}' \mathbf{x}}.$$

**Example**  $\mathbf{x} = (3, 2, 0, 5, 0, 0, 0, 2, 0, 0)$   
 $\mathbf{y} = (1, 0, 0, 0, 0, 0, 0, 1, 0, 2)$

$$\langle \mathbf{x}, \mathbf{y} \rangle = 3 \times 1 + 2 \times 0 + 0 \times 0 + 5 \times 0 + 0 \times 0 + 0 \times 0 + 0 \times 0 + 2 \times 1 + 0 \times 0 + 0 \times 2 = 5$$

$$\|\mathbf{x}\| = \sqrt{3 \times 3 + 2 \times 2 + 0 \times 0 + 5 \times 5 + 0 \times 0 + 0 \times 0 + 0 \times 0 + 2 \times 2 + 0 \times 0 + 0 \times 0} = 6.48$$

$$\|\mathbf{y}\| = \sqrt{1 \times 1 + 0 \times 0 + 1 \times 1 + 0 \times 0 + 2 \times 2} = 2.45$$

$$\cos(\mathbf{x}, \mathbf{y}) = 0.31$$

## Pearson's correlation

$$\text{corr}(\mathbf{x}, \mathbf{y}) = \frac{\text{covariance}(\mathbf{x}, \mathbf{y})}{\text{standard\_deviation}(\mathbf{x}) \times \text{standard\_deviation}(\mathbf{y})} = \frac{s_{xy}}{s_x s_y},$$

$$\text{covariance}(\mathbf{x}, \mathbf{y}) = s_{xy} = \frac{1}{n-1} \sum_{k=1}^n (x_k - \bar{x})(y_k - \bar{y})$$

$$\text{standard\_deviation}(\mathbf{x}) = s_x = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (x_k - \bar{x})^2}$$

$$\text{standard\_deviation}(\mathbf{y}) = s_y = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (y_k - \bar{y})^2}$$

# روش های دیگر

- آنتروبی
- روش کرnel (هسته، **kernel**)

## چارچوب کلی برای یادگیری در علم داده

- استفاده از بخشی از داده ها، معروف به مجموعه آموزشی (training set)،
- استفاده از یک الگوریتم یادگیری (learning algorithm) برای یادگیری با استفاده از داده های آموزشی
- فرآیند استفاده از یک الگوریتم یادگیری برای ساخت یک مدل از داده های آموزشی را القاء (learning a model)، "یادگیری یک مدل (induction)" یا "ساخت یک مدل (building a model)"
- اعمال مدل برای تخصیص برچسبها و طبقات داده های تست (Test Set) برای ارزیابی عملکرد الگوریتم یادگیری.
- این مرحله را استنباط (deduction) نیز می گویند

Training Set

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Learning Algorithm

*Induction:*  
"Learn Model"

Model

*Deduction:*  
"Apply Model"

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
11	No	Married	55K	?
12	Yes	Divorced	80K	?
13	Yes	Single	110K	?
14	No	Single	95K	?
15	No	Married	67K	?

Test Set

# چگونگی ارزیابی یک مدل یا یک الگوریتم یادگیری

Table 3.4. Confusion matrix for a binary classification problem.

		Predicted Class	
		Class = 1	Class = 0
Actual Class	Class = 1	$f_{11}$	$f_{10}$
	Class = 0	$f_{01}$	$f_{00}$

- روش ماتریس سردرگمی (**confusion matrix**)

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

### 5.1.4 Example: Linear Regression

Our definition of a machine learning algorithm as an algorithm that is capable of improving a computer program's performance at some task via experience is somewhat abstract. To make this more concrete, we present an example of a simple machine learning algorithm: **linear regression**. We will return to this example repeatedly as we introduce more machine learning concepts that help to understand the algorithm's behavior.

As the name implies, linear regression solves a regression problem. In other words, the goal is to build a system that can take a vector  $\mathbf{x} \in \mathbb{R}^n$  as input and predict the value of a scalar  $y \in \mathbb{R}$  as its output. The output of linear regression is a linear function of the input. Let  $\hat{y}$  be the value that our model predicts  $y$  should take on. We define the output to be

$$\hat{y} = \mathbf{w}^\top \mathbf{x}, \tag{5.3}$$

where  $\mathbf{w} \in \mathbb{R}^n$  is a vector of **parameters**.

Parameters are values that control the behavior of the system. In this case,  $w_i$  is the coefficient that we multiply by feature  $x_i$  before summing up the contributions from all the features. We can think of  $\mathbf{w}$  as a set of **weights** that determine how each feature affects the prediction. If a feature  $x_i$  receives a positive weight  $w_i$ , then increasing the value of that feature increases the value of our prediction  $\hat{y}$ .

We thus have a definition of our task  $T$ : to predict  $y$  from  $\mathbf{x}$  by outputting  $\hat{y} = \mathbf{w}^\top \mathbf{x}$ . Next we need a definition of our performance measure,  $P$ .

Suppose that we have a design matrix of  $m$  example inputs that we will not use for training, only for evaluating how well the model performs. We also have a vector of regression targets providing the correct value of  $y$  for each of these examples. Because this dataset will only be used for evaluation, we call it the test set. We refer to the design matrix of inputs as  $\mathbf{X}^{(\text{test})}$  and the vector of regression targets as  $\mathbf{y}^{(\text{test})}$ .

One way of measuring the performance of the model is to compute the mean squared error of the model on the test set. If  $\hat{\mathbf{y}}^{(\text{test})}$  gives the predictions of the model on the test set, then the mean squared error is given by

$$\text{MSE}_{\text{test}} = \frac{1}{m} \sum_i (\hat{y}_i^{(\text{test})} - y_i^{(\text{test})})^2. \quad (5.4)$$

Intuitively, one can see that this error measure decreases to 0 when  $\hat{\mathbf{y}}^{(\text{test})} = \mathbf{y}^{(\text{test})}$ . We can also see that

$$\text{MSE}_{\text{test}} = \frac{1}{m} \|\hat{\mathbf{y}}^{(\text{test})} - \mathbf{y}^{(\text{test})}\|_2^2, \quad (5.5)$$

so the error increases whenever the Euclidean distance between the predictions and the targets increases.

To make a machine learning algorithm, we need to design an algorithm that will improve the weights  $w$  in a way that reduces  $\text{MSE}_{\text{test}}$  when the algorithm is allowed to gain experience by observing a training set  $(X^{(\text{train})}, y^{(\text{train})})$ . One intuitive way of doing this (which we justify later, in section 5.5.1) is just to minimize the mean squared error on the training set,  $\text{MSE}_{\text{train}}$ .

To minimize  $\text{MSE}_{\text{train}}$ , we can simply solve for where its gradient is 0:

$$\nabla_w \text{MSE}_{\text{train}} = 0 \quad (5.6)$$

$$\Rightarrow \nabla_w \frac{1}{m} \|\hat{y}^{(\text{train})} - y^{(\text{train})}\|_2^2 = 0 \quad (5.7)$$

$$\Rightarrow \frac{1}{m} \nabla_w \|X^{(\text{train})}w - y^{(\text{train})}\|_2^2 = 0 \quad (5.8)$$

$$\Rightarrow \nabla_w (X^{(\text{train})}w - y^{(\text{train})})^\top (X^{(\text{train})}w - y^{(\text{train})}) = 0 \quad (5.9)$$

$$\Rightarrow \nabla_w (w^\top X^{(\text{train})\top} X^{(\text{train})}w - 2w^\top X^{(\text{train})\top} y^{(\text{train})} + y^{(\text{train})\top} y^{(\text{train})}) = 0 \quad (5.10)$$

$$\Rightarrow 2X^{(\text{train})\top} X^{(\text{train})}w - 2X^{(\text{train})\top} y^{(\text{train})} = 0 \quad (5.11)$$

$$\Rightarrow w = (X^{(\text{train})\top} X^{(\text{train})})^{-1} X^{(\text{train})\top} y^{(\text{train})} \quad (5.12)$$

The system of equations whose solution is given by equation 5.12 is known as the **normal equations**. Evaluating equation 5.12 constitutes a simple learning algorithm. For an example of the linear regression learning algorithm in action, see figure 5.1.

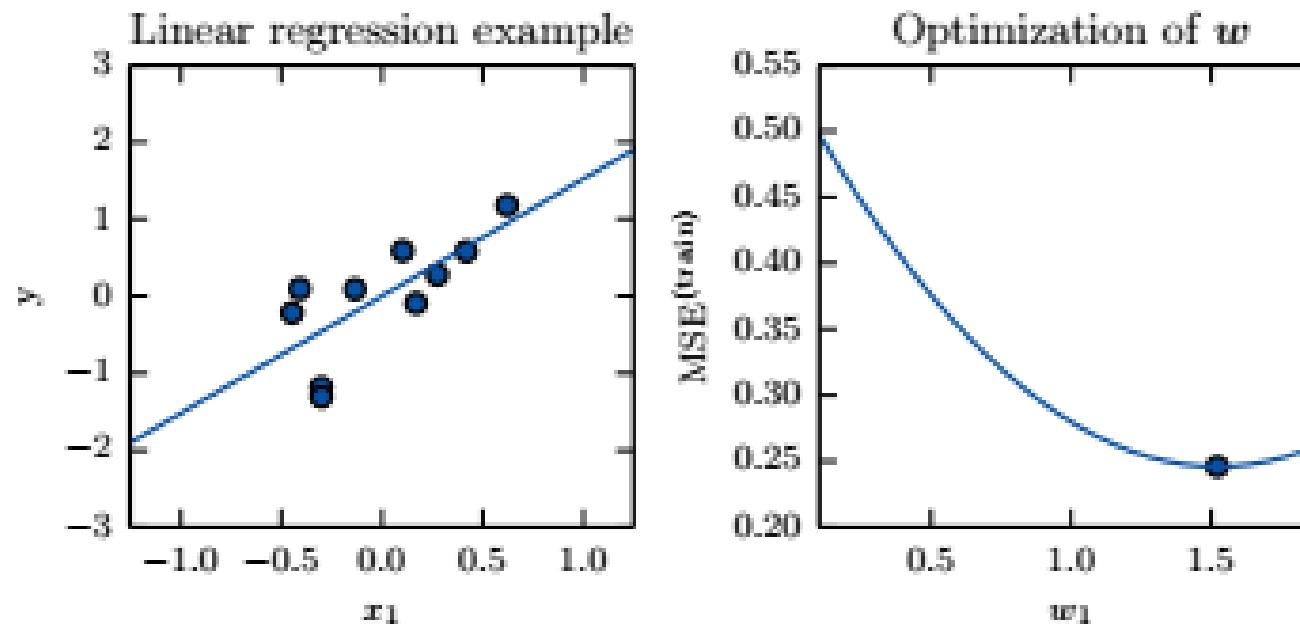


Figure 5.1: A linear regression problem, with a training set consisting of ten data points, each containing one feature. Because there is only one feature, the weight vector  $w$  contains only a single parameter to learn,  $w_1$ . (*Left*) Observe that linear regression learns to set  $w_1$  such that the line  $y = w_1 x$  comes as close as possible to passing through all the training points. (*Right*) The plotted point indicates the value of  $w_1$  found by the normal equations, which we can see minimizes the mean squared error on the training set.

It is worth noting that the term **linear regression** is often used to refer to a slightly more sophisticated model with one additional parameter—an intercept term  $b$ . In this model

$$\hat{y} = \mathbf{w}^\top \mathbf{x} + b, \quad (5.13)$$

so the mapping from parameters to predictions is still a linear function but the mapping from features to predictions is now an affine function. This extension to affine functions means that the plot of the model's predictions still looks like a line, but it need not pass through the origin. Instead of adding the bias parameter  $b$ , one can continue to use the model with only weights but augment  $\mathbf{x}$  with an extra entry that is always set to 1. The weight corresponding to the extra 1 entry plays the role of the bias parameter. We frequently use the term “linear” when referring to affine functions throughout this book.

The intercept term  $b$  is often called the **bias** parameter of the affine transformation. This terminology derives from the point of view that the output of the transformation is biased toward being  $b$  in the absence of any input. This term is different from the idea of a statistical bias, in which a statistical estimation algorithm's expected estimate of a quantity is not equal to the true quantity.

Linear regression is of course an extremely simple and limited learning algorithm, but it provides an example of how a learning algorithm can work. In subsequent sections we describe some of the basic principles underlying learning algorithm design and demonstrate how these principles can be used to build more complicated learning algorithms.

## 5.2 Capacity, Overfitting and Underfitting

The central challenge in machine learning is that our algorithm must perform well on *new, previously unseen inputs*—not just those on which our model was trained. The ability to perform well on previously unobserved inputs is called **generalization**.

Typically, when training a machine learning model, we have access to a training set; we can compute some error measure on the training set, called the **training error**; and we reduce this training error. So far, what we have described is simply an optimization problem. What separates machine learning from optimization is that we want the **generalization error**, also called the **test error**, to be low as well. The generalization error is defined as the expected value of the error on a new input. Here the expectation is taken across different possible inputs, drawn from the distribution of inputs we expect the system to encounter in practice.

We typically estimate the generalization error of a machine learning model by measuring its performance on a **test set** of examples that were collected separately from the training set.

In our linear regression example, we trained the model by minimizing the training error,

$$\frac{1}{m^{(\text{train})}} \|\mathbf{X}^{(\text{train})}\mathbf{w} - \mathbf{y}^{(\text{train})}\|_2^2, \quad (5.14)$$

but we actually care about the test error,  $\frac{1}{m^{(\text{test})}} \|\mathbf{X}^{(\text{test})}\mathbf{w} - \mathbf{y}^{(\text{test})}\|_2^2$ .

Of course, when we use a machine learning algorithm, we do not fix the parameters ahead of time, then sample both datasets. We sample the training set, then use it to choose the parameters to reduce training set error, then sample the test set. Under this process, the expected test error is greater than or equal to the expected value of training error. The factors determining how well a machine learning algorithm will perform are its ability to

1. Make the training error small.
2. Make the gap between training and test error small.

These two factors correspond to the two central challenges in machine learning: **underfitting** and **overfitting**. Underfitting occurs when the model is not able to obtain a sufficiently low error value on the training set. Overfitting occurs when the gap between the training error and test error is too large.

We can control whether a model is more likely to overfit or underfit by altering its **capacity**. Informally, a model's capacity is its ability to fit a wide variety of functions. Models with low capacity may struggle to fit the training set. Models with high capacity can overfit by memorizing properties of the training set that do not serve them well on the test set.

One way to control the capacity of a learning algorithm is by choosing its **hypothesis space**, the set of functions that the learning algorithm is allowed to select as being the solution. For example, the linear regression algorithm has the set of all linear functions of its input as its hypothesis space. We can generalize linear regression to include polynomials, rather than just linear functions, in its hypothesis space. Doing so increases the model's capacity.

A polynomial of degree 1 gives us the linear regression model with which we are already familiar, with the prediction

$$\hat{y} = b + wx. \quad (5.15)$$

By introducing  $x^2$  as another feature provided to the linear regression model, we can learn a model that is quadratic as a function of  $x$ :

$$\hat{y} = b + w_1x + w_2x^2. \quad (5.16)$$

Though this model implements a quadratic function of its *input*, the output is still a linear function of the *parameters*, so we can still use the normal equations to train the model in closed form. We can continue to add more powers of  $x$  as additional features, for example, to obtain a polynomial of degree 9:

$$\hat{y} = b + \sum_{i=1}^9 w_i x^i. \quad (5.17)$$

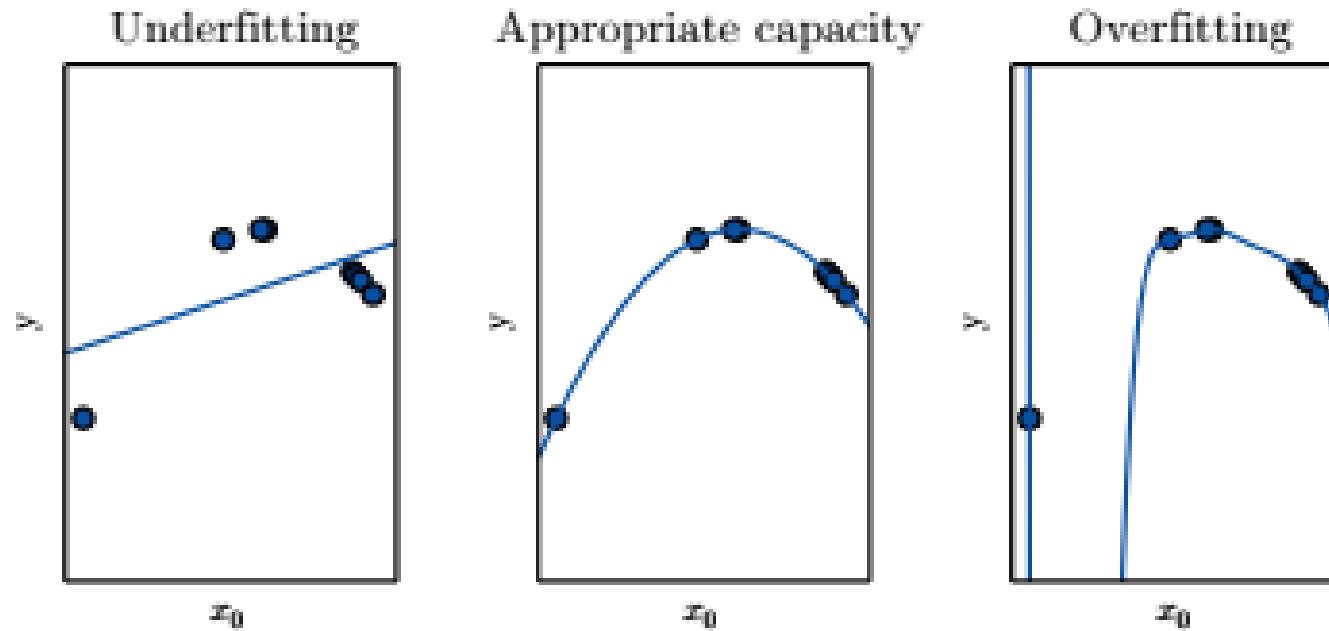


Figure 5.2: We fit three models to this example training set. The training data was generated synthetically, by randomly sampling  $x$  values and choosing  $y$  deterministically by evaluating a quadratic function. (*Left*)A linear function fit to the data suffers from underfitting—it cannot capture the curvature that is present in the data. (*Center*)A quadratic function fit to the data generalizes well to unseen points. It does not suffer from a significant amount of overfitting or underfitting. (*Right*)A polynomial of degree 9 fit to the data suffers from overfitting. Here we used the Moore-Penrose pseudoinverse to solve the underdetermined normal equations. The solution passes through all the training points exactly, but we have not been lucky enough for it to extract the correct structure. It now has a deep valley between two training points that does not appear in the true underlying function. It also increases sharply on the left side of the data, while the true function decreases in this area.

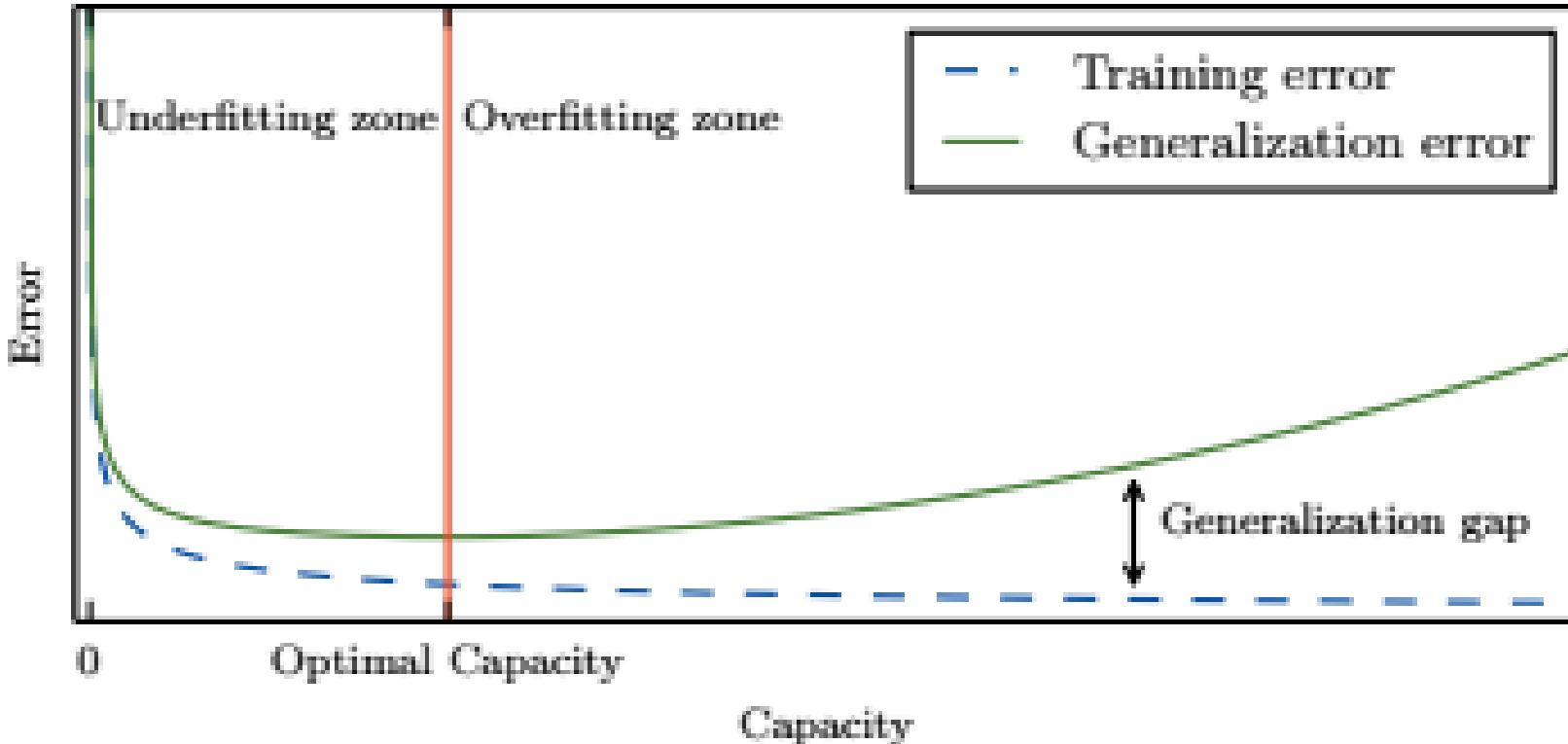


Figure 5.3: Typical relationship between capacity and error. Training and test error behave differently. At the left end of the graph, training error and generalization error are both high. This is the **underfitting regime**. As we increase capacity, training error decreases, but the gap between training and generalization error increases. Eventually, the size of this gap outweighs the decrease in training error, and we enter the **overfitting regime**, where capacity is too large, above the optimal capacity.

# Regularization

So far, the only method of modifying a learning algorithm that we have discussed concretely is to increase or decrease the model's representational capacity by adding or removing functions from the hypothesis space of solutions the learning algorithm is able to choose from. We gave the specific example of increasing or decreasing the degree of a polynomial for a regression problem. The view we have described so far is oversimplified.

The behavior of our algorithm is strongly affected not just by how large we make the set of functions allowed in its hypothesis space, but by the specific identity of those functions. The learning algorithm we have studied so far, linear regression, has a hypothesis space consisting of the set of linear functions of its input. These linear functions can be useful for problems where the relationship between inputs and outputs truly is close to linear. They are less useful for problems that behave in a very nonlinear fashion. For example, linear regression would not perform well if we tried to use it to predict  $\sin(x)$  from  $x$ . We can thus control the performance of our algorithms by choosing what kind of functions we allow them to draw solutions from, as well as by controlling the amount of these functions.

We can also give a learning algorithm a preference for one solution over another in its hypothesis space. This means that both functions are eligible, but one is preferred. The unpreferred solution will be chosen only if it fits the training data significantly better than the preferred solution.

For example, we can modify the training criterion for **linear regression** to include **weight decay**. To perform linear regression with weight decay, we minimize a sum comprising both the mean squared error on the training and a criterion  $J(\mathbf{w})$  that expresses a preference for the weights to have smaller squared  $L^2$  norm. Specifically,

$$J(\mathbf{w}) = \text{MSE}_{\text{train}} + \lambda \mathbf{w}^\top \mathbf{w}, \quad (5.18)$$

where  $\lambda$  is a value chosen ahead of time that controls the strength of our preference for smaller weights. When  $\lambda = 0$ , we impose no preference, and larger  $\lambda$  forces the weights to become smaller. Minimizing  $J(\mathbf{w})$  results in a choice of weights that make a tradeoff between fitting the training data and being small. This gives us solutions that have a smaller slope, or that put weight on fewer of the features. As an example of how we can control a model's tendency to overfit or underfit via weight decay, we can train a high-degree polynomial regression model with different values of  $\lambda$ . See figure 5.5 for the results.

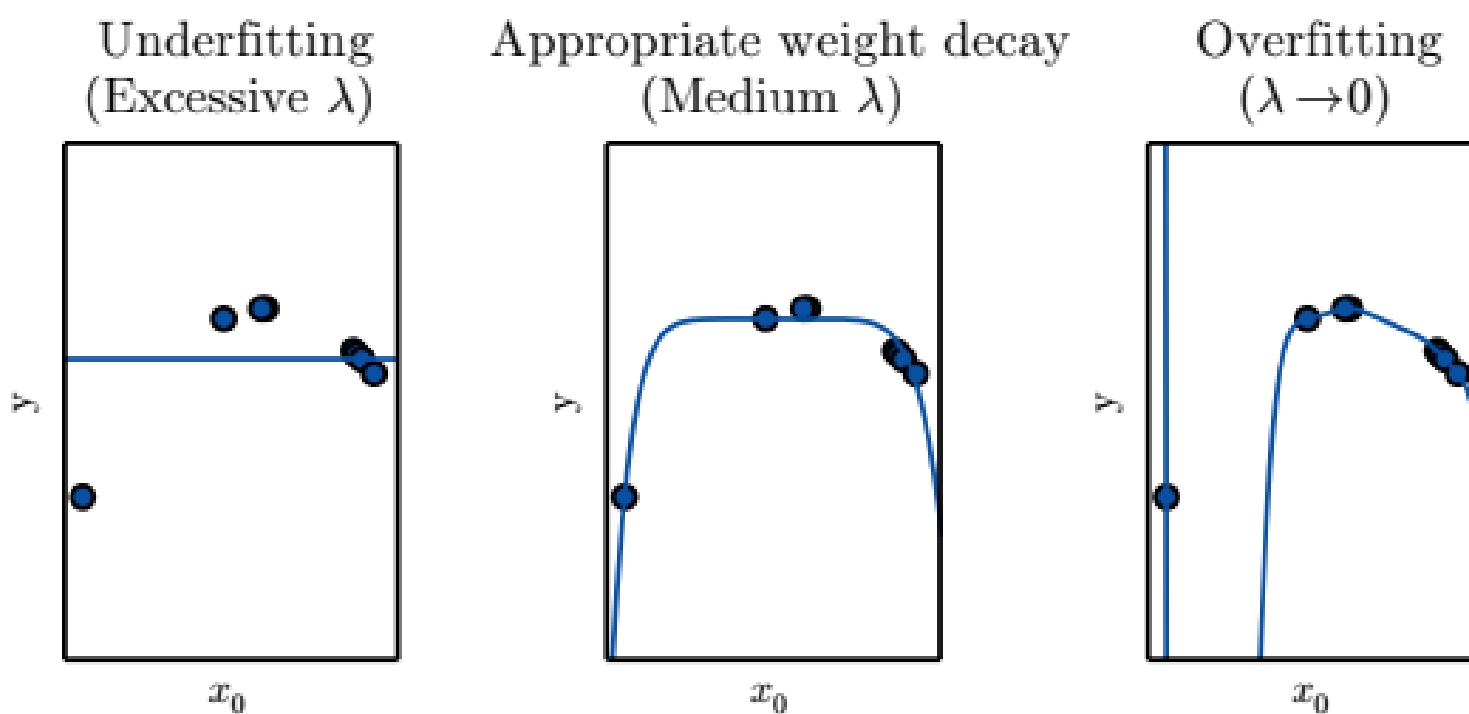


Figure 5.5: We fit a high-degree polynomial regression model to our example training set from figure 5.2. The true function is quadratic, but here we use only models with degree 9. We vary the amount of weight decay to prevent these high-degree models from overfitting. (*Left*) With very large  $\lambda$ , we can force the model to learn a function with no slope at all. This underfits because it can only represent a constant function. (*Center*) With a medium value of  $\lambda$ , the learning algorithm recovers a curve with the right general shape. Even though the model is capable of representing functions with much more complicated shapes, weight decay has encouraged it to use a simpler function described by smaller coefficients. (*Right*) With weight decay approaching zero (i.e., using the Moore-Penrose pseudoinverse to solve the underdetermined problem with minimal regularization), the degree-9 polynomial overfits significantly, as we saw in figure 5.2.

More generally, we can regularize a model that learns a function  $f(\mathbf{x}; \boldsymbol{\theta})$  by adding a penalty called a **regularizer** to the cost function. In the case of weight decay, the regularizer is  $\Omega(\mathbf{w}) = \mathbf{w}^\top \mathbf{w}$ . In chapter 7, we will see that many other regularizers are possible.

In our weight decay example, we expressed our preference for linear functions defined with smaller weights explicitly, via an extra term in the criterion we minimize. There are many other ways of expressing preferences for different solutions, both implicitly and explicitly. Together, these different approaches are known as **regularization**. *Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error.* Regularization is one of the central concerns of the field of machine learning, rivaled in its importance only by optimization.

# Hyperparameters

Most machine learning algorithms have hyperparameters, settings that we can use to control the algorithm's behavior. The values of hyperparameters are not adapted by the learning algorithm itself.

The polynomial regression example in figure 5.2 has a single hyperparameter: the degree of the polynomial, which acts as a **capacity** hyperparameter. The  $\lambda$  value used to control the strength of weight decay is another example of a hyperparameter.

# Validation Sets

## Algorithm 5.1 The $k$ -fold cross-validation algorithm

---

**Define** `KFoldXV( $\mathbb{D}, A, L, k$ )`:

**Require:**  $\mathbb{D}$ , the given dataset, with elements  $z^{(i)}$

**Require:**  $A$ , the learning algorithm, seen as a function that takes a dataset as input and outputs a learned function

**Require:**  $L$ , the loss function, seen as a function from a learned function  $f$  and an example  $z^{(i)} \in \mathbb{D}$  to a scalar  $\in \mathbb{R}$

**Require:**  $k$ , the number of folds

Split  $\mathbb{D}$  into  $k$  mutually exclusive subsets  $\mathbb{D}_i$ , whose union is  $\mathbb{D}$

**for**  $i$  from 1 to  $k$  **do**

$f_i = A(\mathbb{D} \setminus \mathbb{D}_i)$

**for**  $z^{(j)}$  in  $\mathbb{D}_i$  **do**

$e_j = L(f_i, z^{(j)})$

**end for**

**end for**

**Return**  $e$

---

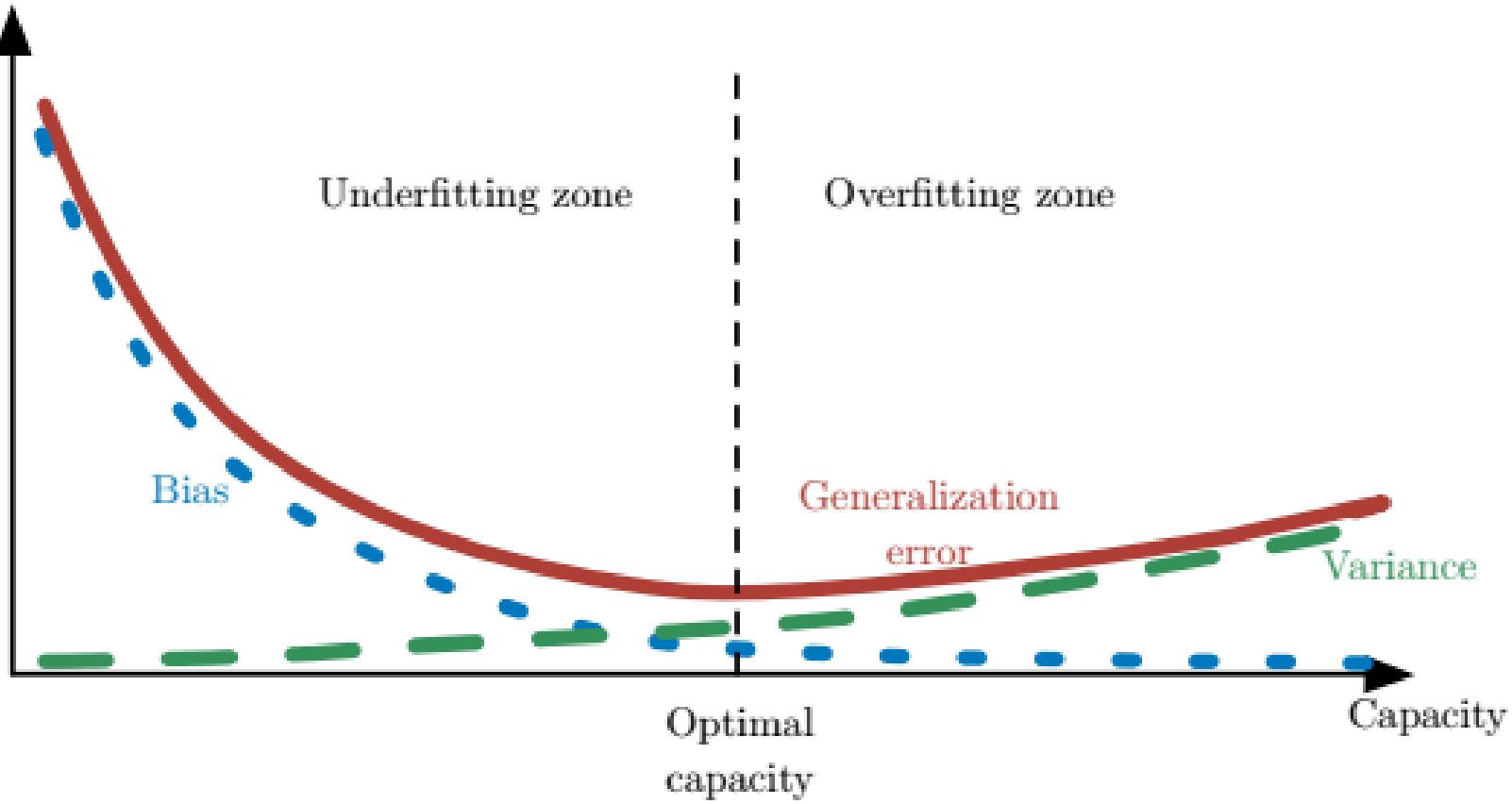


Figure 5.6: As capacity increases ( $x$ -axis), bias (dotted) tends to decrease and variance (dashed) tends to increase, yielding another U-shaped curve for generalization error (bold curve). If we vary capacity along one axis, there is an optimal capacity, with underfitting when the capacity is below this optimum and overfitting when it is above. This relationship is similar to the relationship between capacity, underfitting, and overfitting.

## Maximum Likelihood Estimation

$$\theta_{\text{ML}} = \arg \max_{\theta} p_{\text{model}}(\mathbf{X}; \theta), \quad \theta_{\text{ML}} = \arg \max_{\theta} \sum_{i=1}^m \log p_{\text{model}}(\mathbf{x}^{(i)}; \theta).$$

Because the arg max does not change when we rescale the cost function, we can divide by  $m$  to obtain a version of the criterion that is expressed as an expectation with respect to the empirical distribution  $\hat{p}_{\text{data}}$  defined by the training data:

$$\theta_{\text{ML}} = \arg \max_{\theta} \mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}} \log p_{\text{model}}(\mathbf{x}; \theta). \quad (5.59)$$

One way to interpret maximum likelihood estimation is to view it as minimizing the dissimilarity between the empirical distribution  $\hat{p}_{\text{data}}$ , defined by the training set and the model distribution, with the degree of dissimilarity between the two measured by the KL divergence. The KL divergence is given by

$$D_{\text{KL}}(\hat{p}_{\text{data}} \| p_{\text{model}}) = \mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}} [\log \hat{p}_{\text{data}}(\mathbf{x}) - \log p_{\text{model}}(\mathbf{x})]. \quad (5.60)$$

The term on the left is a function only of the data-generating process, not the model. This means when we train the model to minimize the KL divergence, we need only minimize

$$-\mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}} [\log p_{\text{model}}(\mathbf{x})], \quad (5.61)$$

which is of course the same as the maximization in equation 5.59.

# Conditional Log-Likelihood

The maximum likelihood estimator can readily be generalized to estimate a conditional probability  $P(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta})$  in order to predict  $\mathbf{y}$  given  $\mathbf{x}$ . This is actually the most common situation because it forms the basis for most supervised learning. If  $\mathbf{X}$  represents all our inputs and  $\mathbf{Y}$  all our observed targets, then the conditional maximum likelihood estimator is

$$\boldsymbol{\theta}_{\text{ML}} = \arg \max_{\boldsymbol{\theta}} P(\mathbf{Y} \mid \mathbf{X}; \boldsymbol{\theta}). \quad (5.62)$$

# Bayesian Statistics

**Bayesian estimation** offers two important differences. First, unlike the maximum likelihood approach that makes predictions using a point estimate of  $\boldsymbol{\theta}$ , the Bayesian approach is to make predictions using a full distribution over  $\boldsymbol{\theta}$ . For example, after observing  $m$  examples, the **predicted distribution** over the next data sample,  $x^{(m+1)}$ , is given by

$$p(x^{(m+1)} | x^{(1)}, \dots, x^{(m)}) = \int p(x^{(m+1)} | \boldsymbol{\theta}) p(\boldsymbol{\theta} | x^{(1)}, \dots, x^{(m)}) d\boldsymbol{\theta}. \quad (5.68)$$

The second important difference between the Bayesian approach to estimation and the maximum likelihood approach is due to the contribution of the Bayesian prior distribution. The prior has an influence by **shifting probability mass density towards regions of the parameter space that are preferred a priori**. In practice, the prior often expresses a preference for models that are simpler or more smooth. Critics of the Bayesian approach identify the prior as a source of subjective human judgment affecting the predictions.

## Maximum A Posteriori (MAP) Estimation

The MAP estimate chooses the point of maximal posterior probability (or maximal probability density in the more common case of continuous  $\theta$ ):

$$\theta_{\text{MAP}} = \arg \max_{\theta} p(\theta \mid x) = \arg \max_{\theta} \log p(x \mid \theta) + \log p(\theta). \quad (5.79)$$

We recognize, on the righthand side,  $\log p(x \mid \theta)$ , that is, the standard log-likelihood term, and  $\log p(\theta)$ , corresponding to the prior distribution.

As with full Bayesian inference, MAP Bayesian inference has the advantage of leveraging information that is brought by the prior and cannot be found in the training data. This additional information helps to reduce the variance in the MAP point estimate (in comparison to the ML estimate). However, it does so at the price of increased bias.

Many regularized estimation strategies, such as maximum likelihood learning regularized with weight decay, can be interpreted as making the MAP approximation to Bayesian inference. This view applies when the regularization consists of adding an extra term to the objective function that corresponds to  $\log p(\theta)$ . Not all regularization penalties correspond to MAP Bayesian inference. For example, some regularizer terms may not be the logarithm of a probability distribution. Other regularization terms depend on the data, which of course a prior probability distribution is not allowed to do.

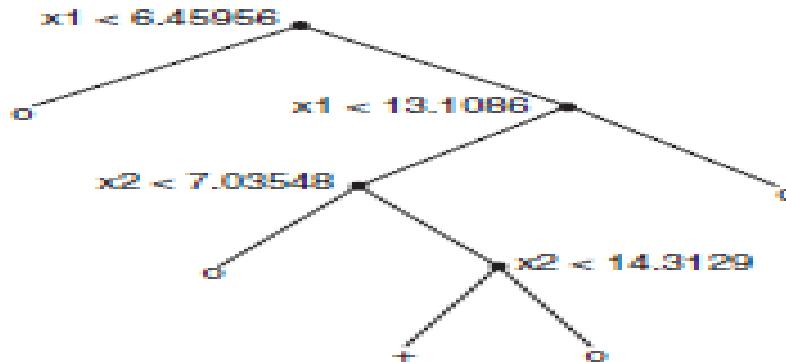
MAP Bayesian inference provides a straightforward way to design complicated yet interpretable regularization terms. For example, a more complicated penalty term can be derived by using a mixture of Gaussians, rather than a single Gaussian distribution, as the prior (Nowlan and Hinton, 1992).

# معرفی مفاهیم اساسی مدل بندی

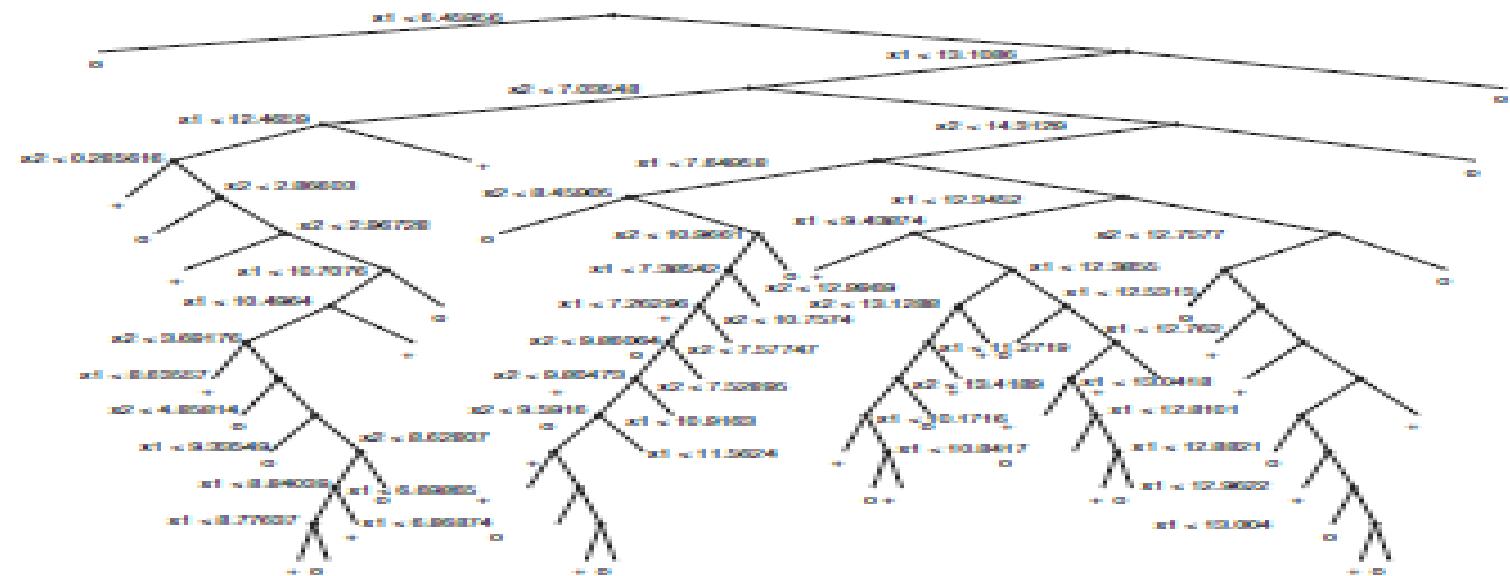
- تعریف و مفهوم مدل بندی
- برآزش بیش از حد مدل model overfitting
- انتخاب مدل model selection
- ارزیابی مدل model evaluation

# بیش برازش Model Overfitting

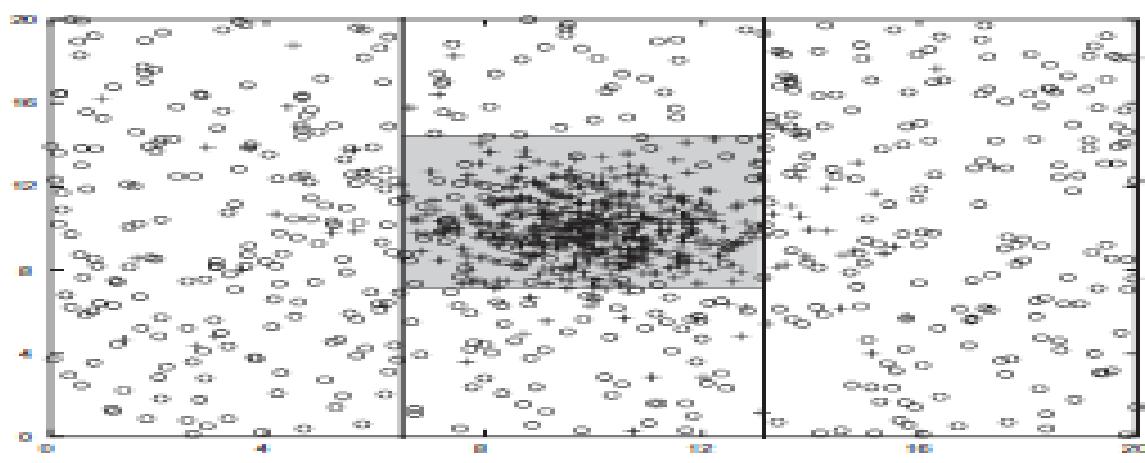
- ما همواره به دنبال یادگیری مدل‌های طبقه‌بندی بر اساس کمترین خطا را در مجموعه آموزشی هستیم
- با این حال، همانطور که در مثال زیر نشان خواهیم داد، حتی اگر یک مدل به خوبی با داده‌های آموزشی مطابقت داشته باشد، باز هم می‌تواند عملکرد تعمیمی ضعیفی را نشان دهد،
- پدیده‌ای که به عنوان **بیش برازش** مدل شناخته می‌شود.



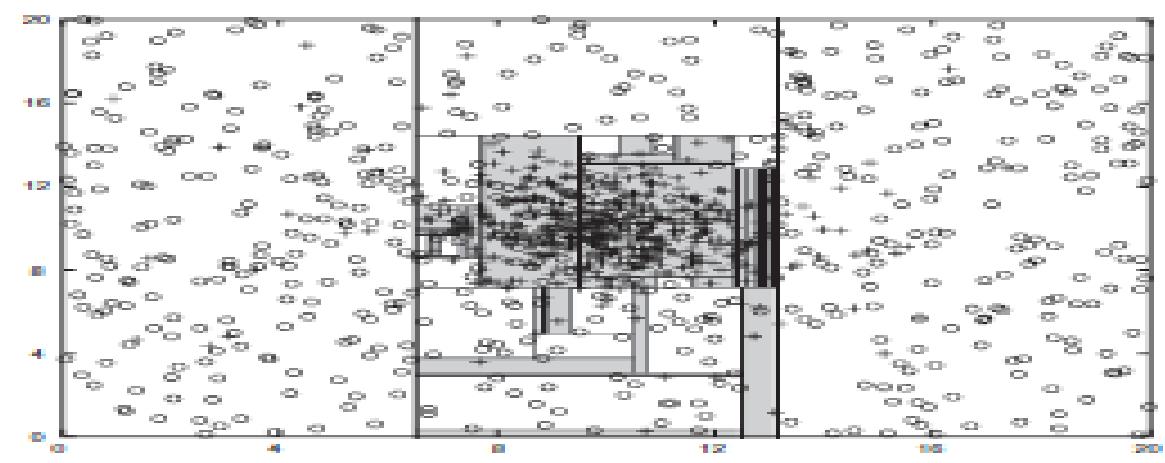
(a) Decision tree with 5 leaf nodes.



(b) Decision tree with 50 leaf nodes.



(c) Decision boundary for tree with 5 leaf nodes.



(d) Decision boundary for tree with 50 leaf nodes.

**Figure 3.24.** Decision trees with different model complexities.

# روشهای اجتناب از بیش برازش

- محدود کردن داده های آموزشی
- تصادفی کردن روند انتخاب داده های آموزشی (تحلیل حساسیت مدل نسبت به داده های آموزشی)
- اجتناب از پیچیدگی

# Model Selection

- بر اساس معیارهای نیکویی برازش
- درجه سادگی و یا پیچیدگی
- مناسب ترین مدل انتخاب می شوند

# Model Evaluation

- انتخاب نمونه های آموزشی و تست ثابت (Cross-Validation)



$$err_{test} = \frac{\sum_{i=1}^k err_{sum}(i)}{N}.$$

Figure 3.33. Example demonstrating the technique of 3-fold cross-validation.

## 5.7 Supervised Learning Algorithms

### 5.7.1 Probabilistic Supervised Learning

Most supervised learning algorithms in this book are based on estimating a probability distribution  $p(y \mid \mathbf{x})$ . We can do this simply by using maximum likelihood estimation to find the best parameter vector  $\boldsymbol{\theta}$  for a parametric family of distributions  $p(y \mid \mathbf{x}; \boldsymbol{\theta})$ .

We have already seen that linear regression corresponds to the family

$$p(y \mid \mathbf{x}; \boldsymbol{\theta}) = \mathcal{N}(y; \boldsymbol{\theta}^\top \mathbf{x}, \mathbf{I}). \quad (5.80)$$

We can generalize linear regression to the classification scenario by defining a different family of probability distributions. If we have two classes, class 0 and class 1, then we need only specify the probability of one of these classes. The probability of class 1 determines the probability of class 0, because these two values must add up to 1.

## 5.7.2 Support Vector Machines

One of the most influential approaches to supervised learning is the support vector machine (Boser et al., 1992; Cortes and Vapnik, 1995). This model is similar to logistic regression in that it is driven by a linear function  $\mathbf{w}^\top \mathbf{x} + b$ . Unlike logistic regression, the support vector machine does not provide probabilities, but only outputs a class identity. The SVM predicts that the positive class is present when  $\mathbf{w}^\top \mathbf{x} + b$  is positive. Likewise, it predicts that the negative class is present when  $\mathbf{w}^\top \mathbf{x} + b$  is negative.

One key innovation associated with support vector machines is the **kernel trick**. The kernel trick consists of observing that many machine learning algorithms can be written exclusively in terms of dot products between examples. For example, it can be shown that the linear function used by the support vector machine can be re-written as

$$\mathbf{w}^\top \mathbf{x} + b = b + \sum_{i=1}^m \alpha_i \mathbf{x}^\top \mathbf{x}^{(i)}, \quad (5.82)$$

where  $\mathbf{x}^{(i)}$  is a training example, and  $\boldsymbol{\alpha}$  is a vector of coefficients. Rewriting the learning algorithm this way enables us to replace  $\mathbf{x}$  with the output of a given feature function  $\phi(\mathbf{x})$  and the dot product with a function  $k(\mathbf{x}, \mathbf{x}^{(i)}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}^{(i)})$  called a **kernel**. The  $\cdot$  operator represents an inner product analogous to  $\phi(\mathbf{x})^\top \phi(\mathbf{x}^{(i)})$ . For some feature spaces, we may not use literally the vector inner product. In some infinite dimensional spaces, we need to use other kinds of inner products, for example, inner products based on integration rather than summation. A complete development of these kinds of inner products is beyond the scope of this book.

After replacing dot products with kernel evaluations, we can make predictions using the function

$$f(\mathbf{x}) = b + \sum_i \alpha_i k(\mathbf{x}, \mathbf{x}^{(i)}). \quad (5.83)$$

This function is nonlinear with respect to  $\mathbf{x}$ , but the relationship between  $\phi(\mathbf{x})$  and  $f(\mathbf{x})$  is linear. Also, the relationship between  $\boldsymbol{\alpha}$  and  $f(\mathbf{x})$  is linear. The kernel-based function is exactly equivalent to preprocessing the data by applying  $\phi(\mathbf{x})$  to all inputs, then learning a linear model in the new transformed space.

The kernel trick is powerful for two reasons. First, it enables us to learn models that are nonlinear as a function of  $\mathbf{x}$  using convex optimization techniques that are guaranteed to converge efficiently. This is possible because we consider  $\phi$  fixed and optimize only  $\boldsymbol{\alpha}$ , that is, the optimization algorithm can view the decision function as being linear in a different space. Second, the kernel function  $k$  often admits an implementation that is significantly more computationally efficient than naively constructing two  $\phi(\mathbf{x})$  vectors and explicitly taking their dot product.

## 5.8 Unsupervised Learning Algorithms

### 5.8.1 Principal Components Analysis

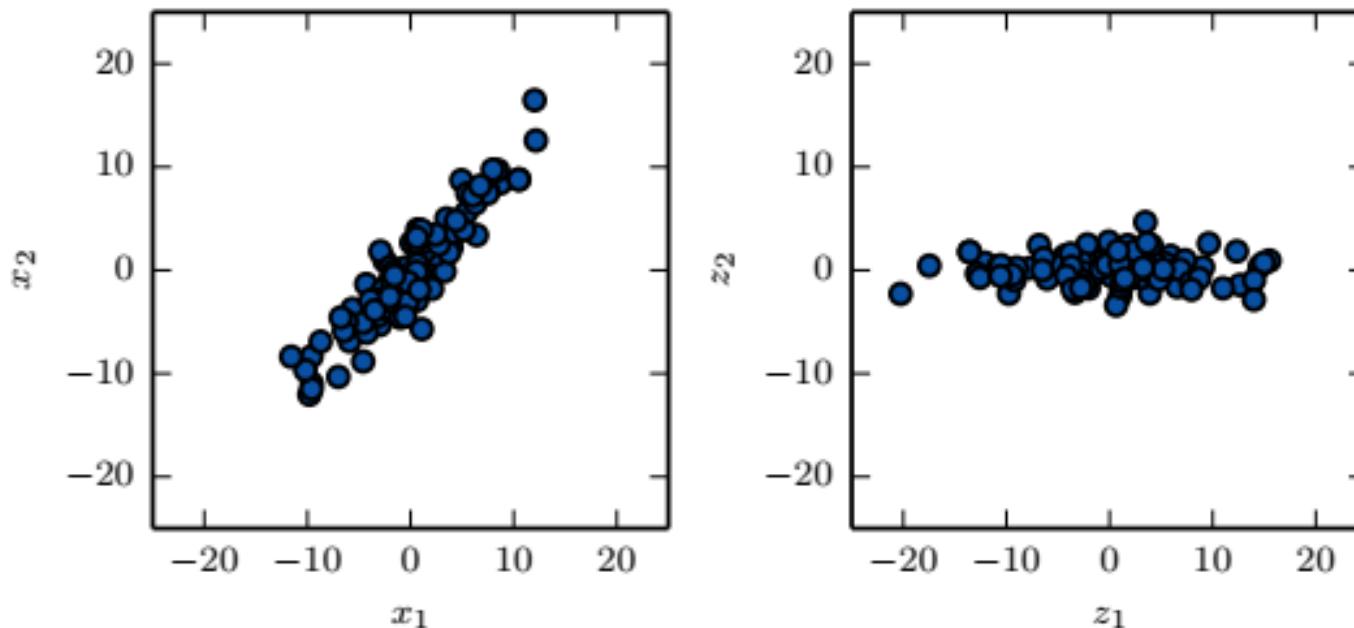


Figure 5.8: PCA learns a linear projection that aligns the direction of greatest variance with the axes of the new space. (*Left*) The original data consist of samples of  $\mathbf{x}$ . In this space, the variance might occur along directions that are not axis aligned. (*Right*) The transformed data  $\mathbf{z} = \mathbf{x}^\top \mathbf{W}$  now varies most along the axis  $z_1$ . The direction of second-most variance is now along  $z_2$ .

Let us consider the  $m \times n$  design matrix  $\mathbf{X}$ . We will assume that the data has a mean of zero,  $\mathbb{E}[\mathbf{x}] = \mathbf{0}$ . If this is not the case, the data can easily be centered by subtracting the mean from all examples in a preprocessing step.

The unbiased sample covariance matrix associated with  $\mathbf{X}$  is given by

$$\text{Var}[\mathbf{x}] = \frac{1}{m-1} \mathbf{X}^\top \mathbf{X}. \quad (5.85)$$

PCA finds a representation (through linear transformation)  $\mathbf{z} = \mathbf{W}^\top \mathbf{x}$ , where  $\text{Var}[\mathbf{z}]$  is diagonal.

In section 2.12, we saw that the principal components of a design matrix  $\mathbf{X}$  are given by the eigenvectors of  $\mathbf{X}^\top \mathbf{X}$ . From this view,

$$\mathbf{X}^\top \mathbf{X} = \mathbf{W} \Lambda \mathbf{W}^\top. \quad (5.86)$$

In this section, we exploit an alternative derivation of the principal components. The principal components may also be obtained via singular value decomposition (SVD). Specifically, they are the right singular vectors of  $\mathbf{X}$ . To see this, let  $\mathbf{W}$  be the right singular vectors in the decomposition  $\mathbf{X} = \mathbf{U}\Sigma\mathbf{W}^\top$ . We then recover the original eigenvector equation with  $\mathbf{W}$  as the eigenvector basis:

$$\mathbf{X}^\top \mathbf{X} = (\mathbf{U}\Sigma\mathbf{W}^\top)^\top \mathbf{U}\Sigma\mathbf{W}^\top = \mathbf{W}\Sigma^2\mathbf{W}^\top. \quad (5.87)$$

The SVD is helpful to show that PCA results in a diagonal  $\text{Var}[z]$ . Using the SVD of  $\mathbf{X}$ , we can express the variance of  $\mathbf{X}$  as:

$$\text{Var}[\mathbf{x}] = \frac{1}{m-1} \mathbf{X}^\top \mathbf{X} \quad (5.88)$$

$$= \frac{1}{m-1} (\mathbf{U}\Sigma\mathbf{W}^\top)^\top \mathbf{U}\Sigma\mathbf{W}^\top \quad (5.89)$$

$$= \frac{1}{m-1} \mathbf{W}\Sigma^\top \mathbf{U}^\top \mathbf{U}\Sigma\mathbf{W}^\top \quad (5.90)$$

$$= \frac{1}{m-1} \mathbf{W}\Sigma^2\mathbf{W}^\top, \quad (5.91)$$

where we use the fact that  $\mathbf{U}^\top \mathbf{U} = \mathbf{I}$  because the  $\mathbf{U}$  matrix of the singular value decomposition is defined to be orthogonal. This shows that the covariance of  $\mathbf{z}$  is diagonal as required:

$$\text{Var}[\mathbf{z}] = \frac{1}{m-1} \mathbf{Z}^\top \mathbf{Z} \quad (5.92)$$

$$= \frac{1}{m-1} \mathbf{W}^\top \mathbf{X}^\top \mathbf{X} \mathbf{W} \quad (5.93)$$

$$= \frac{1}{m-1} \mathbf{W}^\top \mathbf{W} \Sigma^2 \mathbf{W}^\top \mathbf{W} \quad (5.94)$$

$$= \frac{1}{m-1} \Sigma^2, \quad (5.95)$$

where this time we use the fact that  $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$ , again from the definition of the SVD.

The above analysis shows that when we project the data  $\mathbf{x}$  to  $\mathbf{z}$ , via the linear transformation  $\mathbf{W}$ , the resulting representation has a diagonal covariance matrix (as given by  $\Sigma^2$ ), which immediately implies that the individual elements of  $\mathbf{z}$  are mutually uncorrelated.

This ability of PCA to transform data into a representation where the elements are mutually uncorrelated is a very important property of PCA. It is a simple example of a representation that attempts to *disentangle the unknown factors of variation* underlying the data. In the case of PCA, this disentangling takes the form of finding a rotation of the input space (described by  $\mathbf{W}$ ) that aligns the principal axes of variance with the basis of the new representation space associated with  $\mathbf{z}$ .

While correlation is an important category of dependency between elements of the data, we are also interested in learning representations that disentangle more complicated forms of feature dependencies. For this, we will need more than what can be done with a simple linear transformation.

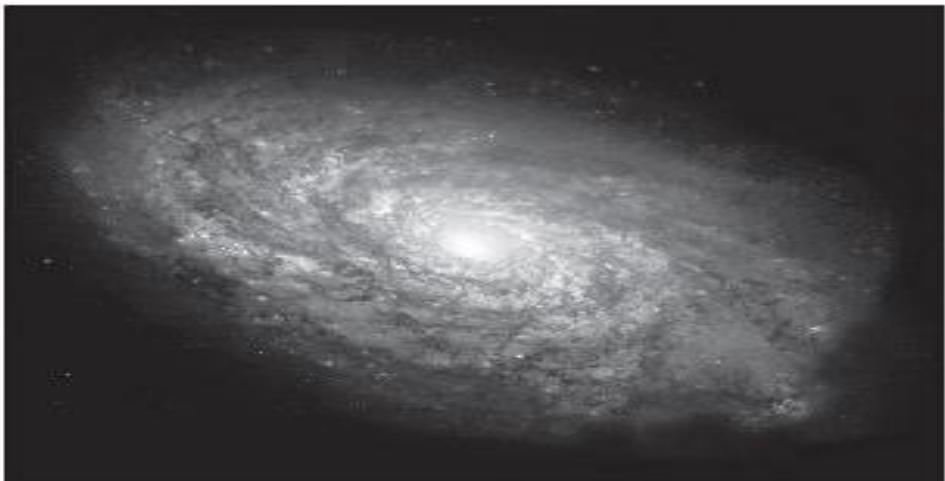
## 5.8.2 $k$ -means Clustering

The  $k$ -means algorithm works by initializing  $k$  different centroids  $\{\mu^{(1)}, \dots, \mu^{(k)}\}$  to different values, then alternating between two different steps until convergence. In one step, each training example is assigned to cluster  $i$ , where  $i$  is the index of the nearest centroid  $\mu^{(i)}$ . In the other step, each centroid  $\mu^{(i)}$  is updated to the mean of all training examples  $x^{(j)}$  assigned to cluster  $i$ .

One difficulty pertaining to clustering is that the clustering problem is inherently ill posed, in the sense that there is no single criterion that measures how well a clustering of the data corresponds to the real world. We can measure properties of the clustering, such as the average Euclidean distance from a cluster centroid to the members of the cluster. This enables us to tell how well we are able to reconstruct the training data from the cluster assignments. We do not know how

# Classification

- انسان ها دارای توانایی ذاتی برای طبقه بندی اشیا به دسته ها هستند
- به عنوان مثال، کارهای پیش پا افتاده مانند فیلتر کردن ایمیل های Spam
- یا کارهای تخصصی تر مانند تشخیص اجرام آسمانی در تصاویر تلسکوپ.



(a) A spiral galaxy.



(b) An elliptical galaxy.

**Figure 3.1.** Classification of galaxies from telescope images taken from the NASA website.

## اهمیت مدل های طبقه بندی در داده کاوی

- یک مدل طبقه بندی دو نقش مهم در داده کاوی ایفا می کند.
- ۱- از آن به عنوان یک مدل پیشگو برای طبقه بندی نمونه های بدون برچسب قبلی استفاده می شود.  
یک مدل طبقه بندی خوب باید پیشگویهای دقیق با زمان پاسخ سریع ارائه دهد.
- ۲- به عنوان یک مدل توصیفی برای شناسایی ویژگی هایی اثرگذار عمل می کند

**Table 3.1.** Examples of classification tasks.

Task	Attribute set	Class label
Spam filtering	Features extracted from email message header and content	spam or non-spam
Tumor identification	Features extracted from magnetic resonance imaging (MRI) scans	malignant or benign
Galaxy classification	Features extracted from telescope images	elliptical, spiral, or irregular-shaped

**Table 3.3.** A sample data for the loan borrower classification problem.

ID	Home Owner	Marital Status	Annual Income	Defaulted?
1	Yes	Single	125000	No
2	No	Married	100000	No
3	No	Single	70000	No
4	Yes	Married	120000	No
5	No	Divorced	95000	Yes
6	No	Single	60000	No
7	Yes	Divorced	220000	No
8	No	Single	85000	Yes
9	No	Married	75000	No
10	No	Single	90000	Yes

## طبقه بندی درخت تصمیم

- درخت دارای سه نوع گره است:
- گره ریشه (root node)، لینک ورودی ندارد ولی لینک خروجی می توان داشته باشد
- گره های داخلی (Internal nodes) که هر کدام دقیقاً یک لینک ورودی و دو یا چند لینک خروجی دارند.
- گره های برگ یا خروجی (Leaf or terminal nodes) که هر کدام دقیقاً یک پیوند ورودی دارند و هیچ پیوند خروجی ندارند.

# مثال تشخیص پستانداران از غیر پستانداران

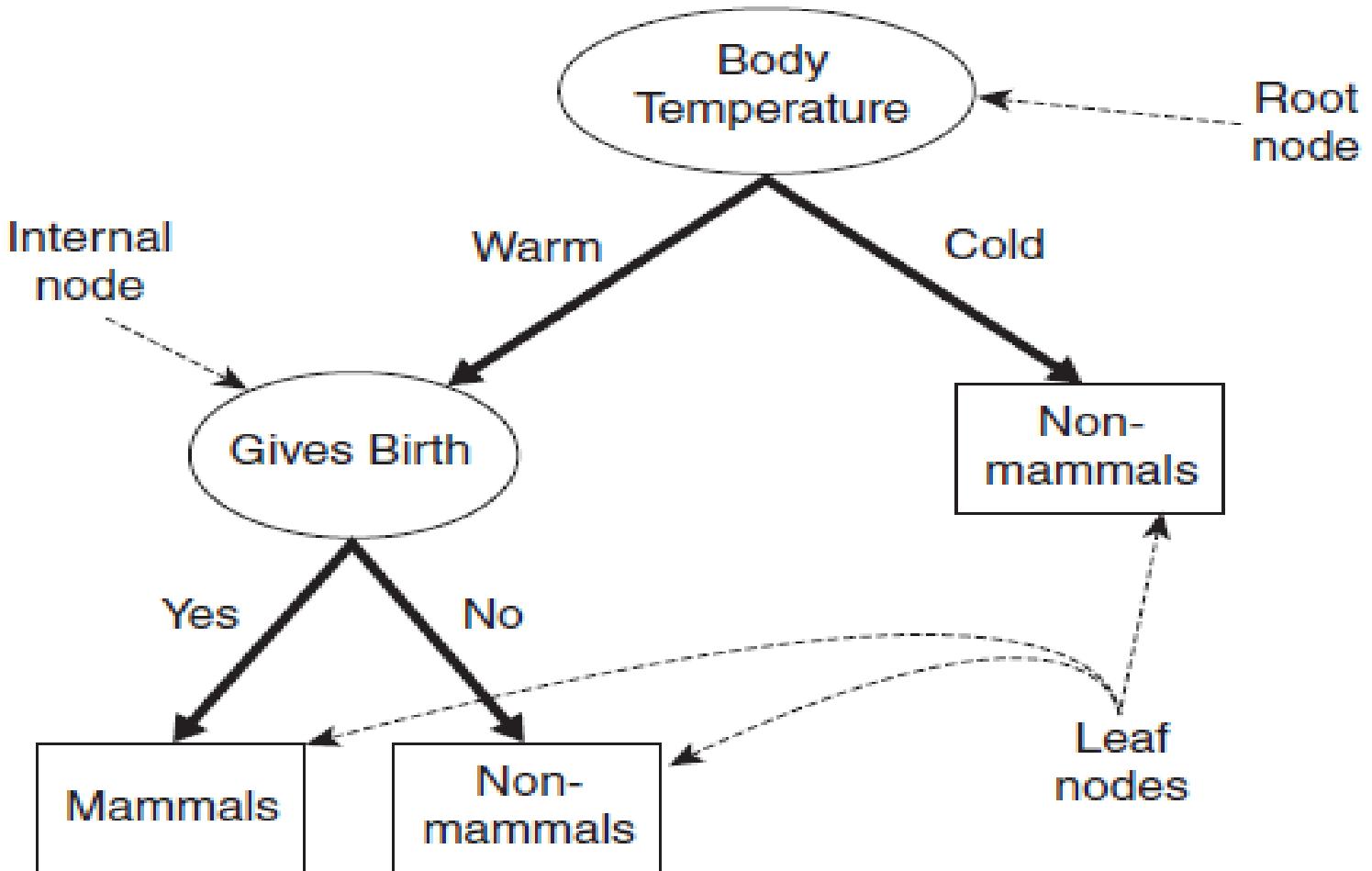
- برای نشان دادن نحوه عملکرد درخت تصمیم،
- مثال طبقه بندی تشخیص پستانداران از غیر پستانداران را در نظر بگیرید.

**Table 3.2.** A sample data for the vertebrate classification problem.

Vertebrate Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard	cold-blooded	scales	yes	yes	no	no	no	fish
shark								
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

## ادامه مثال تشخیص پستانداران از غیر پستانداران

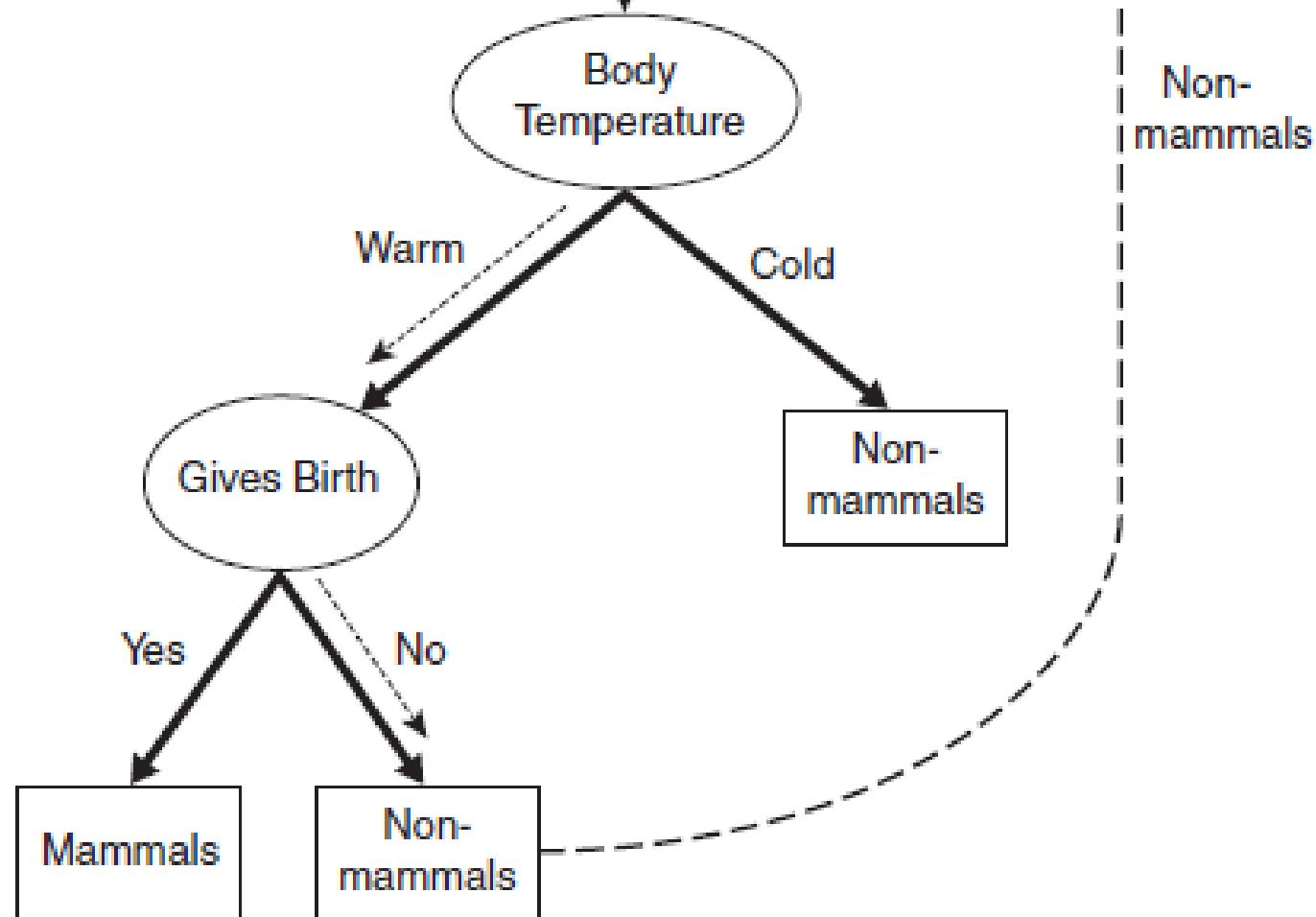
- فرض کنید یک گونه جدید توسط دانشمندان کشف شده است.
- چگونه می توان تشخیص داد که پستاندار است یا غیر پستاندار؟
- یک رویکرد این است که یک سری سوالات در مورد ویژگی های گونه مطرح کنیم.
- اولین سوالی که ممکن است بپرسیم این است که آیا این گونه جدید خونسرد است یا خون گرم.
- اگر خونسرد است پس قطعاً پستاندار نیست.
- در غیر این صورت یا پرنده است یا پستاندار.
- در مورد دوم، باید یک سوال بعدی بپرسیم: آیا ماده های این گونه، بچه های خود را به دنیا می آورند؟ آنها یی که زایمان می کنند قطعاً پستانداران هستند، در حالی که آنها یی که زایمان نمی کنند احتمالاً غیر پستانداران هستند (به استثنای پستانداران تخمگذار مانند پلاتیپوس و مورچه خوار خاردار).



**Figure 3.4.** A decision tree for the mammal classification problem.

Unlabeled  
data

Name	Body temperature	Gives Birth	...	Class
Flamingo	Warm	No	...	?



# پیاده سازی الگوریتم Hunt برای درخت تصمیم

- معیار تقسیم چیست؟ در هر مرحله، یک ویژگی باید انتخاب شود تا نمونه های آموزشی مرتبط با یک گره به زیر مجموعه های کوچکتر مرتبط با گره های فرزند آن تقسیم شود. معیار تقسیم تعیین می کند که کدام ویژگی به عنوان شرط آزمون انتخاب می شود و چگونه نمونه های آموزشی باید در گره های فرزند توزیع شوند. که باید به آنها توجه شود.
- ملاک توقف چیست؟ الگوریتم اصلی گسترش یک گره را تنها زمانی متوقف می کند که تمام نمونه های آموزشی مرتبط با گره دارای برچسب های کلاس یکسان یا دارای مقادیر مشخصه یکسان باشند. اگرچه این شرایط کافی است، دلایلی برای توقف گسترش یک گره خیلی زودتر وجود دارد، حتی اگر گره برگ حاوی نمونه های آموزشی از بیش از یک کلاس باشد. این فرآیند خاتمه زودهنگام نامیده می شود و شرایطی که برای تعیین زمانی که یک گره باید از گسترش متوقف شود، معیار توقف نامیده می شود.

**Table 3.3.** A sample data for the loan borrower classification problem.

ID	Home Owner	Marital Status	Annual Income	Defaulted?
1	Yes	Single	125000	No
2	No	Married	100000	No
3	No	Single	70000	No
4	Yes	Married	120000	No
5	No	Divorced	95000	Yes
6	No	Single	60000	No
7	Yes	Divorced	220000	No
8	No	Single	85000	Yes
9	No	Married	75000	No
10	No	Single	90000	Yes



(a)

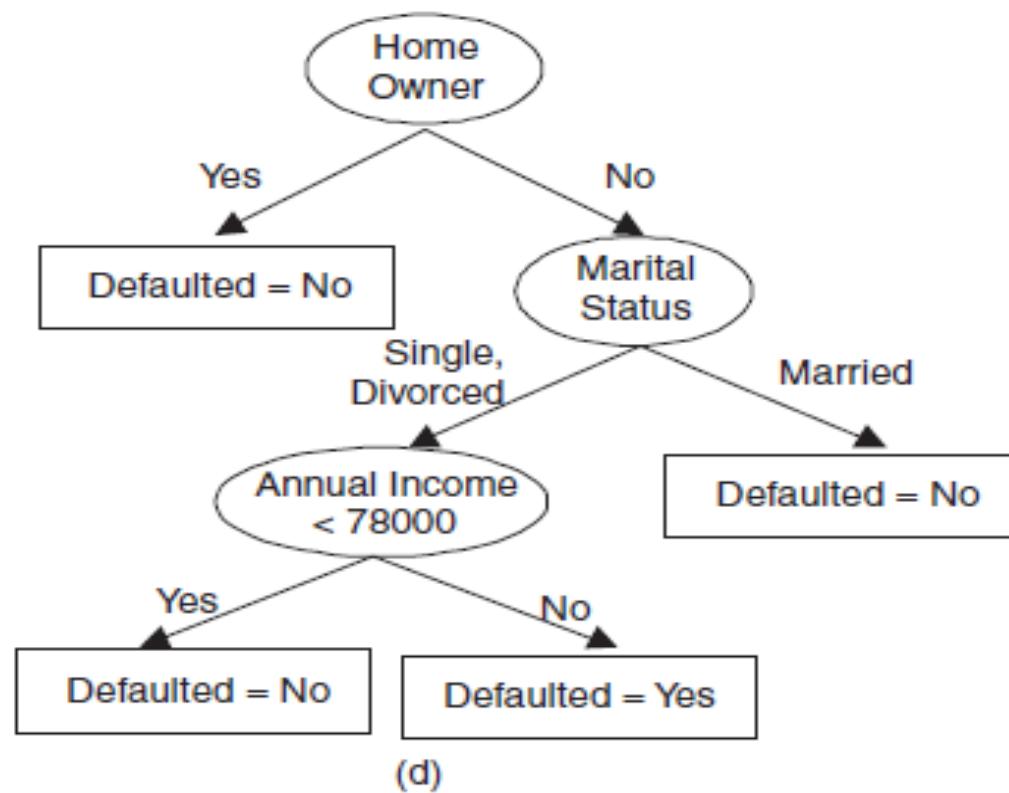
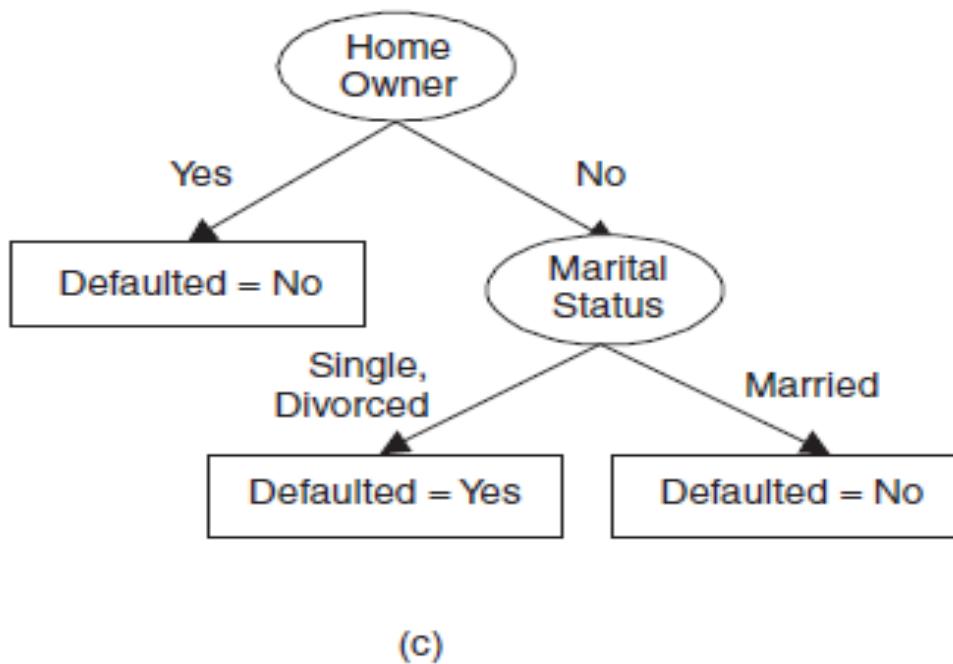
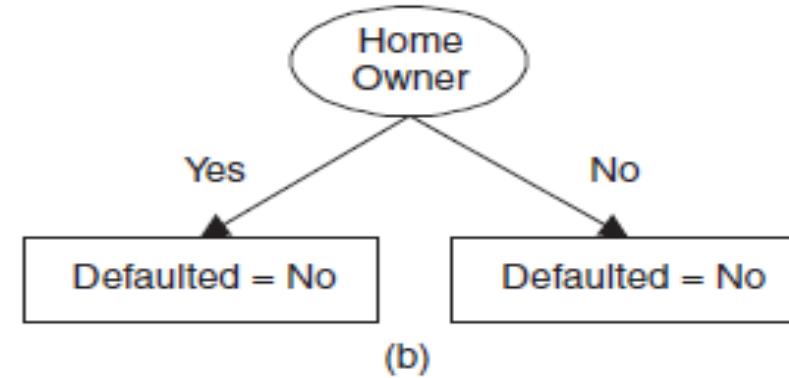
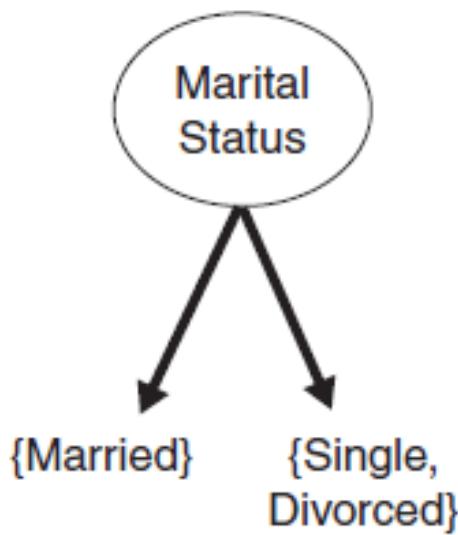
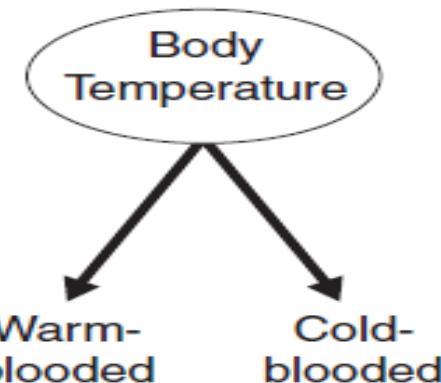
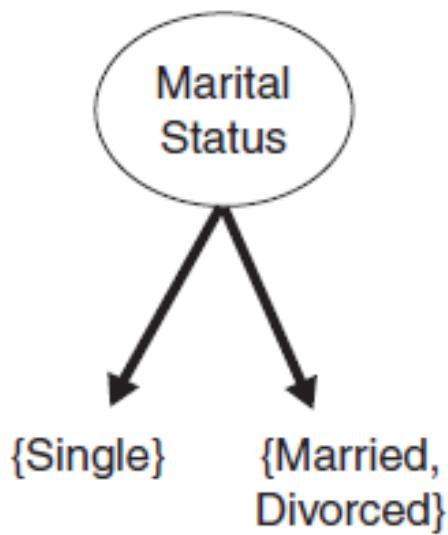


Figure 3.6. Hunt's algorithm for building decision trees.

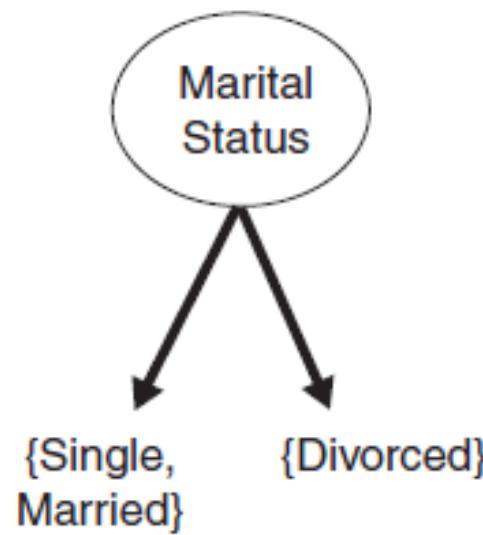
# Binary Attributes



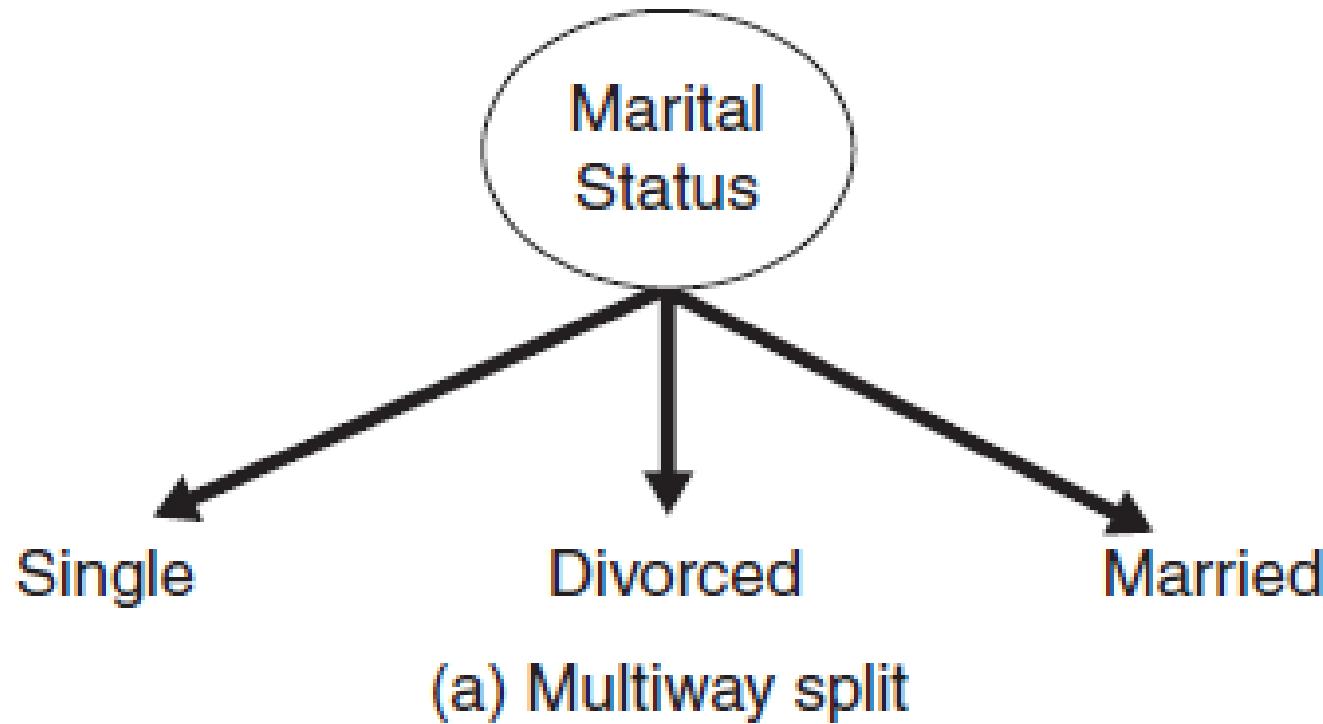
OR



OR



# Nominal Attributes



# Ordinal Attributes

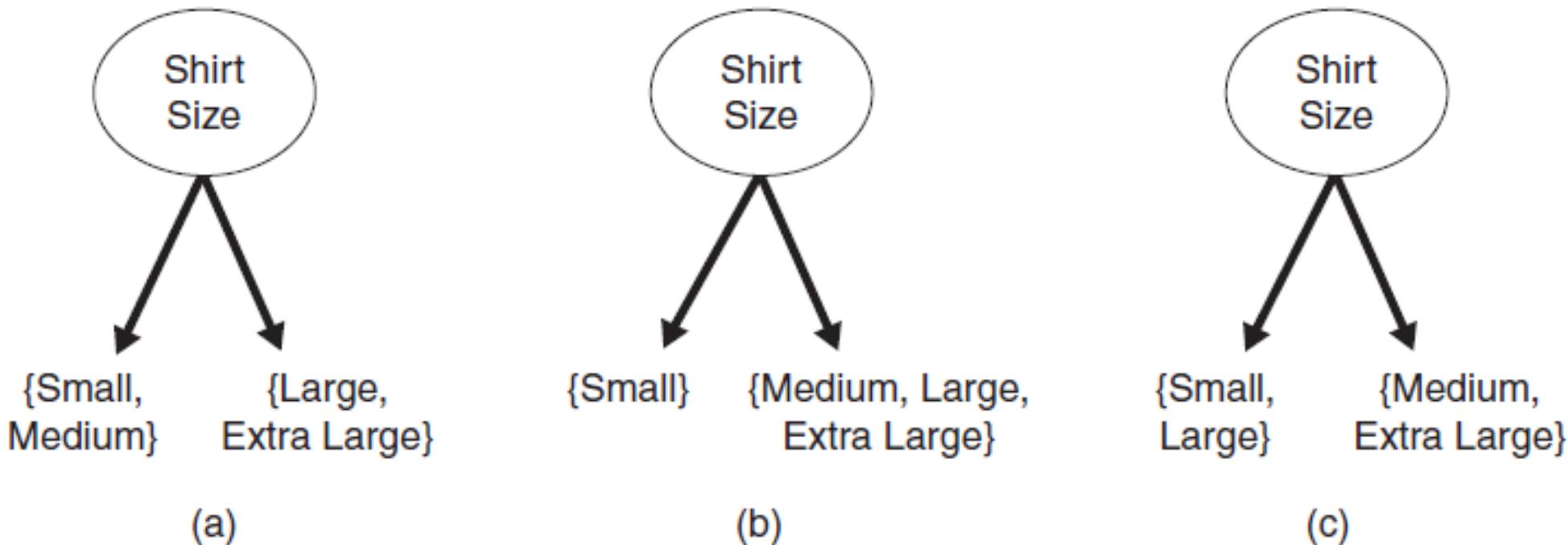
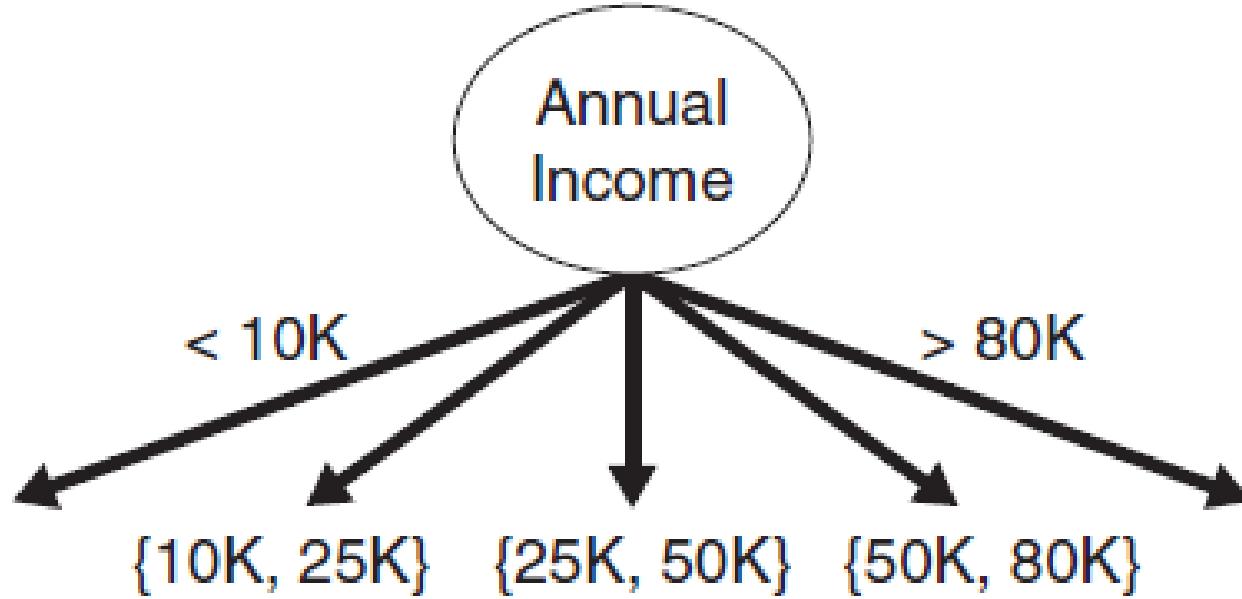
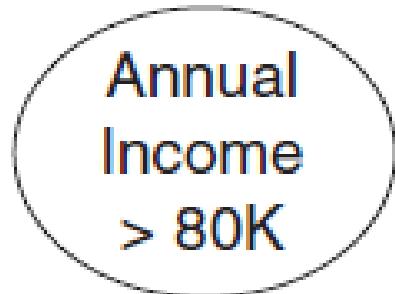


Figure 3.9. Different ways of grouping ordinal attribute values.

# Continuous Attributes



(b)

Figure 3.10. Test condition for continuous attributes.

# تعیین گره والد و گره فرزند

- ناخالصی (Impurity) یک گره میزان تفاوت داده های آن گره به متغیر وابسته اندازه گیری می کند.
- الگوریتم Hunt برای تعیین گروه والد و فرزند:
  - میزان ناخالصی را برای تک تک گره ها محاسبه کنید
  - میزان ناخالصی گره والد و گره های فرزند را محاسبه کنید
  - میزان ناخالصی بیشتر منجر به والد شدن یک گره می شود

## محاسبه ناخالصی برای یک گره

$$\text{Entropy} = - \sum_{i=0}^{c-1} p_i(t) \log_2 p_i(t), \quad (3.4)$$

$$\text{Gini index} = 1 - \sum_{i=0}^{c-1} p_i(t)^2, \quad (3.5)$$

$$\text{Classification error} = 1 - \max_i [p_i(t)], \quad (3.6)$$

where  $p_i(t)$  is the relative frequency of training instances that belong to class  $i$  at node  $t$ ,  $c$  is the total number of classes, and  $0 \log_2 0 = 0$  in entropy

## مثال

Node $N_1$	Count
Class=0	0
Class=1	6

$$\text{Gini} = 1 - (0/6)^2 - (6/6)^2 = 0$$

$$\text{Entropy} = -(0/6) \log_2(0/6) - (6/6) \log_2(6/6) = 0$$

$$\text{Error} = 1 - \max[0/6, 6/6] = 0$$

Node $N_2$	Count
Class=0	1
Class=1	5

$$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$$

$$\text{Entropy} = -(1/6) \log_2(1/6) - (5/6) \log_2(5/6) = 0.650$$

$$\text{Error} = 1 - \max[1/6, 5/6] = 0.167$$

Node $N_3$	Count
Class=0	3
Class=1	3

$$\text{Gini} = 1 - (3/6)^2 - (3/6)^2 = 0.5$$

$$\text{Entropy} = -(3/6) \log_2(3/6) - (3/6) \log_2(3/6) = 1$$

$$\text{Error} = 1 - \max[3/6, 3/6] = 0.5$$

## محاسبه میزان ناخالصی یک گره والد با فرزندش

- فرض کنید یک گروه والد  $k$  فرزند دارد

$$I(\text{children}) = \sum_{j=1}^k \frac{N(v_j)}{N} I(v_j), \quad (3.7)$$

# مثال وام بانکی

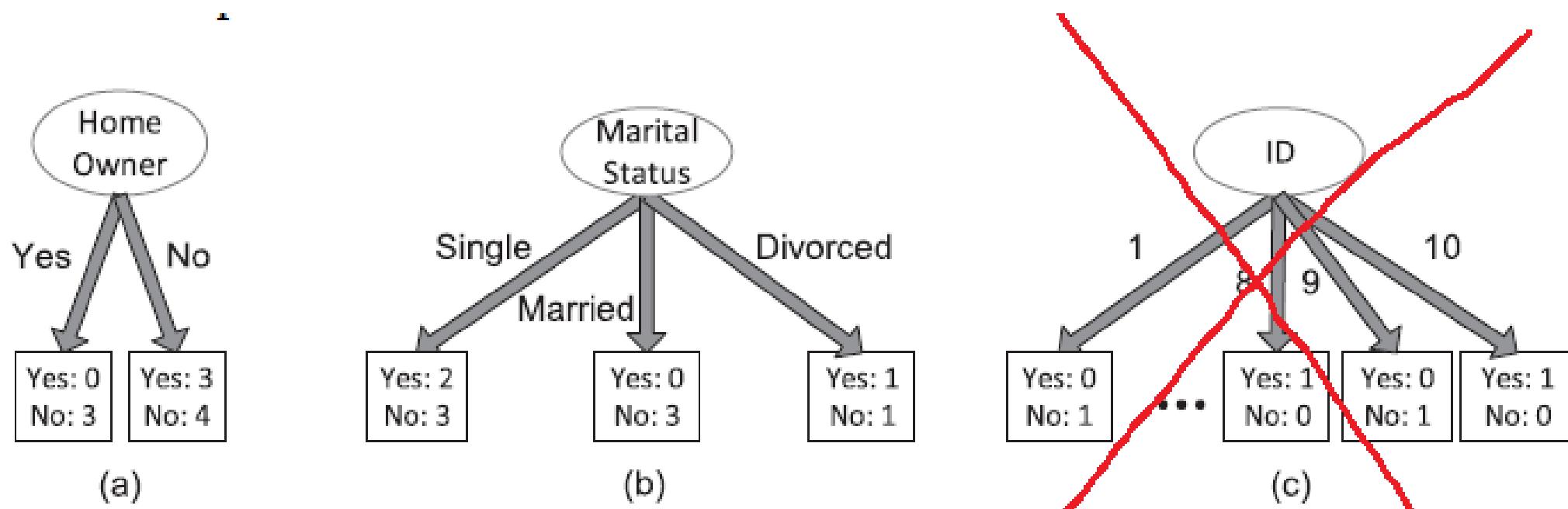
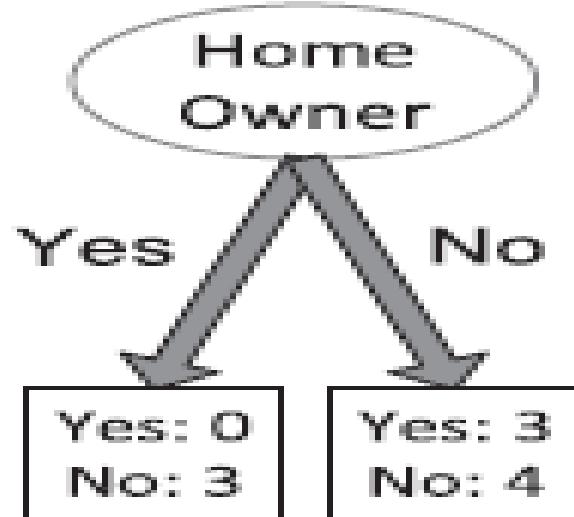


Figure 3.12. Examples of candidate attribute test conditions.

# ادامه مثال وام بانکی

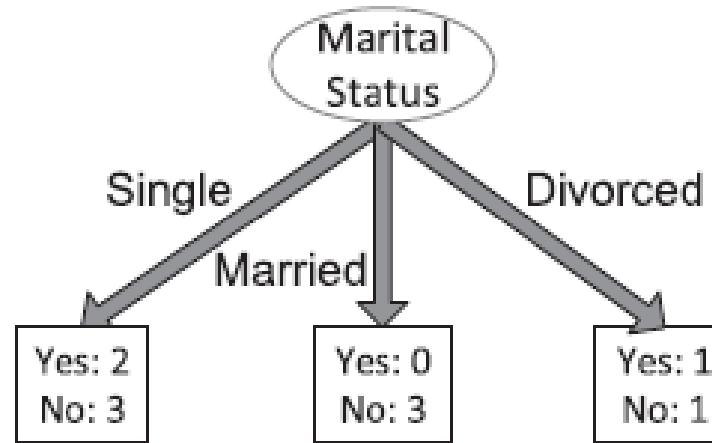


$$I(\text{Home Owner} = \text{yes}) = -\frac{0}{3} \log_2 \frac{0}{3} - \frac{3}{3} \log_2 \frac{3}{3} = 0$$

$$I(\text{Home Owner} = \text{no}) = -\frac{3}{7} \log_2 \frac{3}{7} - \frac{4}{7} \log_2 \frac{4}{7} = 0.985$$

$$I(\text{Home Owner}) = \frac{3}{10} \times 0 + \frac{7}{10} \times 0.985 = 0.690$$

# ادامه مثال وام بانکی

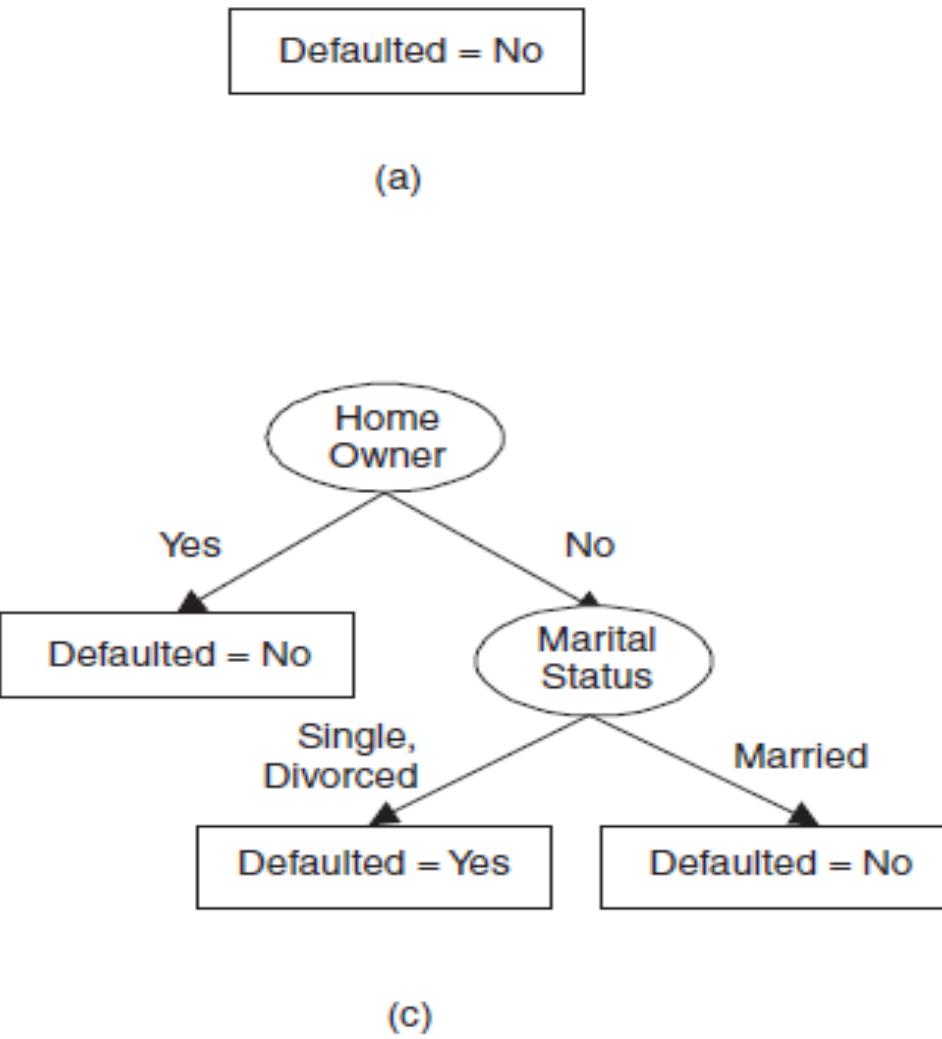


$$I(\text{Marital Status} = \text{Single}) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.971$$

$$I(\text{Marital Status} = \text{Married}) = -\frac{0}{3} \log_2 \frac{0}{3} - \frac{3}{3} \log_2 \frac{3}{3} = 0$$

$$I(\text{Marital Status} = \text{Divorced}) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1.000$$

$$I(\text{Marital Status}) = \frac{5}{10} \times 0.971 + \frac{3}{10} \times 0 + \frac{2}{10} \times 1 = 0.686$$



**Figure 3.6.** Hunt's algorithm for building decision trees.

# تحليل پیوند association analysis

- تحلیل پیوند روشی برای کشف روابط جالب و پنهان در مجموعه داده های بزرگ است.
- این روابط را می توان به صورت مجموعه ای از اقلام موجود در بسیاری از معاملات نشان داد
- که به عنوان مجموعه اقلام مکرر یا قوانین معامله (transaction) شناخته می شوند که روابط بین دو مجموعه اقلام را نشان می دهد.
- به عنوان مثال، قانون زیر را می توان از مجموعه داده های جدول ۴.۱ استخراج کرد:
  - {پوشک} → {آبجو}
- این قانون نشان می دهد که بین فروش پوشک و آبجو رابطه وجود دارد زیرا بسیاری از مشتریانی که پوشک می خرند آبجو نیز می خرند.
- خرده فروشان می توانند از این نوع قوانین برای کمک به شناسایی فرصت های جدید برای فروش متقابل محصولات خود به مشتریان استفاده کنند.

# Binary Representation

Table 4.1. An example of market basket transactions.

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

Table 4.2. A binary 0/1 representation of market basket data.

TID	Bread	Milk	Diapers	Beer	Eggs	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

# Itemset and Support Count

$I = \{i_1, i_2, \dots, i_d\}$  the set of all items in a market basket data

$T = \{t_1, t_2, \dots, t_N\}$  be the set of all transactions.

Each transaction  $t_i$  contains a subset of items chosen from  $I$ .

If an itemset contains  $k$  items, it is called a  $k$ -itemset.

For instance, {Beer, Diapers, Milk} is an example of a 3-itemset.

A transaction  $t_j$  is said to contain an itemset  $X$  if  $X$  is a subset of  $t_j$ .

**Table 4.2.** A binary 0/1 representation of market basket data.

TID	Bread	Milk	Diapers	Beer	Eggs	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

For example, the second transaction shown in Table 4.2 contains the

itemset {Bread, Diapers} but not {Bread, Milk}.

the support count,  $\sigma(X)$ , for an itemset  $X$  can be stated as  $\sigma(X) = \#\{t_i | X \subseteq t_i, t_i \in T\}$

**Association Rule** An association rule is an implication expression of the form  $X \rightarrow Y$ , where  $X$  and  $Y$  are disjoint itemsets, i.e.,  $X \cap Y = \emptyset$ .

$$\text{Support, } s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N};$$

$$\text{Confidence, } c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}.$$

**Table 4.2.** A binary 0/1 representation of market basket data.

TID	Bread	Milk	Diapers	Beer	Eggs	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

$$X \longrightarrow Y$$

**Example 4.1.** Consider the rule  $\{\text{Milk}, \text{Diapers}\} \longrightarrow \{\text{Beer}\}$ . Because the support count for  $\{\text{Milk}, \text{Diapers}, \text{Beer}\}$  is 2 and the total number of transactions is 5, the rule's support is  $2/5 = 0.4$ . The rule's confidence is obtained by dividing the support count for  $\{\text{Milk}, \text{Diapers}, \text{Beer}\}$  by the support count for  $\{\text{Milk}, \text{Diapers}\}$ . Since there are 3 transactions that contain milk and diapers, the confidence for this rule is  $2/3 = 0.67$ . ■

# Association Rule Mining Problem

**Definition 4.1** (Association Rule Discovery). Given a set of transactions  $T$ , find all the rules having support  $> \text{minsup}$  and confidence  $\geq \text{minconf}$ , where  $\text{minsup}$  and  $\text{minconf}$  are the corresponding support and confidence thresholds.

# مثال

$I = \{a, b, c, d, e\}$  candidate itemset  $2^k - 1$  frequent itemsets,

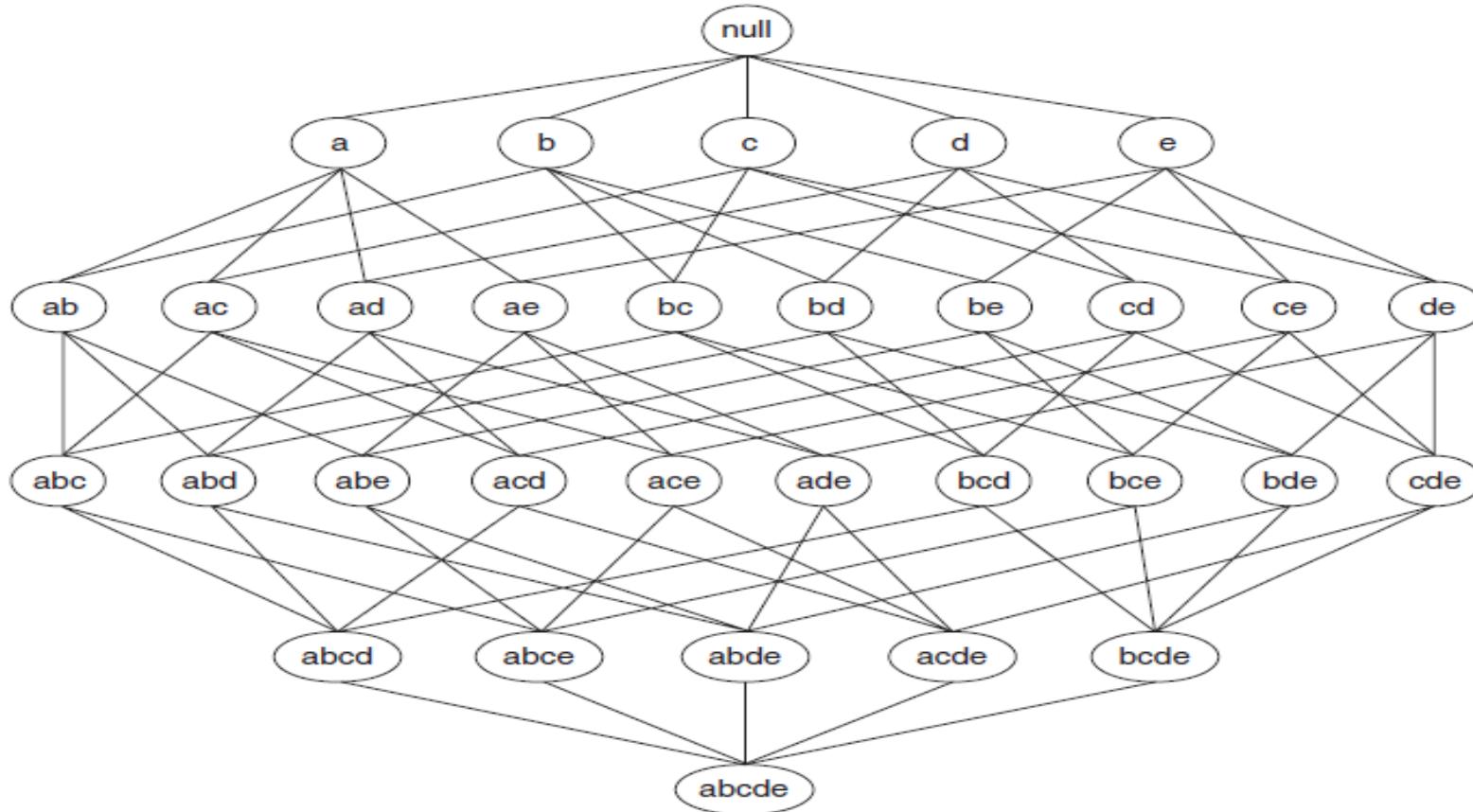


Figure 4.1. An itemset lattice.

# استفاده از قضایا

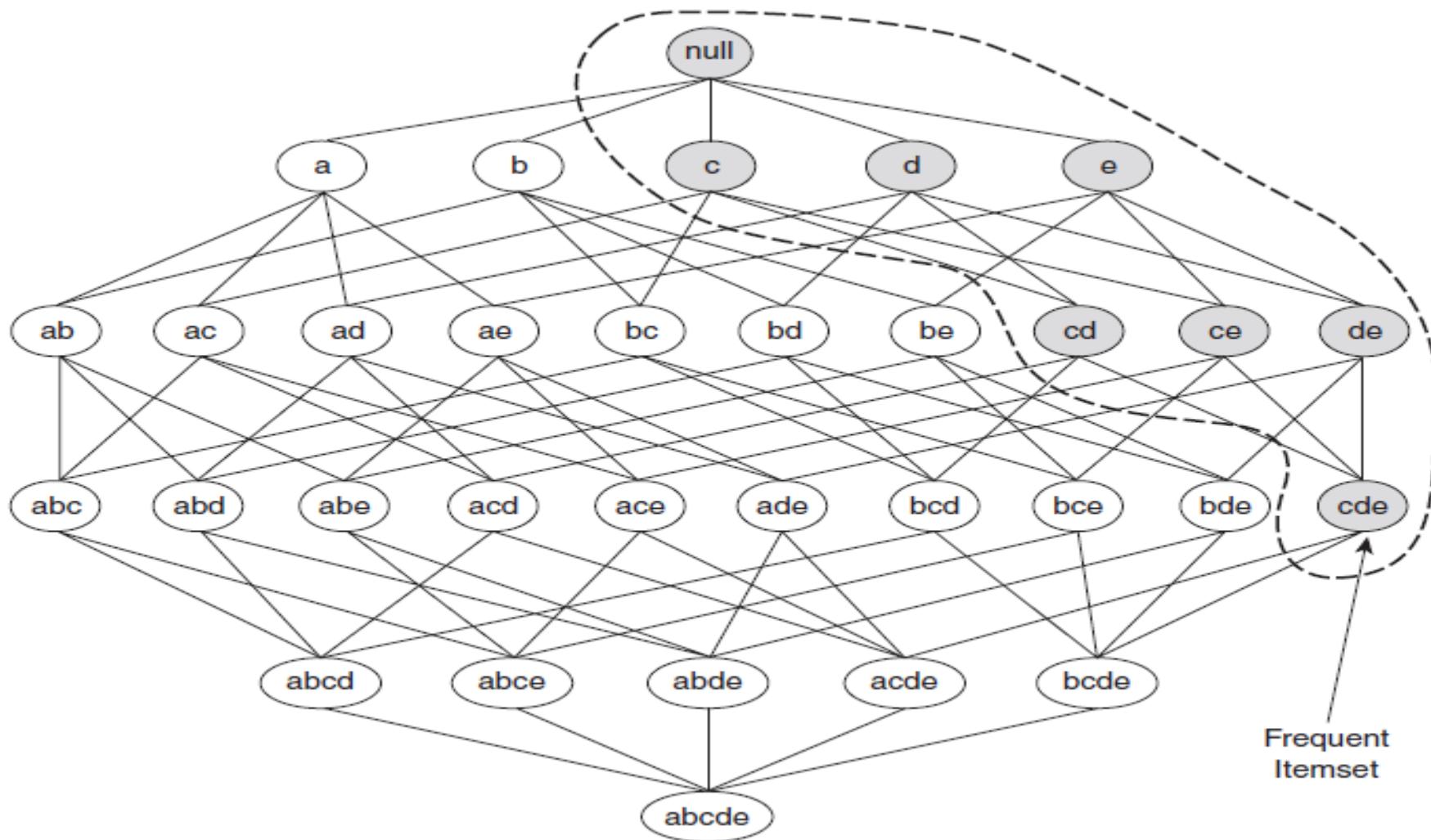
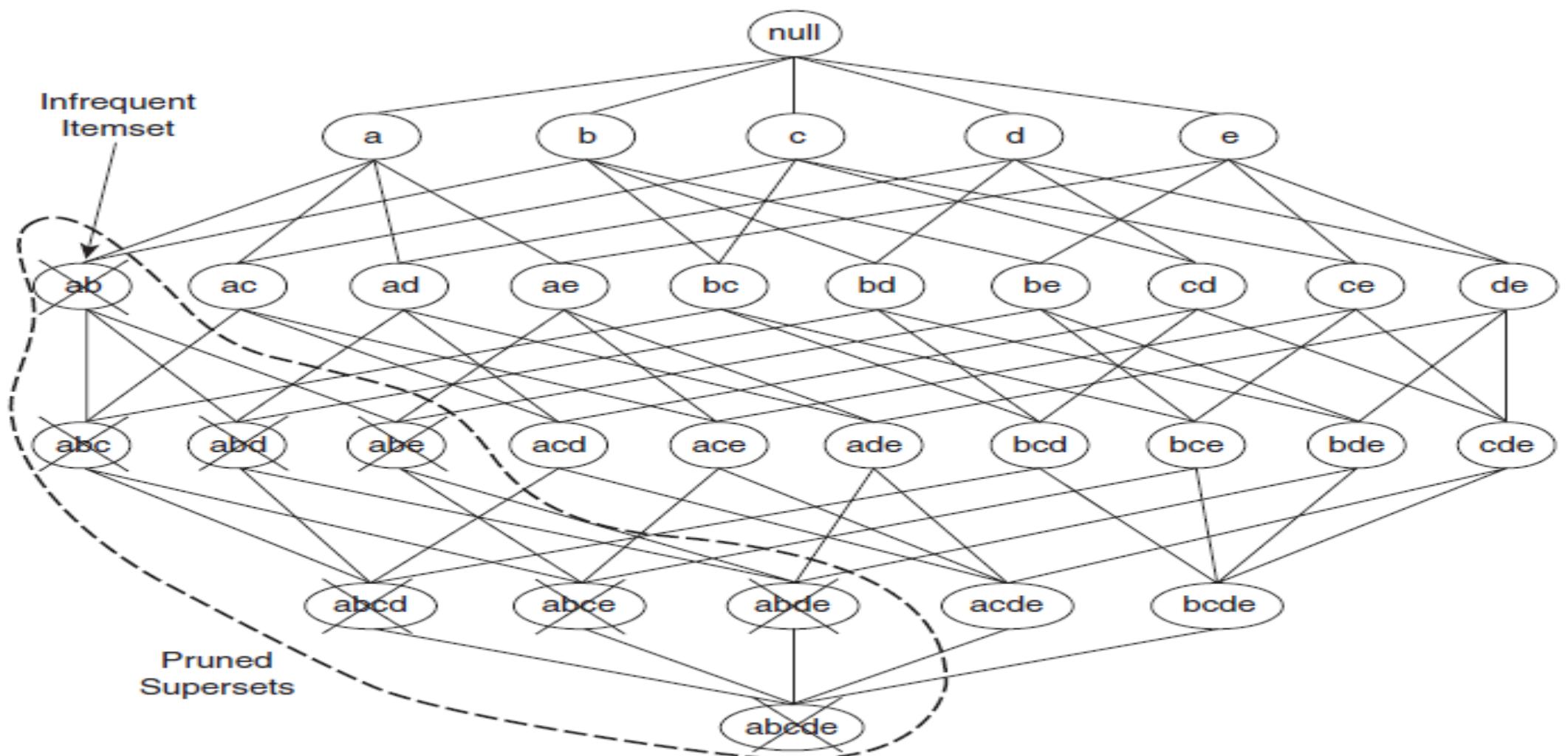


Figure 4.0 - An illustration of the Apriori algorithm using the FPTree structure

# استفاده از قضایا



**Figure 4.4.** An illustration of support-based pruning. If  $\{a, b\}$  is infrequent, then all supersets of  $\{a, b\}$  are infrequent.

---

**Algorithm 4.2** Rule generation of the *Apriori* algorithm.

---

```
1: for each frequent  $k$ -itemset  $f_k$ ,  $k \geq 2$  do
2:    $H_1 = \{i \mid i \in f_k\}$            {1-item consequents of the rule.}
3:   call ap-genrules( $f_k, H_1.$ )
4: end for
```

---

**Algorithm 4.3** Procedure **ap-genrules**( $f_k, H_m$ ).

---

```
1:  $k = |f_k|$     {size of frequent itemset.}
2:  $m = |H_m|$     {size of rule consequent.}
3: if  $k > m + 1$  then
4:    $H_{m+1} = \text{candidate-gen}(H_m).$ 
5:    $H_{m+1} = \text{candidate-prune}(H_{m+1}, H_m).$ 
6:   for each  $h_{m+1} \in H_{m+1}$  do
7:      $\text{conf} = \sigma(f_k)/\sigma(f_k - h_{m+1}).$ 
8:     if  $\text{conf} \geq \text{minconf}$  then
9:       output the rule  $(f_k - h_{m+1}) \rightarrow h_{m+1}.$ 
10:    else
11:      delete  $h_{m+1}$  from  $H_{m+1}.$ 
12:    end if
13:   end for
14:   call ap-genrules( $f_k, H_{m+1}.$ )
15: end if
```

---

# تحلیل خوشة

## Cluster Analysis: unsupervised classification

- تجزیه و تحلیل خوشه ای داده ها را به گروه هایی (خوشه ها) که معنی دار، مفید یا هر دو هستند تقسیم می کند.
- اگر هدف تعیین خوشه های معنادار باشد، خوشه ها باید ساختار طبیعی داده ها را بگیرند.
- در برخی موارد، از تحلیل خوشه ای برای خلاصه سازی داده ها استفاده می شود تا اندازه داده ها کاهش یابد.
- تحلیل خوشه ای چه برای درک و چه برای کاربردهای دیگر، مدت هاست که نقش مهمی در زمینه های مختلف نظیر: روانشناسی، علوم اجتماعی، زیست شناسی، آمار، تشخیص الگو، بازیابی اطلاعات، یادگیری ماشین و داده کاوی، ایفا کرده است

# کاربردهای تحلیل خوشه‌ای در مسائل عملی

- خوشه‌بندی برای درک: کلاس‌ها (گروه‌های معنادار از اشیاء که ویژگی‌های مشترکی دارند) نقش مهمی در نحوه تحلیل و توصیف مردم جهان دارند.
- انسان در تقسیم اشیاء به گروه‌ها (خوشه‌بندی) و اختصاص اشیاء خاص به این گروه‌ها (طبقه بندی) مهارت دارد.
- تحلیل خوشه‌ای مطالعه تکنیک‌هایی برای یافتن خودکار خوشه‌ها است.
- مثلاً درک آب و هوای زمین مستلزم یافتن الگوهایی در جو و اقیانوس است. برای این منظور، تحلیل خوشه‌ای برای یافتن الگوهایی در فشار اتمسفر و دمای اقیانوس که تأثیر قابل توجهی بر آب و هوا دارند، به کار گرفته شده است.

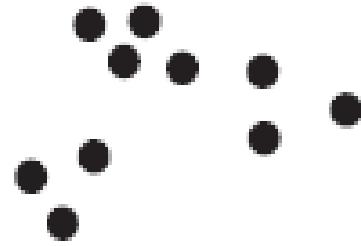
# کاربردهای تحلیل خوشه‌ای در مسائل عملی

خوشه‌بندی برای

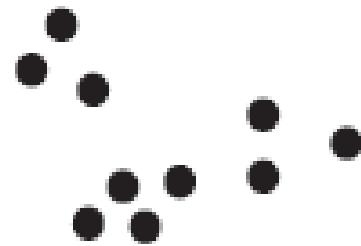
- خلاصه سازی: به منظور استفاده از روش‌های معمول آماری بر روی متغیرهای کمتر مقایسه کردن مجموعه داده‌ها
- یافتن نزدیکترین همسایه‌ها

## نکته مهم

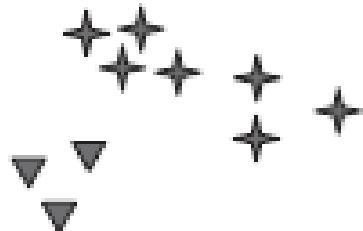
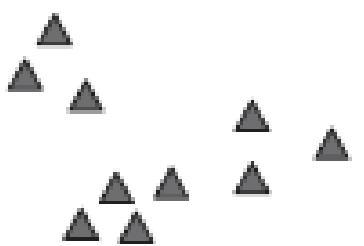
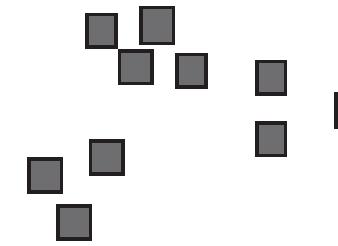
- در اینجا تحلیل خوشه ای داده تنها بر اساس اطلاعاتی که فقط در داده هایی یافت می شود که اشیاء و روابط آنها را توصیف می کند، انجام می شود. بنابراین آنرا **تحلیل اکتشافی** می نامیم.
- اگر خوشه ها بر اساس نظر محقق باشد و تنها با استفاده از داده ها نظر محقق تائید، تصحیح و تایید و کلاً رد می شود، به آن **تحلیل تائیدی** گویند
- همانگونه که شکل زیر نشان می دهد: تعریف خوشه نادقیق است و بهترین تعریف به ماهیت داده ها و نتایج مورد نظر بستگی دارد.



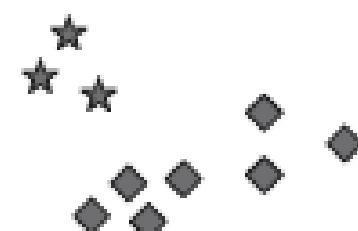
(a) Original points.



(b) Two clusters.



(c) Four clusters.



(d) Six clusters.

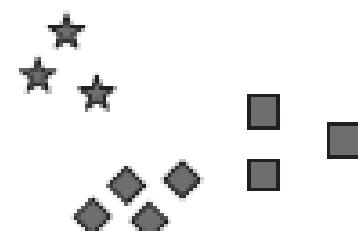
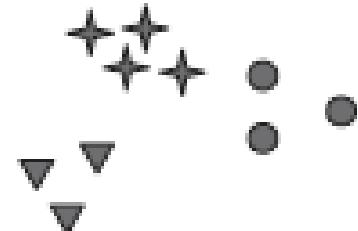
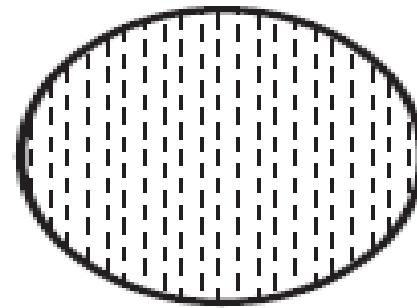
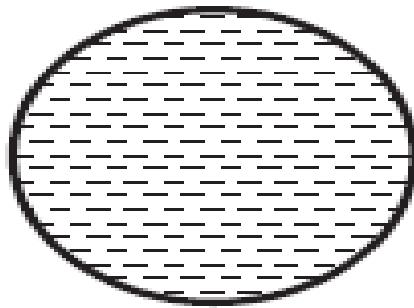


Figure 5.1. Three different ways of clustering the same set of points.

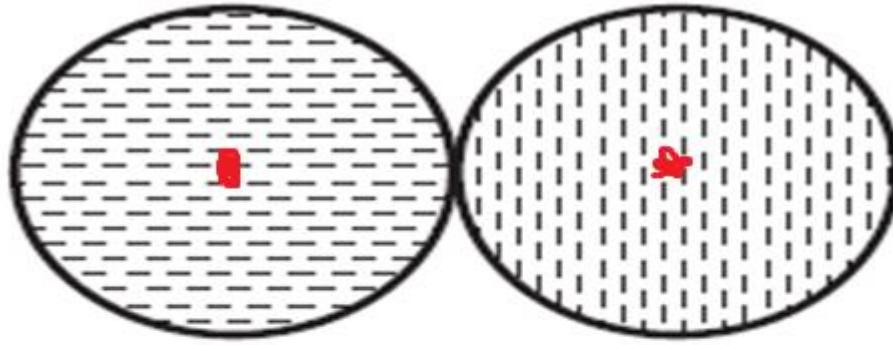
# انواع خوشه بندی

- خوشه بندی **partitional** (unnested) تقسیم داده ها به زیرمجموعه های غیر همپوشان (خوشه ها) است به طوری که هر شی داده دقیقاً در یک زیر مجموعه قرار می گیرد.
- اگر به خوشه ها اجازه دهیم که زیر خوشه هایی داشته باشند، یک خوشه بندی سلسله مراتبی (hierarchical) به دست می آوریم که مجموعه ای از خوشه های تودر تو است که به صورت درختی سازمان دهی شده اند. هر گره (خوشه) در درخت به تعدادی فرزندان (زیره خوشه ها) متصل است. شکل بالا بخش های c و d مثال های از این خوشه بندی هستند.
- اگر هر داده تنها بتواند به یک خوشه متعلق باشد به آن خوشه بندی Exclusive گویند.
- اگر هر داده تنها بتواند همزمان به بیش از یک خوشه متعلق باشد به آن خوشه بندی Overlapping گویند. مثلاً یک فرد می تواند همزمان دانشجو و کارمند دانشگاه باشد.
- اگر عضویت هر داده در یک خوشه با شاخصی با عنوان **میزان تعلق** نشان داده شود، به آن خوشه بندی Fuzzy گویند.
- اگر تمامی داده های در خوشه ها قرار گیرند، خوشه بندی را Complete گویند

## انواع خوش (برای داده های ۲ بعدی)



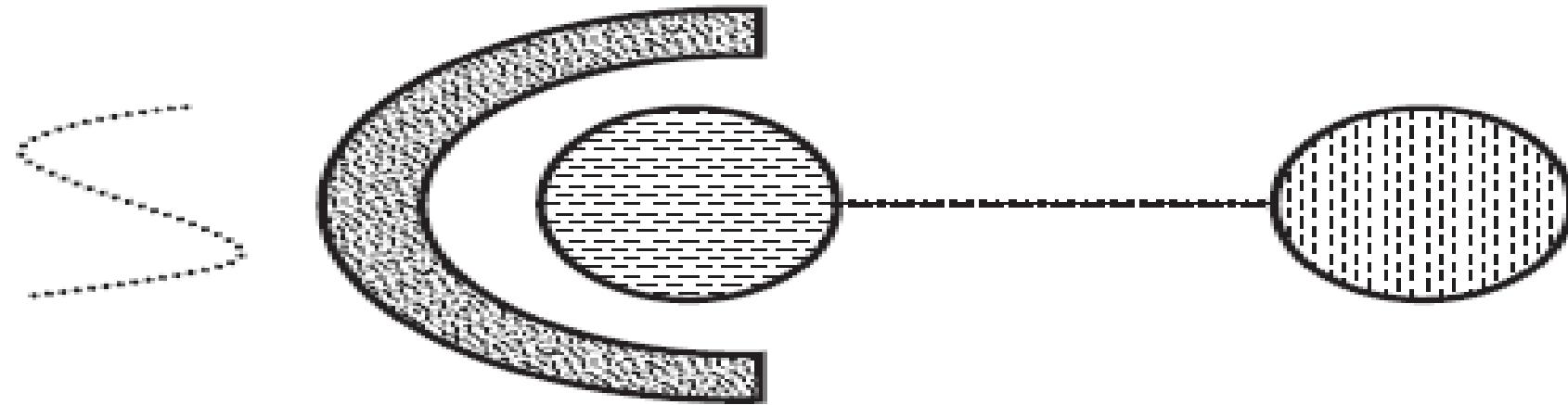
(a) Well-separated clusters. Each point is closer to all of the points in its cluster than to any point in another cluster.



(b) Center-based clusters. Each point is closer to the center of its cluster than to the center of any other cluster.

Prototype-Based      center-based clusters

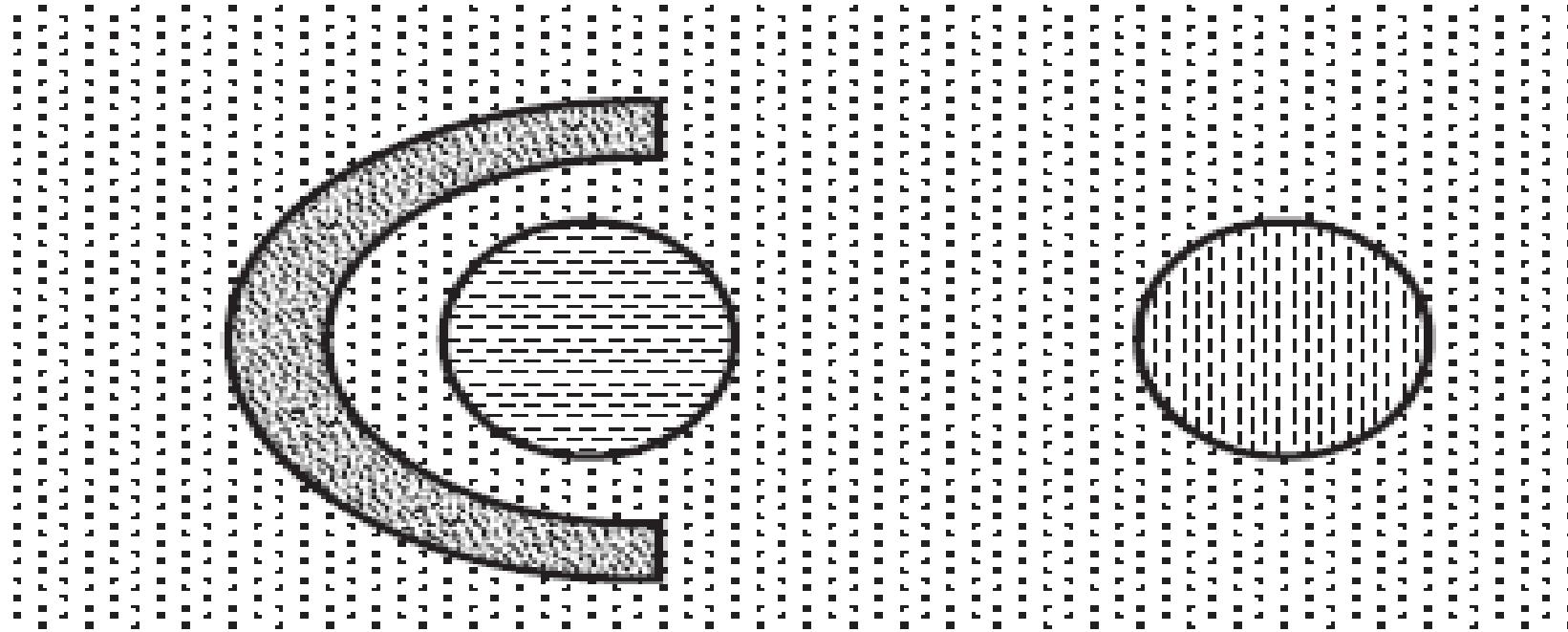
اگر داده ها به صورت نمودار نمایش داده شوند، جایی که گره ها اشیاء هستند و پیوندها نشان دهنده اتصالات بین اشیا هستند، یک خوش می تواند به عنوان یک جزء متصل تعریف شود.



(c) Contiguity-based clusters. Each point is closer to at least one point in its cluster than to any point in another cluster.

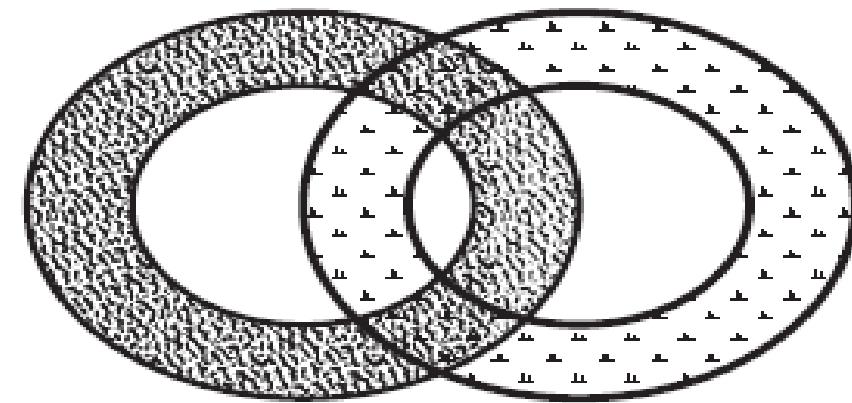
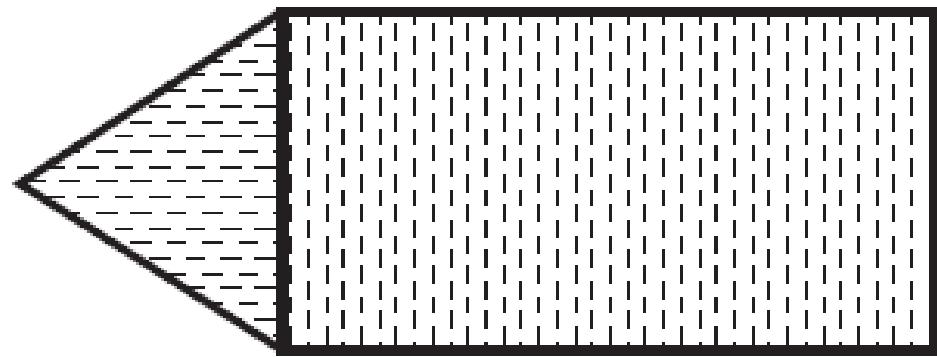
Graph-Based      connected component      contiguity-based cluster

خوش ناحیه متراکمی از اجسام است که توسط ناحیه ای با چگالی کم احاطه شده است.



(d) Density-based clusters. Clusters are regions of high density separated by regions of low density.

می توانیم یک خوش را به عنوان مجموعه ای از اشیاء تعریف کنیم که برخی از ویژگی ها را به اشتراک می گذارند. این تعریف تمام تعاریف قبلی یک خوش را در برمی گیرد..



- (e) Conceptual clusters. Points in a cluster share some general property that derives from the entire set of points. (Points in the intersection of the circles belong to both.)

## Shared-Property (Conceptual Clusters)

# الگوریتم های معرف خوشه بندی

- K-means • این یک تکنیک خوشه بندی **partitional** مبتنی بر prototype-based است که تلاش می کند تعداد خوشه های مشخص شده توسط کاربر ( $k$ ) را پیدا کند که توسط مرکز آنها نشان داده می شوند.
- Agglomerative Hierarchical Clustering • این رویکرد خوشه بندی به مجموعه ای از تکنیک های خوشه بندی نزدیک به هم اشاره دارد، که خوشه بندی سلسله مراتبی را انجام می دهد. بدین گونه که یک خوشه مجزا را در نزدیک ترین خوشه ترکیب می کند و تا زمانی که یک خوشه واحد و فراگیر باقی بماند، یک خوشه بندی سلسله مراتبی ایجاد می کند.
- DBSCAN • یک الگوریتم خوشه بندی مبتنی بر چگالی است، که در آن تعداد خوشه ها به طور خودکار توسط الگوریتم تعیین می شود. نقاط در مناطق کم تراکم به عنوان نویز طبقه بندی و حذف می شوند. بنابراین، یک خوشه بندی کامل ایجاد نمی کند.

# K-means

• خوش بندی بر اساس مرکز (میانگین) هر خوش تعیین می شود

• خوش بندی بر اساس بیشترین فراوانی (نما یا مد) هر خوش تعیین می شود

---

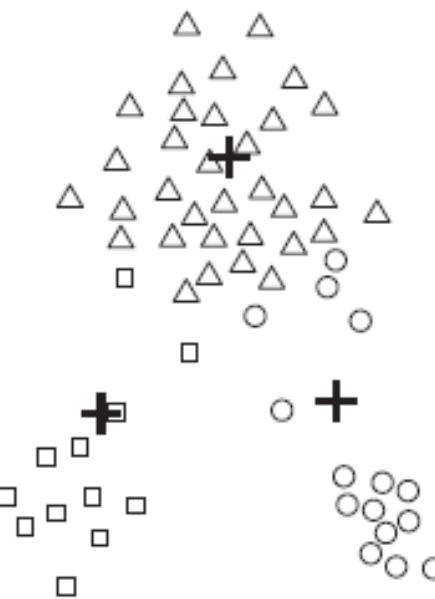
## Algorithm 5.1 Basic K-means algorithm.

---

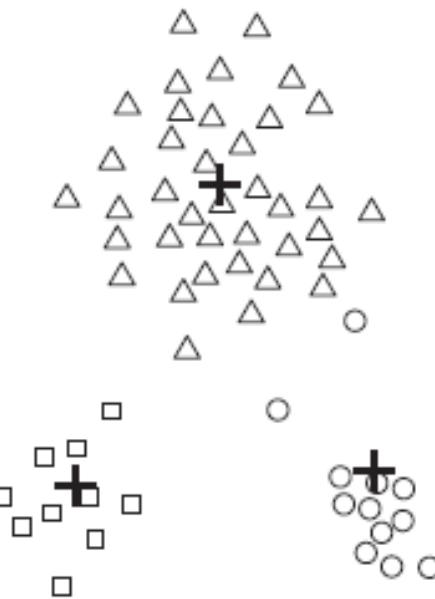
- 1: Select  $K$  points as initial centroids.
  - 2: **repeat**
  - 3:   Form  $K$  clusters by assigning each point to its closest centroid.
  - 4:   Recompute the centroid of each cluster.
  - 5: **until** Centroids do not change.
- 



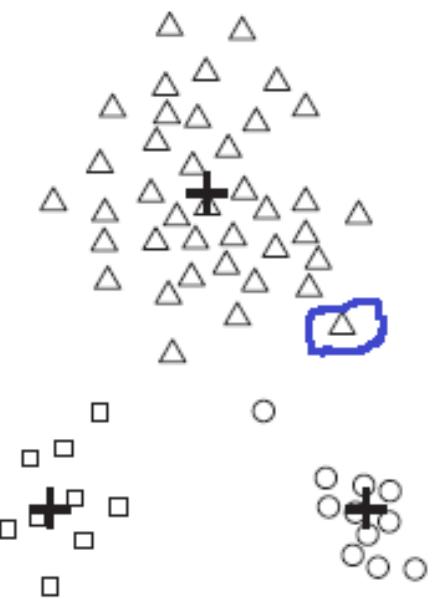
(a) Iteration 1.



(b) Iteration 2.



(c) Iteration 3.



(d) Iteration 4.

# K-means++ الگوریتم

---

## Algorithm 5.2 K-means++ initialization algorithm.

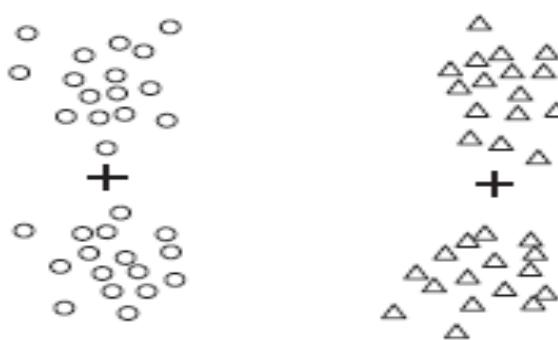
---

- 1: For the first centroid, pick one of the points at random.
  - 2: **for**  $i = 1$  to *number of trials* **do**
  - 3:   Compute the distance,  $d(x)$ , of each point to its closest centroid.
  - 4:   Assign each point a probability proportional to each point's  $d(x)^2$ .
  - 5:   Pick new centroid from the remaining points using the weighted probabilities.
  - 6: **end for**
-

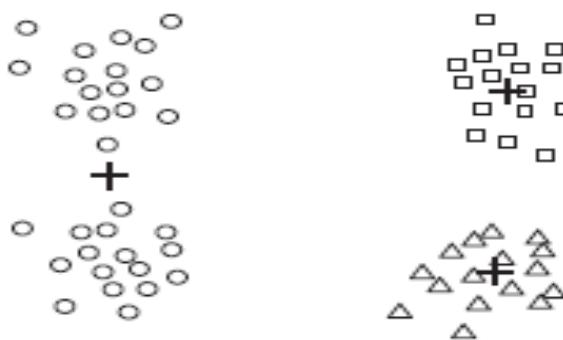
# Bisecting K-means الگوريتم

## Algorithm 5.3 Bisecting K-means algorithm.

```
1: Initialize the list of clusters to contain the cluster consisting of all points.  
2: repeat  
3:   Remove a cluster from the list of clusters.  
4:   {Perform several “trial” bisections of the chosen cluster.}  
5:   for  $i = 1$  to number of trials do  
6:     Bisect the selected cluster using basic K-means.  
7:   end for  
8:   Select the two clusters from the bisection with the lowest total SSE.  
9:   Add these two clusters to the list of clusters.  
10: until The list of clusters contains  $K$  clusters.
```



(a) Iteration 1.



(b) Iteration 2.



(c) Iteration 3.

Figure 5.8. Bisecting K-means on the four clusters example.

# Anomaly Detection تشخیص ناهنجری

- در تشخیص ناهنجری، هدف یافتن داده های است که با الگوهای رفتار عادی مطابقت ندارند.
- اغلب، داده های غیرعادی به عنوان نقاط پرت شناخته می شوند، زیرا در نمودار پراکنش داده ها، آنها از سایر نقاط داده دورتر هستند.
- تشخیص ناهنجری به عنوان تشخیص انحراف نیز شناخته می شود، زیرا داده های غیرعادی دارای مقادیر مشخصه ای هستند که به طور قابل توجهی از مقادیر مشخصه مورد انتظار یا معمولی منحرف می شوند،
- یا به عنوان استخراج استثنایی، زیرا ناهنجری ها به نوعی استثنایی هستند.

## مثال‌های از کاربرد تشخیص ناهنجاری

- **تشخیص تقلب:** رفتار خرید شخصی که کارت اعتباری را می‌دزد دغلب با رفتار مالک اصلی متفاوت است.
- شرکت‌های کارت اعتباری سعی می‌کنند با جستجوی الگوهای خریدی که مشخصه سرقت هستند یا با مشاهده تغییر نسبت به رفتار معمول، سرقت را شناسایی کنند.
- رویکردهای مشابه در بسیاری از حوزه‌ها مانند کشف تقلب در ادعای بیمه و تجارت داخلی مرتبط هستند.
- **تشخیص نفوذ:** حملات به سیستم‌های و شبکه‌های کامپیوتروی امری عادی است. در حالی که برخی از این حملات، مانند حملاتی که برای از کار انداختن یا از کار انداختن رایانه‌ها و شبکه‌ها طراحی شده‌اند، آشکار هستند،
- سایر حملات، مانند حملاتی که برای جمع‌آوری مخفیانه اطلاعات طراحی شده‌اند، دشوار است.
- بسیاری از این نفوذها را فقط می‌توان با نظارت سیستم‌ها و شبکه‌ها برای رفتار غیرعادی تشخیص داد.

## مثال‌های از کاربرد تشخیص ناهنجاری

- اختلالات اکوسیستم: اکوسیستم زمین در چند دهه اخیر به دلایل طبیعی یا انسانی تغییرات سریعی را تجربه کرده است.
- این شامل افزایش تمایل به رویدادهای شدید، مانند امواج گرما، خشکسالی، و سیل است که تأثیر زیادی بر محیط زیست دارد.
- شناسایی چنین رویدادهای شدید از ضبطهای حسگر و تصاویر ماهواره‌ای برای درک منشأ و رفتار آنها و همچنین برای ابداع سیاست‌های سازگاری پایدار مهم است.
- پزشکی و سلامت عمومی: برای یک بیمار خاص، علائم غیرمعمول یا نتایج آزمایش، مانند اسکن غیرعادی MRI، ممکن است نشان دهنده مشکلات بالقوه سلامتی باشد.
- با این حال، غیرعادی بودن یک نتیجه آزمایش خاص ممکن است به بسیاری از ویژگی‌های دیگر بیمار مانند سن، جنس و ساختار ژنتیکی بستگی داشته باشد.
- علاوه بر این، دسته‌بندی یک نتیجه به عنوان غیرعادی، هزینه‌های زیادی را به همراه دارد - آزمایش‌های اضافی غیر ضروری در صورت سالم بودن بیمار و آسیب احتمالی برای بیمار اگر بیماری تشخیص داده نشده و درمان نشود.
- تشخیص شیوع بیماری‌های نو ظهور مانند آنفولانزای H1N1 یا SARS که منجر به نتایج آزمایش غیرعادی و هشداردهنده در یک سری از بیماران می‌شود نیز برای نظارت بر گسترش بیماری‌ها و انجام اقدامات پیشگیرانه مهم است.

## مثال‌های از کاربرد تشخیص ناهنجاری

- **ایمنی هوانوردی:** از آنجایی که هواپیماها سیستم‌های بسیار پیچیده و پویا هستند، به دلیل عوامل مکانیکی، محیطی یا انسانی، مستعد حوادث با عواقب شدید هستند.
- برای نظارت بر وقوع چنین ناهنجاری‌هایی، اکثر هواپیماهای تجاری به تعداد زیادی سنسور برای اندازه‌گیری پارامترهای مختلف پرواز مانند اطلاعات سیستم کنترل، سیستم‌های هوایی و پیشرانه و اقدامات خلبان مجهر هستند.
- شناسایی رویدادهای غیرعادی در این ضبطهای حسگر (به عنوان مثال، یک توالی غیرعادی از اقدامات خلبان یا یک قطعه هواپیما که عملکرد غیرعادی دارد) می‌تواند به جلوگیری از سوانح هواپیما و ارتقای ایمنی هوانوردی کمک کند.

# تعريف ناهنجاری

- یکی از ویژگی مهم تشخیص ناهنجاری، نحوه تعریف آن است.
- از آنجایی که ناهنجاری ها اتفاقات نادری هستند که به طور کامل شناخته نشده اند،
- بسته به نیاز مشکل می توان آنها می توان را به روش های مختلفی تعریف کرد.
- در تعریف زیر سعی شده است، یک تعریف نسبتاً جامع از ناهنجاری ارائه شود
- **تعریف ناهنجاری مشاهدهای** است که با توزیع داده ها برای نمونه های عادی مطابقت ندارد، یعنی وقوع (یا مشاهده) آن داده در حالت نرمال و معمول نامحتمل است.

- در تعریف بالا فرض نمی‌کنیم که توزیع داده‌ها به سادگی بر حسب توزیع‌های آماری شناخته شده، بیان می‌شود. در واقع، دشواری انجام این کار دلیلی است که بسیاری از رویکردهای تشخیص ناهنجاری از رویکردهای غیرآماری استفاده می‌کنند. با این وجود، هدف این رویکردها یافتن داده‌ای است که رایج و معمول نیستند.
- اگر بتوان از نظر مفهومی داده‌ها را با توجه به احتمال وقوع آنها رتبه‌بندی کنیم. هرچه این احتمال کمتر باشد، احتمال ناهنجاری بیشتر خواهد بود.
- دلایل مختلفی می‌تواند برای ایجاد یک ناهنجاری وجود داشته باشد:
  - نویز،
  - این داده از توزیع دیگری پیروی می‌کند
  - یا اینکه شی فقط یک اتفاق نادر را نشان می‌دهد
- همانطور که گفته شد ما علاقه‌ای به ناهنجاری‌های ناشی از نویز نداریم.

## ماهیت داده ها

- ماهیت داده های ورودی نقش کلیدی در تصمیم گیری برای انتخاب روش تشخیص ناهنجاری مناسب دارد.
- اگر داده حاوی یک ویژگی (یک بعدی) واحد باشد، این سؤال که آیا یک شی غیرعادی است یا خیر، به غیرعادی بودن مقدار آن ویژگی بستگی دارد.
- با این حال، اگر داده حاوی بیش از ویژگی باشد، ممکن است غیرعادی بودن ناشی از برخی ویژگی ها باشد در حالی که سایر ویژگی ها عادی هستند.
- علاوه بر این، یک داده ممکن است غیرعادی باشد حتی اگر هیچ یک از ویژگی آن به طور جداگانه غیرعادی نباشد. به عنوان مثال، داشتن افرادی که دو فوت قد دارند (مثلاً کودکان) یا ۱۰۰ پوند وزن دارند، معمول است، اما داشتن یک فردی با قد ۲ فوتی با وزن ۱۰۰ پوندی غیر معمول است.
- بنابراین شناسایی یک ناهنجاری در یک چارچوبهای چند متغیره چالش برانگیز است، به ویژه زمانی که ابعاد داده ها بالا باشد.

- اگر داده ها به صورت یک ماتریس (که در آن هر نمونه داده با استفاده از مجموعه ای از ویژگی ها توصیف می شود) ارائه شود
- برای تشخیص ناهنجاری، اغلب کافی است بدانیم یک نمونه در مقایسه با نمونه های دیگر چقدر متفاوت است.
- از این رو، برخی از روش های تشخیص ناهنجاری بر اساس نمایش متفاوتی از داده های ورودی (معروف به ماتریس مجاورت proximity matrix) عمل می کنند.
- در ماتریس مجاورت که هر عنصر ماتریس نشان دهنده نزدیکی زوجی (شباخت یا عدم شباخت) بین دو نمونه است. توجه داشته باشید که یک ماتریس داده همیشه می تواند با استفاده از یک معیار مجاورت مناسب به یک ماتریس مجاورت تبدیل شود.
- همچنین، یک ماتریس مجاورت (شباخت یا عدم شباخت) را می توان به راحتی با استفاده از یک تبدیل، به یک ماتریس فاصله تبدیل کرد.

- وجود برچسب: برچسب یک داده نشان می دهد که آیا آن داده عادی است یا غیرعادی.
- اگر مجموعه آموزشی با برچسبهایی برای هر نمونه داده داشته باشیم، مشکل تشخیص ناهنجاری به یک مشکل یادگیری نظارت شده (طبقه‌بندی) تبدیل می‌شود.
- در بیشتر کاربردهای عملی مجموعه آموزشی با برچسب‌های دقیق که معرف کلاس‌های عادی و غیرعادی هستند، وجود ندارد.
- لازم به ذکر است به دست آوردن برچسب‌های کلاس غیرعادی به دلیل نادر بودن آنها بسیار چالش برانگیز است.
- بنابراین برای یک متخصص دشوار است که هر نوع ناهنجاری را فهرست بندی کند زیرا ویژگی‌های کلاس ناهنجار اغلب ناشناخته است.
- بنابراین، بیشتر مشکلات تشخیص ناهنجاری ماهیتی **بدون نظارت (unsupervised)** دارند، یعنی داده‌های ورودی هیچ برچسبی ندارند.

- که در غیاب برچسب‌ها، تمایز ناهنجاری‌ها از نمونه‌های عادی با توجه به مجموعه داده‌های ورودی، چالش برانگیز است.
- با این حال، ناهنجاری‌ها معمولاً دارای ویژگی‌هایی هستند که با استفاده از تکنیک‌های می‌توان ناهنجاری‌ها را مشخص نمود.
- دو ویژگی کلیدی عبارتند از:
  - تعداد نسبتاً کوچک: از آنجایی که ناهنجاری‌ها نادر و اکثر داده‌های عادی هستند، برخی از روش‌های تشخیص ناهنجاری بر اساس تعداد مورد انتظار داده‌پرداز در داده‌های ورودی عمل می‌کنند.
  - پراکندگی: ناهنجاری‌ها، بر خلاف داده‌های عادی، اغلب با یکدیگر ارتباطی ندارند و از این رو به صورت پراکنده در فضای صفات توزیع می‌شوند. در واقع، عملکرد موفقیت‌آمیز اکثر روش‌های تشخیص ناهنجاری به این بستگی دارد که ناهنجاری‌ها کاملاً خوشه‌بندی نشده باشند. با این حال، برخی از روش‌های تشخیص ناهنجاری به‌طور خاص برای یافتن ناهنجاری‌های خوشه‌ای طراحی شده‌اند.

# روش های تشخیص ناهمجاري

- Model-based vs. Model-free
- Global vs. Local Perspective
- Label vs. Score

# شش نوع رویکرد تشخیص ناهنجاری

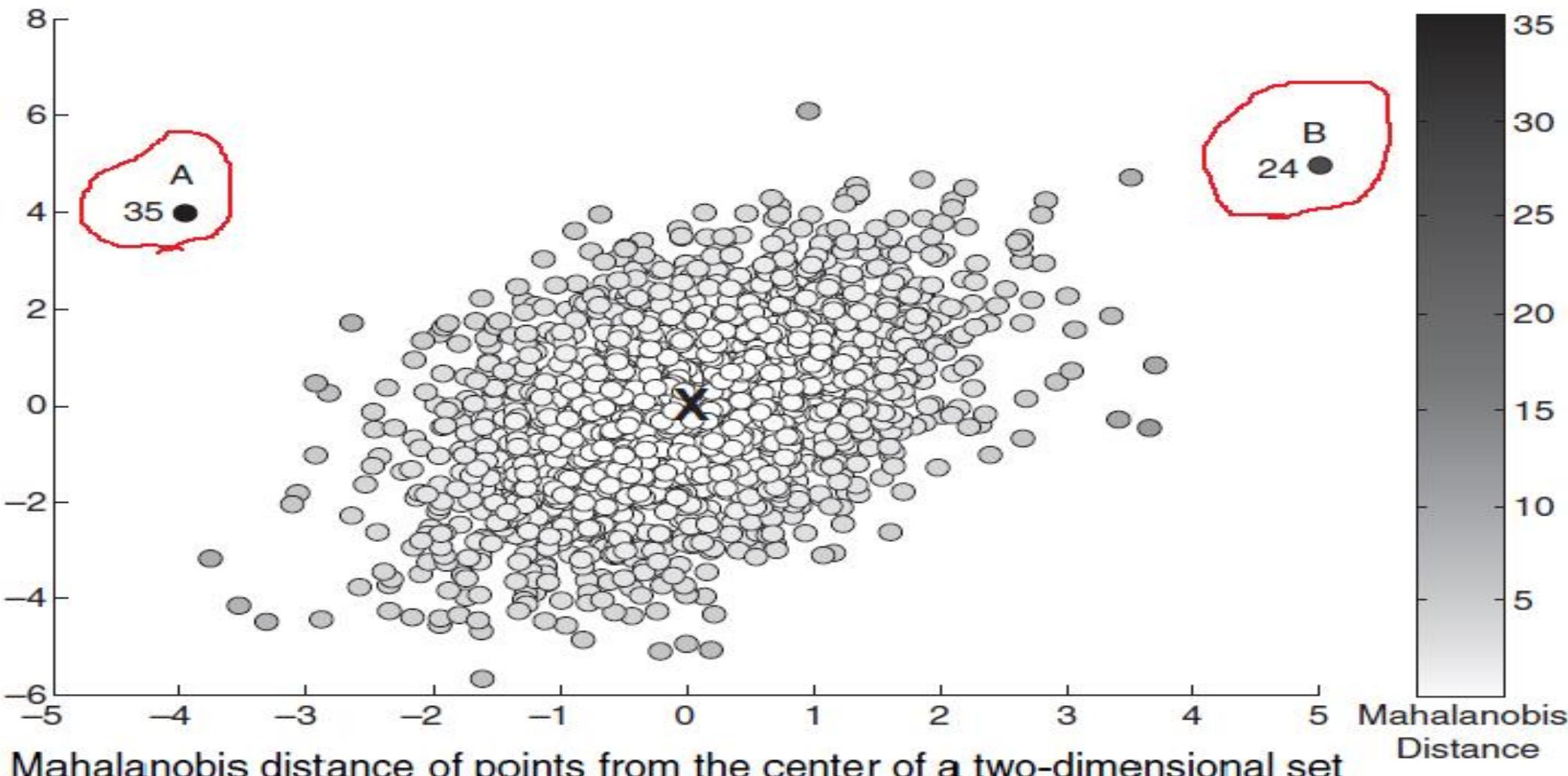
1. Statistical Approaches
2. Proximity-based Approaches
3. Clustering-based Approaches
4. Reconstruction-based Approaches
5. One-class classification approaches
6. Information Theoretic Approaches

# Statistical Approaches

- رویکردهای آماری از توزیع‌های احتمالاتی (نظیر توزیع گاووسی) برای مدل‌سازی کلاس نرمال استفاده می‌کنند. یکی از ویژگی‌های کلیدی چنین توزیع‌هایی این است که آنها یک مقدار احتمال را به هر نمونه داده مرتبط می‌کنند، که نشان می‌دهد چقدر احتمال دارد که نمونه از توزیع تولید شده باشد. سپس ناهنجاری‌ها به عنوان نمونه‌هایی شناسایی می‌شوند که بعید است از توزیع احتمال کلاس نرمال ایجاد شده باشند.
- دو نوع مدل پارامتری و ناپارامتریک در این رویکرد وجود دارد.
- در مدل‌های پارامتریک از خانواده توزیع‌های شناخته شده، استفاده می‌کنند که نیاز به برآورد پارامترها این توزیع‌ها از داده‌ها دارند،
- مدل‌های ناپارامتریک انعطاف‌پذیرتر هستند و توزیع کلاس نرمال را مستقیماً از داده‌های موجود یاد می‌گیرند.

# Parametric Approach: Mahalanobis distance

$$\text{Mahalanobis}(\mathbf{x}, \bar{\mathbf{x}}) = (\mathbf{x} - \bar{\mathbf{x}})\mathbf{S}^{-1}(\mathbf{x} - \bar{\mathbf{x}})^T.$$



# NonParametric Approach:

- یک رویکرد ناپارامتریک ساده، برای مدل‌سازی کلاس عادی، ساختن هیستوگرام داده‌های عادی است.
- سپس می‌توانیم بررسی کنیم که آیا یک نمونه آزمایشی جدید در هر یک از ستون‌های هیستوگرام می‌افتد یا خیر. اگر در هیچ یک از ستونها نیفتد، می‌توانیم آن را به عنوان یک ناهنجاری تشخیص دهیم.
- در غیر این صورت، می‌توانیم از عکس فراوانی ستونی که در آن قرار می‌گیرد به عنوان نمره ناهنجاری آن استفاده کنیم.
- این رویکرد به عنوان رویکرد مبتنی بر فراوانی یا بر اساس شمارش برای تشخیص ناهنجاری شناخته می‌شود.

# Modeling Normal and Anomalous Classes

- The basic idea of this approach is to assume that data is generated with probability  $\lambda$  from the **anomalous class**, which has **uniform distribution,  $p_A$** , and with probability  $1 - \lambda$  and from the **normal class**, which has the **distribution,  $PM(\vartheta)$** , where  $\vartheta$  represents the parameters of the distribution.
- Let  $M_t$  and  $A_t$  be the set of **normal** and **anomalous** objects, respectively, at **iteration t**.

$$\log \mathcal{L}_t(D) = |M_t| \log(1 - \lambda) + \sum_{x_i \in M_t} \log P_M(x_i, \theta_t) + |A_t| \log \lambda + \sum_{x_i \in A_t} \log P_A(x_i)$$

# Modeling Normal and Anomalous Classes

---

## Algorithm 9.1 Likelihood-based outlier detection.

---

- 1: Initialization: At time  $t = 0$ , let  $M_t$  contain all the objects, while  $A_t$  is empty.
  - 2: **for** each object  $\mathbf{x}$  that belongs to  $M_t$  **do**
  - 3:   Move  $\mathbf{x}$  from  $M_t$  to  $A_t$  to produce the new data sets  $A_{t+1}$  and  $M_{t+1}$ .
  - 4:   Compute the new log-likelihood of  $D$ ,  $\log\mathcal{L}_{t+1}(D)$
  - 5:   Compute the difference,  $\Delta = \log\mathcal{L}_{t+1}(D) - \log\mathcal{L}_t(D)$
  - 6:   **if**  $\Delta > c$ , where  $c$  is some threshold **then**
  - 7:     Classify  $\mathbf{x}$  as an anomaly.
  - 8:     Increment  $t$  by one and use  $M_{t+1}$  and  $A_{t+1}$  in the next iteration.
  - 9:   **end if**
  - 10: **end for**
-

# Proximity-based Approaches

- روش‌های مبتنی بر مجاورت، ناهنجاری‌ها را به عنوان نمونه‌هایی شناسایی می‌کنند که بیشترین فاصله را از سایر داده‌ها دارند.
- زیرا در این رویکرد فرض می‌کنیم که نمونه‌های عادی به هم مرتبط هستند و بنابراین نزدیک به هم ظاهر می‌شوند، در حالی که ناهنجاری‌ها با نمونه‌های دیگر متفاوت هستند و از این رو از نمونه‌های دیگر نسبتاً دور هستند.

# Distance-based Anomaly Score

- $dist(x, k)$ =distance x to its  $k$ th nearest neighbor

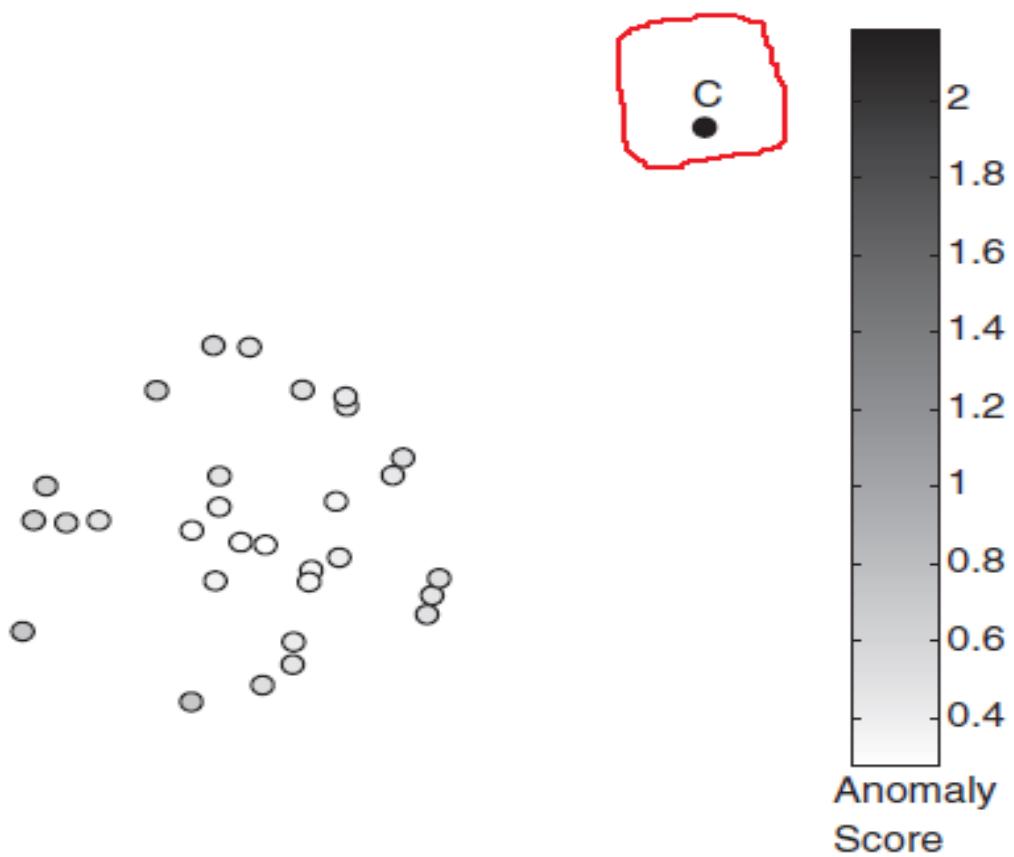


Figure 9.4. Anomaly score based on the distance to fifth nearest neighbor.

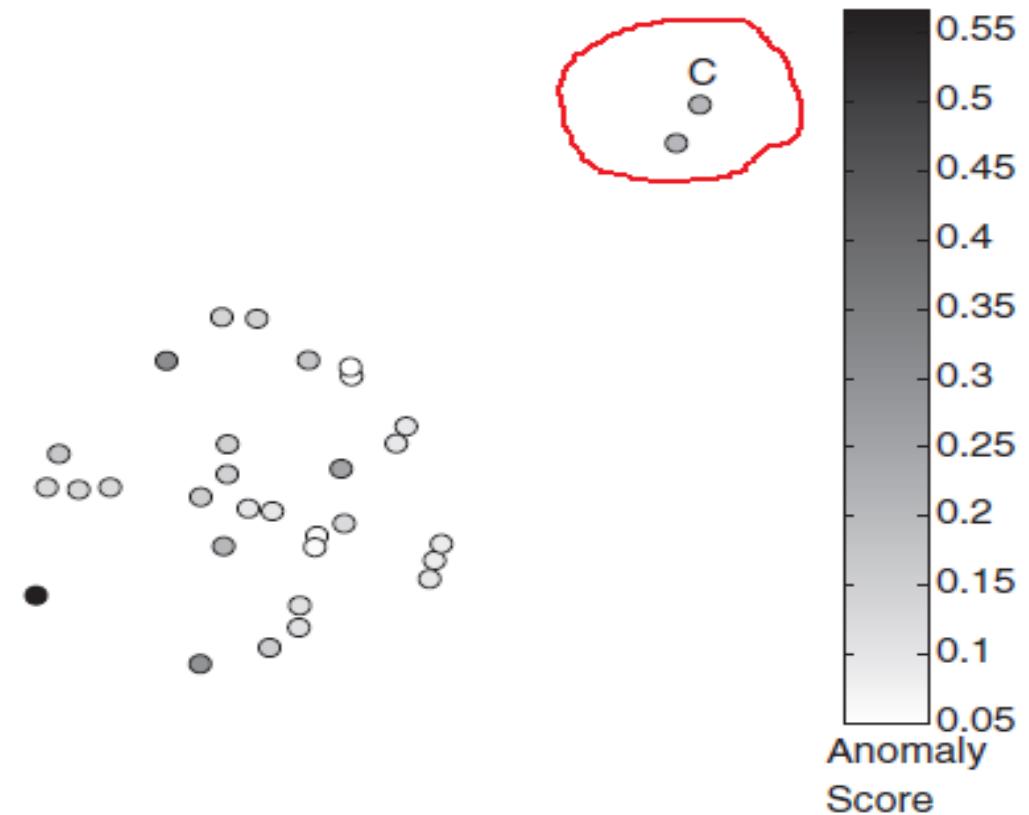


Figure 9.5. Anomaly score based on the distance to the first nearest neighbor.

# Density-based Anomaly Score

- چگالی اطراف یک نمونه را می توان به صورت  $(d/n)/V$  تعریف کرد،
- که در آن  $n$  تعداد نمونه ها در فاصله مشخص  $d$  از نمونه است و  $V$  حجم همسایگی است.
- بنابراین، یک ناهنجاری تعداد نمونه های کمتری در فاصله  $d$  نسبت به یک نمونه معمولی خواهد داشت.

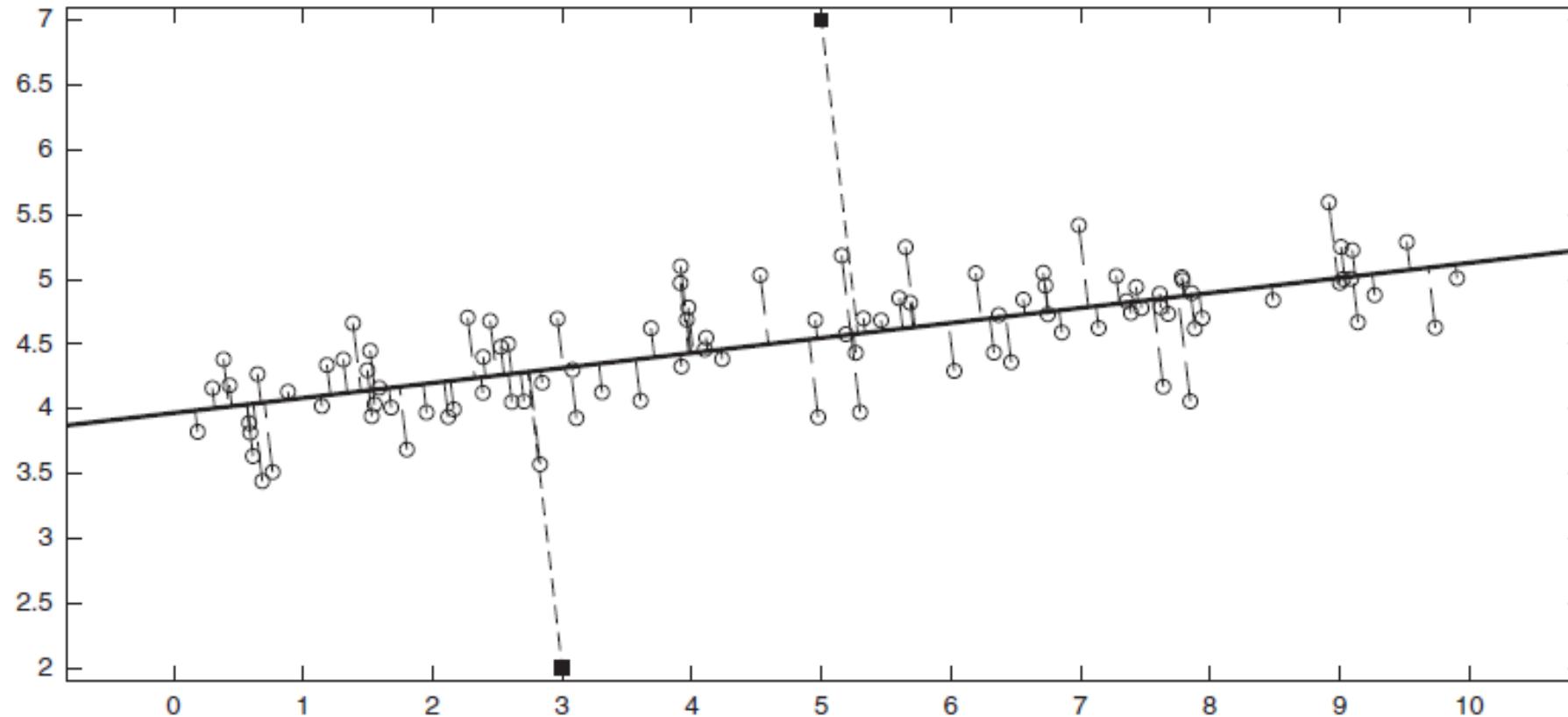
# Clustering-based Approaches

- روش‌های مبتنی بر خوشه‌بندی برای تشخیص ناهنجاری از خوشه‌ها برای نمایش کلاس عادی استفاده می‌کنند.
- این رویکرد بر اساس این فرض استوار است که نمونه‌های عادی نزدیک به یکدیگر ظاهر می‌شوند.
- بنا بر این می‌توانند در خوشه‌ها گروه‌بندی شوند.
- سپس ناهنجاری‌ها به عنوان نمونه‌هایی شناسایی می‌شوند که به خوبی در خوشه‌بندی کلاس عادی قرار نمی‌گیرند، یا در خوشه‌های کوچکی ظاهر می‌شوند که از خوشه‌های کلاس نرمال فاصله زیادی دارند.

# Reconstruction-based Approaches

- تکنیک‌های مبتنی بر بازسازی بر این فرض استوار است که کلاس عادی را می‌توان با ابعاد پایین‌تر از فضای اصلی خلاصه نمود.
- به عبارت دیگر، الگوهایی در توزیع کلاس عادی وجود دارد که می‌توان با استفاده از آنها کاهش بعد، انجام داد.
- *reconstruction of  $\mathbf{x}$  as  $\hat{\mathbf{x}}$*

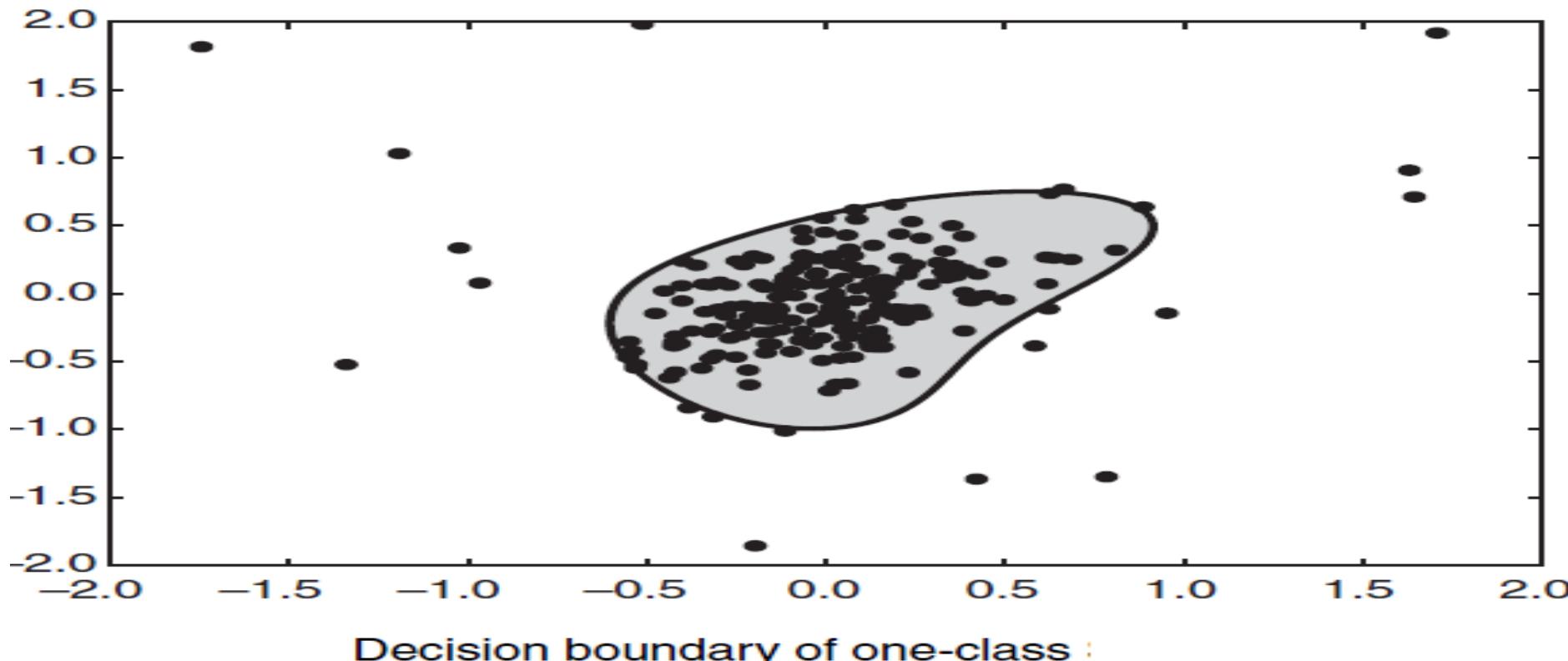
$$\text{Reconstruction Error}(\mathbf{x}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2$$



**Figure 9.11.** Reconstruction of a two-dimensional data using a single principal component (shown as solid black line).

# One-class classification approaches

- رویکردهای طبقه‌بندی تک کلاسی، مرز تصمیم‌گیری را در فضای صفت می‌آموزند
- بر اساس این رویکرد تمام داده‌های عادی را در یک طرف مرز محصور می‌شوند.
- شکل زیر نمونه‌ای از مرز تصمیم را نشان می‌دهد، که در آن نقاط متعلق به یک طرف مرز (سایه دار) به کلاس عادی تعلق دارند.



# Information Theoretic Approaches

- در این رویکرد فرض می شود که کلاس عادی را می توان با استفاده از کدگزارهای فشرده ارائه نمود.
- همچنین به جای یادگیری صریح چنین کدگزارها، تمرکز بر روی مقدار اطلاعات مورد نیاز برای تعیین چنین کدگزارها است.
- اگر کلاس معمولی ساختار یا الگوی خاصی را نشان دهد، می توان انتظار داشت که آن را با استفاده از تعداد کمی اطلاعات کدگزاری کنیم.
- پس می توان ناهنجاری ها را به عنوان نمونه هایی شناسایی کرد که بی نظمی هایی را در داده ها ایجاد می کنند، که محتوای کلی اطلاعات مجموعه داده را افزایش می دهد.

$$Gain(x) = Info(D) - Info(D \setminus x).$$

## Example

**Table 9.2.** Survey data of weight and height of 100 participants.

weight	height	Frequency
low	low	20
low	medium	15
medium	medium	40
high	high	20
high	low	5

entropy of 2.08.

By eliminating these 5 instances, gain of  $2.08 - 1.89 = 0.19$ .

---