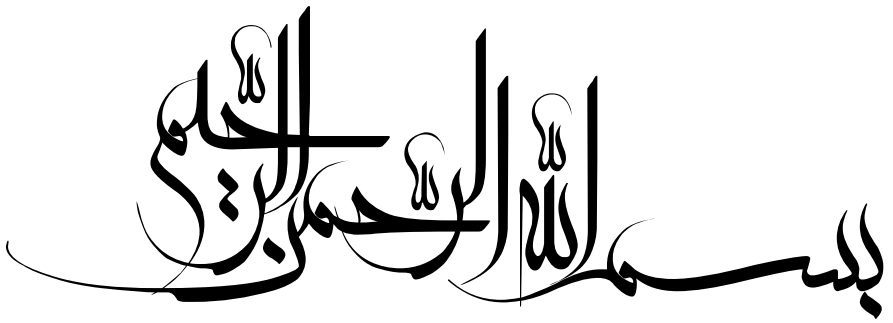


موضوع : یافتن کلمات فارسی در پایگاه داده

نام استاد : مهندس ایزدپرست

نام دانشجو : مبین شاطریان

دانشگاه تربیت معلم تهران

سال ۸۹-۹۰

۱) پیش گفتار :

در این مقاله سعی شده است نخست بر مشکلات ذخیره کلمات فارسی که ناشی از داشتن چندین یونی کد متفاوت برای حروف است صحبت شده است و در ادامه درباره ی خود مشکلات فارسی اعم از چند شکلی بودن کلمات و داشتن چندین رسم الخط برای یک کلمه بحث شده است . در ادامه به معرفی داده کاوی پرداخته و در آخر نیز الگوریتم های جستجو را بررسی کرده‌ایم و در نهایت به یک جمع‌بندی کلی درباره ی چگونگی یافتن کلمات فارسی پرداخته ایم . تا در نهایت بتوانیم راه حلی مناسب برای این امر ارائه دهیم.

بیشتر این مقاله به بررسی مشکلات موجود پرداخته است تا بتواند چراغی برای حل مشکلات فارسی نویسان در محیط مجازی باشد و در آینده شاهد مشکلات کنونی نباشیم .

۱-۱) مقدمه :

از چند سال پیش در کشورمان، استفاده از کامپیوتر با سرعت سرسام‌آوری جای خود را در تمامی عرصه‌ها باز کرد و سیل کامپیوترهای شخصی و تجهیزات جانبی آنها به سوی کشور سرازیر شد. اما بایستی اعتراف کرد که با وجود این که سرعت سوق به سوی تکنولوژی دیجیتال در ایران روند خوبی را طی نموده، اما در زمینه ارائه اطلاعات و پردازش آن به زبان فارسی وقفه‌ای در این میان ایجاد گردید. یکی از عوامل موثر در این ناهماهنگی، نبود الگویی واحد برای ذخیره و پردازش و نمایش اطلاعات بر روی رسانه‌های جدید اطلاع‌رسانی همچون کامپیوتر در سطح ملی است .

نرم‌افزارهای متفاوت، با فرمت‌های مختلف، کدهای فارسی گوناگون و ... درحال استفاده‌اند و روزانه میزان قابل توجهی از اطلاعات را در خود جای می‌دهند. اگر از آن دسته از مراکزی که به دلیل عدم آگاهی کافی اطلاعات را به صورت ناقص جمع‌آوری و وارد می‌کنند (که حدود ۸۰ درصد جامعه مورد نظر را تشکیل می‌دهند) بگذریم به تفرق، اختلاف و اعمال سلیقه‌های مختلف در سایر مراکز خواهیم رسید که برای نمونه به اختلاف در مورد کدنویسه‌های به کار رفته برای حروف فارسی روی کامپیوتر می‌توان اشاره کرد.

در مورد مراکزی که به هرحال مشغول سرمایه‌گذاری در بخش ورود، پردازش و نمایش اطلاعات هستند مسئله به نوع دیگری خود را نشان خواهد داد. این گونه مراکز تا زمانی که پای خود را از محدوده مرکز خود فراتر نگذاشته‌اند مشکلی نخواهند داشت، ولی به محض آنکه بخواهند با مراکز اطلاعاتی و تحقیقاتی دیگر ارتباط برقرار کرده یا به مبادله اطلاعات با این مرکز بپردازند متوجه خواهند شد که سال‌ها سرمایه‌های خود را بر باد داده‌اند.

همین مشکل در سطح ملی برای ایجاد یک مرکز اطلاعات ملی رخ خواهد نمود. زمانی این مشکل ملی بیشتر نمود پیدا می‌کند که بحث شبکه جهانی اینترنت نیز به میان آید.

اینترنت به عنوان کلیدی برای ارتباط با دیگر مراکز اطلاعاتی - به علت در دسترس بودن آسان و همچنین حجم عظیم اطلاعات موجود در آن- یکی از مهم‌ترین موضوعاتی خواهد بود که به علت عدم وجود یک سیستم جهانی برای ذخیره، بازیابی، پردازش و نمایش اطلاعات و به طور کلی مبادله اطلاعات که جنبه‌های ملی نیز داشته باشد، دارای نقاط ضعفی است که ما را از بهره‌برداری مناسب در جهت منافعمان باز می‌دارد.

از زمانی که اولین گزارش «زبان فارسی و کامپیوتر» در سال ۱۳۵۶ در دانشکده ریاضی و کامپیوتر دانشگاه صنعتی شریف ارائه شد، تا امروز

که شبکه اینترنت چهره دیگری به اطلاع‌رسانی داده است، مدت زیادی می‌گذرد. امروزه دیگر محدودیت‌های سخت‌افزاری یا نرم‌افزاری نمی‌تواند مانع پیاده‌سازی یک سیستم ذخیره‌سازی، نمایش، و تبادل اطلاعات چندزبانه گردد. امروزه مؤسسات بزرگ استانداردسازی چون ایزو (ISO) و W۳ Consortium نیز، در استانداردهایشان مشکلات و مسائل مربوط به جهانی‌سازی را در نظر می‌گیرند تا امر تبادل اطلاعات چند زبانه را تسهیل نمایند. اما به نظر می‌رسد که به دلیل عدم حضور ایرانیان و فارسی‌زبان‌ها در این روند، زبان فارسی قدری غریب مانده و کمتر به آن توجه شده است. به عنوان مثال، هنوز در بین صدها مجموعه‌نویسه (Character Set) ثبت شده در اینترنت توسط یانا (Internet Assigned Number Authority)، تنها یک مجموعه‌نویسه ثبت شده متعلق به زبان فارسی است که آن هم کد پیج اختصاصی شرکت آیبیام است.حتی در مورد استاندارد کلی تبادل اطلاعات نیز قالبی که مورد توافق همه باشد وجود ندارد. سه قالب موجود، ایران سیستم، استاندارد ۲۹۰۰ و استاندارد ۳۳۴۲ ، هر یک ایراداتی دارند که سبب شده است شرکت‌ها و مؤسسات داخلی به جدول‌های خاص خود روی آورند تا بتوانند نیازهای خود را تا حدی رفع سازند.

اخیراً راه‌حل‌هایی در هر یک از مسائل خاص مربوط به تبادل اطلاعات برای بین‌المللی‌سازی در نظر گرفته شده است که با وجود این که این موارد کامل‌تر از جداولی است که در ایران برای حل مشکلات تبادل اطلاعات زبان فارسی ایجاد گردیده، ولی به خاطر عدم وجود مراجع موثق در مورد خط و زبان فارسی برای استانداردگذاران، مسائل خاص این زبان یا در نظر گرفته نشده و یا به شکل ناقص منظور شده است. خوشبختانه بسیاری از این استانداردها امکان گسترش بعدی را در نظر گرفته‌اند که روند تصحیت را تسهیل می‌کند.

۲) یونی‌کد چیست؟

یونی‌کد به هر نویسه یک اعداد یکتا اختصاص می‌دهد ، مستقل از محیط ، مستقل از برنامه و مستقل از زبان.

در علم رایانه، هر نویسه یونی‌کد، ۲ بایت جا می‌گیرد، که یک بایت آن مشخص کننده زبان و بایت دیگر مشخص کننده ی خود نویسه است. برای نویسه‌های عادی زبان انگلیسی بایت مشخص کننده ی زبان برابر مقدار تهی (کاراکتر کد صفر) است. (wikipedia.org)

اصولاً کامپیوترها فقط با عددها کار می‌کنند و حروف و نویسه‌های دیگر را با تخصیص عددی به هر یک از آنها ذخیره می‌کنند. تا قبل از اختراع یونی‌کد، صدها سیستم کُدگذاری مختلف برای تخصیص این اعداد وجود داشت. نویسه‌های هیچ کُدگذاری‌ای به‌تنهایی کافی نبود:

یعنی مثلا تصور کنید یک شرکتی برای نمایش یک نویسه مثل a از یک کد و شرکتی دیگر از کدی دیگر استفاده می کرد و این موضوع در مورد تمام زبانها، از جمله زبان فارسی هم صادق بود. و یا اگر استانداردهایی هم برای این منظور طراحی شده بود در حد قابل قبولی نبود، مثلاً اتحادیه اروپا به چندین سیستم کُدگذاری مختلف نیاز داشت تا همه زبان‌های آن اتحادیه را در بر بگیرد. حتی برای زبانی مثل انگلیسی نیز هیچ کُدگذاریی به‌تنهایی برای همه حروف، علائم نقطه‌گذاری و نمادهای فنی متداول، کافی نبود.

از جمله استانداردهای بین‌المللی که کامل‌تر از بقیه استانداردهای موجود به رفع نیازهای مربوط به تبادل اطلاعات چندزبانه پرداخته‌است، می‌توان به استاندارد **یونی‌کد** اشاره کرد.

این استاندارد، تقریباً توسط تمامی شرکت‌های بین‌المللی کامپیوتری، مانند آیبیام، مایکروسافت، و سان، و نیز موسسات ملی استاندارد در کشورهای مختلف جهان برای تبادل اطلاعات چندزبانه مورد توافق قرار گرفته است و سرعت رشد بسیار زیادی نیز در میان کاربران دارد. همین‌طور، در حال حاضر کلیه استانداردهای جدیدی که برای شبکه اینترنت طراحی می‌شوند، این دو استاندارد را به‌عنوان کدپیج پیش‌فرض می‌پذیرند که استاندارد XML و زبان جاوا از آن جمله‌اند.

به زبان ساده می‌توان گفت که یونی‌کد روشی برای تبدیل متون به رشته‌های عددی قابل ذخیره در کامپیوتر است. روش‌های گوناگونی برای

این کار وجود دارند، ولی مزیت یونی‌کد نسبت به آنها، این است که یک روش کامل جهانی است؛ به این معنی که حروف همه زبان‌های دنیا و تمامی علائم مورد استفاده همه مردم جهان در آن آمده‌اند و همچنین در همه‌جا قابل نمایش است و نیاز به امکانات خاصی ندارد. البته یونی‌کد هنوز جوان است ولی امروزه بسیاری نرم‌افزارهای رایج در جهان (از جمله همه مرورگرهای جدید اینترنت) آن را پشتیبانی می‌کنند.

از مهم‌ترین مزایایی که یونی‌کد برای زبان فارسی دارد (مثل بسیاری زبان‌های دیگر) می‌توان موارد زیر را نام برد:

۱. در نسخه استاندارد هر نرم‌افزاری که از این استاندارد پشتیبانی کند، می‌توان فارسی نوشت یا متون فارسی را خواند. بدین ترتیب دیگر نیازی به تأمین نسخه‌های خاص فارسی یا عربی نیست.

۲. برای خواندن متون فارسی که توسط شرکت خاصی نوشته شده‌اند، نیازی به داشتن فونت خاص آن شرکت نداریم و هر متن فارسی که با استاندارد یونی‌کد، کدگذاری شده باشد، با هر فونت یونی‌کدی قابل مشاهده است.

۳. امکان استفاده هم زمان از زبان‌های فارسی و انگلیسی را تأمین می‌کند.

۴. بدون استفاده از فونت‌های خاص امکان استفاده از علائم خاص را فراهم می‌کند.

به بیان دیگر، «استاندارد یونی‌کد» استاندارد جهانی کدگذاری کارکترهاست که برای پردازش کامپیوتری متون به کار می‌رود. این استاندارد همان کاراکترها و کدهای استاندارد ISO/IEC ۱۰۶۴۶ را داراست و کاملا با آن سازگار است. پس در واقع هر پیاده‌سازی سازگار با یونی‌کد، با ISO/IEC ۱۰۶۴۶ نیز سازگار است .

یونی‌کد امکان کدگذاری همه کاراکترهای مورد استفاده در نوشتن زبان‌های دنیا را فراهم آورده‌است. این استاندارد از کدگذاری ۱۶بیتی استفاده می‌کند که برای بیش از ۶۵۰۰۰ نویسه (کاراکتر) جا فراهم می‌کند. اگر چه ۶۵۰۰۰ نویسه برای کدگذاری اکثر نویسه‌هایی که در زبان‌های مهم دنیا استفاده می‌شود کافی است، با این حال یونی‌کد شیوه‌گسترشی به‌نام UTF-۱۶ فراهم‌کرده‌است که امکان اضافه‌کردن حدود یک میلیون نویسه دیگر را نیز می‌دهد. این دامنه برای کلیه نویسه‌های عالم، از جمله پوشش کامل همه خط‌های باستانی (همچون خط میخی) نیز کافی است. (رجوع شود به جدول یونی کد خط میخی پارسی)

یونی‌کد برای کلیه نویسه های مورد استفاده در زبان‌های عمده دنیا کد تعیین‌کرده‌است. به علت گسترده‌بودن فضای تخصیص نویسه، این استاندارد بسیاری از نمادهای لازم برای حروف‌چینی را نیز در بر گرفته‌است. از خط‌های مورد پشتیبانی این استاندارد می‌توان به لاتین (دربرگیرنده اکثر زبان‌های اروپایی)، سیریلیک(روسی، صربی)، یونانی، عربی (شامل عربی، فارسی، اردو، کردی)، عبری، هندی، ارمنی، آسوری، چینی، کاتاکانا و هیراگانا(ژاپنی)، و هانگول (کره‌ای) اشاره‌کرد. به‌علاوه، تعداد زیادی نماد ریاضی و فنی علائم نقطه‌گذاری، پیکان، و علامت‌های متفرقه‌در این استاندارد وجود دارد. این استاندارد برای علامت‌های ترکیب‌شونده یا اعراب‌ها نیز کدهایی در نظر گرفته‌است که از جمله آنها علامت‌هایی چون «◌ٓ» (مد)هستند که در ترکیبب حروف پایه را می‌سازند.

به طور کلی، بعضی از مشخصات یونی‌کد به شرح زیر است:

۱. نویسه‌های شانزده‌بیتی

۲. یکی‌سازی (اختصاص یک کد به نویسه‌های مشترک در چند زبان مختلف)

۳. نویسه ، نه شکل (یک «ع»، و نه چهارتا: ع اول « عـ » ، ع وسط « ـعـ » ، ع چسبان آخر « ـع »، «ع» تنها)

۴. بار معنایی (حرف‌بودن، مقدار عددی، ...)

در استاندارد یونی‌کد، نویسه‌های فارسی در بلوک مربوط به‌خط عربی قرار دارند. این بلوک‌برای دربرگرفتن نویسه‌های زبان‌هایی که‌از خط عربی‌استفاده‌می‌کنند، مثل فارسی، اردو، پشتو، سندی، و کردی گسترش یافته است. این بلوک نشانه‌های قرآنی از قبیل نشانه‌های سجده و پایان آیه، و علائم وقف را نیز در بردارد.

در یونی‌کد با وجود یکی‌سازی کدهای حروف مشترک، برای حروف فارسی که بار معنایی یا نمایشی متفاوت با حروف عربی دارند، نویسه‌های جداگانه درنظر گرفته‌شده‌است. یعنی کلیه حروف خاص فارسی (پ، چ، ژ، گ) و نیز «ک» و «ی» فارسی که‌با حرف مشابه‌در عربی‌تفاوت نمایشی دارند، مکان‌جداگانه‌ای به‌خود اختصاص داده‌اند. کلیه اعراب‌های متداول حضور دارند و میان شکل‌فارسی/اردو و عربی ارقام نیز به‌علت شکل و رفتار متفاوت، تفاوت‌هایی منظور گشته است.

از طرف دیگر، علائم نقطه‌گذاری چون نقطه و فاصله که‌شکلی کسانی‌در خط‌های لاتین و عربی دارند، کد یکسان دارند. علائمی‌چون پرانتز نیز، بسته به جهت متن، آینه‌ای می‌شوند، به طور مثال، نویسه ۰۰۲۸ نماینده «پرانتز باز» است، و نه«پرانتز سمت‌چپ». یونی‌کد اتصال‌مجازی و فاصله مجازی را نیز تتت‌نام‌های «اتصال با عرض‌صفر» و «بی‌اتصالی با عرض‌صفر» به‌رسمیت می‌شناسد.

بدین ترتیب ملاحظه می‌شود که برای حل مشکلات موجود، و نیز رفتن به سوی یک استاندارد مقبول و همه‌جانبه، استاندارد یونی‌کد، روشی مناسب به نظر می‌رسد.

۲-۱) کنسرسیوم یونی‌کد :

کنسرسیوم یونی‌کد سازمان غیرانتفاعی‌ای است که برای بهبود، گسترش، و ترویج استفاده از استاندارد یونی‌کد تأسیس شده است، استانداردی که شیوه ی بازنمایی متون را در محصولات نرم‌افزاری و استانداردهای امروزی مشخص می‌کند. اعضای این کنسرسیوم طیف گسترده‌ای را از شرکت‌ها و سازمان‌های فعال در صنعت پردازش اطلاعات، در بر می‌گیرند. پشتیبانی مالی این کنسرسیوم صرفاً از طریق حق عضویت اعضا است. عضویت در کنسرسیوم یونی‌کد برای سازمان‌ها و افراد هر جای دنیا که استاندارد یونی‌کد را پشتیبانی کنند و بخواهند در گسترش و پیاده‌سازی آن کمک کنند، آزاد است.

۲-۲) یونی‌کد در ویندوز :

در ویندوز ای‌پی‌آی توابع یونی‌کد با پسوند W می‌آیند. (مثال: CreateWindowExW) پسوند W حرف اول عبارت wide character است که در زبان‌های برنامه نویسی انواع داده‌ای که یونی‌کد را پشتیبانی می‌کنند اسامی مشابهی مانند WCHAR دارند و گاهی به آن نویسه چندبایتی (به انگلیسی: *multibyte character*) نیز گفته می‌شود.

۲-۳) ایجاد کنندگان یونی کد فارسی و اصلاح آن :

روزبه پورنادر (دانشگاه شریف)، بهداد اسفهبد (دانشگاه شریف)، هومن پورناصح (مایکروسافت) و کمیل بهمن‌پور(چرتکه) از جمله ایرانیانی هستند که در توسعه ی استاندارد یونی‌کد بخصوص در زبان‌های راست به چپ فارسی، عربی و عبری به کنسرسیوم یونی‌کد، تولیدکنندگان سیستم‌عامل، و شرکت‌های عظیم وب کمک شایانی کرده‌اند. (wikipedia)

۲-۴) صفحه‌کلید استاندارد فارسی :

چینش صفحه‌کلید استاندارد فارسی در ویندوز، با چینش استاندارد ملی ایران، متفاوت است. در حالی که در توزیع‌های گنو/لینوکس این استاندارد صفحه کلید به صورت پیش‌فرض رعایت گردیده است. استاندارد ملی شماره ۲۹۰۱ ایران(تجدید نظرشده) به صورت زیر است: (سایت ویکی پدیا)

حالت عادی

حالت با تبدیل

این استاندارد بعد از انتشار استاندارد ملی ایران ۹۱۴۷ باطل اعلام شد. و این استاندار همچنان در سیستم عامل لینوکس رعایت می‌شود .

برای مشاهده ی یونی کد فارسی و عربی و همچنین یونی کد خط میخی پارسی به آخر مقاله مراجعه شود.

۲-۵) نتیجه ی فصل ۲ :

تا اینجای کار به بررسی یونی کد پرداختیم و متوجه شدیم برای اینکه بتوانیم کلمات فارسی را به سهولت ذخیره و بازیابی کنیم باید از یونی کد یکسان برای تمامی حروف فارسی استفاده شود . یعنی اگر قرار است حروفی با یونی کد عربی وارد پایگاه داده ی ما شود با ابتدا آن را به یونی کد فارسی (با استفاده از جدول یونی کد فارسی – عربی) تبدیل کنیم یا به عبارتی کلمات ورودی را یکسان سازی کنیم .

این تبدیل یونی کد بسیار حائز اهمیت است زیرا در صورت عدم این کار ما چندین جواب متفاوت برای یک یافتن یک کلمه ی مورد نظر داریم . مثلاً حرف «ی» که یونی کد فارسی آن با یونی کد عربی آن متفاوت است و تفاوت آن در نقطه ی زیر «ی» در عربی است که این کار باعث به دست آوردن نتایج کم و ناکافی می‌باشد .

راه حل دیگر برای حل این معذل تبدیل کلمات فارسی به کلمات لاتین است (سایت دهخدا) . که در آنجا برای حل مشکل جندین یونی کد کلمات فارسی را به کلمات لاتین تبدیل می‌کند و عمل جستجو را روی کلمات لاتین انجام می‌دهد .

لازم به تذکر می‌باشد در صورت یکسان سازی استاندارد کی بورد ها فارسی برای اپراتورها این مشکل می‌تواند تا حد زیادی کاهش یابد .

وهمچنین سیستم عامل لینوکس از کی بورد فارسی پشتیبانی کامل به عمل می‌آورد و با توجه به رایگان بودن آن می‌توان این سیستم عامل را به راحتی جایگزین روش‌های فعلی نمود تا یونی کد های یکسانی وارد پایگاه داده‌های فارسی شوند .

۲-۶) مراجع این فصل :

۱. unicode.org
۲. wikipedia.org
۳. http://www.nofa.ir/
۴. مرکز محاسبات دانشگاه صنعتی شریف sharif.ac.ir
۵. سازمان مدیریت و برنامه‌ریزی کشور
۶. شورای عالی انفورماتیک

۳) معایب خط فارسی

وقتی درباره زبان سخن به میان می‌آید، اکثر مردم به طور ناآگاه به خط و نوشته فکر می‌کنند. ولی باید همواره به خاطر داشت که زبان اصل و خط فرع است و خط، به عنوان یک وسیله ثانوی، برای نمایاندن زبان به وجود آمده است.(دکتر محمد رضا باطنی ، ۲۶ خرداد ۱۳۸۷)

با توجه به اینکه خط فارسی برای نشان دادن صداهای زبان فارسی ناتوان است. معایب مهم خط فارسی را به اختصار می‌توان چنین برشمرد:

۱. مصوت های /a/ /e/ /o/ در خط نمایانده نمی‌شوند. مثلا «کَش»، «کِش»، «کُش» را در خط یک جور می‌نویسند. اشتباهات تلفظی و سوءتفاهماتی که در خواندن یک متن از این رهگذر پیش می‌آید بسیار است. چون مصوت‌های سه گانه بالا در خط وارد نمی‌شوند بسیاری از کلمات که تلفظ و معنی مختلف دارند یک صورت نوشته پیدا می‌کنند. مثلا صورت‌های: سقط، مهر، اقدام، اعمال، اخبار، اعلام، علم، اتباع، مجاز، رب، اسناد، ملک، پر، عده، چک، نسبی، حکم، ظهر، خلق، مثل، ترکه، در، مرد، خرد، تف، کرم، گرد، چرا، مسلم، سحر، سفت، صفر، و صدها صورت دیگر که گاهی چندین تلفظ و چندین معنی کاملا مختلف دارند. البته در مورد بعضی از کلمات آشنا از سیاق عبارت یا فحوای کلام می‌توان معنی آن‌ها را حدس زد و آن‌ها را درست تلفظ کرد، ولی در مورد کلماتی که برای خواننده آشنا نباشند، و مخصوصا در مورد کلمات خارجی و اسم‌های خاص که به خط فارسی نوشته شده باشند، فحوای کلام نیز نمی‌تواند دشواری تلفظ را بر طرف نماید.

۲. چون بعضی از صامت‌ها مکرر هستند، یعنی تشدید دارند و در عمل علامت تشدید آن‌ها نوشته نمی‌شود، اغلب دو کلمه کاملا متفاوت، یک صورت پیدا می‌کنند. صورت‌های «بر»، «کره» و «ماده» با تشدید و بدون تشدید کلمات متفاوت با معانی متفاوتی هستند. گاهی اوقات دشواری تلفظ و اختلاط معنی ناشی از ترکیب دو نقض بالا است: مثلا اختلاط کلماتی که به صورت، محرم، رویه، مقدم، سر، در خط ظاهر می‌شوند، ناشی از فقدان مصوت و تشدید با یکدیگر است.

۳. برای بعضی صداها بیش از یک علامت وجود دارد. مثلا برای صدای /s/ حروف «س»، «ص» و «ث» و برای صدای /z/ حروف «ز»، «ذ»، «ظ» و «ض» و برای صدای /t/ حروف «ت» و «ط» و برای صدا /h/ حروف «هـ» و «ح» و برای صدای /q/ حروف «ق» و «غ» به کار برده می‌شود. هم چنین دو حرف «ع» و «ء» نیز نماینده یک صدا هستند. این تعدد حروف از آن جا است که کلماتی که از عربی به فارسی آمده‌اند، بعضی دارای صداهایی بوده‌اند که در فارسی وجود نداشته است، در نتیجه فارسی زبانان نزدیک‌ترین صدایی را که در زبان خودشان وجود داشته، به جای آن‌ها نشانده‌اند. بدین ترتیب اختلاف املای کلمه که نماینده اصل عربی آن‌ها است حفظ شده، ولی تلفظ آن‌ها فارسی و یکنواخت شده است. گذشته از این که بسیاری از این کلمات از اصل فارسی به عربی رفته و دوباره به فارسی برگشته‌اند، از آن جایی که سال‌ها است این کلمات در فارسی به کار برده می‌شوند و در نتیجه تلفظ و معنی آن‌ها تغییر کرده است، باید آن‌ها را فارسی پنداشت. نوشتن آن‌ها به صورت اصلی عربی، نه تنها فارسی زبانان را به خارجی بودن آن‌ها حساس می‌کند – که خود کار نادرستی است زیرا این‌ها جزو واژگان فارسی شده‌اند – بلکه از نظر صوت شناسی نیز نامناسب است زیرا، چنان که گفتیم، هر دسته ازاین حروف نماینده یک صدا هستند و تمایز اصلی آن‌ها در فارسی از بین رفته است.

۴. بعضی حروف نمایده بیش از یک صدا هستند. مثلا «و» می‌تواند نماینده /o/ باشد مانند «تو» /to/، یا نماینده /u/ باشد، مانند «رو» /ru/ یا نماینده /v/ باشد، مانند «ولی» /vali/، یا نماینده /ow/ باشد، مانند «روشن» /rowšan/. «ی» می‌تواند نماینده /y/ باشد، مانند «یار» یا نماینده /i/ باشد، مانند «کی» /ki/. در نتیجه صورت نوشته «دین» می‌تواند دو تلفظ و دو معنی داشته

باشد: /deyn/ و /din/؛ هم چنین صورت‌های نوشته «کی»، «نیل» و مانند آن. «ا» در میان یا پایان کلمه نماینده صدای /â/ است، مانند «رفاه»، «جفا» و غیره ولی در آغاز به تنهایی می‌تواند نماینده سه صدا باشد مانند /a/ در «احمد»، /e/ در «اجبار» و /o/ در «الفت»؛ به صورت «آ» می‌تواند نماینده /â/ باشد، مانند «آب». به کمک حروف «و» و «ی» می‌تواند به ترتیب نماینده صداهای /u/ و /i/ باشد، مانند «او» /u/ و «ایران» /iran/. بدین ترتیب، حرف «ا» به تنهایی یا با کمک حروف «و» و «ی» می‌تواند در آغاز نماینده شش مصوت باشد.

۵. حرف بسته به جای خود در کلمه و بسته به حروفی که در پس و پیش آن قرار می‌گیرد لااقل به چهار صورت نوشته می‌شود: ع ـع ـعـ ع بعضی حروف مانند «ی» چهار صورت نیز بیشتر پیدا می‌کنند.

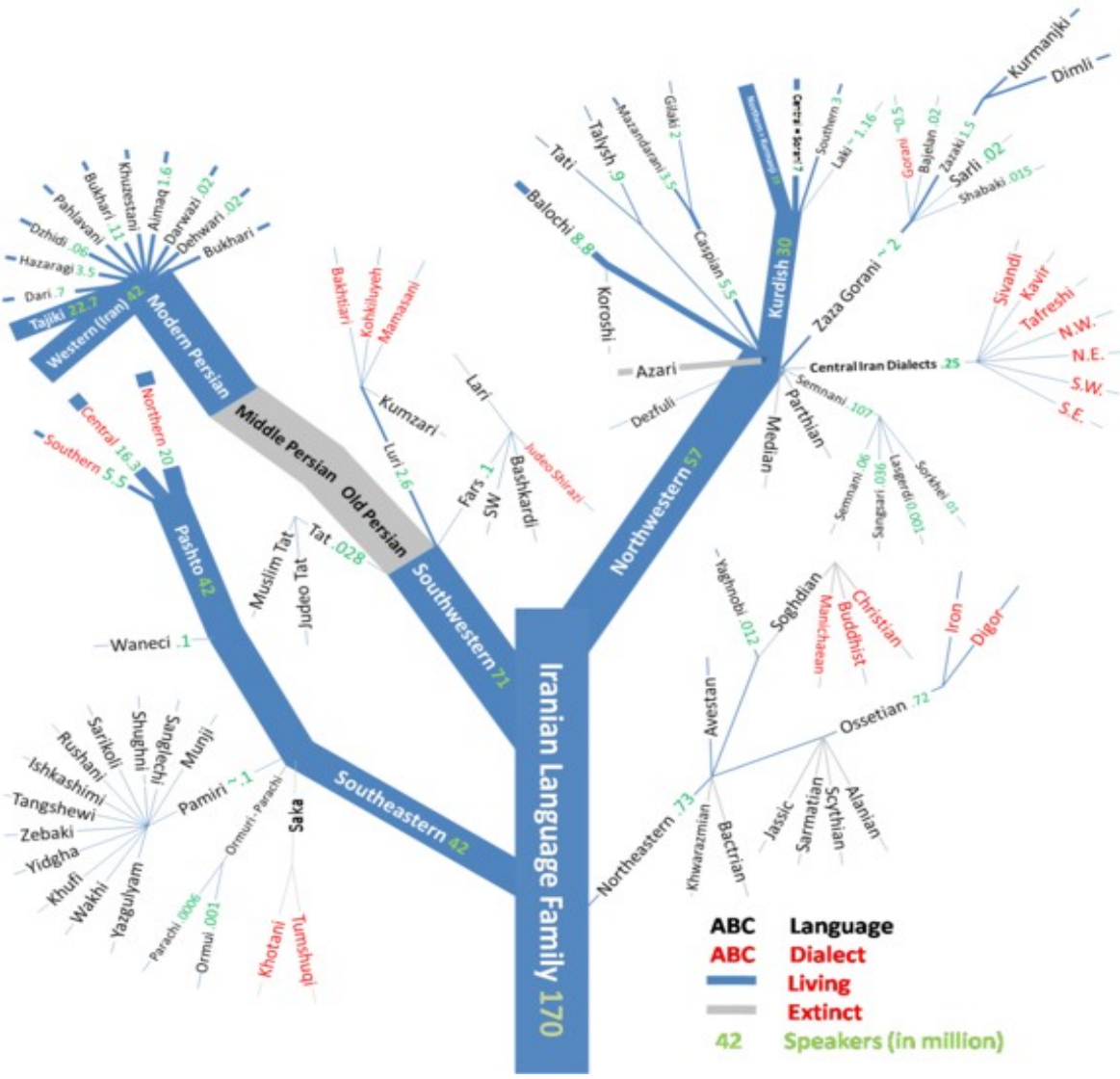
۶. مسئله قطع و وصل حروف، که بعضی به هم می‌چسبند و بعضی نمی‌چسبند، مشکل دیگری است. در مورد حروفی که به هم می‌چسبند نیز مرز مشخصی وجود ندارد، به همین جهت تصمیم آن اغلب به سلیقه فردی نویسنده واگذار می‌شود. بعضی اوقات دیده می‌شود که چندین کلمه را به هم می‌چسبانند، مانند «اینستکه» (این است که) «طوریستکه» (طوری است که) و مانند آن. اشکال دیگری که از قطع و وصل حروف ناشی می‌شود این است که گاهی فاصله بین حروف منفصل یک کلمه، در نوشتن یا در چاپ، کم و زیاد می‌شود، در نتیجه خواندن آن کلمه مشکل می‌شود و یا اصلا معنی دیگری پیدا می‌کند. مثلا اگر فاصله بین حروف در جمله «ما در آن جا هستیم» به هم بخورد جمله می‌تواند معنی دیگری داشته باشد: «مادر آن جا هستیم».

۷. اضافه که یکیاز عناصر بسیار فعال دستوری در زبان فارسی است، معمولا به صورت مصوت /e/ در تلفظ ظاهر می‌شود و چون این مصوت در خط فارسی وارد نمی‌شود بنابراین اضافه هم نوشته نمی‌شود. در نتیجه این نقص بزرگ گاهی خواننده جمله را تا آخر می‌خواند و آن را بی معنی حس می‌کند. ناچار برمی‌گردد و دقیق‌تر نگاه می‌کند و پی می‌برد که اضافه‌ای در تلفظ وارد نکرده یا بی جهت وارد کرده است. گاهی اوقات نیز کم و زیاد کردن اضافه جمله را بی معنی نمی‌کند، بلکه معنی آن را تغییر می‌دهد. مثلا در جمله «مردم دیگر این کار را نمی‌کنند» اگر اضافه‌ای پس از «مردم» قرار گیرد یا نگیرد معنی جمله به کلی فرق می‌کند. هم چنین بود و نبود اضافه پس از کلمه «اغلب» در جمله «اغلبِ مردم این طور فکر می‌کنند» معنی آن را تغییر می‌دهد.

۸. فراوانی نقطه‌ها و پس و پیش شدن آن‌ها و گاهی اوقات حذف آن‌ها اشکالات فراوانی ایجاد می‌کند که درباره آن‌ها داستان‌هایی ساخته شده است.

۹. علاوه بر معایب فوق، رسم الخط‌های مختلف، شکسته نویسی و قوانین پیچیده و اختلاف سلیقه‌ها در مورد املای همزه و مانند آن بر اشکال خواندن و نوشتن خط فارسی می‌افزاید.

۱۰. وجود لهجه ها و گونه های بسیار متفاوت در زبان فارسی



شکل ۳-۱) لهجه های زبان فارسی

۱۱. از دید یک برنامه نویس، جدا از عیب های گفته شده، عاملهای پیچیده دیگر در خط ما، مانند صورتهای گوناگون حرفها، نوشتن از راست به چپ و چسبیدن حرفها به یکدیگر موجب کند شدن روند بومی سازی نرم افزارها و ابزارهای برنامه نویسی است. که خود موجب گسترش به کارگیری الفبای بیگانه و ناهمگونی خط در استفاده روزمره است. از آن جمله می شود به پدیده فینگلیش یا استفاده از زبانهایی جز پارسی در ابزارها و ابزارک ها اشاره کرد. (فرامرز ۲۳ دی ۱۳۸۹ persianlanguage.ir)

۳-۱) مثالهایی از اشکالات زبان فارسی :

الف) ابهام :

معلم خطاب به دانش آموز : آیا می‌دانی کلمه ی « لا » در عربی به چه معنا است ؟

دانش آموز : نه آقا .

همانگونه که ملاحظه می‌کنید عبارت « نه آقا » دارای ۲ معنی می‌باشد که باعث ابهام در جمله می‌گردد

ب) غلط املایی :

مدیر محترم دانشگاه تربیت معلم:

بنا به درخواست هیئت عامل شرکت « صبا » مبنی بر دعوت شما و خانواده ی گرامیتان ؛ از شما عاجزان { به جای عاجزانه } تقاضا دارم که فردا راس ساعت ۱۰ در مجلس عروسی به صرف شیرینی و شام شرکت نمایید .

همان‌گونه که ملاحظه می‌فرمایید جا افتادن یک حرف «ه» باعث تغییر کل معنی جمله می‌شود وحالت نامه را از محترمانه به توهین آمیز تغییر می‌دهد .

پ)جستجوی کلمات با یک رسم الخط با معنی های متفاوت :

یافتن کلمه ی دین(مذهب) و دین (قرض) در موتور های جستجو ی مختلف

google :

فقط عبارات مرتبط با دین (مذهب) را پیدا کرده است .

دین - ویکیپدیا

دین یا کیش یا طریقت یا شریعت، مجموعه‌ای از معارف، عقاید، باورها و قانونها و دستورهایی برای راهنمایی و پرورش انسان و پیشبرد او به سوی تکامل و رستگاری است. ...

۱ تعریف دین - ۲ ریشه شناسی و دین - ۳ اهمیت دین - ۴ انتقادات از دین

fa.wikipedia.org/wiki/دین - Cached – Similar

دین | تاریخ تمدن باستان

گراردو جنولی ؛ علی رضا شجاعی روشن نیست که آیین زُروان دین ایرانیِ ماقبل اسلامی و مستقل از دین زرتشت و از جهاتی رقیب آن است[۱]، یا اندیشه رایج فلسفی و دینی ...

Cached - *tarikhema.ir/words/دین/*

دین و اندیشه

آخرین مطالب دین و اندیشه. جهانی شدن فرهنگی و مهدویت ۱ · بلاگردان اهل زمین (انتظار، انواع و آثار) · کجایی ای ستون آسمانها · پیامکهای مهدوی عصر جمعه ...

- Cached – Similar *www.tebyan.net › صفحه اصلی › دین و اندیشه*

wikipedia :

فقط به دین و مذهب اشاره کرده است.

Bing :

فقط به دین و مذهب اشاره کرده است.

دین - Wiktionary

Azeri: religion · religion

en.wiktionary.org/wiki/دین

نظرهای علمی درباره دین، پیدایش و ...

نوشته دکتر احمد ایرانی

-/www.scribd.com/doc/19491542

دین - *ویکی‌پدیا – Wikipedia*

تعریف دین
ریشه‌شناسی ...
اهمیت دین

دین یا کیش یا طریقت یا شریعت، مجموعه‌ای از معارف، عقاید، باورها و قانون‌ها و دستورها یی برای ...

fa.wikipedia.org/wiki/دین

۳-۲) مسئله ۱ :

همانطور که ملاحظه کردید حتی موتورهای جستجوی بزرگ نیز فقط اقدام به پیدا کردن لغاتی به کار می‌روند که کاربرد بیشتری دارد . مثلاً در مثال بالا اصلاً به دنبال کلمه ی دین به معنای قرض نبوده و تنها معنی مذهب آن را پیدا کرده است .

۳-۳) راه حل ۱ :

این مشکل در زبان‌های دیگر حتی در زبان انگلیسی هم وجود دارد البته نه به این شدت . در ویکی پدیا وقتی به دنبال یک کلمه ی به‌خصوص می‌گردیم که چند جواب متفاوت دارد از ما سؤال می‌کند که منظور ما کدام کلمه است .

مثلاً وقتی کلمه ی lamp را در ویکی پدیا جسنجو می‌کنیم با این عبارت رو به رو می‌شویم :

Lamps

Lamp may refer to one of the following:

- Oil lamp, the original use of the term
- Kerosene lamp, a lamp burning liquid petroleum
- Lamp (electrical component) is a replaceable component that produces light, such as:
    - Arc lamp
    - Fluorescent lamp
    - Gas discharge lamp
    - Incandescent light bulb, also known as an incandescent lamp
- A lampshade is a part of a luminaire that serves to block direct view of the lamp
- A light fixture, or luminaire, often colloquially known as a lamp
- Signal lamp, a device used for visual communication between ships
- A fuel burning illumination or signal lantern
- A safety lamp or Davy lamp, is an explosion-proof lantern used in mining
    - See also Category:Lamps
- *Lamp (advertisement)*, a 2002 television advertisement directed by Spike Jonze
- *GET LAMP*, a documentary

LAMP

As an Acronym, **LAMP** may refer to:

Computing

- LAMP (software bundle), a platform consisting of Linux, Apache, MySQL and Perl/PHP/Python.
- Library Access to Music Project, a free music library for MIT students

Science

- Lyman Alpha Mapping Project, an instrument to be used on the NASA Lunar Reconnaissance Orbiter
- Loop-mediated isothermal AMPlification, a single tube technique for the amplification of DNA
- *Lysosome-associated membrane glycoprotein*
    - LAMP1 - Lysosomal-associated membrane protein 1

- LAMP2 - Lysosomal-associated membrane protein 2
- LAMP3 - Lysosomal-associated membrane glycoprotein 3

Other

- Lutheran Association of Missionaries and Pilots, a cross-cultural ministry organisation
- Loveless Academic Magnet Program, a magnet high school in Montgomery, Alabama.
- LAMP Community, a Los Angeles-based nonprofit organization located in Skid Row

(REF =wikipedia.org)

همانطور که ملاحضه کردید کلمه ی lamp چندین معنی مختلف در پایگاه داده ی ویکی پدیا دارد لذا این سای از کاربر خو می خواد که مشخص کند که منظوراش کدام lamp است .

۳-۴) راه حل ۲ :

برای حل مشکل فوق می‌توان یک راه ابتکاری دیگر نیز استفاده کرد .

استفاده از حروف لاتین برای ذخیره ی کلمه ی فارسی در پایگاه داده مثال :

علی → ali
دین → deyn
دین → din

دادن یک کد لاتین اختصاصی به هر کلمه . از این روش در لغت‌نامه ی دهخدا استفاده می‌شود
http://www.loghatnaameh.com

۳-۵) مسأله ۲ :

زبان و خط فارسی به سبب استفاده از زبان یا خط انگلیسی (لاتین) در محاورات رایانه‌ای یا پیام‌های کوتاه توسط فارسی زبانان، مورد تهدید واقع شده است. این امر بیشتر ناشی از عدم پشتیبانی نرم‌افزار های موجود از زبان فارسی می‌باشد که باعث شده است کاربران فارسی زبان بیشتر به فانگیلیش نویسی و عدم استفاده از حروف فارسی بپردازند . البته لازم به ذکر است این موضوع در کشورهای عربی زبان کمتر مشاهده می‌شود زیرا بسیاری از برنامه‌های کاربردی از زبان عربی پشتیبانی می‌کنند و این امر باعث استقبال گسترده ی عربی زبانها از لغات عربی در محیط های مجازی است .

۳-۶) نرم‌افزار های فارسی زبان :

۳-۶-۱) خطایاب املایی ویرا :

خطایاب املایی ویرا دارای کامل‌ترین بانک لغات فارسی با بیش از یک‌صدهزار مدخل ریشه‌یابی‌شدهٔ املایی است. این مداخل املایی در طول سه سال گردآوری شده و چندین مرتبه توسط مصححین مرور شده است تا حتی‌الامکان خطای املایی نداشته باشد. همچنین مجهز بودن ویرا به موتور ریشه‌یاب سپنتا که در این محصول به کار گرفته شده است، به ویرا اجازه می‌دهد دامنهٔ وسیعی از لغات انشقاقی فارسی را که ریشهٔ آنها در بانک لغات وجود دارد، شناسایی کند.

http://www.spellchecker.ir/default.aspx

۳-۶-۲) نویسا: تایپ گفتاری فارسی :

نسخهٔ جاری سیستم نویسا دارای دقت تشخیص ۹۵% در محیط اداری، قابلیت استفاده در همه ویرایشگرها یا قسمت‌هایی که امکان تایپ فارسی دارند، قابلیت تغییر/ افزایش و تخصصی نمودن دایره کلمات است.

با سیستم دیکتهٔ گفتاری زبان فارسی دیگر نیازی به تایپ یا نوشتن متن نیست! تنها با خواندن متن، آن را تایپ نمایید. مزیت‌های استفاده از سیستم تایپ گفتاری نویسا:

- صرفه‌جویی در زمان
- کاهش هزینه
- افزایش سرعت تایپ و ورود اطلاعات
- حفظ امنیت اطلاعات در هنگام ورود داده‌ها
- قابلیت استفاده در بسیاری از سیستم‌های (مستندسازی، ترجمه گفتاری و ...)
- جلوگیری از اشتباهات تایپی

۳-۶-۳) جوملا! به زبان فارسی :

جوملا! (Joomla!) نام یک نرم‌افزار آزاد و متن‌باز برای مدیریت محتوای اینترنتی است. جوملا! به زبان PHP نوشته شده‌است و از پایگاه دادهٔ MySQL استفاده می‌کند.
قابلیت‌های جوملا! شامل امکان بارگذاری موقت در حافظه برای بهبود کارایی (caching)، ایجاد فهرست خودکار، ارسال خبر از طریق پروتکل RSS، ارائهٔ نسخه قابل چاپ، بخش‌های کوتاه خبری، تالار گفتگو، نظرسنجی، تقویم، جستجوی اینترنت و پشتیبانی از زبان‌های متعدد (از جمله فارسی) است.

۳-۶-۴) ویراستار – غلط یاب فارسی :

غلط‌یاب فارسی «ویراستیار» نرم‌افزاری است توانا برای اصلاح خطاهای املائی، اشتباهات ویرایشی، نشانه‌گذاری و نیز استانداردسازی متون فارسی. این نرم‌افزار توسط «دبیرخانه‌ی شورای عالی اطلاع رسانی» تهیه شده و در قالب یک افزونه (Add in) ارائه می‌گردد. نرم‌افزار «ویراستیار ۱ » امکان پشتیبانی از نسخه‌های (Microsoft Word) برای نرم‌افزار مایکروسافت وُرد متفاوت مایکروسافت وُرد و ویندوز را دارا است. ویراستیار توسط «شورای عالی اطلاع‌رسانی» تهیه شده است.

۳-۶-۵) نرم افزار واژه شناس (OCR فارسی) :

OCR(Optical Character Recognition) از لحاظ لغوی به معنی تشخیص متون موجود در تصاویر می باشد و به یک تعبیر ساده تبدیل تصاویر اسناد مکتوب به متن کامپیوتری است.

جهت آشنایی سعی شده است تا بعضی از کاربردهای این نرم افزار در مراکز مختلف ارائه گردد:

- سیستم دبیرخانه بدون کاغذ
- سیستم بایگانی یا آرشیو اسناد و قراردادها

- موسسات فرهنگی و آموزشی و شرکت هایی که نرم افزارها و فعالیت های آموزشی دارند
- بانک ها و موسسات مالی و اعتباری (قرض الحسنه ، سرمایه گذاری و ...)
- کتابخانه ها ، موسسات و شرکت های اطلاع رسانی
- دانشگاه ها و مراکز آموزشی
- روزنامه ها و نشریات
- سازمان های قضایی و دادگاهی

۳-۷) مراجع این فصل :

- مقاله ی دکتر محمد رضا باطنی در مورد معایب زبان فارسی ۲۶ خرداد ۱۳۸۷
- مقاله ی حسن بهزادیان‌نژاد در مورد لزوم حفظ زبان فارسی در دنیای فن‌آوری‌های وب و تلفن همراه ۲۸ اردیبهشت ۱۳۸۷
- http://persianlanguage.ir
- ویکی پدیا

۴) روش های جستجو هوش مصنوعی

۴-۱) مقدمه :

تا اینجای کار درمورد یونی کد پارسی و مسائل و مشکلات کلمات فارسی صحبت کردیم . در این فصل می‌خواهیم از موضوعات فصل های قبل فاصله بگیریم و در مورد روش‌های جستجو بپردازیم . لازم به ذکر است که مقوله ی جستجو یک مقوله بسیار وسیع است و در بسیاری از موارد کاربرد دارد همچنین مقوله ی جستجو به صورت مجزا در درس الگوریتم بررسی شده است و خود شامل دنیای وسیعی می‌باشد و در اینجا به مختصر بررسی آن می‌پردازیم و از پروا گویی در مورد آن اجتناب می‌کنیم .

۴-۲) جستجو چیست ؟

ما به عنوان یک عامل هوشمند، در مسائل روزمره ای که برایمان پیش می آید با استفاده از تکنیک های جستجو اقدام به یافتن راه حلی برای مسئله می کنیم. به عنوان مثال خانه ای را فرض کنید که در آن به دنبال شی خاصی هستیم. همچنین فرض کنید هیچ اطلاعی در

مورد خانه و محل قرارگیری اشیا در خانه نداریم. در چنین مواردی برای یافتن شی موردنظر، خانه را اتاق به اتاق و در هر اتاق همه بخش های آن اتاق را جستجو می کنیم. نتیجه جستجو تنها در صورتی می تواند به دست آید که جواب مسئله در فضایی که آن را جستجو می کنیم، موجود باشد.به عنوان یک مثال دیگر فرض کنید می خواهیم از تبریز به سمت شیراز حرکت کنیم. هدف ما نیز رسیدن به شیراز در کمترین زمان ممکن می باشد (مسأله ی دوره‌گرد ، دور دنیا در ۸۰ روز ، TSP) . یکبار دیگر نیز ما به عنوان یک عامل هوشمند، با در دست داشتن نقشه راه های کشور براحتی مسیر حرکت خود را مشخص می کنیم. به طور حتم هنگام جستجو بر روی نقشه برای یافتن کوتاهرین مسیر حرکت از تبریز به شیراز، از جستجوی مسیر حرکت تبریز-ارومیه-شیراز صرف نظر می‌کنیم . حال سوالی که اینجا مطرح می شود این است از به جستجوی راه حل در مسیر تبریز-ارومیه-شیراز نمی پردازیم.مهمترین تفاوت دو مثال ذکر شده را ناآگاهانه بودن جستجو در مثال اول و آگاهانه بودن جستجو در مثال دوم می باشد. بدین معنی که در مثال اول ما از ابتدا کل فضای جستجو را برای یافتن جواب مسئله جستجو می کنیم و در واقع برای حرکت سریع تر در فضای جستجو (حرک بین اتاق ها و بخش های مختلف هر اتاق) از آگاهی خاصی بهره مند نیستیم. در نقطه مقابل و در مثال دوم برای حرکت در فضای جستجو به منظور یافتن کوتاهترین مسیر از تبریزبه شیراز، از روش خاصی برای انتخاب شهر بعدی استفاده می کنیم. به عبارت دیگر در مثال دوم علاوه بر اینکه مسیری از مبدا به مقصد پیدا می کنیم، همچنین کوتاهی مسیر نیز برایمان از اهمیت ویژه ای برخوردار می باشد. در واقع عملی که برای سنجش کوتاهی مسیر انجام می دهیم موجب آگاهانه بودن جستجوی ما در فضای جستجو می شود. در مثال دوم از تابعی استفاده می کنیم که فاصله شهر انتخابی تا مقصد را برایمان **تخمین** می زند که به آن تابع هیوریستیک یا تابع مکاشفه خواهیم گفت.

حال ممکن است این تصویر به وجود آید که جستجوی آگاهانه بهتر از جستجوی ناآگاهانه می باشد. اما چنین تصوری را نمی توان قطعا درست تلقی کرد. چراکه در صورت ناآگاهانه بودن ماهیت مسئله چاره جز جستجوی ناآگاهانه نخواهیم داشت. همچنین سریعتر بودن جستجوی آگاهانه نسبت به جستجوی ناآگاهانه نیز بستگی به شرایط مسئله خواهد داشت و موارد بسیاز زیادی می توان یافت که در آن جستجوی ناآگاهانه سریعتر از جستجوی آگاهانه ما را به جواب مسئله می رساند.

:در حالت کلی روش های جستجو را می توان به سه دسته زیر تقسیم کرد

- ناآگاهانه
- هیوریستیکی

- متاهیوریستیکی

مسائلی وجود دارند که فضای جستجوی آنها به اندازه ای بزرگ می باشد که استفاده از روش های جستجوی ناآگاهانه و هیویریستیکی را برایمان غیرممکن می سازد. در چنین مواردی از روش های متاهیوریستیکی یا فرامکاشفه ی استفاده خواهیم کرد.

۴-۳) انواع مسائل در هوش مصنوعی

مسائل مطرح شده در هوش مصنوعی را از چند دیدگاه مختلف می توان مورد بررسی قرار دارد :

- هدف یافتن تنها یک جواب برای مسئله است یا همه جواب های ممکن برای مسئله باید جستجو شوند؟
- پاسخ دهی الگوریتم جستجو برای حل مسئله بلادرنگ خواهد بود یا زمان بیشتری برای حل مساله می توان اختیار کرد
- آیا در مسئله محدودیت هایی وجود دارد یا نه ؟

همه اینها سوالاتی هستند که هنگام طراحی الگوریتمی برای پویش در فضای جستجو باید به آنها توجه کنیم. به عنوان مثال در صورتی که همه جواب های مسئله مدنظر ما باشد، در اینصورت حتی در صورت آگاهانه بودن ماهیت مسئله نیز نمی توان از روش های هیوریستیکی استفاده کرد. به عنوان مثال در صورتی که هدف یافتن همه مسیرهای ممکن از تبریز به شیراز باشد، استفاده از هیوریستیک تنها موجب پیچیده شدن الگوریتم و حتی موجب ناقص بودن جواب های مسئله خواهد شد.

سیستم های پویا سیستم هایی هستند که در آنها به طور مداوم ورودی ها و حالت مسئله در حال تغییر است و هربار تغییر در یکی از پارامترها موجب خواهد شد که عمل جستجوی مجددی در فضای جستجو انجام پذیرد. در چنین مواردی ممکن است مدت زمان پاسخ دهی سیستم برای حل مساله بسیار کم باشد. همچنین ممکن است سیستم مدت زمان کافی برای حل مساله در اختیار داشته باشد که هریک از این موارد تاثیر بسزایی در طراحی الگوریتم جستجو خواهند داشت.

فرض کنید می خواهیم در یک صفحه شطرنج استاندارد، ۸ وزیر را به گونه بر روی صفحه شطرنج قرار دهیم که هیچ یک از آنها وزیر دیگری راتحدید نکنند. در این مسئله که به مسئله ۸ وزیر معروف است، حالتی وجود دارد که هیچ کدام از آه وزیر یکدیگر را تحدید نکنند . به صورت تجربی می توان چنین گفت که بیشتر مسائلی که با آنها سروکار داریم دارای محدویت هایی در مسئله هستند. به چنین مسائلی

مسائل ارضای محدودیت نیز می گوییم. وجود محدودیت در مسئله قدری موجب پیچیدگی مسئله می شود اما در اکثر موارد نیز موجب خواهد شد فضای جستجو مسئله بسیار کوچکتر از حالتی باشد که در آن مسئله هیچ محدودیتی در خود ندارد.

مسائلی که در عمل با آنها مواجه هستیم ترکیبی از موارد فوق را در خود دارند. به عنوان مثال در مسئله ی ۸ وزیر ، مسئله دارای محدودیت بوده ولی در مقابل هدف یافتن تنها یک جواب در مدت زمان کافی است. و در واقع یافتن جواب در آن نیازی به بلادرنگ بودن ندارد. در ادامه ، به بررسی روش های جستجو ناآگاهانه و نحوه پیاده سازی آنها خواهیم پرداخت.

۴-۴) الگوریتم های جستجوی ناآگاهانه

روش های جستجوی ناآگاهانه ای که در این بخش با آنها آشنا خواهیم شد، به قرار زیر هستند :

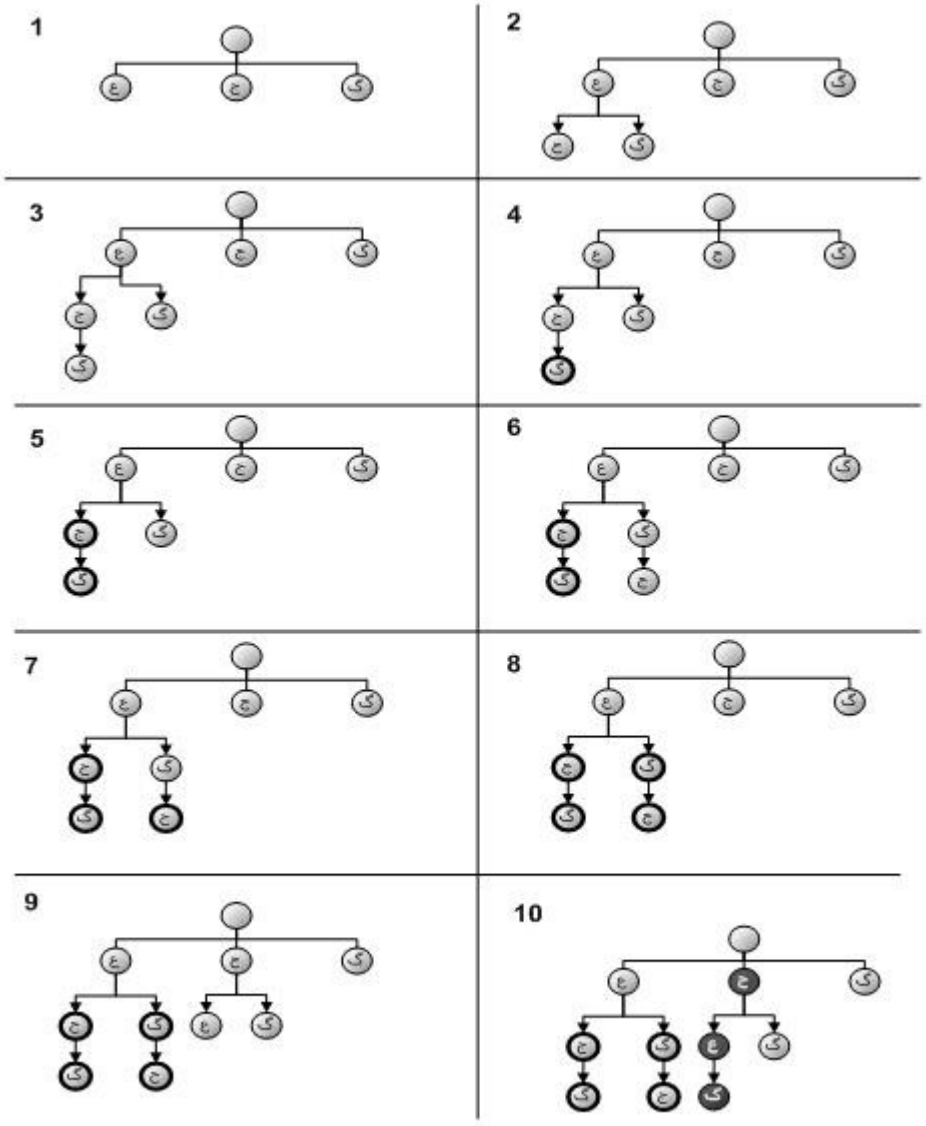
- جستجوی عمقی
- جستجوی سطحی
- جستجوی عمقی محدود شده
- جستجوی عمقی تکرار شونده

در همه روش های جستجو تابعی به نام تابع گسترش وجود دارد که وظیفه این تابع تولید فرزندان یک گره می باشد. تنها تفاوت روش های جستجوی فوق در نحوه پیمایش فضای جستجو خواهد بود. مسلم است که مرتبه زمانی و مکانی هریک از این روش ها برای یک مسئله مشخص متفاوت از یکدیگر خواهد بود. در حالت کلی مقایسه روش های جستجو برای یک مسئله و انتخاب روش جستجوی مناسب برای آن با استفاده از فاکتورهای زیر تعیین می شود :

- فاکتور انشعاب : حداکثر تعداد فرزندانی که در هر سطح از گراف وجود خواهد داشت.
- عمق درخت : حداقل عمقی که در آن عمق از گراف جوابی برای مسئله پیدا خواهد شد.

۴-۵) جستجوی عمقی (پیمایش FLR)

از اسم این روش جستجو پیداست که پیمایش گراف فضای جستجو با حرکت در عمق انجام می پذیرد. به عنوان مثال فرض کنید با سه کلمه "جستجو" ، "گراف" و "عمقی" می خواهیم **یک** جمله معنی دار بسازیم. شکل زیر مراحل مختلف تشکیل گراف هنگام جستجوی عمقی گراف را نشان می دهد :



شکل) ۴-۱

الگوریتم جستجو چنین است :

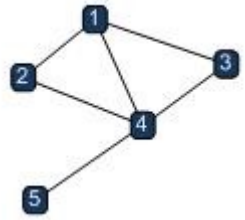
ابتدا فرزندان ریشه درخت تولید شده و در سطح بعدی درخت قرار می گیرند. در ابتدای کار هریک از کلمات می تواند برای شروع جمله بکار روند. سپس اولوین گره ارای کلمه "عمقی" در سطح ۱ از درخت انتخاب شده و همه فزرندان قابل تولید آن در سطح ۲ درخت تشکیل می شوند(شکل ۴-۱-۲). سپس در سطح ۲ از درخت گره شامل کلمه "جستجو" انتخاب شده و همه فرزندان آن در سطح ۳ درخت تشکیل می شوند(شکل ۴-۱-۳). در نهایت نیز کلمه "گراف" انتخاب شده ور همه فزرندان آن تولید می شوند. با توجه به اینکه دیگر هیچ فرزندی برای این گره وجود ندارد، بنابراین جمله "عمقی جستجوی گراف" که از پیمایش درخت از ریشه تا گره فعلی به دست می آید بررسی می شود تا صحیح بودن این جمله را تعیین کند. همانطور که می دانیم این جمله از نظر قواعدی جمله صحیحی نمی باشد. بنابراین گره فعلی از حافظه حذف شده (شکل ۴-۱-۴) و گره "جستجو" در سطح ۲ بررسی می شود تا اگر فزرند دیگری برای این گره وجود داشته باشد، ادامه جستجو از آن فرزند ادامه یابد. اما فرزند دیگری برای این گره وجود ندارد. از اینرو این گره نیز از حافظه حذف می شود (شکل ۴-۱-۵)

حال فرزندان گره "گراف" در سطح ۲ تولید می شوند (شکل ۴-۱-۶). همانطور که می بینیم فرزند دیگری برای گره "جستجو" در سطح ۳ وجود ندارد. بنابراین این گره نیز به همراه گره پدر از حافظه حذف خواهد شد (شکل ۴-۱-۷ و ۴-۱-۸). سپس به سطح ۱ بازگشته و فرزندان گره "جستجو" را گسترش می دهیم(شکل ۴-۱-۹). سپس فرزندادن گره "عمقی" در سطح ۲ گسترش یافته صحت جمله ساخته شده از آن کلمات بررسی می شوند. در این مرحله جمله از نظر قواعدی درست تشخیص داده شده و عبارت "جستجوی عمقی درخت" به عنوان جواب مسئله پیدا می شود (شکل ۴-۱-۱۰).

در مسئله فوق از دو تابع گسترش و تست هدف برای گسترش دادن فرزندان گره و تعیین صحت جواب مسئله استفاده کردیم. این توابع در مسائل مختلف به روش های گوناگونی پیاده سازی می شوند و الگوریتم کلی برای آن ها وجود ندارد.

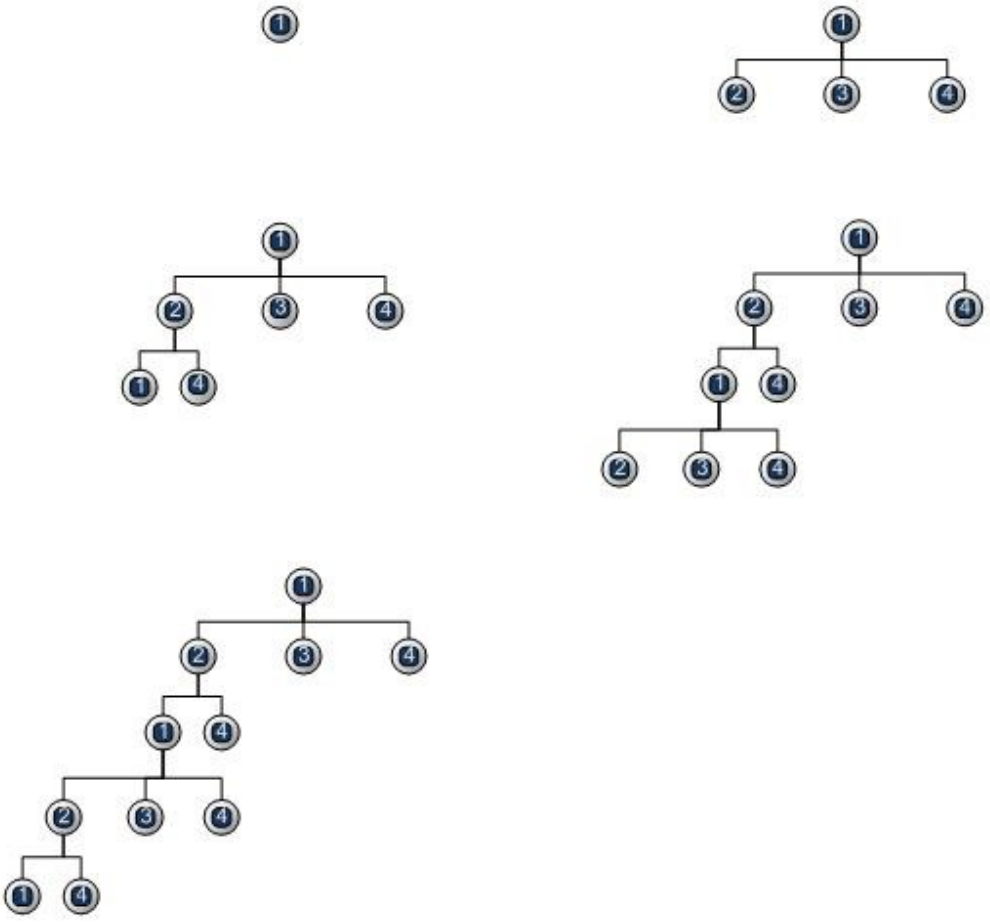
۴-۶) جستجوی عمقی محدود شده:

مهمترین مشکل روش جستجوی عمقی را که در برخی از مسائل با آن مواجه خواهیم بود، قرار گرفتن این روش جستجو در حلقه بینهایت می باشد. به عنوان مثال گراف شکل ۴-۲ را در نظر بگیرید :



شکل ۴-۲

فرض کنید می خواهیم مسیری از گره ۱ به گره ۵ با استفاده از روش جستجوی عمقی پیدا کنیم. شکل ۴-۳ مراحل مختلف تشکیل درخت جستجو را نشان می دهد .



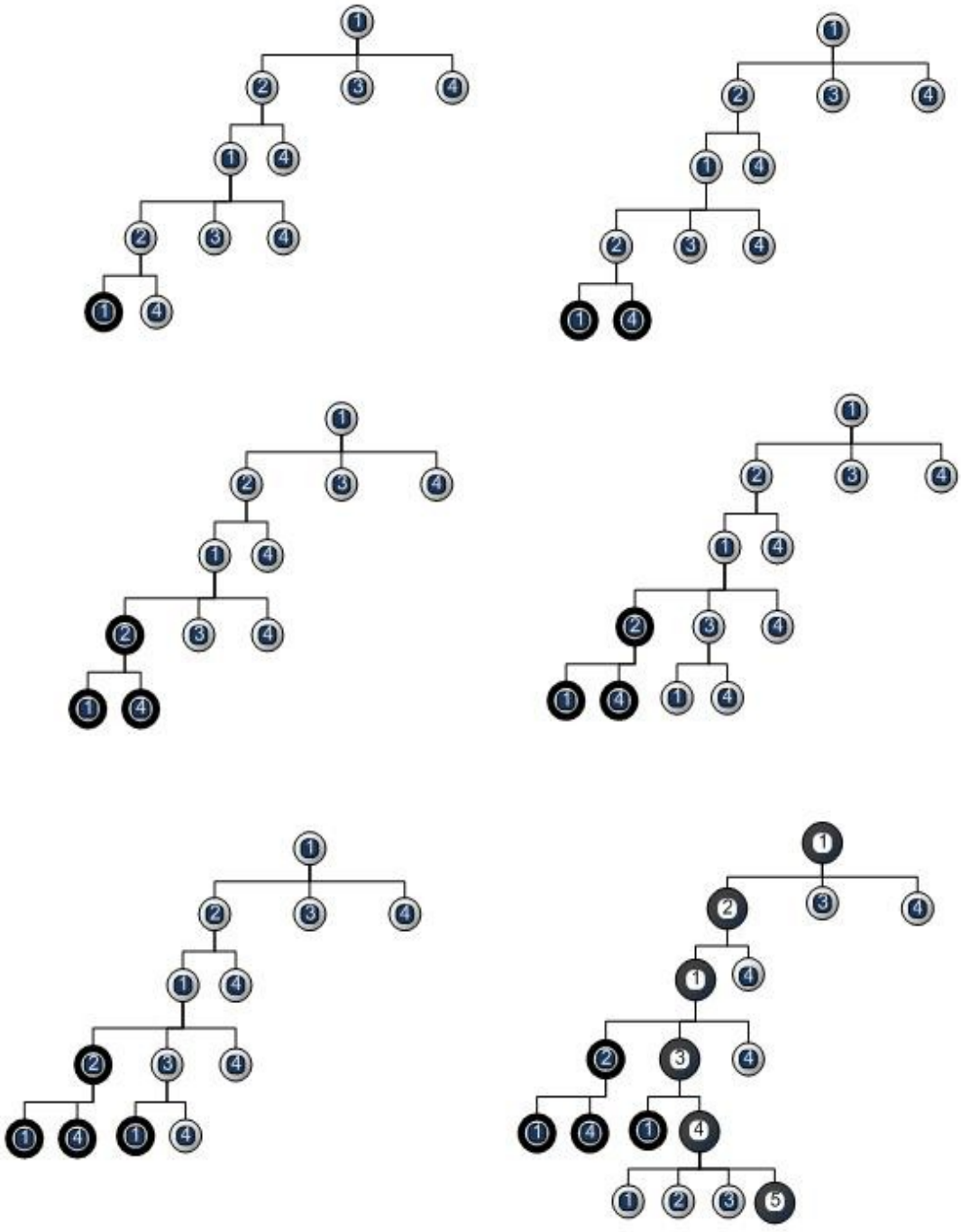
شکل ۴-۳

همانطور که از شکل پیداست ، استفاده از روش جستجوی عمقی برای حل این مساله هیچگاه به یافتن جواب منجر نخواهد شد. چرا که حلقه بینهایتی از گسترش گره های ۱ و ۲ تشکیل می شود. مشکل قرار گرفتن در حلقه بینهایت از گسترش حالت های تکراری در هنگام جستجو حاصل شده است. بنابراین یکی از روش های حل این مشکل تشخیص حالت های تکراری هنگام گسترش فرزندان یک گره می باشد.

تشخیص حالت های تکراری یکی از مهمترین نکات طراحی الگوریتم های هوش مصنوعی می باشد که وابستگی بسیاری به نحوه نمایش راه حل مساله دارد. نحوه نمایش مساله و تشخیص حالت های تکراری را در انتهای فصل و در مثال های مختلف مورد بررسی قرار خواهیم داد. در حال حاضر با روش دیگری به رفع این مشکل خواهیم پرداخت.

می دانیم در صورتی که اگر مسیری از گره ۱ به گره ۵ در گراف شکل ۱ وجود داشته باشد، طول این مسیر نمی تواند بیش از ۵ یال باشد. بنابراین می توان با اتخاذ یک محدودیت به روش جستجوی عمقی از گسترش گره هایی که در عمق ۵ از درخت جستجو قرار دارند، جلوگیری کرد. به عبارت دیگر برای گره هایی که در عمق ۵ از درخت جستجو قرار دارند ، هیچ فرزندی تولید نکنیم. پیمایش در فضای جستجوی مساله با این روش را روش جستجوی عمقی محدود شده می نامیم. یکبار دیگر با استفاده از روش جستجوی عمقی محدود شده به عمق ۵ به حل مساله می پردازیم. شکل زیر مراحل مختلف تشکیل درخت جستجو را نشان می دهد (ادامه درخت جستجوی شکل ۴-۴) :

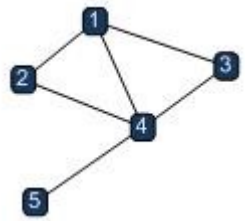
شکل ۴-۴



با توجه به گراف شکل ۴-۴ ، مسیر ۵-۴-۳-۱-۲-۱ ما را از گره ۱ به گره شماره ۵ هدایت می کند. با وجود اینکه محدود کردن عمق درخت مشکل حلقه بینهایت را رفع کرد، اما هنوز مشکل حالت تکراری در روش جستجوی عمقی محدود شده وجود دارد. برای حل همزمان مشکل حلقه بینهایت و حالت تکراری، می توان از روش جستجوی سطحی استفاده کرد که در بخش بعد به شرح آن پرداخته ایم.

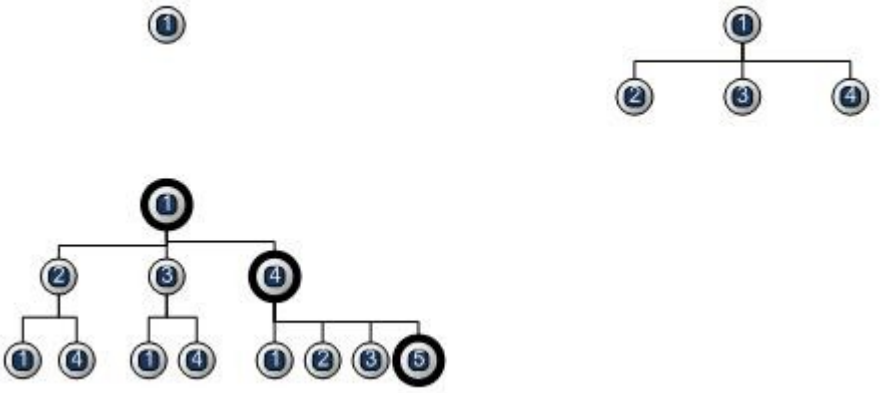
۴-۷) جستجوی سطحی

همانطور که در بخش جستجوی عمقی محدود شده بررسی کردیم، این روش جستجو مشکل حلقه بینهایت و یال های تکراری را حل کرد. البته این بدان معنا نیست که مشکل حالت های تکراری در جستجوی عمقی محدود شده به طور کامل رفع شده است. بلکه هیچ یک از روش های جستجو در مقابل حالت های تکراری مصون نیستند. این وظیفه طراح الگوریتم است که با اتخاذ روشی از بروز حالت های تکراری جلوگیری کند. یکبار دیگر گراف زیر را در نظر بگیرید :



شکل ۴-۵

می خواهیم مسیری از گره ۱ به گره ۵ پیدا کنیم. در صورتی که از جستجوی سطحی برای پیمایش فضای جستجو بخواهیم استفاده کنیم، بدین روش عمل می کنیم : در هر سطح از درخت جستجو ، به ازای هر گره در آن سطح همه فرزندان گره جاری تولید شده و در سطح بعدی درخت قرار می گیرند. با این تعریف حال می توانیم نحوه پیمایش درخت جستجوی شکل ۴-۶ را نشان دهیم :



شکل ۴-۶

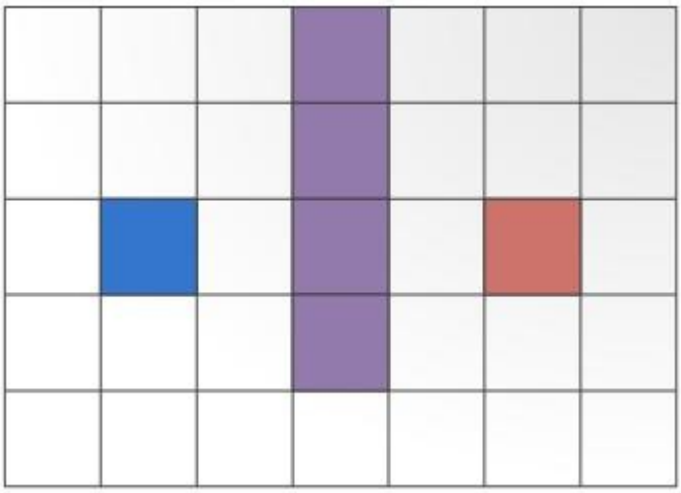
در ابتدا همه فرزندان گره ۱ تولید شده و به سطح بعدی درخت اضافه می گردند. در سطح بعدی درخت نیز به ازای هر گره فرزندان آن گره تولید و به درخت جستجو اضافه می شوند. همانطور که از شکل پیداست، در سطح ۳ از درخت مسیری از گره ۱ به گره ۵ پیدا می شود. همچنین مشکل یال های تکراری روش های قبلی نیز در این درخت وجود ندارد. یکی از بزرگترین مشکلات روش جستجوی سطحی مرتبه مکانی یا میزان حافظه مصرفی این روش جستجو است. در روش های جستجو عمقی و عمقی محدود شده در هر لحظه تنها گره های مربوط به شاخه فعلی در حافظه وجود دارد. این در حالی است که روش جستجوی سطحی همه گره های بالاتر از سطح فعلی از فضای جستجو را در حافظه نگه خواهد داشت. می توان روش جستجوی عمقی و سطحی را باهم ترکیب کنیم تا با ترکیب این دو ، معایب موجود در هریک از روش ها را رفع کرده و مزایای آن ها را باهم ترکیب کنیم. این روش جستجو راجستجوی عمقی تکرار شونده می نامیم.

۴-۸) جستجوی هزینه یکنواخت :

در هیچ یک از روش های جستجوی عمقی ، سطحی ، عمقی محدود شده و عمقی تکرار شونده هزینه ای که تا به حال از ریشه تا گره فعلی صرف شده است ، در نظر گرفته نمی شود. همین امر موجب می گردد تا در مسائل بهینه سازی نتوان از این روش ها برای جستجوی فضای جستجو استفاده کرد. چرا که هیچ استراتژی برای گسترش گره های بعدی وجود ندارد. از دیگر روش های جستجوی ناآگاهانه که بر روی درخت های وزن دار می توانیم از آن استفاده کنیم، روش جستجوی هزینه یکنواخت می باشد.

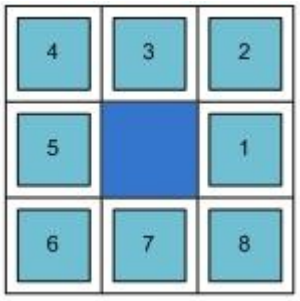
جستجوی هزینه یکنواخت را از آن جهت که در هر لحظه کم هزینه ترین گره را گسترش می دهد ، نه می توان در رده روش های جستجوی عمقی دانست ، نه می توان آن را یک روش جستجوی سطحی تلقی کرد. در این روش جستجو هر گره هزینه مصرف شده از

ریشه تا گره فعلی را در خود نگه می دارد. هنگامی که می خواهیم فرزندان گره گسترش نیافته ای را تولید کنیم، گـره ای از درخـت بـرای گسترش انتخاب می شود که کم هزینه ترین گره در میان گره های گسترش نیافته باشد. به عنوان مثال شکل ۴-۷ را در نظر بگیرید :



شکل ۴-۷

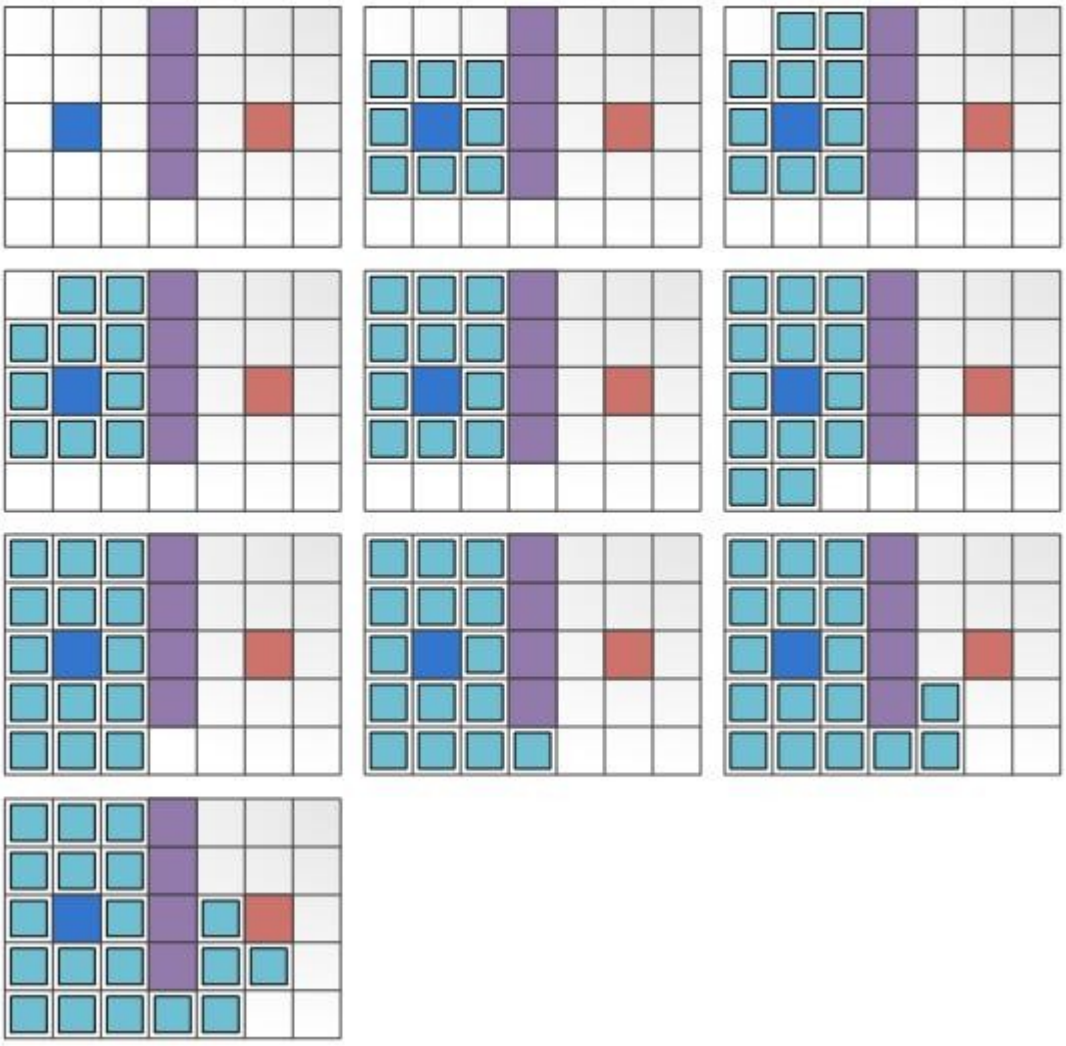
مسئله یافتن مسیری از مستطیل آبی رنگ به مستطیل قرمز رنگ می باشد. همچنین مستطیل های بنفش نشان دهنده دیوار هستند که نمی توان بر روی آن ها حرکت کرد. در این مسئله به ازای هر خانه ۸ خانه همسایه وجود دارد که در شکل ۴-۸ نشان داده شده اند (اعداد ترتیب گسترش گره ها را نشان می دهند) :



شکل ۴-۸

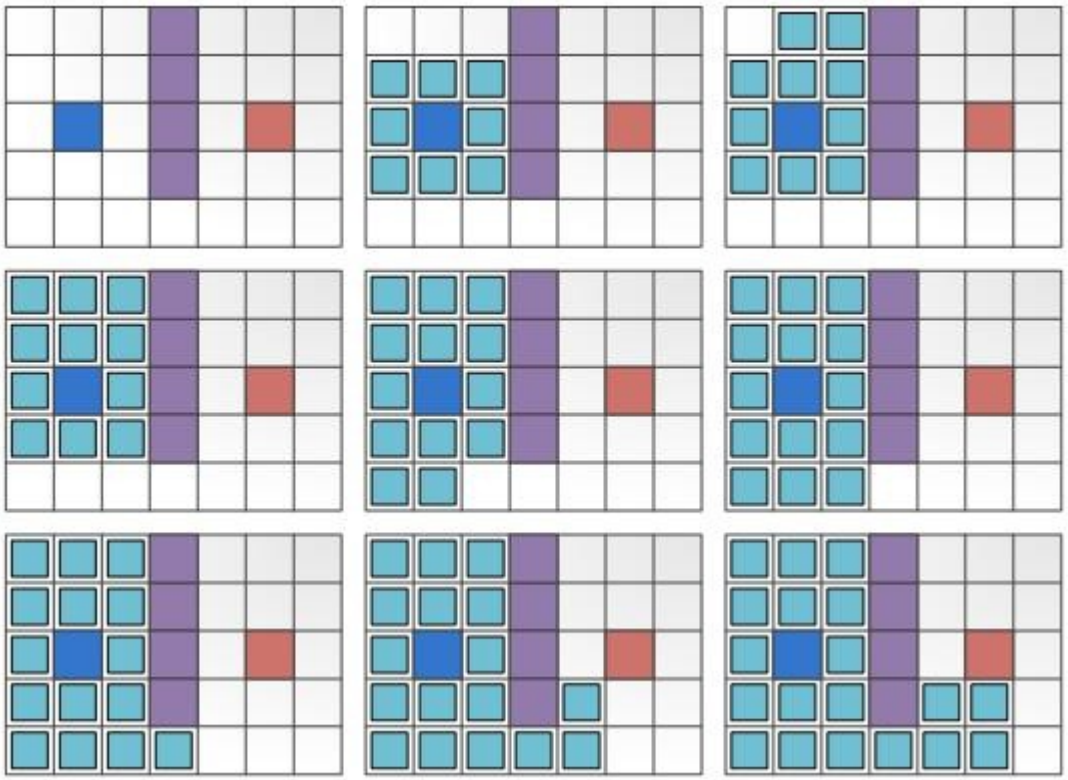
این بدان معناست که هنگام تولید فرزندان یک گره ، ۸ خانه همسایه برای گسترش وجود خواهد داشت. از طرف دیگر حرکات قطری مجـاز بوده و از یک خانه می توان چندبار عبور کرد. قبل از بررسی روش جستجوی هزینه یکنواخت با روش هـای جسـتجوی عمقـی ، سـطحی و عمقی محدود شده به حل این مساله می پردازیم. برای روش جستجوی عمقی بسته به اینکه ترتیب گسترش گره ها به چه نحـوی باشـد و

همچنین مبدا و مقصد در کدام قسمت نقشه قرار گیرند، امکان قرار گرفتن در حلقه بینهایت وجـود خواهـد داشـت. بنـابراین از ایـن روش استفاده نمی کنیم. در صورتی که از جستجوی سطحی برای حل مساله استفاده کنیم ، مسیر پیدا شده برای بدین صورت خواهد بود :



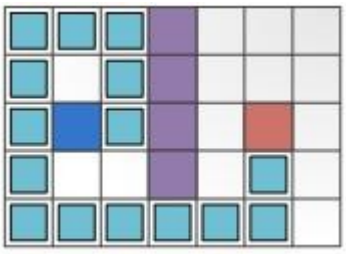
شکل ۴-۹

خانه های اضافی بسیاری را نیز برای یافتن مسیر گسترش می دهد. که این امر در مورد مسائل پیچیده تر خوشایند نخواهد بود. حال فرض کنید از روش جستجوی عمقی محدود شده به عمق ۲۰ برای یافتن مسیر استفاده کنیم. شکل ۴-۱۰ نحوه گسترش گره ها را نشان می دهد :



شکل ۴-۱۰

بنابراین مسیر پیدا شده از مبدا به مقصد با روش جستجوی عمقی محدوده شده به صورت زیر خواهد بود :



شکل ۴-۱۱

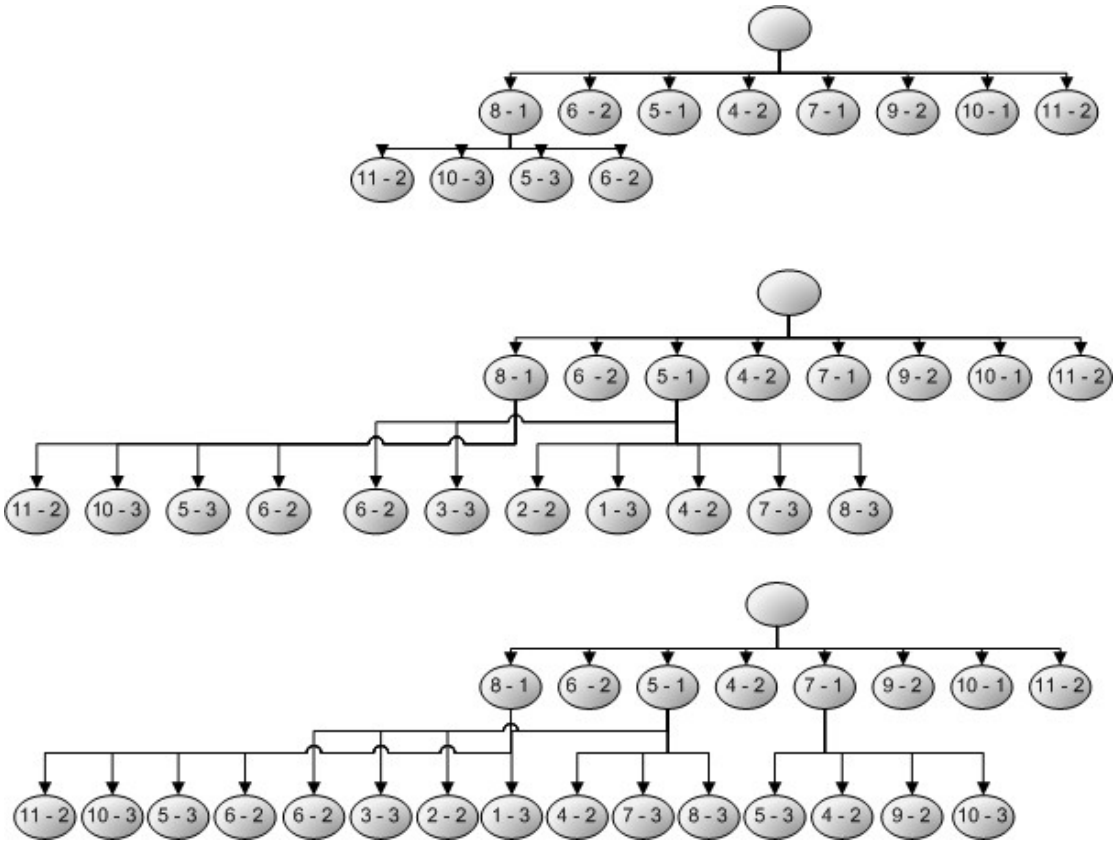
در مورد جستجوی عمقی محدود شده نیز با دو مشکل مواجه هستیم ، مشکل اول تعیین عمق محدود شده و مشکل دوم عدم بهینگی مسیر پیدا شده از مبدا به مقصد می باشد. در هیچ یک از روش های جستجو فوق هزینه از ریشه تا گره فعلی در نظر گرفته نشده است. حال فرض کنید هزینه انتقال از یک خانه به خانه دیگر در جهت افقی و عمودی ۱ و در جهت قطری ۱.۵ باشد. حال با این تعریف می توانیم از روش جستجوی هزینه یکنواخت برای یافتن مسیر استفاده کنیم. همانطور که در ابتدای بخش بیان کردیم ، در روش جستجوی

هزینه یکنواخت در هر لحظه گره ای از درخت گسترش می یابد که کمترین هزینه را در میان دیگر گره ها داشته باشد. قبل از نمایش نحوه تشکیل درخت جستجو به هر خانه شماره ای را انتساب می دهیم.

1	2	3		27	28	29
4	5	6		26	25	24
7		8		22		23
9	10	11		21	20	19
12	13	14	15	16	17	18

شکل ۴-۱۲

درختی که از جستجو به روش هزینه یکنواخت در هر مرحله به دست می آید به شکل ۴-۱۳ خواهد بود :



شکل ۴-۱۳

به دلیل بزرگ بودن درخت جستجو با استفاده از روش جستجوی هزینه یکنواخت در این مسئله ، تنها بخش کوچکی از آن را در اینجا نشان دادیم. بقیه گره ها نیز به همین ترتیب گسترش یافته تا اینکه در نهایت به گره هدف می رسیم. پس از ایجاد درخت خواهیم دید که مسیر بهینه ۲۱-۱۵-۱۱ به عنوان مسیر حرکت از مبدا به مقصد انتخاب می شود.

نکته قابل توجه در مورد جستجوی هزینه یکنواخت این است که این روش جستجو در صورتی جواب بهینه را پیدا می کند که هزینه های هر گام به درستی انتخاب شوند. مشکلی که همچنان در این روش باقی مانده ، گسترش گره های اضافی است که این امر موجب می شود پیدا کردن جواب برای مسئله زیاد سریع نباشد. به جای جستجوی هزینه یکنواخت می توان از جستجوی حریصانه نیز برای حل این مساله استفاده کرد که در بخش بعد به بررسی آن خواهیم پرداخت.

۴-۹) روش های جستجوی هیوریستیک:

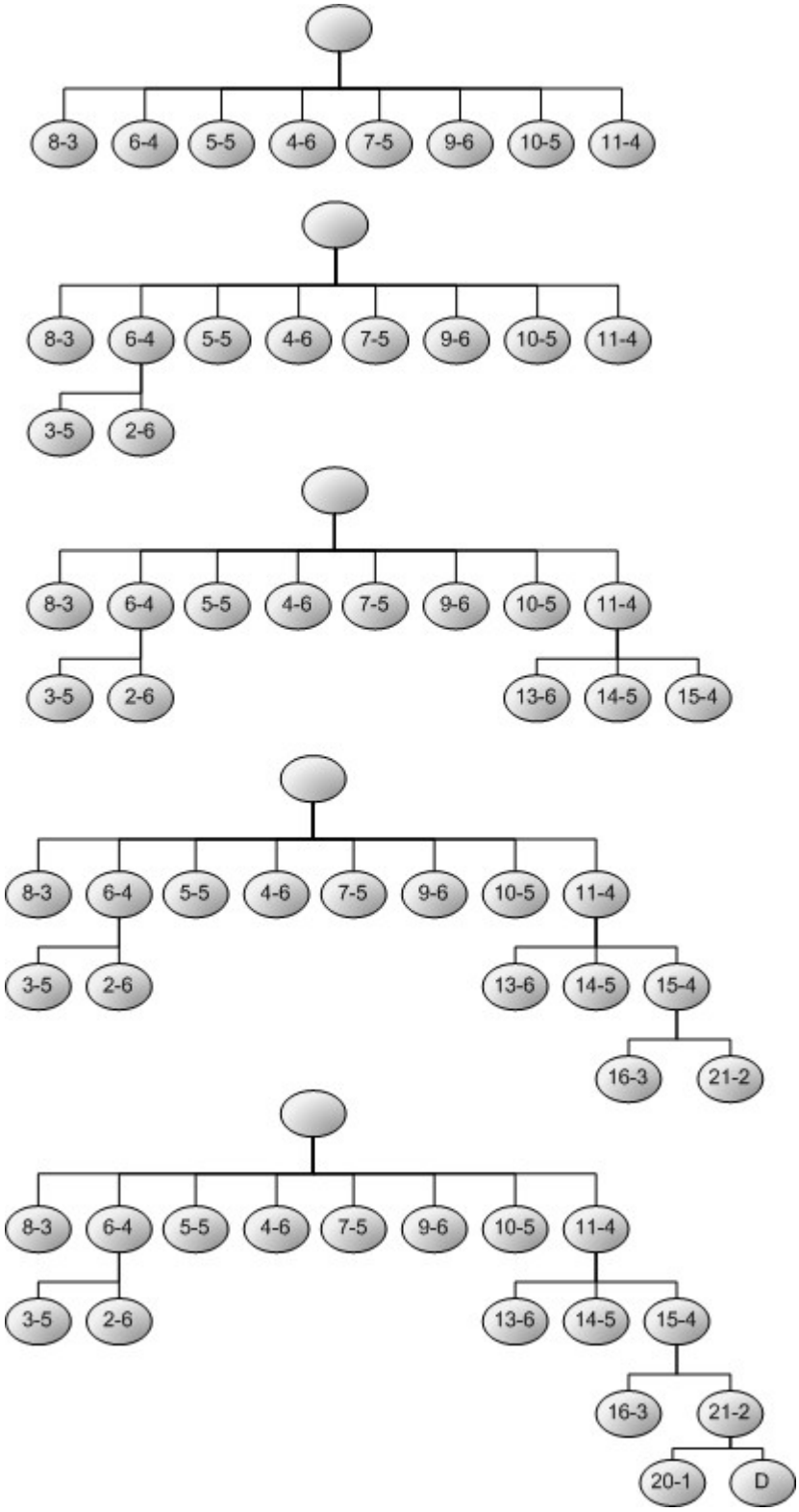
در روش های جستجوی عمقی، سطحی ، عمقی محدود شده ، عمقی تکرار شونده و هزینه یکتنواخت از هیچ تابعی برای تخمین میزان بهینگی هر گره استفاده نکردیم. تنها در جستجوی هزینه یکنواخت هزینه هر گام را برای سنجش میزان بهینگی گره ها به کار بردیم. اما در این روش جستجو نیز از اطلاعات مسئله برای تخمین میزان بهینگی هر گره بهره نبردیم. تابعی که عمل تخمین بهینگی هر گره را انجام می دهد ، تابع هیوریستیک می‌نامیم .

۴-۱۰) جستجوی حریصانه :

مسئله مسیریابی مطرح شده در بخش جستجوی هزینه یکنواخت را در نظر بگیرید. می توانیم به جای اینکه هزینه پیموده شده تا گره فعلی را به عنوان سنجه ای برای میزان بهینگی گره در نظر بگیریم، فاصله گره فعلی تا مقصد را به عنوان تخمینی برای سنجش میزان بهینگی هر گره انتخاب کنیم. به عبارت دیگر هنگام گسترش ، گره ای از درخت را گسترش دهیم که کمترین فاصله را تا مقصد دارد. در اینجا فرض می کنیم از هر خانه تنها یکبار می توانیم عبور کنیم. بنابراین تابع هیوریستیک به صورت زیر تعریف خواهد شد :

$$H(\text{گره}) = |X_{\text{گره}} - X_{\text{مقصد}}| + |Y_{\text{گره}} - Y_{\text{مقصد}}|$$

در این زیر مسئله گره ای برای گسترش مطلوب تر ر است که مقدار H آن کمتر باشد. درختی که با استفاده از این تابع هیوریستیک به دست می آید به شکل ۴-۱۴ خواهد بود :



شکل ۴-۱۴

همانطور که در ابتدای مساله فرض کردیم ، از هر خانه تنها یکبار می توان عبور کرد. این بدان معنی است که هر گره تنها یکبار می تواند در درخت می تواند گسترش یابد. علت گسترش نیافتن گره ۸ در سطح ۱ از درخت نیز به همین دلیل است. چرا که همه فرزندان این گره قبلا در درخت (سطح ۱) گسترش یافته اند.

درخت تشکیل شده از جستجوی حریصانه را با درخت تشکیل شده از جستجوی هزینه یکنواخت مقایسه کنید. می بینیم که جستجوی حزیصانه با گسترش گره های کمتر ، سریعتر به جواب مساله دست پیدا می کند. با این حال هنگام استفاده از جستجوی حریصانه باید دقت کنیم چرا که جستجوی حریصانه با دو مشکل بزرگ همراه است. مشکل اول اینکه نمی توان با استفاده از جستجوی حریصانه مطمئن بود که همیشه به جواب بهینه دست پیدا خواهیم کرد. همچنین جستجوی حریصانه ممکن است در یک حلقه بینهایت به دام افتد. به عنوان مثال در مسئله مسیریابی فرض کنید از هر خانه می توان چندین بار عبور کرد و با این فرض درخت جسنجو را تشکیل دهید. مشاهده خواهید کرد که گره شماره ۸ در یک حلقه بینهایت گسترش یافته و در واقع هیچگاه به جواب مسئله دست پیدا نخواهیم کرد. حال می توان علت نامگذاری این روش به جستجوی حریصانه را حدس زد.

دیدیم که جستجوی هزینه یکنواخت در صورتی که هزینه گام ها به درستی انتخاب شود، موجب یافتن جواب بهینه مسئله خواهد شد. در مقابل این روش با مشکل کند بودن عمل جستجو همراه است. در مقابل جستجوی حریصانه در یافتن جواب مسئله بسیار سریع بوده اما با مشکل حلقه بینهایت و عدم بهینگی جواب مواجه است (لازم به ذکر است که بهینگی جستجوی حریصانه در برخی مسائل همانند الگوریتم های پریم و کروسکال اثبات شده است. اما در حالت کلی نمی توان ادعا کرد همیشه جستجوی حریصانه منجر به یافتن جواب بهینه خواهد شد). حال این سوال مطرح می شود که چگونه می توان این دو روش را باهم ترکیب کرده و روش جستجویی را طراحی کنیم که هم سریع بوده و هم جواب بهینه مسئله را بتواند پیدا کند؟ جواب مسئله در روش جستجوی A است

۴-۱۱) منابع :

- **روش های جستجو در هوش مصنوعی ، مهلا قوی اندام**
- **آشنایی با الگوریتم CLRS ، کورمن ، لایسرسون ، روست ، استاین**
- **طراحی الگوریتم ها ، نیپولیتان ، نعیمی**

۵) مقدمه ای بر داده کاوی :

دو دهه قبل توانایی های فنی بشر در برای تولید و جمع آوری داده ها به سرعت افزایش یافته است. عواملی نظیر استفاده گسترده از بارکد برای تولیدات تجاری، به خدمت گرفتن کامپیوتر در کسب و کار، علوم، خدمات دولتی و پیشرفت در وسائل جمع آوری داده، از اسکن کردن متون و تصاویر تا سیستمهای سنجش از دور ماهواره ای، در این تغییرات نقش مهمی دارند.

بطور کلی استفاده همگانی از وب و اینترنت به عنوان یک سیستم اطلاع رسانی جهانی ما را مواجه با حجم زیادی از داده و اطلاعات می کند. این رشد انفجاری در داده های ذخیره شده، نیاز مبرم وجود تکنولوژی های جدید و ابزارهای خودکاری را ایجاد کرده که به صورت هوشمند به انسان یاری رسانند تا این حجم زیاد داده را به اطلاعات و دانش تبدیل کند: داده کاوی به عنوان یک راه حل برای این مسائل مطرح می باشد. در یک تعریف غیر رسمی داده کاوی فرآیندی است، خودکار برای استخراج الگوهایی که دانش را بازنمایی می کنند، که این دانش به صورت ضمنی در پایگاه داده های عظیم، انباره داده و دیگر مخازن بزرگ اطلاعات، ذخیره شده است. داده کاوی بطور همزمان از چندین رشته علمی بهره می برد نظیر: تکنولوژی پایگاه داده، هوش مصنوعی، یادگیری ماشین، شبکه های عصبی، آمار، شناسایی الگو، سیستم های مبتنی بر دانش، حصول دانش، بازیابی اطلاعات، محاسبات سرعت بالا و بازنمایی بصری داده . داده کاوی در اواخر دهه ۱۹۸۰ پدیدار گشته، در دهه ۱۹۹۰ گامهای بلندی در این شاخه از علم برداشته شده و انتظار ی رود در این قرن به رشد و پیشرفت خود ادامه دهد.واژه های «داده کاوی» و «کشف دانش در پایگاه داده» اغلب به صورت مترادف یکدیگر مورد استفاده قرار می گیرند.

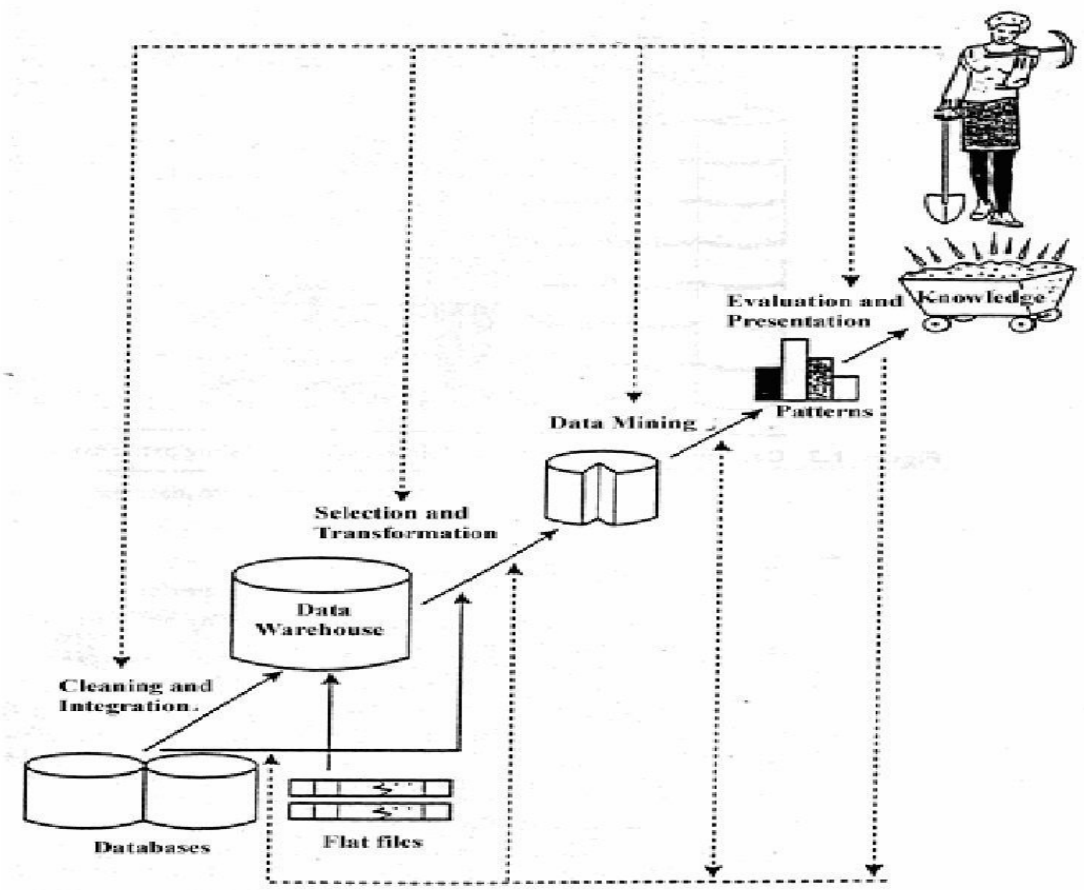
کشف دانش به عنوان یک فرآیند در شکل ۵-۱ نشان داده شده است.

کشف دانش در پایگاه داده فرایند شناسایی درست، ساده، مفید، و نهایتا الگوها و مدلهای قابل فهم در داده ها می باشد. داده کاوی، مرحله ای از فرایند کشف دانش می باشد و شامل الگوریتمهای مخصوص داده کاوی است، بطوریکه، تحت محدودیتهای مؤثر محاسباتی قابل قبول، الگوها و یا مدلها را در داده کشف می کند . به بیان ساده تر، داده کاوی به فرایند استخراج دانش ناشناخته، درست، و بالقوه مفید از داده اطلاق می شود. تعریف دیگر اینست که، داده کاوی گونه ای از تکنیکها برای شناسایی اطلاعات و یا دانش تصمیم گیری از قطعات داده می باشد، به نحوی که با استخراج آنها،در حوزه های تصمیم گیری، پیش بینی، پیشگویی، و تخمین مورد استفاده قرار گیرند. داده ها اغلب حجیم ، اما بدون ارزش می باشند، داده به تنهایی قابل استفاده نیست، بلکه دانش نهفته در داده ها قابل استفاده می باشد. به این دلیل اغلب به داده کاوی، تحلیل داده ای ثانویه گفته می شود.

۵-۱-۱) چه چیزی سبب پیدایش داده کاوی شده است؟

اصلی ترین دلیلی که باعث شد داده کاوی کانون توجهات در صنعت اطلاعات قرار بگیرد، مساله در دسترس بودن حجم وسیعی از داده ها و نیاز شدید به اینکه از این داده ها اطلاعات و دانش سودمند استخراج کنیم. اطلاعات و دانش بدست آمده در کاربردهای وسیعی از مدیریت کسب و کار وکنترل تولید و تحلیل بازار تا طراحی مهندسی و تحقیقات علمی مورد استفاده قرار می گیرد. داده کاوی را می توان حاصل سیر تکاملی طبیعی تکنولوژی اطلاعات دانست، که این سیر تکاملی ناشی از یک سیرتکاملی در صنعت پایگاه داده می باشد، نظیر عملیات: جمع آوری داده ها وایجاد پایگاه داده، مدیریت داده و تحلیل و فهم داده ها. در شکل ۵-۲ این روند تکاملی در پایگاه های داده نشان داده شده است .

کامل تکنولوژی پایگاه داده و استفاده فراوان آن در کاربردهای مختلف سبب جمع آوری حجم فراوانی داده شده است. این داده های فـراوان باعث ایجاد نیاز برای ابزارهای قدرتمند برای تحلیل داده ها گشته، زیرا در حال حاضر به لحـاظ داده ثروتمنـد هسـتیم ولـی دچـار کمبـود اطلاعات می باشـیم. ابزارهـای داده کـاوی داده هـا را آنـالیز مـی کننـد و الگوهـای دادهـای را کشـف مـی کننـد کـه مـی تـوان از آن در کاربردهایی نظیر: تعیین استراتژی برای کسب و کار، پایگاه دانش و تحقیقات علمی و پزشکی، استفاده کرد. شکاف موجـود بیـن داده هـا و اطلاعات سبب ایجاد نیاز برای ابزارهای داده کاوی شده است تا داده های بی ارزش را به دانشی ارزشمند تبدیل کنیم.



شکل ۵-۱ داده کاوی به عنوان یک مرحله از فرآیند کشف دانش

به طور ساده داده کاوی به معنای استخراج یا« معدن کاری» دانش از مقدار زیادی داده خام است. البته این نامگذاری برای این فرآیند تا حدی نامناسب است، زیرا به طور مثال عملیات معدن کاری برای استخراج طلا از صخره و ماسه را طلا کاوی می نامیم، نه ماسه کـاوی یـا صخره کاوی، بنابراین بهتر بود به این فرآیند نامی شبیه به «استخراج دانش از داد » می دادیم که متاسفانه بسـیار طـولانی اسـت. «دانـش کاوی» به عنوان یک عبارت کوتاهتر به عنوان جایگزین، نمی تواند بیانگر تاکید و اهمیت بر معدن کاری مقدار زیاد داده باشد. معدن کـاری عبارتی است که بلافاصله انسان را به یاد فرآیندی می اندازد که به دنبال یافتن مجموعه کوچکی از قطعات ارزشمند از حجم بسیار زیادی از مواد خام هستیم .

با توجه به مطالب عنوان شده، با اینکه این فرآیند تا حدی دارای نامگذاری نـاقص اسـت ولـی ایـن نامگـذاری یعنـی داده کـاوی بسـیار

عمومیت پیدا کرده است. البته اسامی دیگری نیز برای این فرآیند پیشنهاد شده که بعضا بسیاری متفاوت با واژه داده کاوی است، نظیر: استخراج دانش از پایگاه داده، استخراج دانش ،فرآیند داده / الگو، باستان شناسی داده، و لایروبی داده‌ها .

۵-۱-۲) مراحل کشف دانش کشف دانش دارای مراحل تکراری زیر است:

۱.پاکسازی داده ها (از بین بردن نویز و ناسازگاری داده ها).

۲. یکپارچه سازی داده (چندین منبع داده ترکیب می شوند).

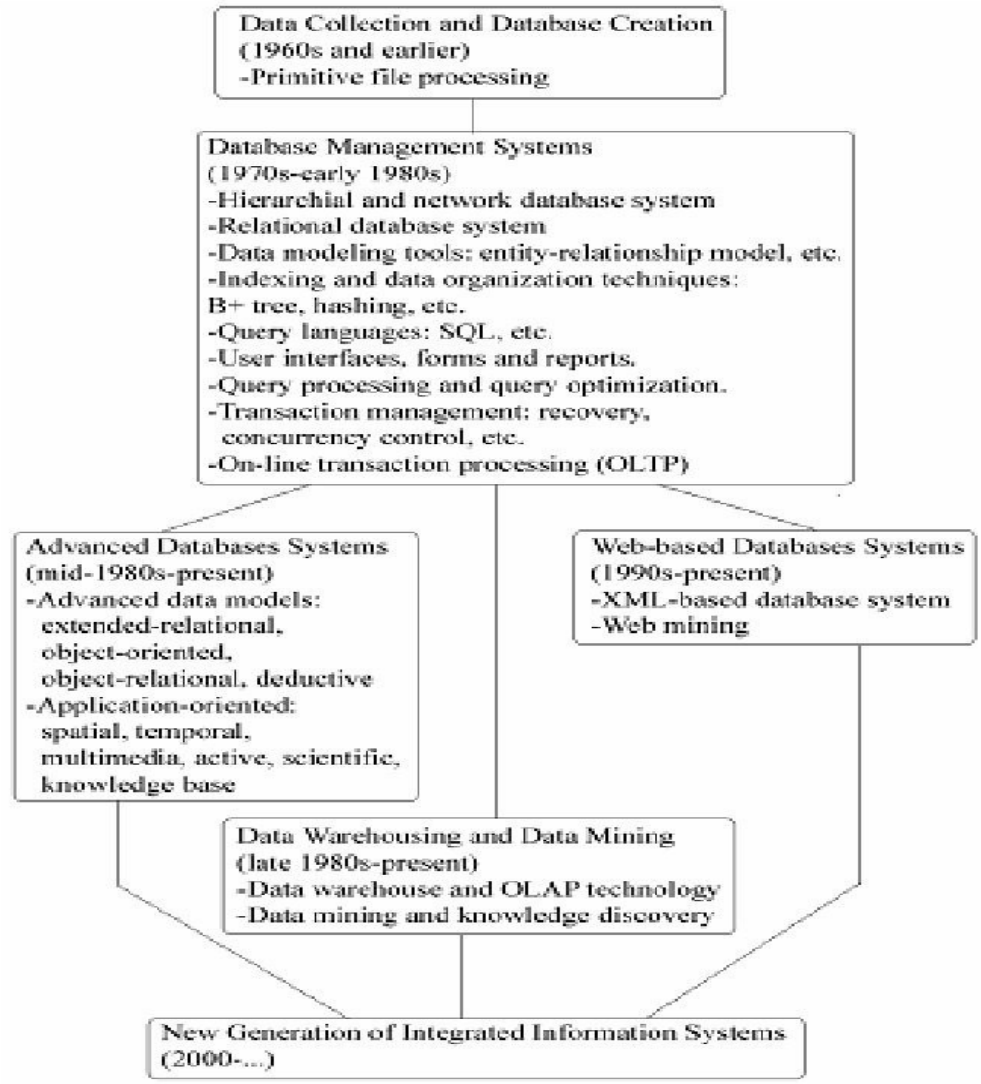
۳ .انتخاب داده‌ها (داده های مرتبط بآنالیزازپایگاه داده بازیابی می شوند)

۴ .تبدیل کردن داده‌ها (تبدیل داده ها به فرمی که مناسب برای داده کاوی باشد مثل خلاصه سازی و همسان سازی)

۵ . داده کاوی(فرایند اصلی که روالهای هوشمند برای استخراج الگوها از داده ها به کار گرفته می شوند) .

۶ . ارزیابی الگو (برای مشخص کردن الگوهای صحیح و مورد نظربه وسیله معیارهای اندازه گیری)

۷ . ارائه دانش (یعنی نمایش بصری، تکنیکهای بازنمایی دانش برای ارائه دانش کشف شده به کاربر استفاده می شود)



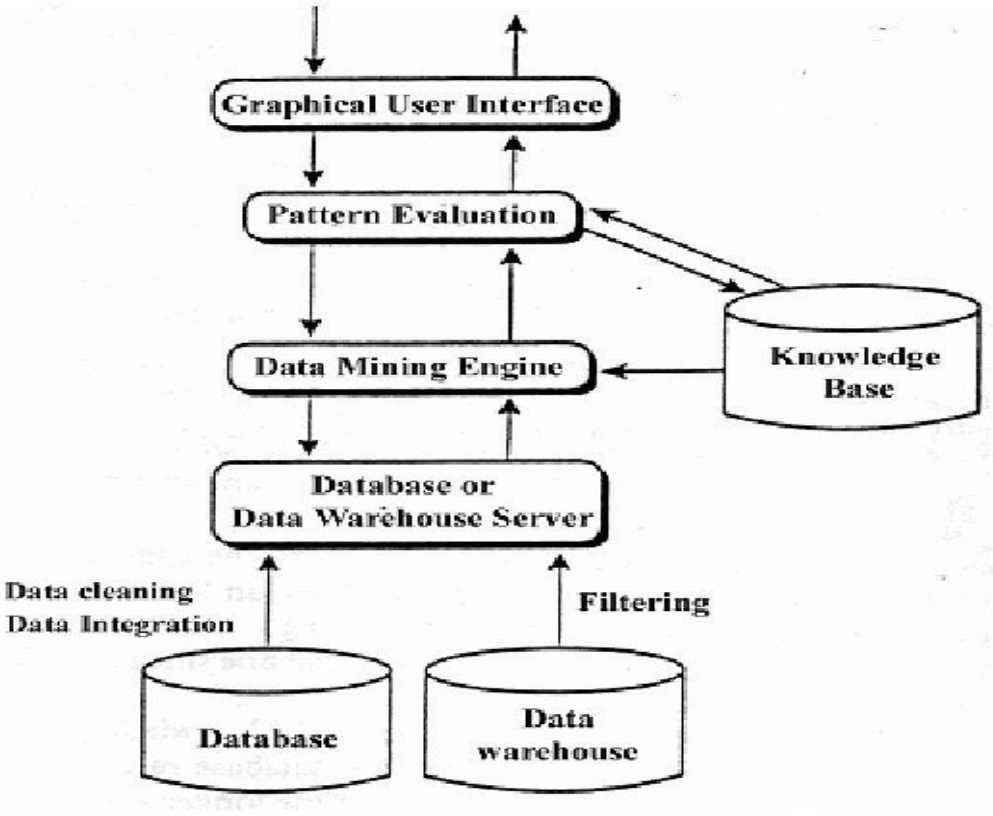
شکل ۲-۵ سیر تکاملی صنعت پایگاه داده

هر مرحله داده کاوی باید با کاربر یا پایگاه دانش تعامل داشته باشد. الگوهای کشف شده به کاربر ارائه می شوند و در صورت خواست او به عنوان دانش به پایگاه دانش اضافه می شوند. توجه شود که بر طبق این دیدگاه داده کاوی تنها یک مرحله از کل فرآیند است، البته به عنوان یک مرحله اساسی که الگوهای مخفی را آشکار می سازد. با توجه به مطالب عنوان شده، دراینجا تعریفی از داده کاوی ارائه می دهیم:

"داده کاوی عبارتست از فرآیند یافتن دانش از مقادیر عظیم داده های ذخیره شده در پایگاه داده، انباره داده ویا دیگر

مخازن اطلاعات"

براساس این دیدگاه یک سیستم داده کاوی به طور نمونه دارای اجزاء اصلی زیر است که شکل ۵-۳ بیانگر معماری سیستم است.



شکل ۵-۳معماری یک نمونه سیستم داده کاوی

۱. پایگاه داده، انباره داده یا دیگر مخازن اطلاعات: که از مجموعه ای از پایگاه داده ها، انباره داده، صفحه گسترده، یا دیگر انواع مخازن اطلاعات. پاکسازی داده ها و تکنیکهای یکپارچه سازی روی این داده ها انجام می شود.

۲ . سرویس دهنده پایگاه داده یا انباره داده: که مسئول بازیابی داده های مرتبط بر اساس نوع درخواست داده کاوی کاربر می باشد.

۳ . پایگاه دانش: این پایگاه از دانش زمینه تشکیل شده تا به جستجو کمک کند، یا برای ارزیابی الگوهای یافته شده ازآن استفاده می شود.

۴ . موتور داده کاوی : این موتور جزء اصلی از سیستم داده کاوی است و به طور ایدّال شامل مجموعه ای از پیمانه هایی نظیر توصیف، تداعی، کلاسبندی،آنالیزخوشه ها، وآنالیز تکامل وانحراف، است.

۵. پیمانه ارزیابی الگو : این جزء معیارهای جذابیت را به کار می بندد و با پیمانهء داده کاوی تعامل می کند بدینصورت که تمرکزآن بر جستجو بین الگوهای جذاب می باشد، و از یک حدآستانه جذابیت استفاده می کند تا الگوهای کشف شده را ارزیابی

کند.

۶ . واسط کاربرگرافیکی : این پیمانه بین کاربر و سیستم داده کاوی ارتباط برقرار می کند، به کاربر اجازه می دهد تا با سیستم داده کاوی از طریق پرس وجو ارتباط برقرار کند، این جزء به کاربر اجازه می دهد تا شمای پایگاه داده یا انباره داده را مرور کرده، الگوهای یافته شده را ارزیابی کرده و الگوها را در فرمهای بصری گوناگون بازنمایی کند. با انجام فرآیند داده کاوی، دانش، ارتباط یا اطلاعات سطح بالا از پایگاه داده استخراج می شود و قابل مرور از دیدگاههای مختلف خواهد بود. دانش کشف شده در سیستم های تصمیم یار، کنترل فرآیند، مدیریت اطلاعات و پردازش پرس وجود قابل استفاده خواهد بود .

بنابراین داده کاوی به عنوان یکی از شاخه های پیشرو در صنعت اطلاعات مورد توجه قرار گرفته و به عنوان یکی از نوید بخش ترین زمینه های توسعه بین رشته ای در صنعت اطلاعات است.

۵-۱-۳) جایگاه داده کاوی در میان علوم مختلف :

ریشه های داده کاوی در میان سه خانواده از علوم، قابل پیگیری می باشد . مهمترین این خانواده ها،آمار کلاسیک می باشد. بدون آمار، هیچ داده کاوی وجود نخواهد داشت، بطوریکه آمار، اساس اغلب تکنولوژی هایی می باشد که داده کاوی بر روی آنها بنا می شود.آمار کلاسیک مفاهیمی مانند تحلیل رگرسیون، توزیعاستاندارد ، انحراف استاندارد، واریانس، تحلیل خوشه، و فاصله های اطمینان را که همه این موارد برای مطالعه داده و ارتباط بین داده هامی باشد، را در بر می گیرد. مطمئنا تحلیل آماری کلاسیک نقش اساسی در تکنیکهای داده کاوی ایفا می کند.دومین خانواده ای که داده کاوی به آن تعلق دارد هوش مصنوعی می باشد. هوش مصنوعی که بر پایه روشهای ابتکاری می باشد و باآمار ضدیت دارد، تلاش دارد تا فرایندی مانند فکر انسان، را برای حل مسائل آماری بکار بندد.چون این رویکرد نیاز به توان محاسباتی بالایی دارد، تا اوایل دهه ۱۹۸۰ عملی نشد. هوش مصنوعی کاربردهای کمی رادر حوزه های علمی و حکومتی پیدا کرد، اما نیاز به استفاده از کامپیوترهای بزرگ باعث شد همه افراد نتوانند ازتکنیکهای ارائه شده استفاده کنند. سومین خانواده داده کاوی، یادگیری ماشین می باشد، که به مفهوم دقیقتر، اجتماع آمار و هوش مصنوعی می باشد.درحالیکه هوش مصنوعی نتوانست موفقیت تجاری کسب کند، یادگیری ماشین در بسیاری از موارد جایگزینآن گردید.از یادگیری ماشین به عنوان تحول هوش مصنوعی یاد شد، چون مخلوطی از روشهای ابتکاری هوش مصنوعی به همراه تحلیل آماری پیشرفته می باشد. یادگیری ماشین اجازه می دهد تا برنامه های کامپیوتری در مورد داده ای که آنها مطالعه می کنند، مانند برنامه هایی که تصمیمهای متفاوتی بر مبنای کیفیت داده مطالعه شده می گیرند، یادگیری داشته باشند و برای مفاهیم پایه ایآن ازآمار استفاده می کنند و از الگوریتمها و روشهای ابتکاری هوش مصنوعی رابرای رسیدن به هدف بهره می گیرند.داده کاوی در بسیاری از جهات، سازگاری تکنیکهای یادگیری ماشین با کاربردهای تجاری است. بهترین توصیف از داده کاو ی بوسیله اجتماع آمار، هوش مصنوعی و یادگیری ماشین بدست میآید. این تکنیکها سپس با کمک یکدیگر، برای مطالعه داده و پیدا کردن الگوهای نهفته درآنها استفاده می شوند.بعضی از کاربردهای داده کاوی به شرح زیر است:

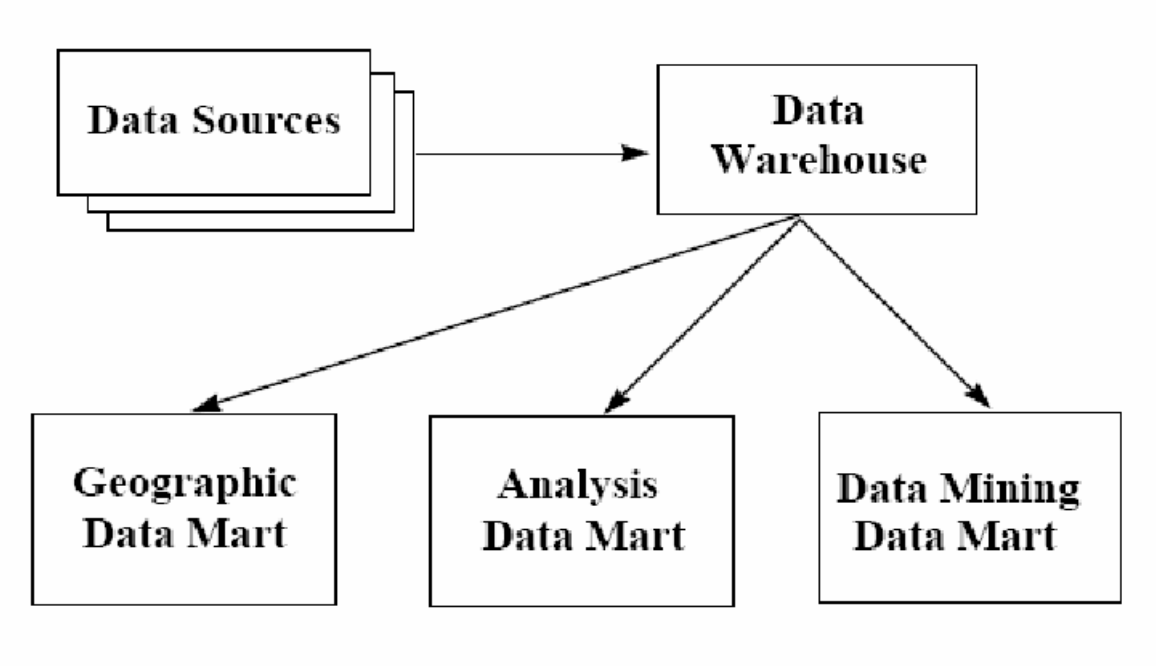
- کاربردهای معمول تجاری: از قبیل تحلیل و مدیریت بازار، تحلیل سبد بازار، بازاریابی هدف، فهم رفتار مشتری، تحلیل و مدیریت ریسک؛
- مدیریت و کشف فریب: کشف فریب تلفنی، کشف فریبهای بیمه ای و اتومبیل، کشف حقه های کارت اعتباری، کشف تراکنشهای مشکوک مالی (پولشویی) ؛
- وب کاوی : پیشنهاد صفحات مرتبط، بهبود ماشینهای جستجوگر یا شخصی سازی حرکت در وب سایت؛

۵-۱-۴) داده کاوی چه کارهایی نمی تواند انجام دهد؟

داده کاوی فقط یک ابزار است و نه یک عصای جادویی. داده کاوی به این معنی نیست که شما راحت به کناری بنشینید و ابزارهای داده کاوی همه کار را انجام دهد. داده کاوی نیاز به شناخت داده ها و ابزارهای تحلیل و افراد خبره در این زمینه ها را از بین نمی برد. داده کاوی فقط به تحلیلگران برای پیدا کردن الگوها و روابط بین داده ها کمک می کند و در این مورد نیز روابطی که یافته می شود باید به وسیله داده های واقعی دوباره بررسی و تست گردد.

۵-۱-۵) داده کاوی و انبار داده ها:

معمولا داده هایی که در داده کاوی مورد استفاده قرار می گیرند از یک انبار داده استخراج می گردند و در یک پایگاه داده یـا مرکـز داده ای ویژه برای داده کاوی قرار می گیرند. اگر داده های انتخابی جزیی از انبار داده ها باشند بسیار مفید است چون بسیاری از اعمالی که برای ساختن انباره داده ها انجام می گیرد با اعمال مقدماتی داده کاوی مشترک است و در نتیجه نیاز به انجام مجدد این اعمال وجود نـدارد ، از جمله این اعمال پاکسازی داده ها می باشد. پایگاه داده مربوط به داده کاوی می تواند جزیی از سیستم انبار داده ها باشد و یا می تواند یک پایگاه داده جدا باشد.

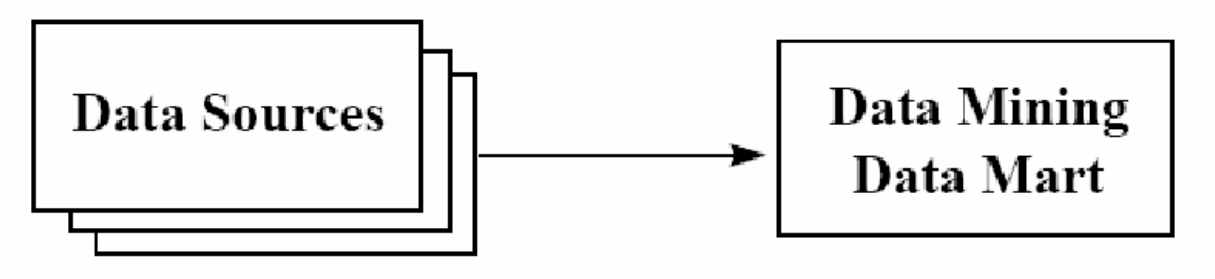


شکل ۴ . *داده ها از انباره داه ها استخراج می گردند*

شکل ۵-۴

ولی با این حال وجود انباره داده ها برای انجام داده کاوی شرط لازم نیست و بدون آن هم اگر داده ها دریک یا چنـدین پایگـاه داده باشـند می توان داده کاوی را انجام دهیم و بدین منظور فقط کافیست داده ها را در یک پایگاه داده جمع آوری کنیم و اعمال جـامعیت داده هـا و

پاکسازی داده ها را روی آن انجام دهیم. این پایگاه داده جدید مثل یک مرکز داده ای عمل می کند.



شکل ۵ .داده ها از چند پایگاه داده استخراج شده اند

شکل ۵-۵

۴-۱-۶) داده کاوی و OLAP

بسیاری فکر می کنند که داده کاوی و OLAP دو چیز مشابه هستند در این بخش سعی می کنیم این مسئله را بررسی کنیم و همانطور که خواهیم دید این دو ابزار های کاملا متفاوت می باشند که می توانند همدیگر را تکمیل کنند.

OLAP جزیی از ابزارهای تصمیم گیری می باشد.سیستم های سنتی گزارش گیری و پایگاه داده‌ای آنچه را که در پایگاه داده بود توضیح می دادند حالآنکه در OLAP هدف بررسی دلیل صحت یک فرضیه است.

بدین معنی که کاربر فرضیه ای در مورد داده ها و روابط بین آنها ارائه می کند و سپس به وسیله ابزار OLAP با انجام چند Query صحتآن فرضیه را بررسی می کند. اما این روش برای هنگامی که داده ها بسیار حجیم بوده و تعداد پارامترها زیاد باشد نمی تواند مفید باشد چون حدس روابط بین داده ها کار سخت و بررسی صحتآن بسیار زمانبر خواهد بود.

تفاوت داده کاوی با OLAP در این است که داده کاوی برخلاف OLAP برای بررسی صحت یک الگوی فرضی استفاده نمی شود بلکه خود سعی می کند این الگوها را کشف کند. درنتیجه داده کاوی و OLAP می توانند همدیگر را تکمیل کنند و تحلیل گر می تواند به وسیله ابزار OLAP یک سری اطلاعات کسب کند که در مرحله داده کاوی می تواند مفید باشد و همچنین الگوها و روابط کشف شده در مرحله داده کاوی می تواند درست نباشد که با اعمال تغییرات درآنها می توان به وسیله OLAP بیشتر بررسی شوند.

۵-۱-۷) کاربرد یادگیری ماشین وآمار در داده کاوی

داده کاوی از پیشرفت هایی که در زمینه هوش مصنوعی وآمار رخ می دهد بهره می گیرد . هر دو این زمینه ها در مسائل شناسایی الگو و طبقه بندی داده ها کار می کنند و بالطبع در داده کاوی استفاده مستقیم خواهند داشت. و هر دو گروه در شناخت و استفاده از شبکه های

عصبی و درخت های تصمیم گیری فعال می باشند.
داده کاوی جانشین تکنیک هایآماری سابق نمی باشد بلکه وارثآنها بوده و در واقع تغییر و گسترش تکنیک های سابق برای متناسب ساز یآنها با حجم داده ها و مسٔایل امروزی می باشد. تکنیک های کلاسیک برای داده های محدود و مسائل ساده مناسب بوده اند حالآنکه با پیچیده شدن مسائل و رشد روزافزون داده ها نیاز به تغییرآنها کاملا طبیعی است.به عبارت دیگر داده کاوی ترکیب تکنیک های کلاسیک با الگوریتم های جدید مثل شبکه های عصبی و درخت تصمیم گیری می باشد.
مهمترین نکته این است که داده کاوی راهکاری است برای مسائل تجاری امروز به کمک تکنیک هایآماری و هوش مصنوعی برای افراد حرفه ای که قصد دارند یک مدل پیش بینی ایجاد نمایند.

۵-۲) توصیف داده ها در داده کاوی :

۵-۲-۱) خلاصه سازی و به تصویر در آوردن داده ها :

قبل از اینکه بتوان روی مجموعه ای از داده ها ،داده کاوی انجام بدهیم و یک مدل پیش بینی مناسب ایجاد کنیم ، باید بتوان داده ها را به خوبی شناخت که برا ی شروع این کار می توان از پارامترهایی مثل میانگین , انحراف معیار و... استفاده کنیم.
ابزارهای تصویرسازی داده ها و گراف سازی برای شناخت داده ها بسیار مفید می باشند و نقش آنها در آماده سازی داده ها بسیار مفید و غیر قابل انکار است ، مثلا با استفاده از این ابزار می توان توزیع مقادیر مختلف داده ها را در یک نمودار مشاهده کرد و میزان داده های دارای خطا را به طور تقریبی حدس زد.
مهمترین مشکل این ابزار این است که معمولا تحلیل ها دارای تعداد زیادی پارامتر هستند که به هم مربوطند و باید رابطه این پارامترها را که چند بعدی می باشد در دو بعد نمایش دهند که این کار اگر هم عملی باشد برای استفاده از آنها نیاز به افراد خبره می باشد.

۵-۲-۲) خوشه بندی :

هدف از خوشه بندی این است که داده های موجود را به چند گروه تقسیم کنند و در این تقسیم بندی داده های گروه های مختلف باید حداکثر تفاوت ممکن را به هم داشته باشند و داده های موجود در یک گروه باید بسیار به هم شبیه باشند .
برخلاف کلاس بندی (که در ادامه خواهیم دید) در خوشه بندی ، گروه ها از قبل مشخص نمی باشند و همچنین معلوم نیست که بر حسب کدام خصوصیات گروه بندی صورت می گیرد. درنتیجه پس از انجام خوشه بندی باید یک فرد خبره خوشه های ایجاد شده را تفسیر کند و در بعضی مواقع لازم است که پس از بررسی خوشه ها بعضی از پارامترهایی که در خوشه بندی در نظر گرفته شده اند ولی بی ربط بوده یا اهمیت چندانی ندارند حذف شده و جریان خوشه بندی از اول صورت گیرد .
پس از اینکه داده ها به چند گروه منطقی و توجیه پذیر تقسیم شدند از این تقسیم بندی می توان برای کسب اطلاعات در مورد داده ها یا تقسیم داده ها جدید استفاده کنیم.
از مهمترین الگوریتم هایی که برای خوشه بندی استفاده می شوند می توان Kohnen و الگوریتم K-means را نام برد.

۵-۲-۳) تحلیل لینک :

تحلیل داده ها یکی از روش های توصیف داده هاست که به کمک آن داده ها را بررسی کرده و روابط بین مقادیر موجود در بانک اطلاعاتی را کشف می کنیم.از مهمترین راههای تحلیل لینک کشف وابستگی و کشف ترتیب می باشد.
منظور از کشف وابستگی یافتن قوانینی در مورد مواردی است که با هم اتفاق می افتند مثلا اجناسی که در یک فروشگاه احتمال خرید

همزمان آنها زیاد است.
کشف ترتیب نیر بسیار مشابه می باشد ولی پارامتر زمان نیز در آن دخیل می باشد.
وابستگی ها به صورت A->B نمایش داده می شوند که به A مقدم و به B موخر یا نتیجه گفته می شود.مثلا اگر یک قانون به صورت زیر داشته باشیم :

" اگر افراد چکش بخرند آنگاه آنها میخ خواهند خرید "
در این قانون مقدم خرید چکش و نتیجه خرید میخ می باشد .

۵-۳) مدل های پیش بینی داده ها :

۵-۳-۱) Classification:

در مسائل classification هدف شناسایی ویژگیهایی است که گروهی را که هر مورد به آن تعلق دارد را نشان دهند. از این الگو می توان هم برای فهم دادههای موجود و هم پیشبینی نحوه رفتار مواد جدید استفاده کرد.
داده کاوی مدلهای classification را با بررسی داده های دسته بندی شده قبلی ایجاد می کند و یک الگوی پیش بینی کننده را بصورت استقرایی مییابند. این موارد موجود ممکن است از یک پایگاه داده تاریخی آمده باشند.

۵-۳-۲) Regression

Regression از مقادیر موجود برای پیشبینی مقادیر دیگر استفاده میکند. در سادهترین فرم، regression از تکنیکهای آماری استاندارد مانند linear regression استفاده میکند. متاسفانه، بسیاری مسائل دنیای واقع تصویرخطی سادهای از مقادیر قبلی نیستند. بنابراین تکنیکهای پیچیده تری(logistic regression، درختهای تصمیم، یا شبکههای عصبی)ممکن است برای پیش بینی مورد نیاز باشند.
انواع مدل یکسانی را میتوان هم برای regression و هم برای classification استفاده کرد. برای مثال الگوریتم درخت تصمیم CARTرا میتوان هم برای ساخت درختهای classification و هم درختهای regression استفاده کرد. شبکههای عصبی را نیز میتوان برای هر دو مورد استفاده کرد .

۵-۳-۳) Time series

پیش بینی های Time series مقادیر ناشناخته آینده را براساس یک سری از پیشبینی گرهای متغیر با زمان پیش بینی می کنند. و مانند regression، از نتایج دانسته شده برای راهنمایی پیشبینی خود استفاده می کنند. مدلها باید خصوصیات متمایز زمان را در نظر گیرند و بویژه سلسله مراتب دوره ها را.

۵-۴) مدل ها و الگوریتم های داده کاوی :

در این بخش قصد داریم مهمترین الگوریتم ها و مدل های داده کاوی را بررسی کنیم. بسیاری از محصولات تجاری داده کاوی از مجموعه از این الگوریتم ها استفاده می کنند و معمولا هر کدام آنها در یک بخش خاص قدرت دارند وبرای استفاده از یکی از آنها باید بررسی های لازم در جهت انتخاب متناسب ترین محصول توسط گروه متخصص در نظر گرفته شود. نکته مهم دیگر این است که در بین این الگوریتم ها

و مدل ها ، بهترین وجود ندارد و با توجه به داده ها و کارایی مورد نظر باید مدل انتخاب گردد.

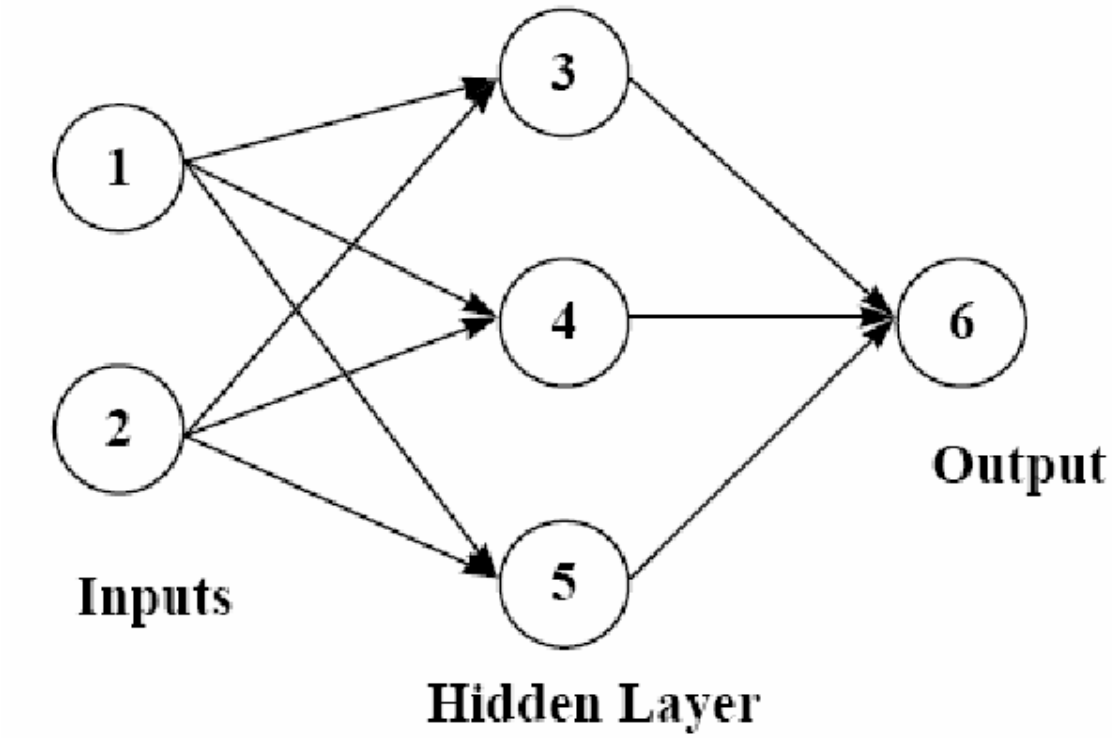
۵-۴-۱) شبکه های عصبی :

شبکه های عصبی از پرکاربردترین و عملی ترین روش های مدل سازی مسائل پیچیده و بزرگ که شامل صدها متغیر هستند می باشد. شبکه های عصبی می توانند برای مسائل کلاس بندی(که خروجی یک کلاس است) یا مسائل رگرسیون(که خروجی یک مقدار عددی است) استفاده شوند.

هر شبکه عصبی شامل یک لایه ورودی می باشد که هر گره در این لایه معادل یکی از متغیرهای پیش بینی می باشد. گره های موجود در لایه میانی وصل می شوند به تعدادی گره در لایه نهان . هر گره ورودی به همه گره های لایه نهان وصل می شود.

گره های موجود در لایه نهان می توانند به گره های یک لایه نهان دیگر وصل شوند یا می توانند به لایه خروجی وصل شوند .

لایه خروجی شامل یک یا چند متغیر خروجی می باشد.



شبکه عصبی با یک لایه نهان

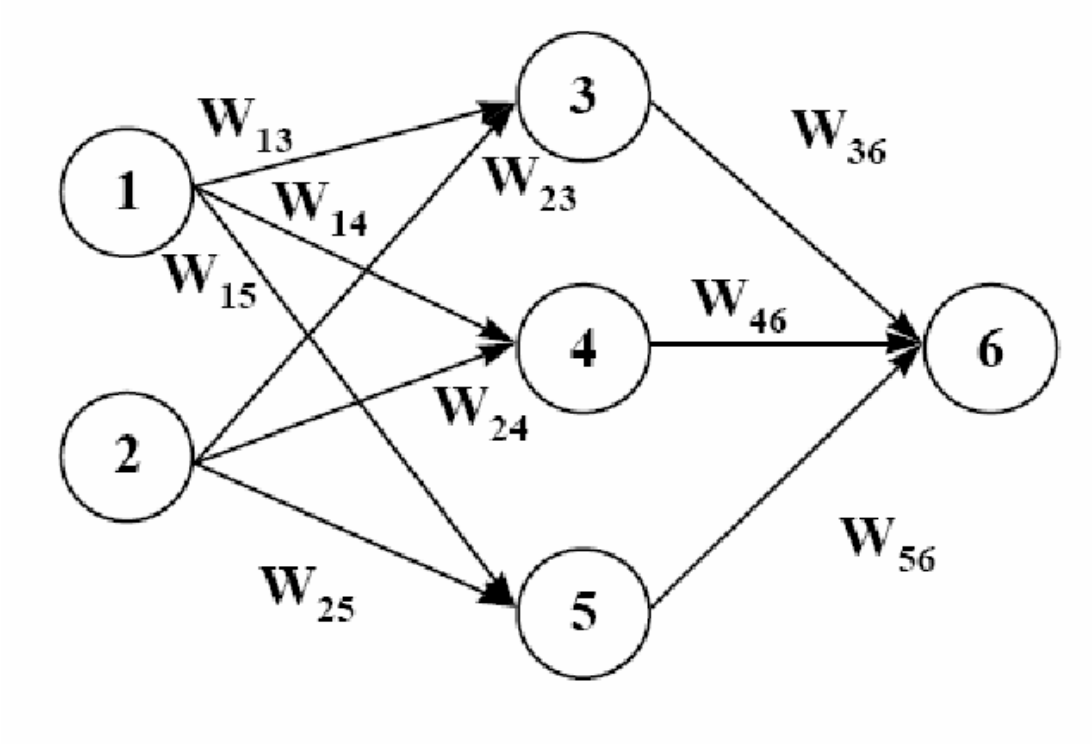
شکل ۵-۶

هر یال که بین نود های X,Y می باشد دارای یک وزن است که با Wx,y نمایش داده می شود. این وزن ها در محاسبات لایه های میانی استفاده می شوند و طرز استفاده آنها به این صورت است که هر نود در لایه های میانی (لایه های غیر از لایه اول) دارای چند ورودی از چند یال مختف می باشدکه همانطور که گفته شد هر کدام یک وزن خاص دارند .

هر نود لایه میانی میزان هر ورودی را در وزن یال مربوطه آن ضرب می کند و حاصل این ضرب ها را با هم جمع می کند و سپس یک تابع از پیش تعیین شده)تابع فعال سازی(روی این حاصل اعمال می کند و نتیجه را به عنوان خروجی به نودهای لایه بعد می دهد.

وزن یال ها پارامترهای ناشناخته ای هستند که توسط تابع آموزش و داده های آموزشی که به سیستم داده می شودتعیین می گردند.

تعداد گره ها و تعداد لایه های نهان و نحوه وصل شدن گره ها به یکدیگر معماری)توپولوژی(شبکه عصبی را مشخص می کند. کاربر یا نرم افزاری که شبکه عصبی را طراحی می کند باید تعداد نودها ، تعداد لایه های نهان ، تابع فعال سازی و محدودیت های مربوط به وزن یال ها را مشخص کند.



$W_{x,y}$ وزن یال بین X و Y است.

شکل ۵-۷

از مهمترین انواع شبکه های عصبی Feed-Forward Backpropagation می باشد که در اینجا به اختصار آنرا توضیح می دهیم.

Feed-Forward به معنی این است که مقدار پارامتر خروجی براساس پارامترهای ورودی و یک سری وزن های اولیه تعیین می گردد. مقادیر ورودی با هم ترکیب شده و در لایه های نهان استفاده می شوند و مقادیر این لایه های نهان نیز برای محاسبه مقادیر خروجی ترکیب می شوند.

Backpropagation : خطای خروجی با مقایسه مقدار خروجی با مقدار مد نظر در داده های آزمایشی محاسبه میگردد و این مقدار برای تصحیح شبکه و تغییر وزن یال ها استفاده می گردد و از گره خروجی شروع شده و به عقب محاسبات ادامه می یابد.

این عمل برای هر رکورد موجود در بانک اطلاعاتی تکرار می گردد.

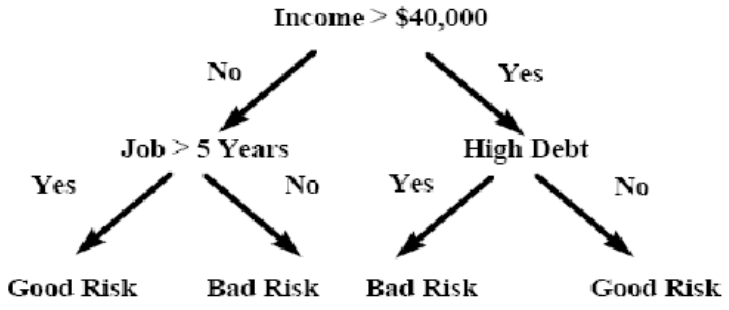
به هر بار اجرای این الگوریتم برای تمام داده های موجود در بانک یک دوره گفته می شود. این دوره ها آنقدر ادامه می یابد که دیگر مقدار خطا تغییر نکند. از آنجایی که تعداد پارامترها در شبکه های عصبی زیاد می باشد محاسبات این شبکه ها می تواند وقت گیر باشد. ولی اگر این شبکه ها به مدت کافی اجرا گردند معمولا موفقیت آمیز خواهند بود. مشکل دیگری که ممکن است به وجود بیاید Overfittingمی باشد و آن بدین صورت است که که شبکه فقط روی داده ها آموزشی خوب کار می کند و برای سایر مجموعه داده ها مناسب نمی باشد. برای رفع این مشکل ما باید بدانیم چه زمانی آموزش شبکه را متوقف کنیم.یکی از راه ها این است که شبکه را علاوه بر داده های آزمایشی روی داده های تست نیز مرتبا اجرا کنیم و جریان تغییر خطا را در آنها بررسی کنیم.اگر در این داده ها به جایی رسیدیم که میزان خطا رو به افزایش بود حتی اگر خطا در داده های آزمایشی همچنان رو به کاهش باشد آموزش را متوقف کنیم.

از آنجایی که پارامترهای شبکه های عصبی زیاد است یک خروجی خاص می تواند با مجموعه های مختلفی از مقادیر پارامترها ایجاد گردد درنتیجه این پارامترها مثل وزن یالها قابل تفسیر نبوده و معنی خاصی نمی دهند .

یکی از مهمترین فواید شبکه های عصبی قابلیت اجرای آنها روی کامپیوترهای موازی می باشد.

Decision trees (۲-۴-۵

درختهای تصمیم روشی برای نمایش یک سری از قوانین هستند که منتهی به یک رده یا مقدار میشوند. برای مثال، میخـواهیم متقاضـیان وام را به دارندگان ریسک اعتبار خوب و بد تقسیم کنیم. شکل یک درخت تصمیم را که این مسئله را حل می کنـد نشـان میدهـد و همـه مؤلفه های اساسی یک یک درخت تصمیم در آن نشان داده شده است : نود تصمیم، شاخه ها و برگها.



درخت تصمیم گیری

شکل ۵-۸

براساس الگوریتم، ممکن است دو یا تعداد بیشتری شاخه داشته باشد. برای مثال، CART درختانی با تنها دو شاخه در هر نود ایجاد میکند. هر شاخه منجر به نود تصمیم دیگر یا یک نود برگ میشود. با پیمایش یک درخت تصمیم از ریشه به پایین به یک مورد یک رده یا مقدار نسبت میدهیم. هر نود از دادههای یک مورد برای تصمیم گیری دربارهآن انشعاب استفاده میکند.

درختهای تصمیم از طریق جداسازی متوالی دادهها به گروههای مجزا ساخته میشوند و هدف در این فرآیند افزایش فاصله بین گروههـا در هر جداسازی است.

یکی از تفاوتها بین متدهای ساخت درخت تصمیم اینستکه این فاصله چگونه اندازهگیری میشود. درختهای تصمیمی که برای پیشبینی متغیرهای دستهای استفاده میشوند، درختهای classification نامیده میشوند زیرا نمونه ها را در دسته ها یا رده ها قرار میدهند. درختهای تصمیمی که برای پیشبینی متغیرهای پیوسته استفاده میشوند درختهای regression نامیده میشوند.

هر مسیر در درخت تصمیم تا یک برگ معمولا قابل فهم است. از این لحاظ یک درخت تصمیم میتواند پیش بینی های خود را توضیح دهد، که یک مزیت مهم است. با این حال این وضوح ممکن است گمراه کننده باشد. برای مثال،جداسازی های سخت در درختهای تصمیم دقـتی را نشان میدهند که کمتر در واقعیت نمود دارند. (چرا باید کسی که حقوق او ۴۰۰۰۰۱ است از نظر ریسک اعتبـار خـوب باشـد درحـالیکه کسی که حقوقش ۴۰۰۰۰ است بد باشد. بعلاوه، از آنجاکه چندین درخت میتوانند دادههای مشابهای را با دقـت مشـابه نشـان دهنـد، چـه تفسیری ممکن است از قوانین شود؟

درختهای تصمیم تعداد دفعات کمی از دادهها گذر میکنند(برای هر سطح درخت حداکثر یک مرتبه) و با متغیرهای پیش بینی کننده زیـاد بخوبی کار میکنند. درنتیجه، مدلها بسرعت ساخته میشوند، که آنها را برای مجموعهداده های بسیار مناسب می سازد. اگر به درخـت اجـازه دهیم بدون محدودیت رشد کند زمان ساخت بیشتری صرف میشود که

غیرهوشمندانه است، اما مسئله مهمتر اینستکه با داده ها overfit می شوند. اندازه درختها را میتوان از طریق قوانین توقف کنترل کرد. یک قانون معمول توقف محدود کردن عمق رشد درخت است.

راه دیگر برای توقف هرس کردن درخت است. درخت میتواند تا اندازه نهایی گسترش یابد، سپس با استفاده از روش های اکتشافی توکار یـا با مداخله کاربر، درخت به کوچکترین اندازهای که دقت در آن از دست نرود کاهش مییابد.

یک اشکال معمول درختهای تصمیم اینستکه آنها تقسیمکردن را براساس یک الگوریتم حریصـانه انجـام میدهنـد کـه در آن تصـمیمگیری اینکه براساس کدام متغیر تقسیم انجام شود، اثرات این تقسیم در تقسیمهای آینده را درنظر نمی گیرد.

بعلاوه الگوریتمهایی که برای تقسیم استفاده می شوند، معمولا تک متغیری هستند: یعنی تنها یک متغیر را در هر زمـان در نظـر میگیرنـد. درحالیکه این یکی از دلایل ساخت سری مدل است، تشخیص رابطه بین متغیرهای پیش بینی کننده راسخت تر می کند.

۵-۴-۳) MARS (Multivariate Adaptive Regression Splines) :

در میانه های دهه ۸۰ یکی از مخترعین CART، Jerome H. Friedman_، متدی را برای برطرف کردن این کاستی ها توسعه داد.

کاستیهای اساسی که او قصد برطرف کردن آنها را داشت عبارتند از :

- پیشبینی های غیرپیوسته(تقسیم سخت)
- وابستگی همه تقسیم ها به تقسیم های قبلی

به این دلیل او الگوریتم MARS را توسعه داد. ایده اصلی MARS نسبتا ساده است، درحالیکه خود الگوریتم نسبتاً پیچیده است. بسیار ساده ایده عبارت است از :

- جایگزینی انشعابهای غیرپیوسته با گذر های پیوسته که توسط یک جفت از خطهای مستقیم مدل میشوند.

در انتهای فرآیند ساخت مدل، خطوط مستقیم در هر نود با یک تابع بسیارهموار که spline نامیده میشودجایگزین میشوند.

- عدم نیاز به اینکه تقسیمهای جدید وابسته به تقسیمهای قدیمی باشند.

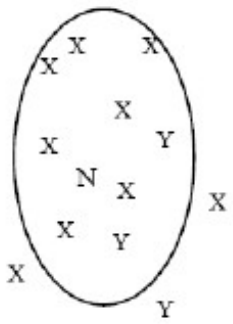
متأسفانه این به معنی اینست که MARS ساختار درختی CART را ندارد و نمیتواند قوانینی را ایجاد کند. از طرف دیگر، MARS به صورت خودکار مهمترین متغیرهای پیشبینی کننده و همچنین تعامل میان آنها را مییابد. همچنین MARS وابستگی میان پاسخ و هر پیشبینی کننده را معین میکند. نتیجه ابزار رگرسیون اتوماتیک، خودکار wise step است . MARS، مانند بیشتر الگوریتمهای شبکههای عصبی و درخت تصمیم، تمایل به overfit شدن برای داده های آموزش دهنده دارد. که میتوان آنرا به دو طریق درست کرد. اول اینکه، cross validation بصورت دستی انجام شود و الگوریتم برای تولید پیشبینی خوب روی مجوعه تست تنظیم شود. دوم اینکه، پارامترهای تنظیم متفاوتی در خود الگوریتم وجود دارد که cross validation درونی را هدایت می کند.

Rule induction (۴-۴-۵

استنتاج قوانین متدی برای تولید مجموعهای از قوانین است که موارد را دسته بندی می کند. اگرچه درختهای تصمیم میتوانند مجموعهای از قوانین را ایجاد کند، متدهای استنتاج قوانین مجموعهای از قوانین مستقل را ایجاد میکند. که لزوما یک درخت را ایجاد نمیکنند. از آنجا که استنتاجگر قوانین اجباری به تقسیم در هر سطح ندارد، و میتواند به آینده بنگرد، قادر است الگوهای متفاوت و گاها بهتری برای ردهبندی بیابد. برخلاف درختان، قوانین ایجاد شده ممکن است برخلاف درختان، قوانین ممکن است در پیشبینی متعارض باشند، که در هر » همه موارد ممکن را نپوشاند. همچنین مورد باید قانونی را برای دنبال کردن انتخاب کرد. یک روش برای حل این تعارضات انتصاب یک میزان اطمینان به هر قانون است و استفاده از قانونی است که میزان اطمینان بالاتری دارد.

K-nearest neibour and memory-based reansoning(MBR) (۵-۴-۵

هنگام تلاش برای حل مسائل جدید، افراد معمولا به راهحل های مسائل مشابه که قبلا حل شده اند مراجعه میکنند. (K-nearest neighbor(k-NN یک تکنیک دسته بندی است که از نسخهای از این متد استفاده میکند. دراین روش تصمیم گیری اینکه یک مورد جدید در کدام دسته قرار گیرد با بررسی تعدادی (k) از شبیه ترین موارد یا همسایه ها انجام میشود. تعداد موارد برای هر کلاس شمرده می شوند، و مورد جدید به دستهای که تعداد بیشتری از همسایه ها به آن تعلق دارند نسبت داده می شود.



محدوده همسایگی (بیستر همسایه ها در دسته X قرار گرفته اند)

شکل ۵-۸

اولین مورد برای بکاربردن k-NN یافتن معیاری برای فاصله بین صفات در دادهها و محاسبه آن است. در حالیکه این k-NN اولین مورد برای بکاربردن عمل برای دادههای عددی آسان است، متغیرهای دستهای نیاز به برخورد خاصی دارند. هنگامیکه فاصله بین موادمختلف را توانستیم اندازه گیریم، میتوانیم از مجموعه مواردی که قبلا دست هبندی شده اند را بعنوان پایه دسته بندی موارد جدید استفاده کنیم، فاصله همسایگی را تعیین کنیم، و تعیین کنیم که خود همسایه ها را چگونه بشماریم.

K-NN بار محاسباتی زیادی را روی کامپیوتر قرار می دهد زیرا زمان محاسبه بصورت فاکتوریلی از تمام نقاط افزایش مییابد. درحالیکه بکابردن درخت تصمیم یا شبکه عصبی برای یک مورد جدید فرایند سریعی است ، K-NN نیاز به محاسبه جدیدی برای هر مورد جدید دارد. برای افزایش سرعت K-NN معمولا تمام دادهها در حافظه نگهداری می شوند.

فهم مدلهای K-NN هنگامی که تعداد متغیرهای پیشبینی کننده کم است بسیار ساده است. آنها همچنین برای ساخت مدلهای شامل انواع داده غیر استاندارد هستند، مانند متن بسیار مفیدند. تنها نیاز برای انواع داده جدید وجود معیار مناسب است.

۵-۴-۶) رگرسیون منطقی

رگرسیون منطقی یک حالت عمومی تر از رگرسیون خطی می باشد.قبلا این روش برای پیش بینی مقادیر باینری یامتغیرهای دارای چند مقدار گسسته (کلاس) استفاده می شد. از آنجایی که مقادیر مورد نظر برای پیش بینی مقادیر گسسته می باشند نمی توان آنرا به روش رگرسیون خطی مدلسازی کرد برای این منظور این متغیرهای گسسته را به

روشی تبدیل به متغیر عددی و پیوسته می کنیم وبرای این منظور مقدار لگاریتم احتمال متغیر مربوطه را در نظر می گیریم و برای این منظور احتمال پیشامد را بدین صورت در نظر می گیریم :

احتمال اتفاق نیفتادن پیشامد/ احتمال اتفاق افتادن پیشامد

و تفسیر این نسبت مانند تفسیری است که در بسیاری از مکالمات روزمره در مورد مسابقات یا شرط بندی ها یه مواردمشابه به کار می رود .مثلا وقتی می گوییم شانس بردن یک تیم در مسابقه ۳ به ۱ است در واقع از همین نسبت استفاده کرده و معنی آن این است که احتمال برد آن تیم ۷۵ ٪ است.وقتی که ما موفق شدیم لگاریتم احتمال مورد نظر را بدست آوریم با اعمال لگاریتم معکوس می توان نسبت مورد نظر و از روی آن کلاس مورد نظر را مشخص نمود.

۵-۴-۷) تحلیل تفکیکی

این روش از قدیمی ترین روش های ریاضی وار گروه بندی داده ها می باشد که برای اولین بار در سال ۱۹۳۶ توسط فیشر استفاده گردید. روش کار بدین صورت است که داده ها را مانند داده های چند بعدی بررسی کرده و بین دادهها مرزهایی ایجاد می کنند (برای داده ها دو بعدی خط جدا کننده، برای داده های سه بعدی سطح جدا کننده و ..) که این مرزها مشخص کننده کلاس های مختلف می باشند و بعد برای مشخص کردن کلاس مربوط به داده های جدید فقط باید محل قرارگیری آن را مشخص کنیم.

این روش از ساده ترین و قابل رشدترین روش های کلاس بندی می باشد که در گذشته بسیار استفاده می شد.

این روش به سه دلیل محبوبیت خود را از دست داد :اول اینکه این روش فرض می کند همه متغیرهای پیش بینی به صورت نرمال توزیع شده اند که در بسیاری از موارد صحت ندارد . دوم اینکه داده هایی که به صورت عددی نمی باشندمثل رنگها در این روش قابل استفاده نمی باشند. سوم اینکه در این روش فرض می شود که مرزهای جدا کننده داده هابه صورت اشکال هندسی خطی مثل خط یا سطح می باشند حال اینکه این فرض همیشه صحت ندارد.

نسخه های اخیر تحلیل تفکیکی بعضی از این مشکلات را رفع کرده اند به این طریق اجازه می دهند مرزهای جدا کننده بیشتر از درجه ۲ نیز باشند که باعث بهبود کارایی و حساسیت در بسیاری از موارد می گردد.

۵-۴-۸) مدل افزودنی کلی GAM

این روش ها در واقع بسطی بر روش های رگرسیون خطی و رگرسیون منطقی می باشند. به این دلیل به این روش افزودنی می گویند که فرض می کنیم می توانیم مدل را به صورت مجموع چند تابع غیر خطی (هر تابع برای یک متغیر پیش بینی کننده) بنویسیم. GAM می تواند هم به منظور رگرسیون و هم به منظور کلاس بندی داده ها استفاده گردد. این ویژگی غیر خطی بودن توابع باعث می شود که این روش نسبت به روشهای رگرسیون خطی بهتر باشد .

۵-۴-۹) Boosting

در این روش ها مبنی کار این است که الگوریتم پیش بینی را چندین بار و هر بار با داده های آموزشی متفاوت (که باتوجه به اجرای قبلی انتخاب می شوند) اجرا کنیم و در نهایت آن جوابی که بیشتر تکرار شده را انتخاب کنیم. این روش اگر چه وقت گیر است ولی جواب های آن مطمئن تر خواهند بود. این روش اولین بار در سال ۱۹۹۶ استفاده شد و در این روزها با توجه به افزایش قدرت محاسباتی کامپیوترها بر مقبولیت آن افزوده گشته است.

۵-۵) سلسله مراتب انتخابها

هدف داده کاوی تولید دانش جدیدی است که کاربر بتواند از آن استفاده کند. این هدف با ساخت مدلی از دنیای واقع براساس داده های جمع آوری شده از منابع متفاوت بدست میآید. نتیجه ساخت این مدل توصیفی از الگوها و روابط داده هاست که میتوان آنرا برای پیش بینی استفاده کرد. سلسه انتخابهایی که قبل از آغاز باید انجام شود به این شرح است:

- هدف تجاری
- نوع پیش بینی
- نوع مدل
- الگوریتم
- محصول

در بالاترین سطح هدف تجاری قرار دارد: هدف نهایی از کاوش دادهها چیست؟ برای مثال، جستجوی الگوها در داده هاممکن است برای حفظ مشتریهای خوب باشد، که ممکن است مدلی برای سودبخشی مشتریها و مدل دومی برای شناسایی مشتری هایی که ممکن از دست دهیم می سازیم. اطلاع از اهداف و نیازهای سازمان ما را در فرموله کردن هدف سازمان یاری می رساند.

مرحله بعدی تصمیم گیری درباره نوع پیشبینی مناسب است:

۱- classification :پیش بینی اینکه یک مورد درکدام گروه یا رده قرار می گیرد. یا

۲-regression :پیش بینی اینکه یک متغیر عددی چه مقداری خواهد داشت .

مرحله بعدی انتخاب نوع مدل است: یک شبکه عصبی برای انجام regression ر و یک درخت تصمیم برای classification . همچنین روشهای مرسوم آماری برای مانند logistic regression، analysis discriminant و یا مدلهای خطی عمومی وجود دارد.

الگوریتمهای بسیاری برای ساخت مدلها وجود دارد. میتوان یک شبکه عصبی را با backpropagation ر یا توابع radial bias ساخت. برای درخت تصمیم، میتوان از میان CART، C۵,۰، Quest و یا CHAID انتخاب کرد.

هنگام انتخاب یک محصول داده کاوی، باید آگاه بود که معمولا پیاده سازیهای متفاوتی از یک الگوریتم دارند. این تفاوتهای پیاده سازی میتواند بر ویژگی های عملیاتی مانند استفاده از حافظه و ذخیره داده و همچنین ویژگی های کارایی مانند سرعت و دقت اثر گذارند.

در مدلهای پیش گویانه، مقادیر یا رده هایی که ما پیش بینی میکنیم متغیرهای پاسخ، وابسته، یا هدف نامیده می شوند. مقادیری که برای

پیشبینی استفاده میشوند متغیرهای مستقل یا پیش بینی کننده نامیده می شوند.مدلهای پیش گویانه با استفاده از داده هـایی کـه مقـادیر متغیرهای پاسخ برای آنها از قبل دانسته شده است ساخته یا آمـوزش داده میشـوند. ایـن نحـوه آمـوزش supervised learning نامیده میشود، زیرا که مقادیر محاسبه شده یا تخمین زده شده با نتایج معلـومی مقایسـه میشـوند.(در مقابـل، تکنیکهـای توصـیفی ماننـد clustering، unsupervised learning نامیده میشوند زیرا که هیچ نتیجه از پیش معلومی برای راهنمایی الگوریتم وجود ندارد.)

۵-۶) منابع :

- مقدمه‌ای بر داده کاوی ، احسان زنجانی
- Two Crows Corporation, **Introduction to Data Mining and Knowledge Discover** , ۱۹۹۹
- David Hand, Heikki Mannila , **Padhraic Smyth**. **Principles of Data Mining**. The MIT Press , ۲۰۰۱
- J.Han, and M.Kamber, "**Data Mining**: **Concepts and Techniques**",
- San Diego Academic Press, ۲۰۰۱
- Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth.From **Data Mining to Knowledge Discovery in Databases**. ۱۹۹۶

۶) کاوش پایگاه داده های وب

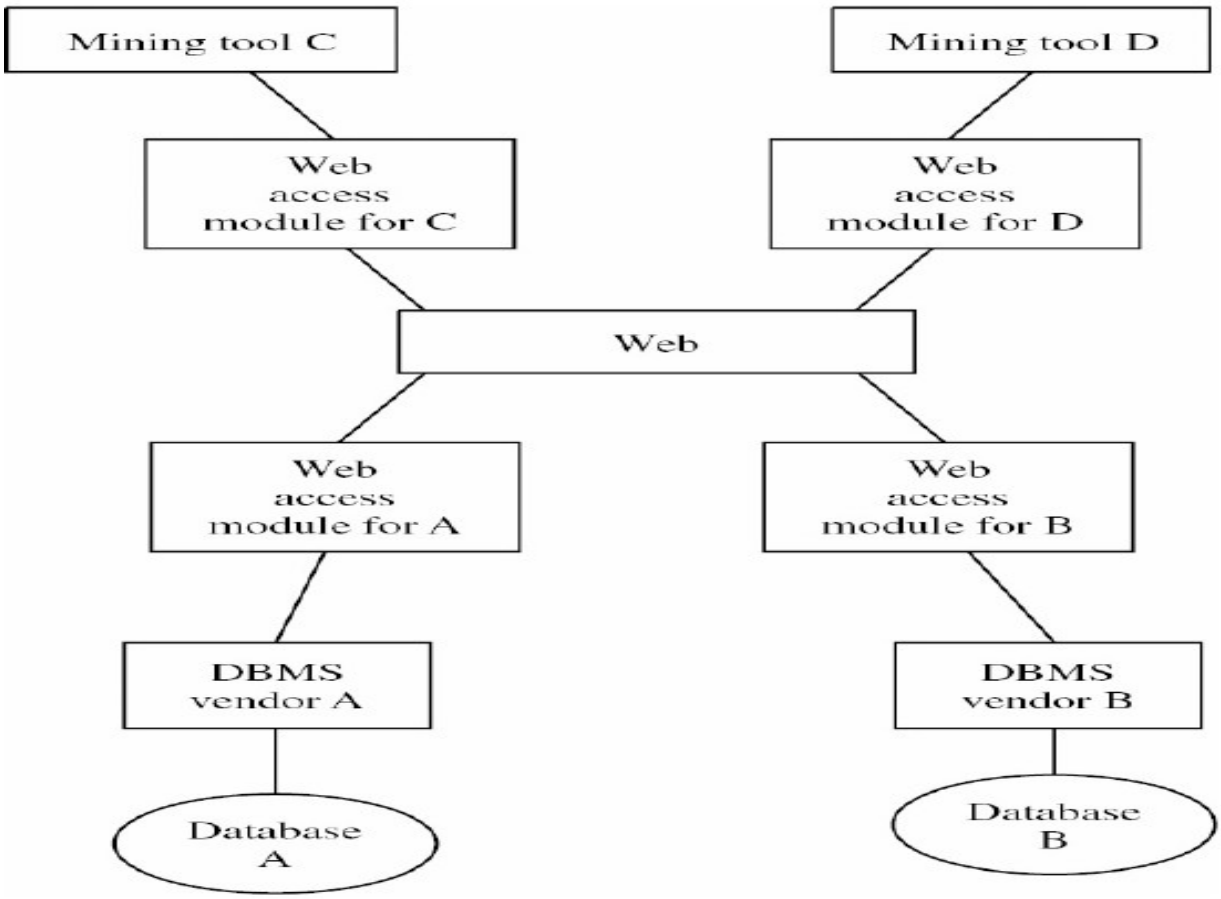
۶-۱) مقدمه:

در فصل قبل یک پیش زمینه از data mining web به دست آمد. در این بخش درباره جستجو در پایگاه داده های موجود در وب صحبت می کنیم.

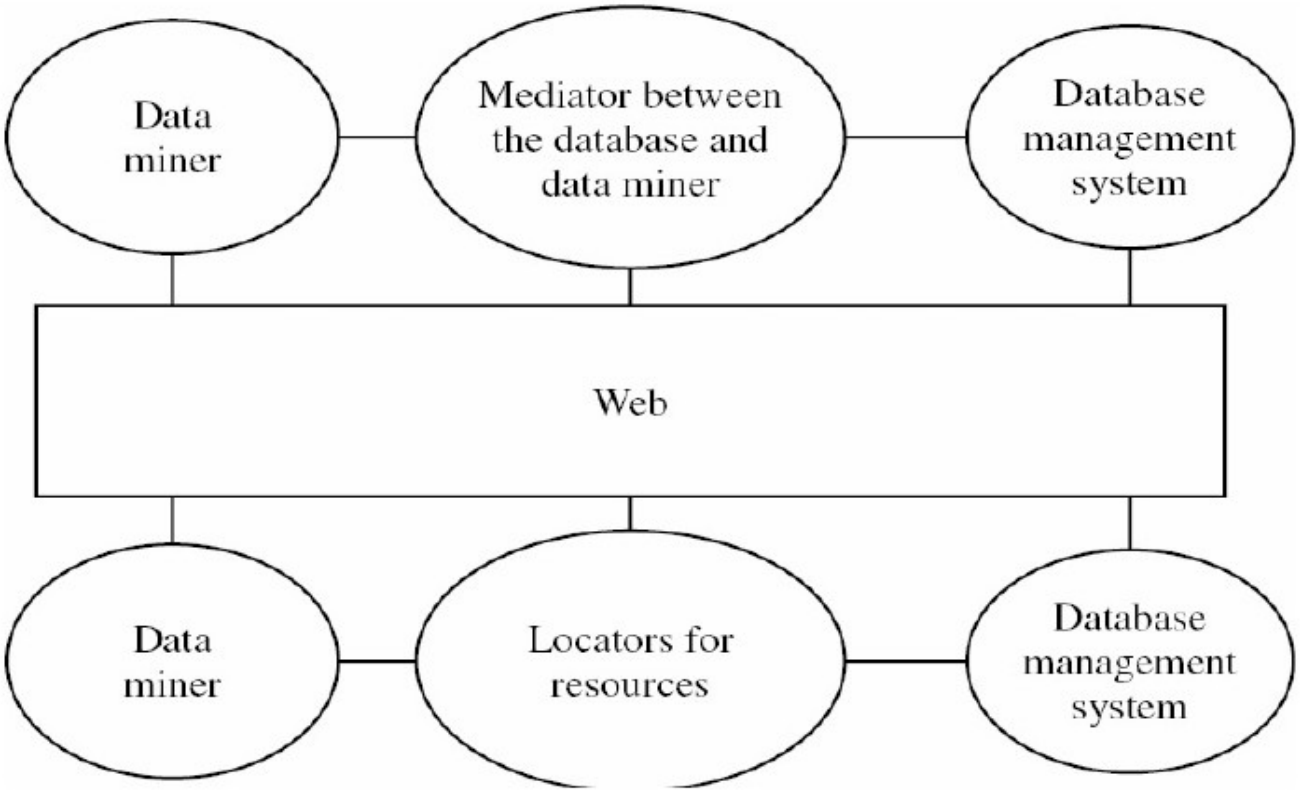
همانطور که گفته شد، حجم بالایی از داده را روی وب داریم. برخی در فایلها و برخی در سایر منابع داده ای وجـود دارنـد. پایگاههـای داده ممکن است نیمه ساختیافته یا رابطه ای شی گرا یا چند رسانه ای باشد. با جستجوی این پایگاه های داده اطلاعات مفیدی می توان از آنهـا استخراج کرد.

۶-۲) مفهوم کلی جستجوی پایگاه داده های وب:

یک شمای ساده از جستجوی پایگاه داده ای در وب در شکل ۶-۱ نشان داده شده است. نظریه این است که پایگاه داده روی وب قرار دارد وبرای استخراج الگوها و روشها کاوش در پایگاه داده انجام می گیرد



شکل 1 : جستجو در پایگاه دادهها در وب



شکل 2: مکانیابي و وساطت براي کاوش در پایگاه دادة روي وب.

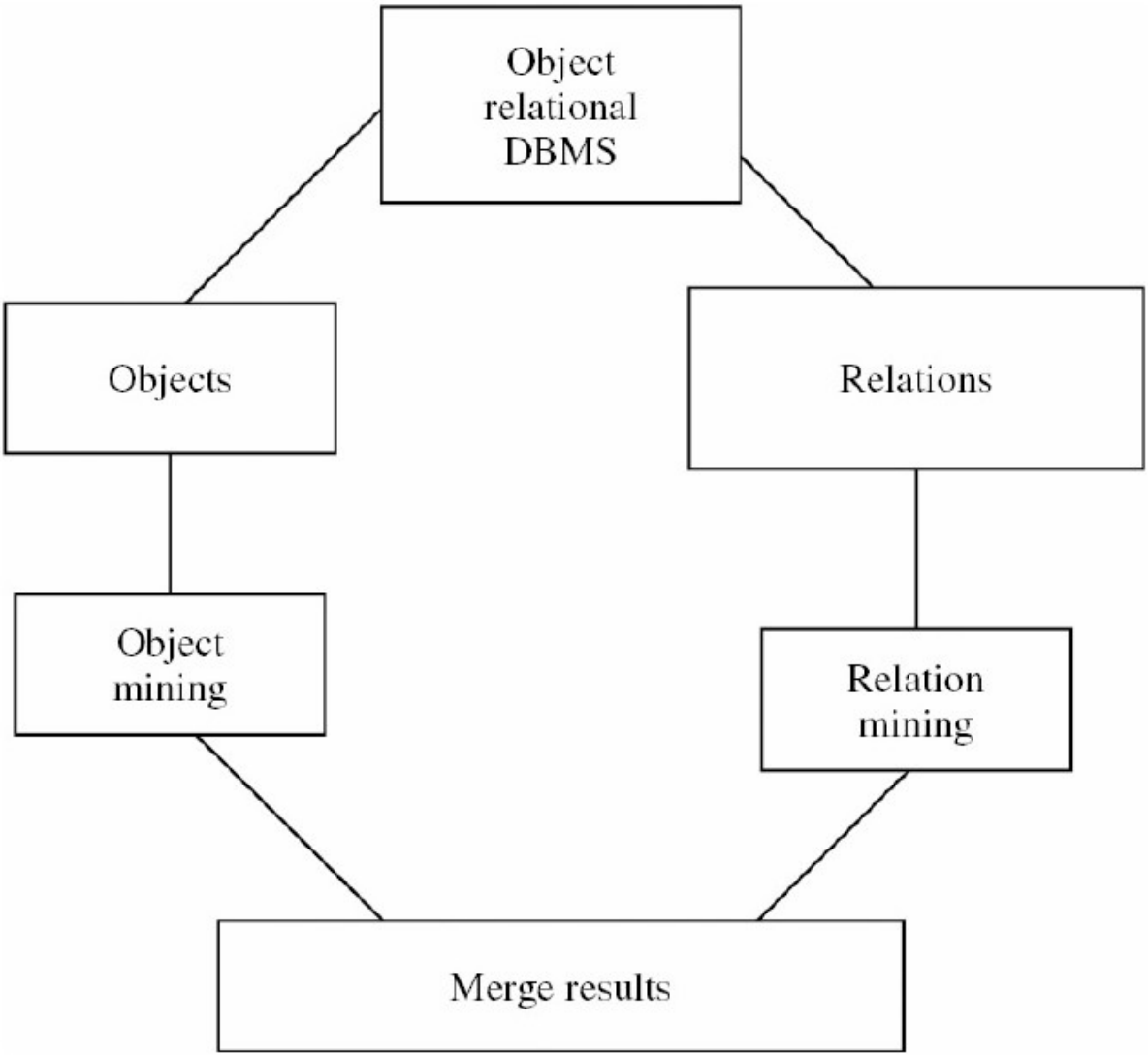
وقتی می توانیم بسیاری از تکنیکهای داده کاوی را برای جستجو در پایگاه داده در وب استفاده کنیم. مشکل این خواهد بود که چگونه مکانیابی پایگاه داده را روی وب انجام دهیم. از این گذشته، ممکن است پایگاه داده در فرمتی که ما برای کاوش داده احتیاج داریم نباشد. پس شاید به واسطی بین پایگاه داده روی وب کاوشگر داده احتیاج داشته باشیم. این مطلب در شکل ۶-۲ نمایش داده شده است.

در بخش بعدی درباره توابع پایگاه داده وب و داده کاوی بحث خواهیم کرد.

۶-۳) توابع مدیریت پایگاه داده وب و داده کاوی:

در این قسمت به بررسی و آزمایش برخورد داده کاوی می پردازیم. همانطورکه قبلاً گفته شد، داده می تواند در یک پایگاه داده رابطه ای، شیئی یا نیمه ساختیافته باشد. کاوش داده درون پایگاه داده نیمه ساختیافته در بخش بعدی مطرح خواهد شد.در این بخش به بررسی داده درون پایگاه داده ی شیئی،رابطه ای و شیئی- رابطه ای می پردازیم.

ممکن است بصورت مستقیم از ابزارهای داده کاوی روی پایگاه داده استفاده کنیم و یا در ابتدا اطلاعات را از داده استخراج کنیم و سپس ا اطلاعات استخراج شده را جستجو کنیم. هر چند به علت اینکه داده درون وب ممکن است در چند منبع متفاوت وجود داشته باشد ممکن است کامل نباشد، یا متناقض باشد. بنابراین ما بدنبال پیدا کردن علت نقص یا نادرستی خواهیم بود. به عنوان مثال یک پایگاه داده شیئی-رابطه ی در شکل ۶-۳ نمایش داده شده است.



شکل 3: داده کاوي شيئي – رابطه‌اي

توابع پایگاه داده، شامل پردازش Query و مدیریت واکنشهاست. پردازش Query شامل تکنیکهای ویژه بهینه سازی و زبانهایی برای وب می باشد. باید داده کاوی را در زبانها بسازیم همان طوری که باید تکنیکهای داده کاوی را در الگوریتمهای بهینه سازی Query بکار

بگیریم. داده کاوی می تواند از دو راه در مدیریت واکنشها شرکت داشته باشد: کاوش در واکنش ثبت وقایع در وب و جستجو وقتی که واکنش به صورت بلادرنگ در حال انجام است. یک کار کوچک روی داده کاوی وجود دارد به علت اینکه داده کاوی معمولاً برای تحلیل انجام می شود. ساختن مدلها به صورت بلادرنگ مشکل است. اما در کاربردهایی استفاه از آن قطعی به نظر می رسد مثلاً تشخیص تجاوز که باید پایگاههای داده را به صورت بلادرنگ جستجو کنیم. ساختن مدلها در حالت بلادرنگ یک چالش مهم است. مدیریت حافظه نیز به خوبی از تابع مدیریت پایگاه داده در وب است. در اینجا، داده کاوی برای تعیین و تشخیص ساختمان و سازمان پایگاه داده به ما کمک می کند. البته احتیاج به تحقیقات بیشتری در این زمینه وجود دارد. شکل ۶-۴ برخی کاربردهای داده کاوی در توابع مدیریت پایگاه داده های موجود در وب را نشان می دهد.

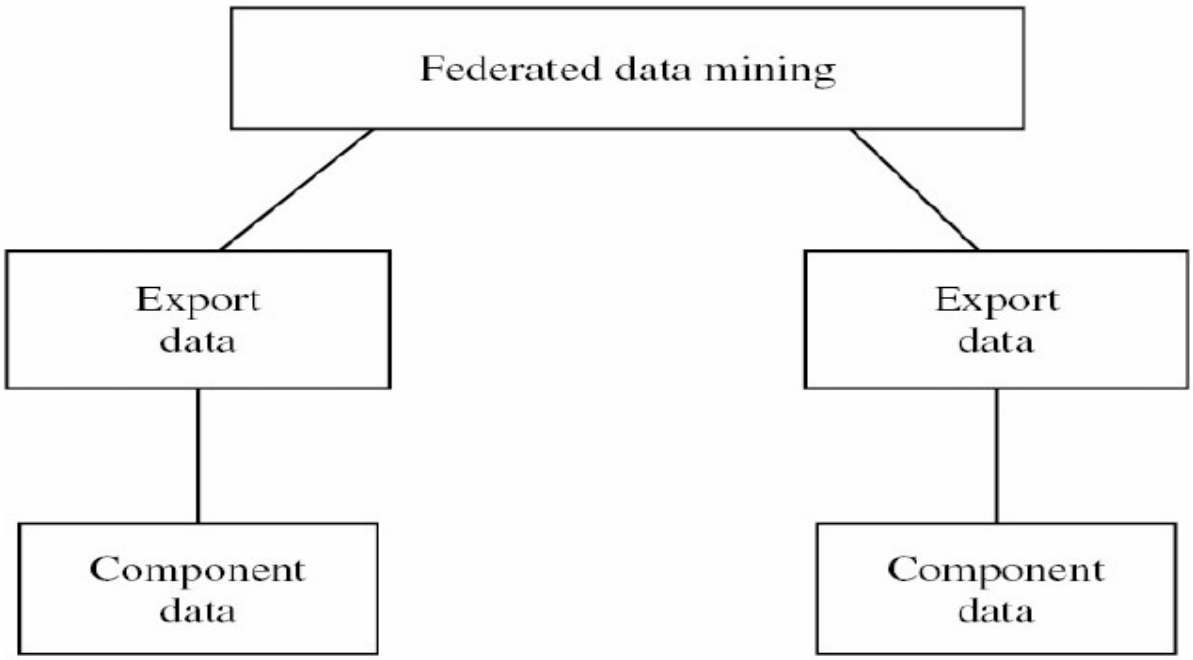
شکل 4: توابع پایگاه داده وب و کاوش

۶-۴) اشتراک داده در مقابل داده کاوی در وب:

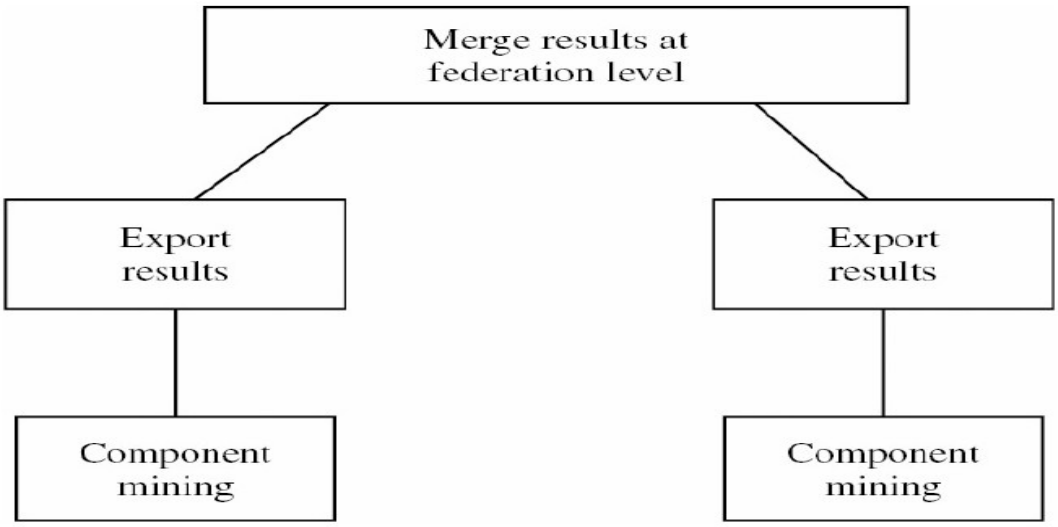
همانطورکه گفته شد یک چالش در داده کاوی گرفتن داده آماده برای کاوش است. معنی اش اینست که سازمان هایی که می خواهند داده را به اشتراک بگذارند، در بسیاری موارد داده خصوصی یا حساس است. بنابراین، سازمانها و آژانسها ممکن است تمایلی برای اشتراک گذاری داده ها نداشته باشند. داده کاوی روی یک داده بد نتایجی خوبی نخواهد داشت حتی اگر از ابزارهای بسیار عالی داده کاوی استفاده شود.

بنابراین سئوال این خواهد بود که چگونه می توانیم داده ای خوب را به اشتراک گذاریم به طوریکه بتوانیم آن را جستجو کنیم، در حقیقت ما باید یک داده کاوی متحد را انجام دهیم. در یک پیاده‌سازی می توانیم داده قطعی و مدل را به یک حالت همگانی و متحد بفرستیم و سپس کاوش را انجام دهیم. که در شکل ۶-۵ نمایش داده شد.

در یک پیاده سازی دیگر، کاوش را در سطح اجزا انجام می دهیم وسپس قطعات را در کنار هم قرار می دهیم. این پیاده سازی در شکل ۶-۶ نشان داده شده است. ما در باره کاوش پایگاه داده های توزیع شده، ناهمگن و وراثتی و متحد در بخش بعدی صحبت خواهیم کرد.



شکل 5: داده کاوي متحد شده : پیاده سازي I



شکل 6: داده کاوي متحد شده: پیاده سازي II

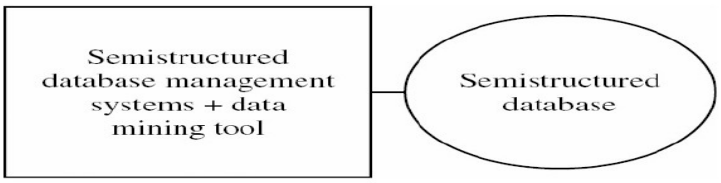
۶-۵) کاوش پایگاههای داده نیمه ساختیافته:

مامی توانیم از پایگاههای نیمه ساختیافته به جای پایگاه داده XML استفاده کنیم یا برعکس، البته پایگاه داده نیمه ساختیافته گسترده تراند و شامل پایگاه داده های RDF هم می باشد.

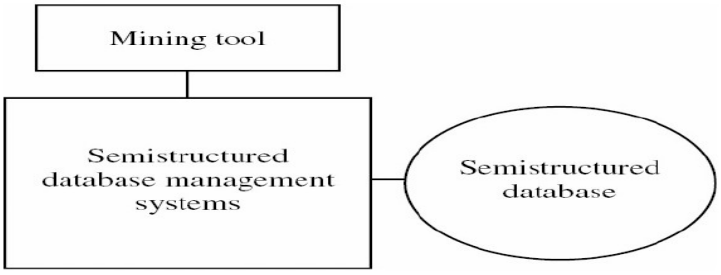
در مثال در رابطه با کاوش پایگاه داده شیئی- رابطه ای، نظریه ای در باره چگونگی کاوش اشیاء و روابط و الحاق نتایج به دست آمده می دهیم. بامدیریت پایگاه داده های نمیه ساختیافته، دو پیاده سازی برای مدیریت مستندات نیمه ساختیافته وجود دارد. یکی از آنها، توسعه سیستم مدیریت پایگاه داده برای مدیریت مسنتدات نیمه ساختیافته می باشد. که بعنوان یک پیاده سازی شناخته شده و قاطع است. پیاده سازی دیگر یک واسط کاربر می سازد، مثلاً در یک پایگاه داده رابطه ای مستندات نیمه ساختیافته را مدیریت کند. که بعنوان یک پیاده سازی ضعیف شناخته شده است.

داده کاوی پیاده سازیهای متفاوت دارد. در یک پیاده سازی می توانیم سیستم مدیریت پایگاه داده نیمه ساختیافته با یک داده کاو و یا کاوش مستندات توسعه بدهیم، (شکل ۶-۷)، یا می توانیم یک واسط کاربر برای سیستم مدیریت پایگاه داده های نیمه ساختیافته بسازیم (شکل ۶-۸). قبلاً حالت ارتباط قوی بین داده کاو و سیستم مدیریت پایگاه داده (DBMS) و بعداً حالت ارتباط ضعیف بین داده کاو و DBMS قرار دارد. در حالت ارتباط ضعیف برای مدیریت پایگاه داده نیمه ساختیافته می توانیم روابط و مستندات نیمه ساختیافته را کاوش کنیم و در نهایت تنایج را با هم جمع کنیم. (شکل ۶-۹) پیاده سازی دیگری نیز وجود دارد که در آن ساختار مستندات نیمه

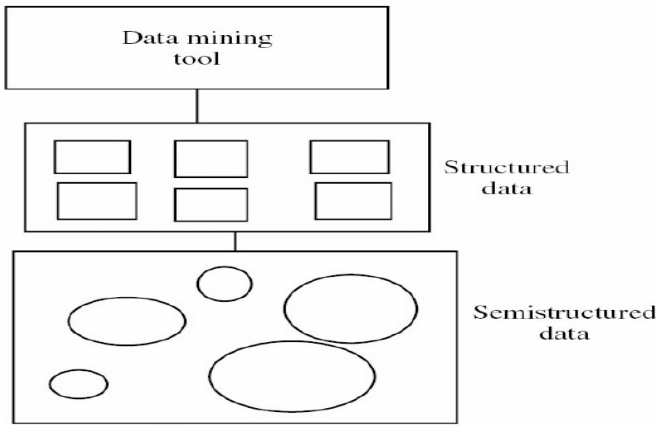
ساختیافته استخراج می شود و سپس مستندات ساختیافته کاوش می شود. (شکل۶-۱۰) باید توجه کرد که جایگشت ها و ترکیبهای مختلفی از این پیاده سازیها وجود دارد. مثلاً حتی در پیاده سازی ضعیف، می توانیم یک داده کاو مشابه یک واسط برای سیستم مدیریت پایگاه داده بسازیم. (شکل۶-۱۱). نکته ای که باید به آن اشاره کرد اینست که دو راه برای مدیریت پایگاههای داده نیمه ساختیافته وجود دارد: پیاده سازی ضعیف و پیاده سازی قوی. همچنین در نوعی از داده کاوی می توانیم حالت قوی یا حالت ضعیف را با داده کاو داشته باشیم.



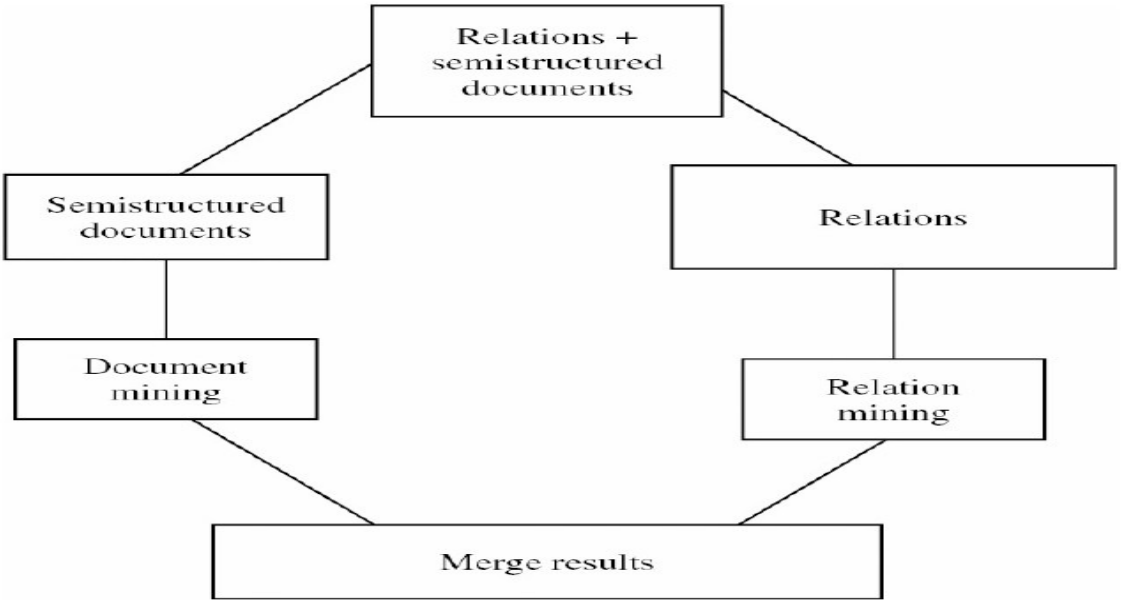
شکل 7: اتصال قوي بين داده كاو و DBMS



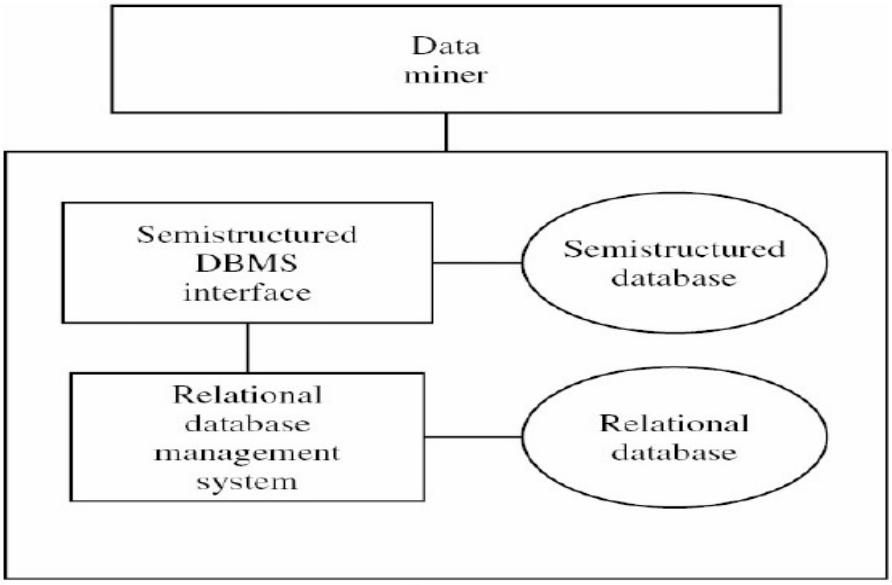
شکل 8: اتصال ضعيف بين داده كاو و DBMS



شکل ۶-۹: استخراج ساختار و سپس کاوش



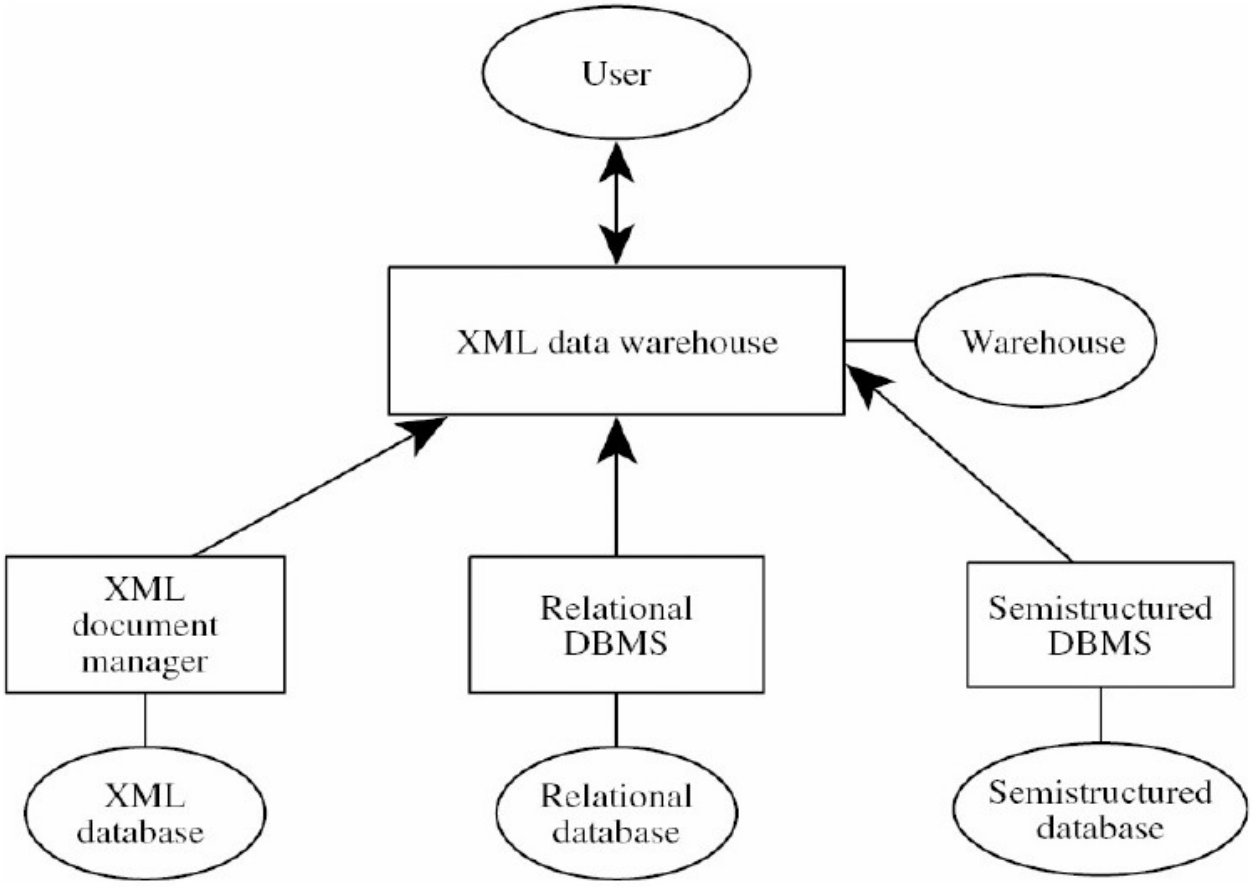
شکل 10: کاوش و سپس ادغام



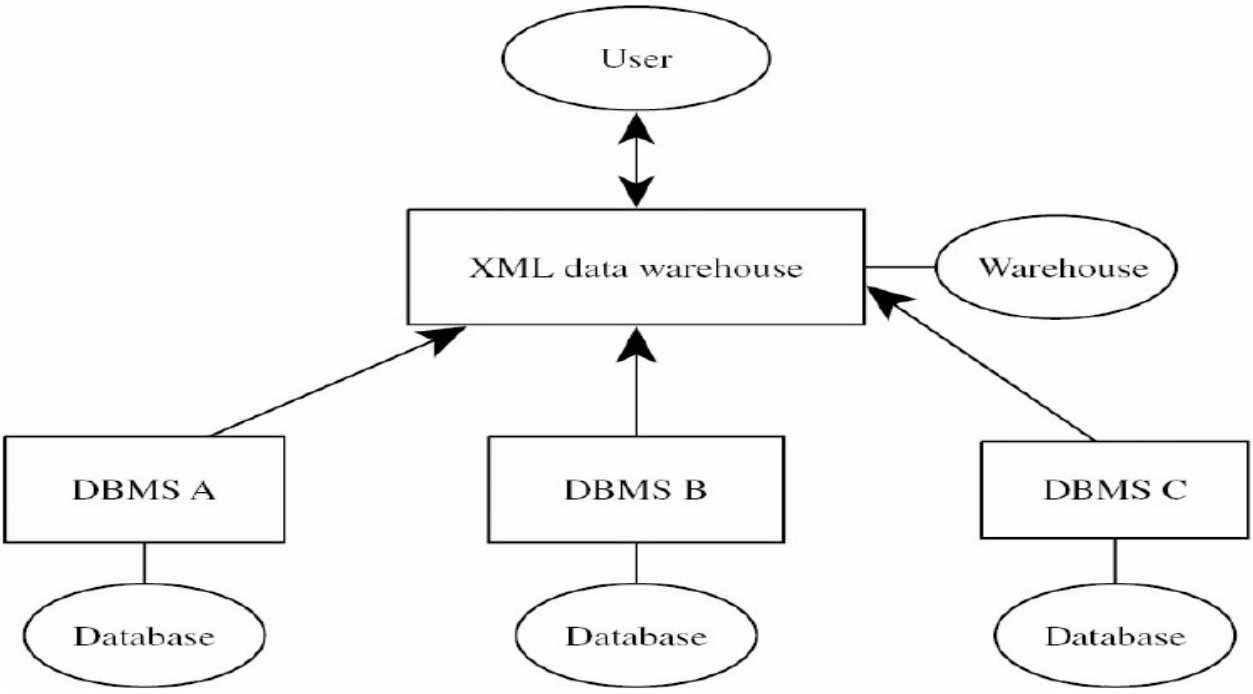
شکل 11: داده کاو مشابه یك واسط در یك ارتباط ضعیف DBMS نیمه ساختیافته

راجع به ذخیره داده ها، دو حالت وجود دارد. اول اینکه مستندات XML، پایگاه داده های نیمه ساختیافته و پایگاه داده های رابطه ای ودیگر منابع داده در یک مخزن جمع شده اند .حال بیشترین کار اینست که پایگاه داده های رابطه ای را بر اساس یک مدل رابطه ای در یک مخزن جمع کنیم..

وقتی پایگاه داده ها مستندات XML هستند مثل پایگاه داده های نیمه ساختیافته این سوال وجود دارد که چگونه آنها را در یک مخزن ذخیره کنیم؟ چه مدلی برای یک مخزن وجود دارد؟ به علت اینکه اکنون نگاشتی بین SQL و XML وجود دارد، چگونه می توانیم مدلی بر پایه SQL برای مخزن خود داشته باشیم؟ شکل دیگری از نمایش مخزن مجموعه ای از مستندات XML می باشد. در این حالت، مثلاً احتیاج به نگاشتی بین منبع داده بر پایه مدلهای شیئی و رابطه ای و مدل های داده XML داریم. همچنین احتیاج به توسعه تکنیکهای دسترسی، پرس و جو وشاخص گذاری منبع داریم. این دو نوع از ذخیره داده در شکل ۱۲ و۱۳ نشان داده شده اند.

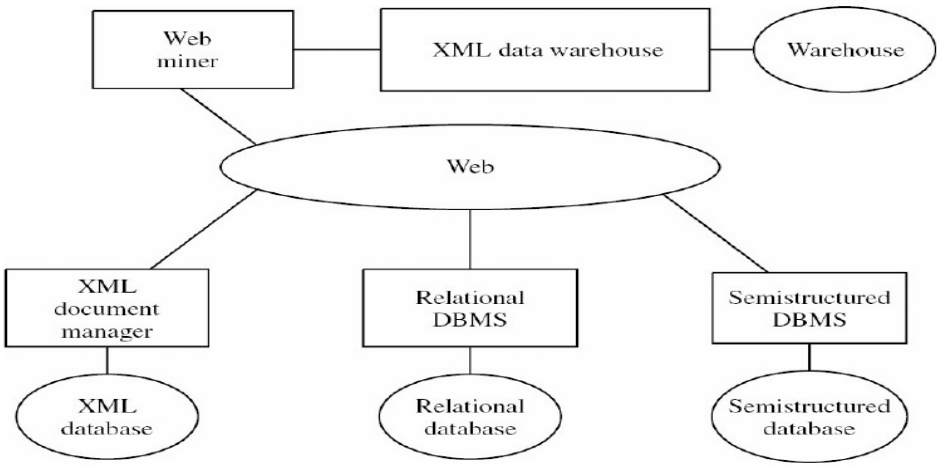


شکل 12: مخزن داده بر پایهٔ XML: پیاده سازی 1



شکل 13: مخزن داده بر پایهٔ XML : پیاده سازی 2

اوش مستدات XML دو تا حالت دارد. اولی کاوش مستندات برای استخراج اطلاعات مفید مثلاً اگر الگوها و گرایشها، است. مثلاً مستندات XML ممکن است برای تجارت هوشمند کاوش شوند. اول اینکه می توان این مستندات را برای مدیریت ارتباط با مشتری کاوش کرد. شکل دیگری کاوش پیوندها در یک مستند XML واستخراج اطلاعات از این پیوندها است. دراینجا هنوز کار زیادی وجود دارد. شکل۶-۱۴ کاوش مخزن بر پایه XML نشان داده شده در شکلهای۶-۱۲ و۶-۱۳ را نشان می دهد.

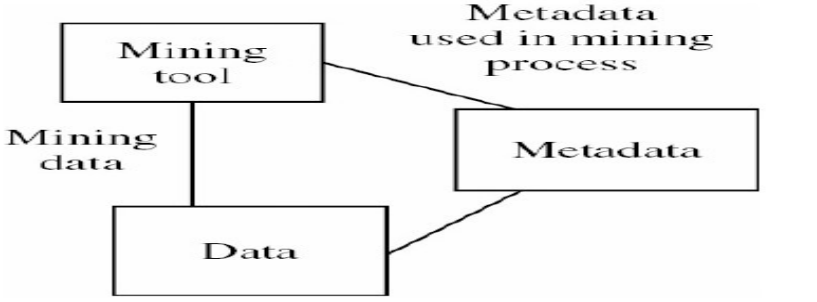


شکل 14: کاوش مخازن وب

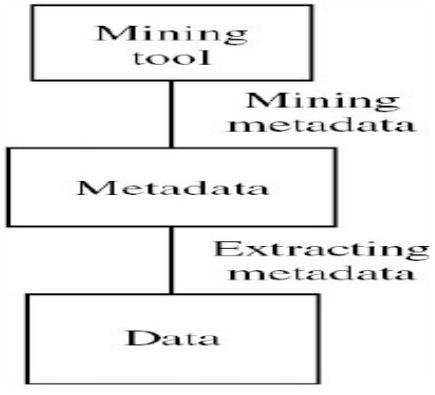
۶-۶) Web mining و Meta data :

Meta data خودش یک تکنولوژی کاربردی برای کارهای مختلفی مثل مدیریت داده، ذخیره داده، جستجوی وب، پردازش اطلاعات چند رسانه ای و داده کاوی است. به علت اینکه متادیتا با پایگاه داده ارتباط نزدیکی دارد، در باره تکنولوژی متادیتا در web data mining در اینجا صحبت می کنیم.

متادیتا در داده کاوی نقش مهمی را بازی می کند و راهنمایی برای پردازش داده کاوی می باشد. ابزار داده کاوی می تواند از پایگاه متادیتا برای شناسایی نوع پرس وجوهای موجود در DBMS استفاده کند. متادیتا ممکن است در طور پردازش کاوش بروز شود. مثلاً اطلاعات گذشته و استاتیک ممکن است جمع شود و متادیتا تغییرات محیط را منعکس کند. نقش متادیتا در راهنمایی وهدایت پردازش داده کاوی در شکل۶- ۱۵ نمایش داده شده است . استخراج متادیتا ازداده و سپس کاوش آن در شکل۶-۱۶ نشان داده شده است.



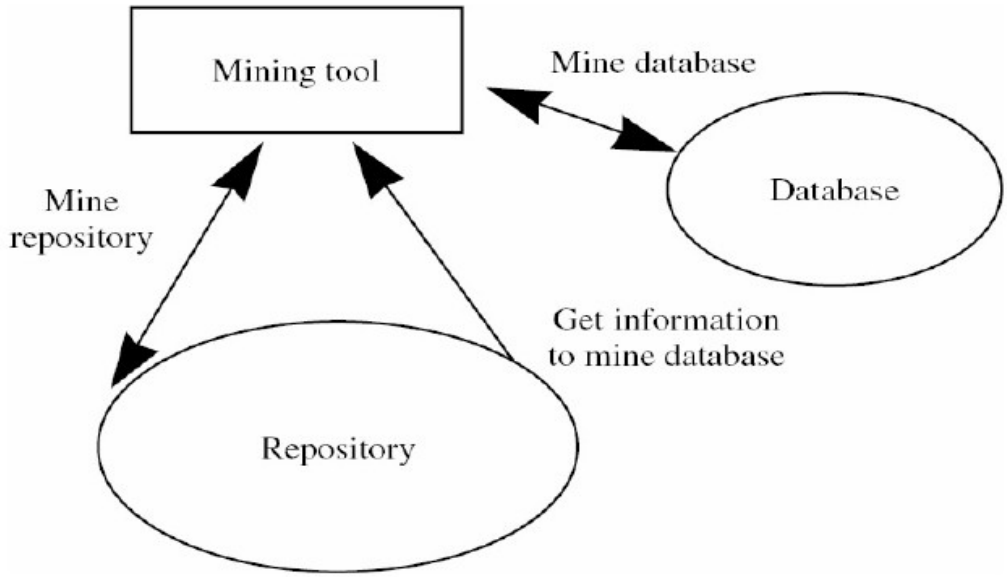
شکل 15: متادیتای استفاده شده در داده کاوی



شکل 16: کاوش متادیتا

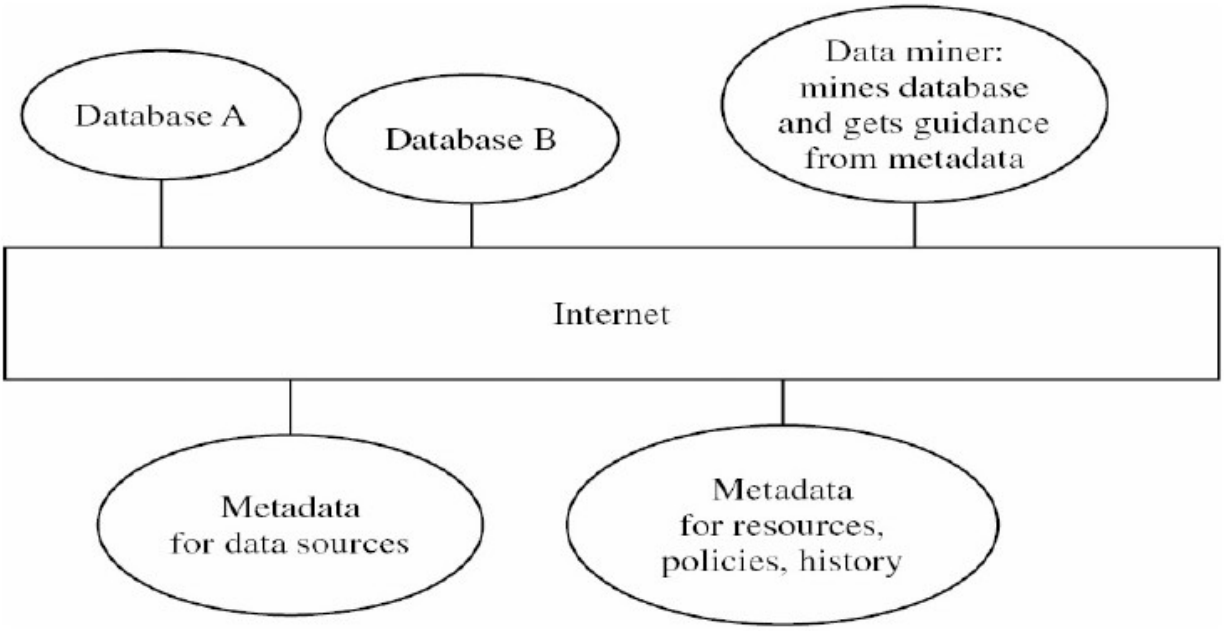
چالشهای زیادی در اینجا وجود دارد. مثلاً چه زمانی برای کاوش متادیتا مناسبتر است؟ چه تکنیکهایی برای کاوش متادیتا است؟ چگونه ساختار متادیتا داده کاوی را آسان می کند؟ محققین در باره این سؤالات کار می کنند.

با متادیتا مفهومی مثل انبار همراه است. یک انبار یک پایگاه داده است که همه متادیتاهای ممکن در آن ذخیره شده اند. نگاشت میان منابع مختلف داده وقتی که منابع داه ناهمگن با همه جمع می شوند، احتیاج به اطلاعاتی برای به کار بردن ناهمگن های مفهومی مثل "قایق X و زیردریایی Y موجودیتی یکسان هستند" دارد. بنابراین ابزار داده کاوی ممکن است از انبار برای اجرای کاوش استفاده کند. از طرف دیگر، خود انبار ممکن است کاوش شود. هر دو این حالت در شکل ۱۷ نشان داده شده است.



شکل 17: انبار و کاوش

متادیتا در web mining هم بحرانی است. بعلت اینکه اطلاعات و داده های زیادی در وب وجود دارد، کاوش این داده ها به صورت مستقیم، خودش یک چالش است. بنابراین، ما احتمالاً به استخراج متادیتا از داده احتیاج داریم. و سپس باید این متادیتا را کاوش کنیم و یا از این متادیتا برای هدایت پردازش کاوش استفاده کنیم که در شکل ۱۹ نشان داده شده است. باید توجه شود که زمانی مثل XML که می خواهیم به طور خلاصه در قسمت بعدی توضیحش دهیم، نقش را در توضیح متادیتا در مستندات وب بازی می کند.

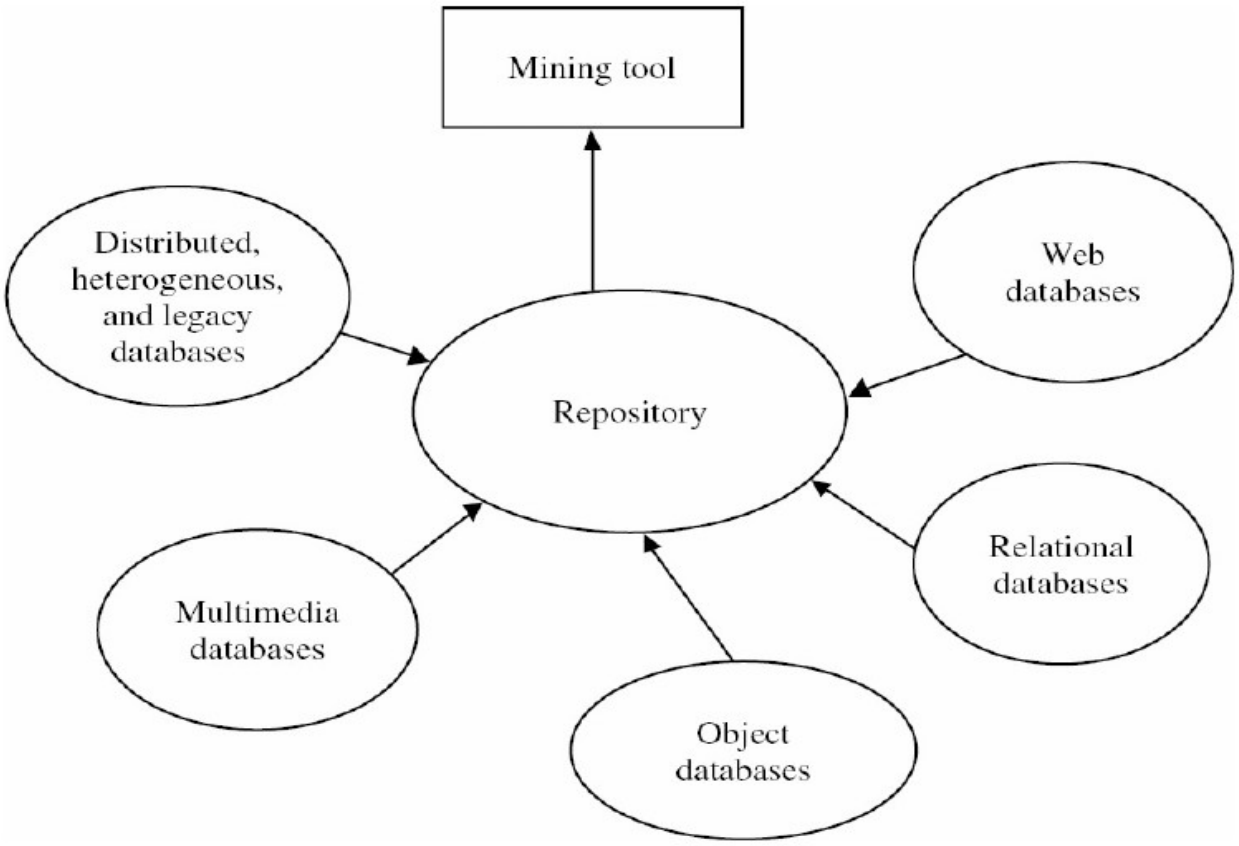


شکل 19: متا دیتا براي web mining

سیاستها و روشهای که مورد بحث هستند برای تعیین محدوده ای که ما می توانیم حوزه وب شخصی خود را محافظت کنیم. این سیاستها و روشها می توانند به قسمتی از متادیتا وابسته باشند. بنابراین این چنین متادیتایی پردازش داده کاوی را هدایت می کند به طوریکه عمل حفاظت در طول کاوش به خطر نمی افتد.

تقریباً در هر نوعی از کاوش، متادیتا نقشی حیاتی را بازی می کند حتی در نوعی از ذخیره سازی داده، که ما در مراحل اولیه کاوش محافظت، را داریم، مجموعه کردن متادیتا در مراحل مختلف مهم است. مثلاً در این نوع ذخیره سازی داده ها، داده از منابع چند گانه جمع می شود. متادیتا پردازش تبدیل را از لایه به لایه در ساختن مخزن هدایت می کند. متادیتا همچنین در اداره کردن مخزن داده کمک می کند. متادیتا در استخراج پاسخها به پرس وجوهای مختلف نیز استفاده می شود.

به علت اینکه متادیتا برای همه انواع پایگاه داده ها مثل رابطه ای، شیئی، چند رسانه ای، توزیع شده ناهمگن، وراثتی و پایگاه داده های وب، کلیدی است. ممکن است با یک انبار متادیتا مواجه شویم که شامل متادیتایی از پایگاه داده های متفاوت باشیم و سپس برای استخراج الگوها متادیتا را کاوش کنیم. این پیاده سازی در شکل۶- ۲۰ نشان داده شده است و می تواند یک پیشنهاد خوب باشد وقتی که کاوش مستقیم پایگاه داده ها مشکل باشد.



شکل 20 : متادیتا یك انبار مرکزي براي کاوش

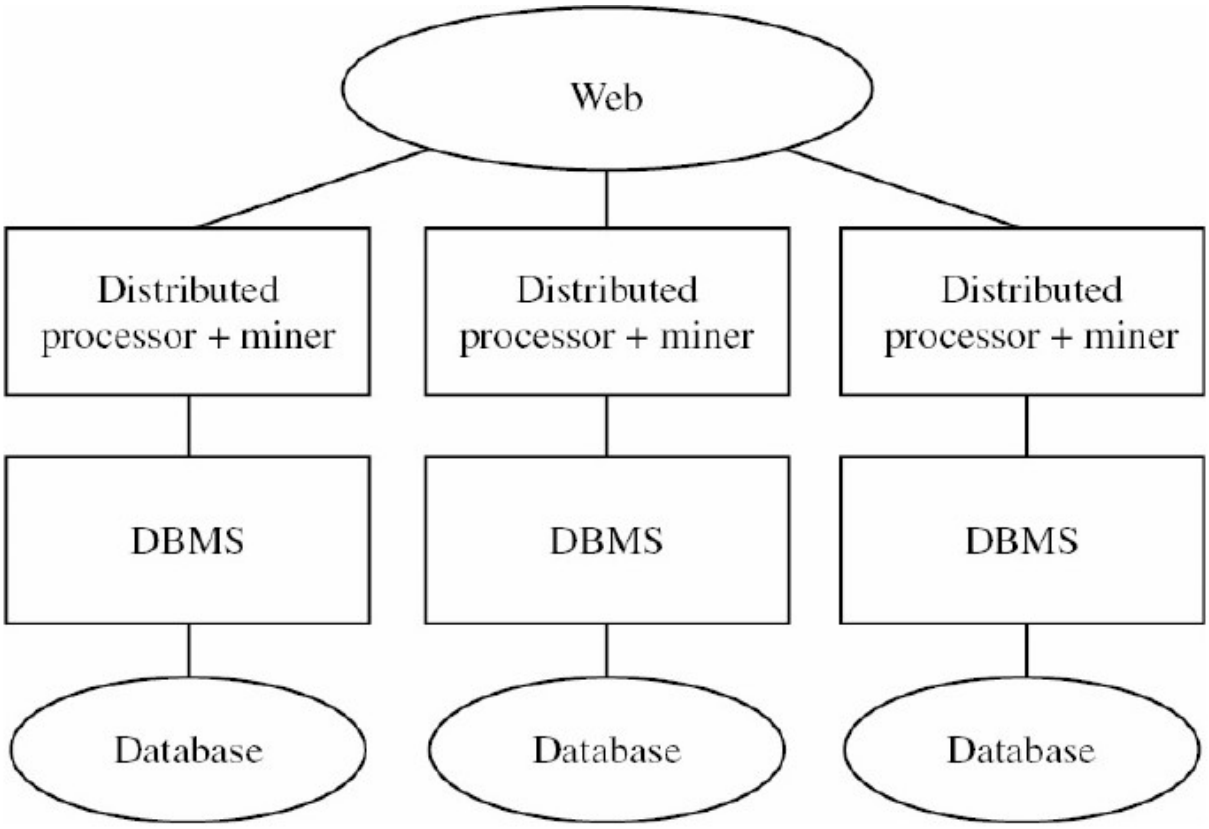
۶-۷) کاوش پایگاه داده های توزیع شده، ناهمگن ، وراثتی و متحد در وب:

بسیاری از کاربردها به مجموعه ای از منابع داده و پایگاههای داده احتیاج دارند. این منابع داده ممکن است نیاز باشد که برای کشف الگوها کاوش شوند. از این گذشته الگوهای جالب توجه ممکن است در پایگاه داده های چند گانه پیدا شود. کاوش منابع داده توزیع شده و ناهمگن، توجه کمی دریافت می کند.

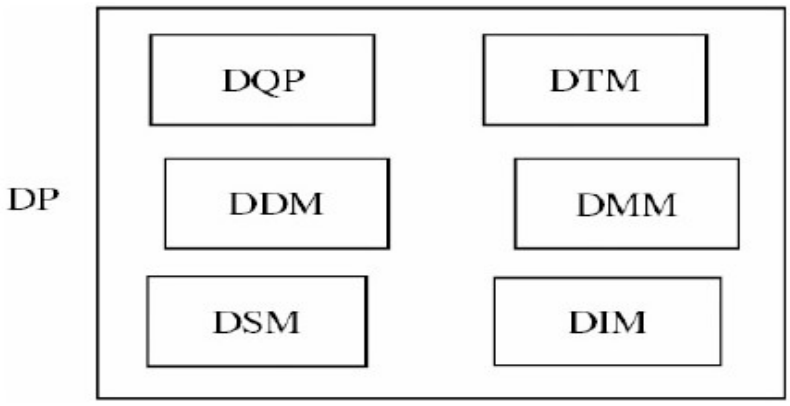
در نوع پایگاه داده توزیع شده، یک روش دارای ابزار داده کاوی است که شبیه قسمتی از پردازشگر توزیع شده است، که هر پردازشگر توزیع شده (DP) یک چنین جزئی برای کاوش دارد، که این مطلب در شکل ۶-۲۱ نشان داده شده است. از این راه، جزء داده کاوی می تواند داده را در پایگاه داده محلی کاوش کند و DP همه نتایج را می تواند باهم ترکیب کند. این یک مطلب چالش انگیز اساسی خواهد بود. که ارتباط بین قطعات مختلف ارتباطی یا اشیاء به منظور کاوش مؤثر باقی بماند. همچنین ابزار داده کاوی می تواند در بهینه سازی پرس وجو

در DQP (پردازشگر پرس و جو توزیع شده) جاسازی شود. د ر حقیقت با این روش DP یک ماژول اضافه دارد داده کاو توزیع شده دارد.

در شکل ۶-۲۲ این مطلب نمایش داده شده است.

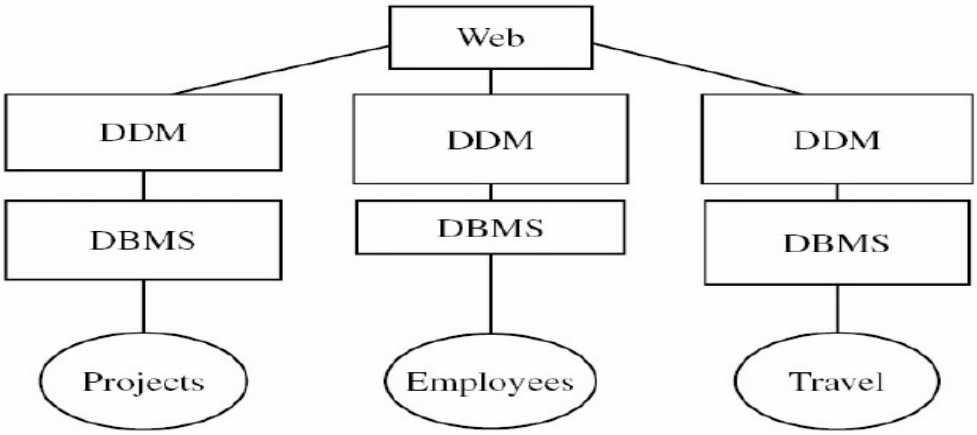


شکل 21: پردازش و کاوش توزیع شده



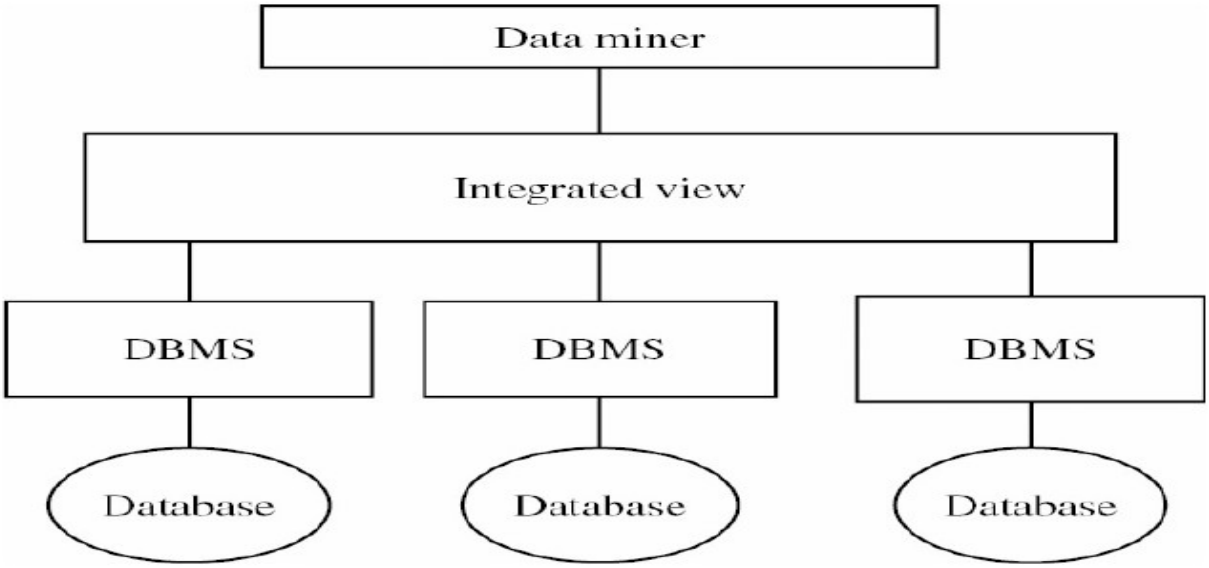
شکل 6-22: ماژولهایی از DP برای داده کاوی

داده کاوی توزیع شده با یک مثال در شکل۶- ۲۳ نشان داده شده است. هر DDM داده را از داخل پایگاه داده خاصی کاوش می کند. این پایگاه داده ها حاوی اطلاعاتی در مورد پروژه ها، کارکنان و سفرها می باشند. DDM ها می توانند اطلاعاتی مثل زیر را کاوش کنند و بگیرند. علی و محمد در پروژه XXX لااقل ۱۰ بار در سال به تهران با هم سفر می کنند. رضا در حداقل ۴ بار در سال به آنها ملحق می شود.



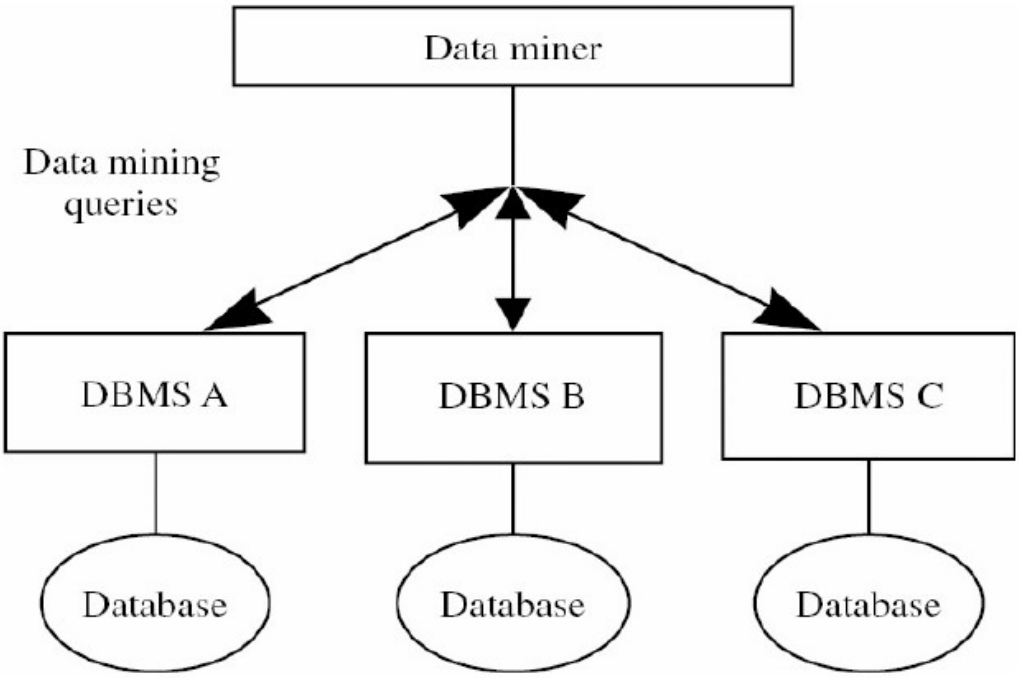
شکل 23: مثال داده کاوي توزيع شده

یک روش پیشنهادی به کاربردن ابزار داده کاوی روی سیستم توزیع شده است. تا آنجا که به ابزار کاوش مربوط می شود، پایگاه داده یک موجودیت یکپارچه است. داده در این پایگاه داده کاوش می شود و الگوی مفید از آن استخراج می شود.

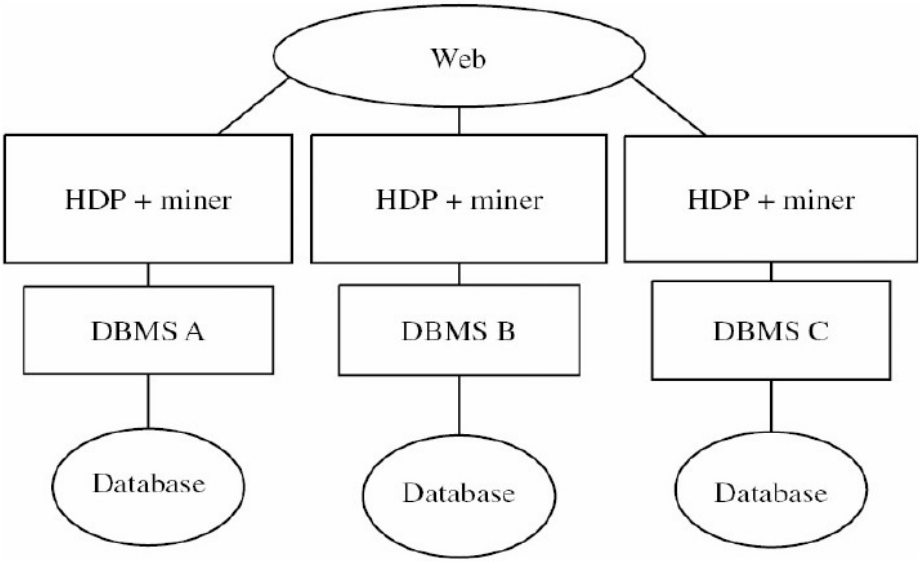


شکل 24: داده کاوي روي پايگاه داده توزيع شده

در نوع پایگاه داده ناهمگن، ما می توانیم پایگاههای داده را جمع کنیم و سپس ابزارهای داده کاوی را برای آنها به کار ببریم، شکل۶- ۲۵، یا ابزارهای داده کاوی را برای موجود در منابع مختلف داده به کاربریم و سپس نتایج آنها را با هم جمع کنیم، شکل ۶-۲۶، نکته اینجاست که اگر ما ابتدا پایگاههای داده را با هم جمع کنیم و سپس متدهای مجتمع برای پایگاه داده های ناهمگن با متدهایی برای تولید یک شمای مجتمع شده دریک پایگاه داده توزیع شده متفاوت است. از این گذشته، ممکن است برای منابع داده متفاوت پرس وجوی مشابهی فرستاده شود و سپس نتایج جمع شوند، که ای مطلب در شکل۶- ۲۵ نشان داده شده است. اگر داده جمع نشود یک داده کاو به پردازشگر توزیع شده ناهمگن متصل می شود. شکل ۶-۲۶ اگر هر منبع داده یک داده کاو داشته باشد، هر داده کاو می تواند مستقل عمل کند. ما پرس وجو مشابهی را برای منابع داده متفاوت نمی فرستیم در نتیجه داده کاو تعیین می کند که روی داده چه عملیاتی انجام شود. چالشی که اینجا وجود دارد، اجتماع نتایج ابزارهای متفاوت کاوش به کار رفته در منابع داده شخصی است، به طوریکه الگوهای منابع داده پیدا شوند.

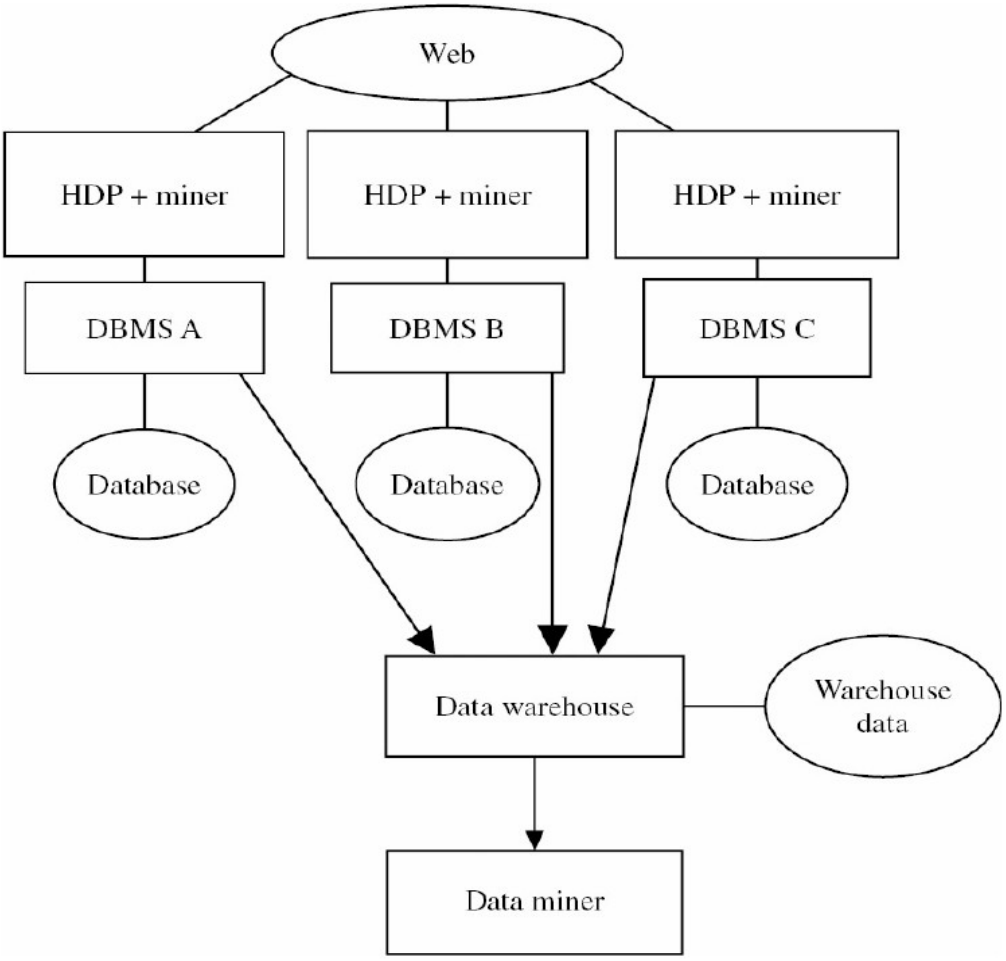


شکل 25: داده کاوي روي منابع دادة نا همگن



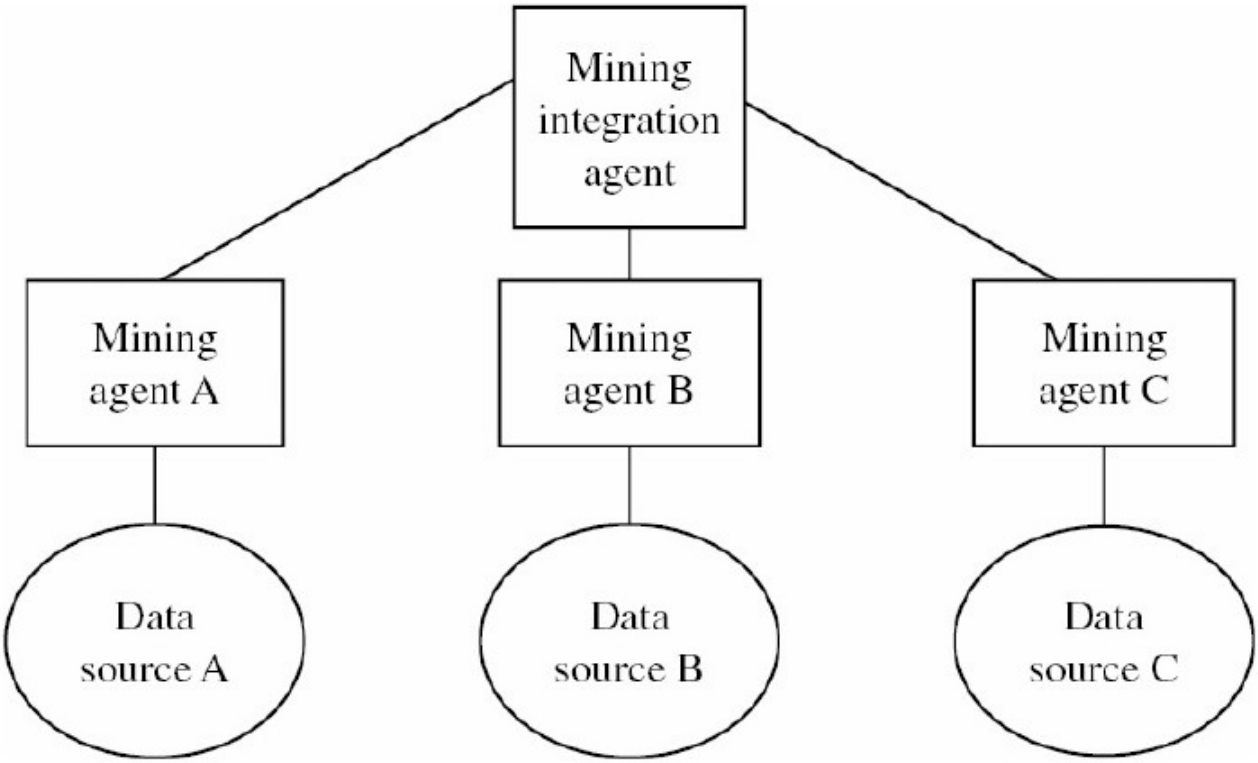
شکل 26: کاوش و سپس اجتماع

اگر ما منابع داده را جمع کنیم و سپس ابزارهای داده کاوی را به کار ببریم، سئوال این خواهد بود که، آیا ما یک مخزن داده را توسعه دهیم و منبع را کاوش کنیم یا با یک سیستم پایگاه داده interoperating کاوش را انجام دهیم. نکته جالب اینجاست که در نوعی از روش ذخیره سازی، همه داده های داخل منابع داده ناهمگن در مخزن آورده نشده است. فقط داده پشتیبان تصمیم در مخزن آورده شـده اسـت. اگر interoperating با ذخیره سازی استفاده شود، داده کاو می تواند HDP و مخزن را با هم الحاق کند، که این در شـکل ۶-۷ نشان داده شده است.

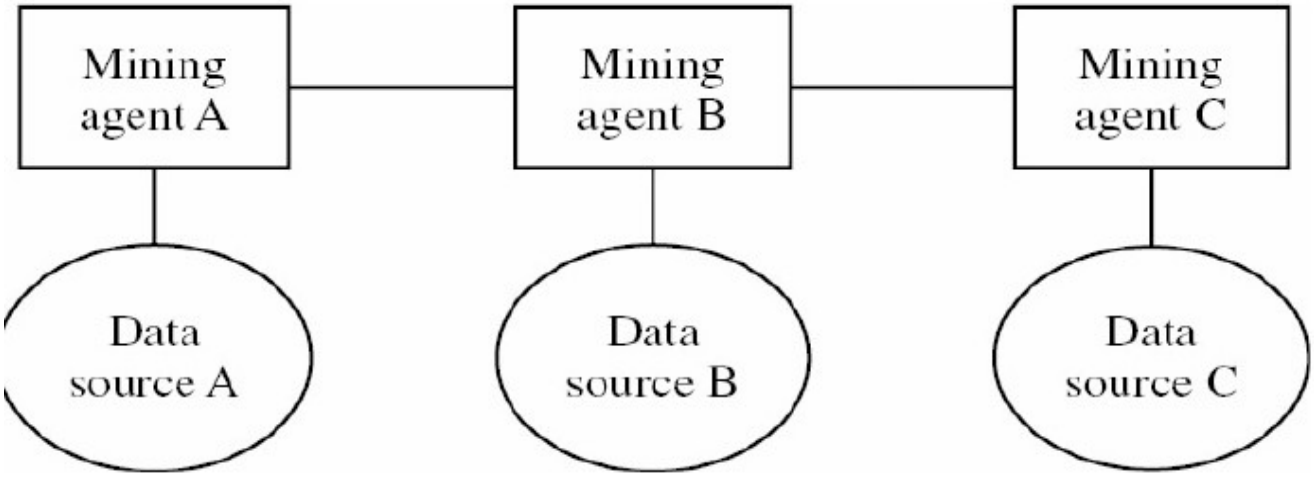


شـکل 27: کاوش interoperating و مخزن

میتوان از ابزارهایی پیچیده تر هم مثل عاملها برای کاوش منابع ناهمگن استفاده کرد، شکل۶-۲۸، که یک عامل مجتمع سازی نتایج کاوش همه عاملها را جمع می کند. عامل مجتمع سازی ممکن است یک بازگشت به عاملهای کاوش داشته باشد برای اینکه عاملهای کاوش ممکن است سؤالات بیشتری از پرس وجو از منابع داده ایجاد کنند و اطلاعات قابل توجهی رابه دست آورد. ارتباطی دو طرفه بین عامل مجتمع سازی عاملهای کاوش وجود دارد. پیشنهاد دیگر عامل مجتمع سازی ندارد اما به جای آن عاملهای کاوش متفاوتی دارد که با همدیگر الگوهای قابل توجه موجود در منابع داده مختلف را کشف می کنند. شکل۶-۲۹، روش اخیر، به نام داده کاوی تعاونی هم نامیده می شود. در این روش، محاسبه تعاوتی، داده حاوی و پایگاه داده ناهمگن باهم کار می کنند.



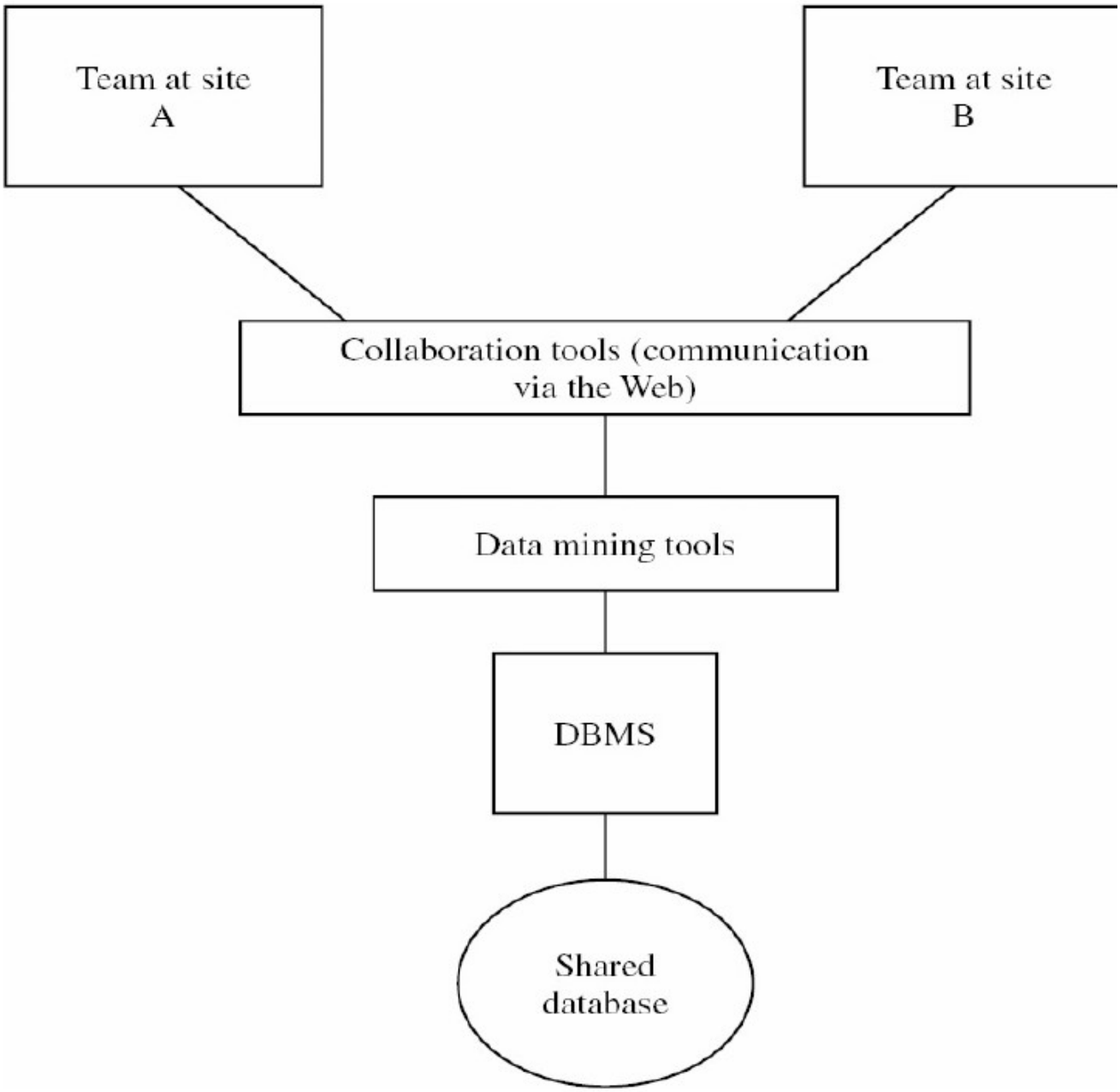
شکل 28: عاملهاي داده کاوي اجتماعي



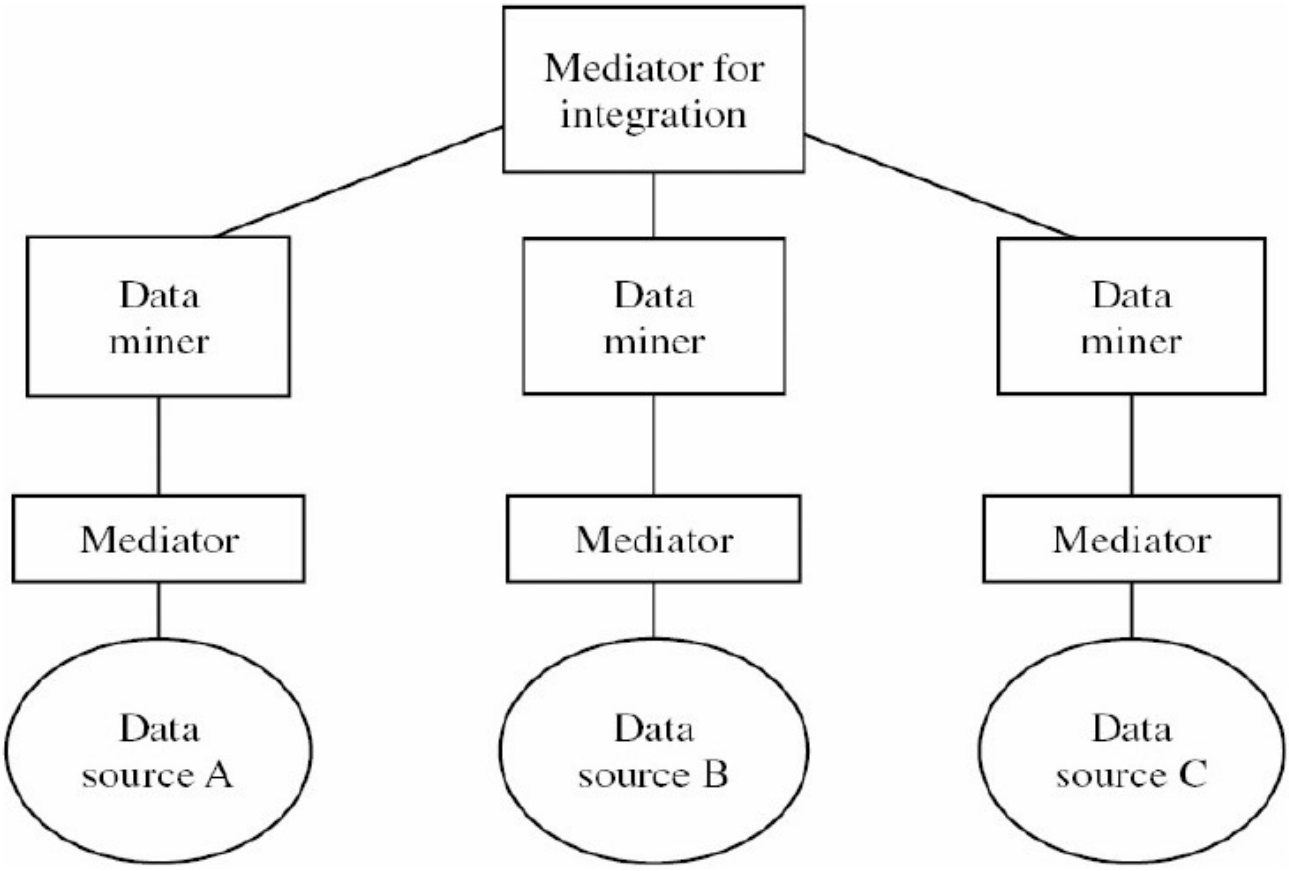
شکل 29: همکاري ميان عاملهاي کاوش

یک روش ویژه برای کاوش منابع ناهمگن، چه از یک عامل مجتمع سازی استفاده کنید یا همکاری بین عاملهای کاوش وجود داشته باشد، هنوز در حال شناخته شدن هستند. هر دو تا روش یا حتی روش دیگری ممکن است استفاده شود نکته اینجاست که ناهمگنی ممکن است د رمدلهای داده نوع داده ها و زبانها هم وجود داشته باشد. این چالش دیگر در پردازش داده کاوی است.تحقیقات بیشتری در این باره در حال انجام است.

شکل دیگر ازهمکاری در داده کاوی در شکل ۶-۳۰ نشان داده شده است. در اینجا دو تیم از دو سایت مختلف از اتحاد و ابزارهای کاوش برای کاوش پایگاه داده مشترکی استفاده می کنند. میتوان از واسطه ها نیز برای کاوش منابع داده ناهمگن استفاده کرد. شکل ۶-۳۱ یک مثال را نشان می دهد که ما فرض کردیم که داده کار همه منظوره و واسطه ها بین داده کاو و منابع داده قرار گرفته اند. همچنین از یک واسط برای جمع کردن نتایج از داده کاوهای متفاوت استفاده کرده ایم.



شكل 30: سيستمها كاوش را روي پايگاه دادة اشتراكي انجام مي‌دهند.

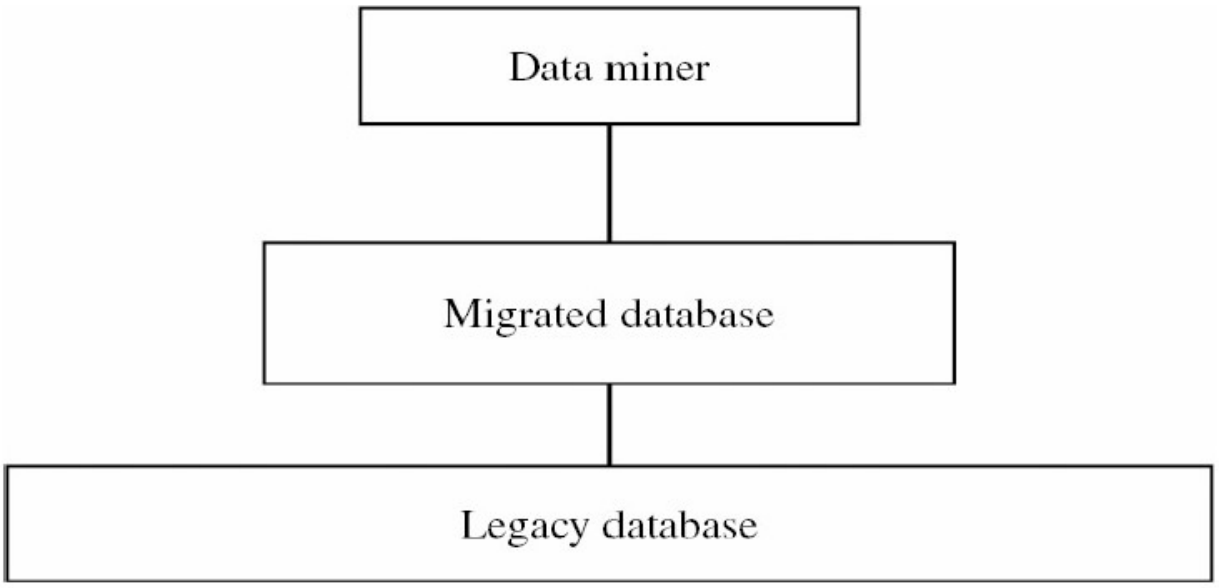


شکل 31: واسط براي مجتمع سازي

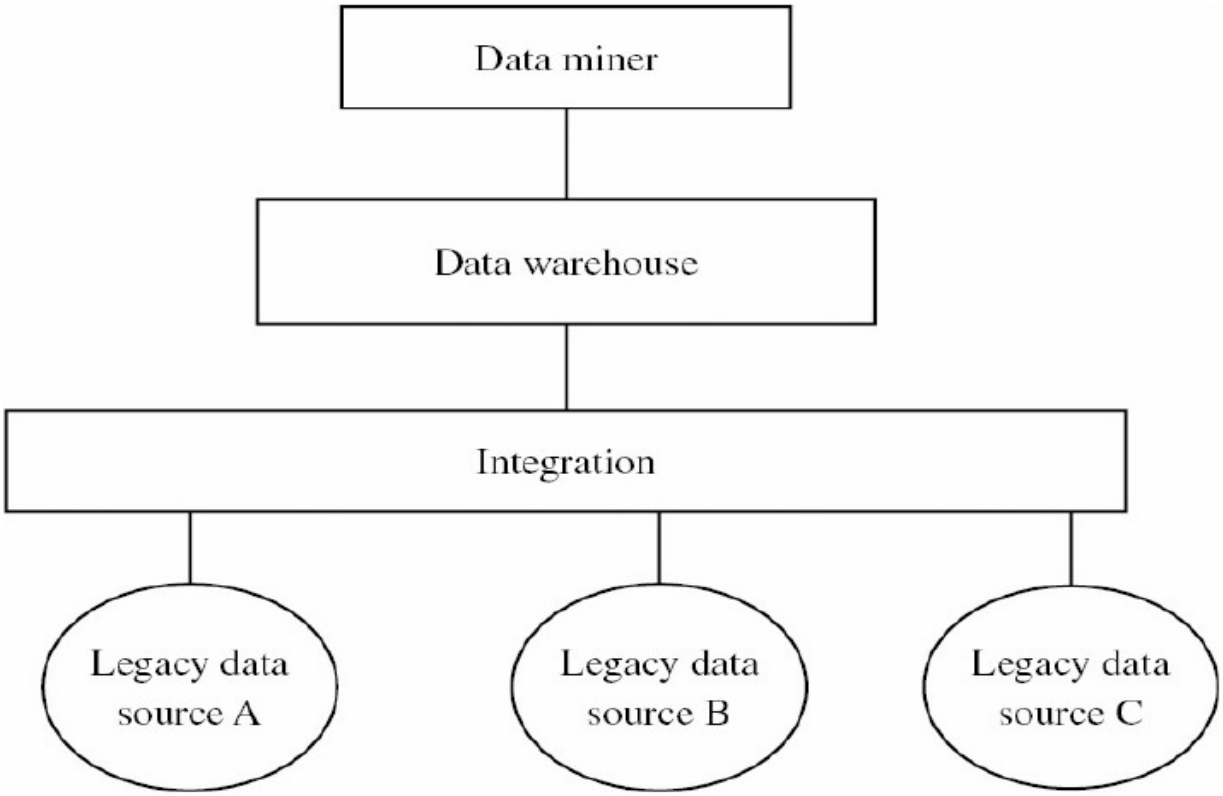
اکنون برروی پایگاه داده های وراثتی بحث خواهیم کرد. یک چالش در اینجا پاسخ به این سؤال است که چگونه پایگاه داده وراثتی را کاوش کنیم؟ آیا می توانیم به داده موجود در این پایگاه داده ها استفاده کنیم؟ آیا سازمان و قالب این داده ارزشی دارد و بویژه اینکه آیا از سیستم نوینی آمده است؟ اما ابزارهای توسعه دادن برای کاوش پایگاه داده های وراثتی ارزش دارند؟ چگونه جمع کردن پایگاه داده های وراثتی در مخازن داده ساده است؟ این چند حالت وجود دارد. اول اینکه پایگاه داده های وراثتی به سیستم جدیدی انتقال یابند و داده ها در سیستم جدید کاوش شوند. (شکل ۶-۳۲). روش دیگر اینست که پایگاه داده ها وفرمی از یک مخزن داده بر پایه معماری جدید و تکنولوژیهای جدید، باهم مجتمع شده و سپس داده در مخزن کاوش شود. (شکل ۶-۳۳). در اصل، کاوش مستقیم داده وراثتی فکر خوبی

نیست، به علت اینکه این داده احتمالاً به زودی خارج می شود یا ممکن است کامل نباشد، یا نامعلوم باشد. بنابراین کاوش گران تمام شود.

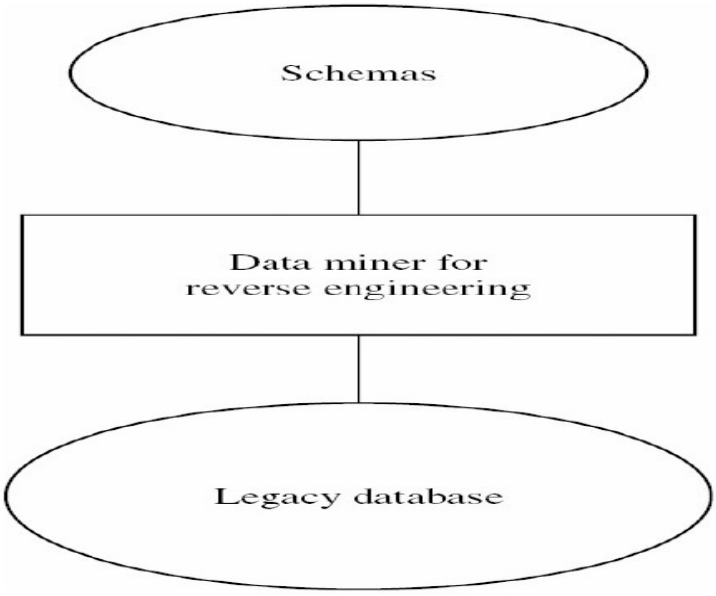
نکته اینجاست که کاوش می تواند برای مهندسی معکوس و استخراج الگو از پایگاه داده های وراثتی استفاده شود. (شکل۶-۳۴)



شکل 32: مهاجرت و سپس کاوش

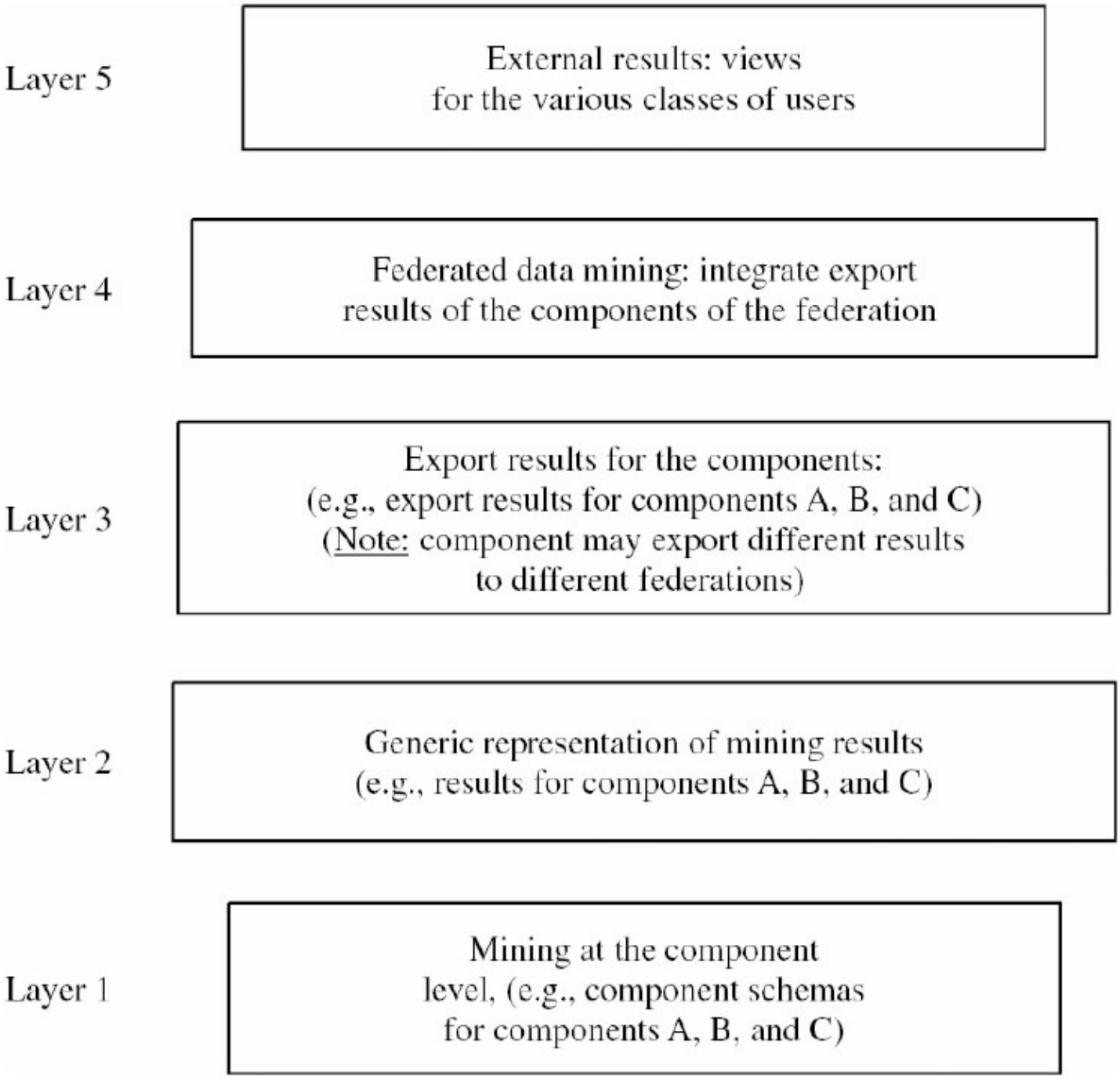


شکل 33: کاوش پایگاه داده‌های وراثتی



شکل 34: استخراج الگو از پایگاه داده‌ها وراثتي

در نهایت، معماری های متحد و داده کاوی را امتحان خواهیم کرد که به آن داده کاوی وابسته می گوییم. در اینجا، داده کاوها در سایتهای محلی احتیاج به استقلال و همچنین به اشتراک گذاری اطلاعات با سایتهای خارجی دارند. Seth و Larson، با یک معماری قابل توجه برای پایگاه داده های متحد مطرح شدند. اکنون ما احتیاج به وفق دادن آن با داده کاوی داریم. شکل ۶-۳۵ نظریه اولیه ای در باره داده کاوی متحد را نشان می دهد.



شکل 35: معماري 5 مرحله‌اي براي داده کاوي متحد

۶-۸) خلاصه: در این فصل ما جنبه های مختلف کاوش پایگاه داده های وب را بررسی کردیم. اول درباره پی آمد کاوش پایگاههای داده مثل داده کاوی مجتمع شده در بهینه سازی پرس وجوی وب بحث کردیم. سپس کاوش پایگاه داده های نیمه ساختیافته را بررسی کردیم. کاوش متادیتا بحث بعدی بود. در قسمت بعد کاوش به کاوش پایگاه داده های ناهمگن و توزیع شده نگاهی کردیم وهمچنین روشها داده

کاوی متحد را مورد بحث قرار دادیم ودر نهایت جنبه های معماری web data mining راتوضیح دادیم. هدف ما گرفتن اطلاعات حیاتی در بالاترین سطح است. ما به جزئیات الگوریتمهای بهینه سازی پرس و جو و کاوش متادیتا نمی پردازیم. هدفمان تولید جزئیات کافی است به طوریکه خواننده شروع فکری درباره به کارگیری تکنولوژی ما برای کاربردهای حساس مثل counterterrorism داشته باشد.

۶-۹) منابع :

- **Web Data Mining ، سیما رشیدی ، مهسا یغمایی**
- **Web Data Mining and Applications in Business Intelligence and Counter- Terrorism , Bhavani Thuraisingham , CRC Press, Boca Raton,۲۰۰۳.**
- [MITC۹۷] Mitchell, T., Machine Learning, McGraw-Hill, New York, ۱۹۹۷.
- BERR۹۷] Berry, M. and Linoff, G., Data Mining Techniques for Marketing, Sales, and Customer Support, John Wiley, New York, ۱۹۹۷.

۷) جدول یونی کد فارسی عربی :

Arabic

Range: 0600–06FF

The Unicode Standard, Version 6.0

This file contains an excerpt from the character code tables and list of character names for
The Unicode Standard, Version 6.0.

Characters in this chart that are new for The Unicode Standard, Version 6.0 are shown in conjunction with any existing characters. For ease of reference, the new characters have been highlighted in the chart grid and in the names list.

This file will not be updated with errata, or when additional characters are assigned to the Unicode Standard.
See http://www.unicode.org/errata/ for an up-to-date list of errata.

See http://www.unicode.org/charts/ for access to a complete list of the latest character code charts.
See http://www.unicode.org/charts/PDF/Unicode-6.0/ for charts showing only the characters added in Unicode 6.0.
See http://www.unicode.org/Public/6.0.0/charts/ for a complete archived file of character code charts for Unicode 6.0.

Disclaimer

These charts are provided as the online reference to the character contents of the Unicode Standard, Version 6.0 but do not provide all the information needed to fully support individual scripts using the Unicode Standard. For a complete understanding of the use of the characters contained in this file, please consult the appropriate sections of The Unicode Standard, Version 6.0, online at http://www.unicode.org/versions/Unicode6.0.0/, as well as Unicode Standard Annexes #9, #11, #14, #15, #24, #29, #31, #34, #38, #41, #42, and #44, the other Unicode Technical Reports and Standards, and the Unicode Character Database, which are available online.

See http://www.unicode.org/ucd/ and http://www.unicode.org/reports/

A thorough understanding of the information contained in these additional sources is required for a successful implementation.

Fonts

The shapes of the reference glyphs used in these code charts are not prescriptive. Considerable variation is to be expected in actual fonts. The particular fonts used in these charts were provided to the Unicode Consortium by a number of different font designers, who own the rights to the fonts.

See http://www.unicode.org/charts/fonts.html for a list.

Terms of Use

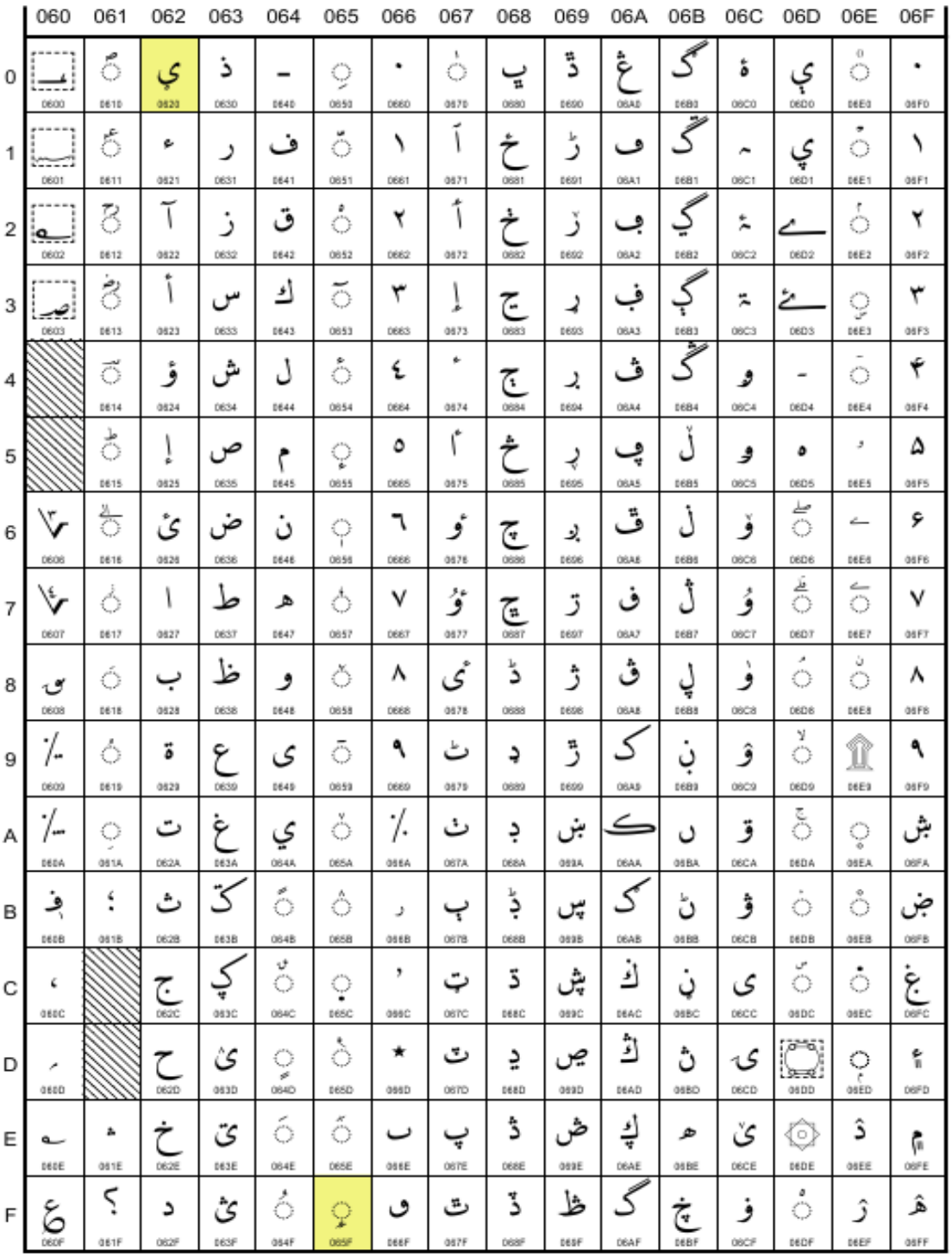
You may freely use these code charts for personal or internal business uses only. You may not incorporate them either wholly or in part into any product or publication, or otherwise distribute them without express written permission from the Unicode Consortium. However, you may provide links to these charts.

The fonts and font data used in production of these code charts may NOT be extracted, or used in any other way in any product or publication, without permission or license granted by the typeface owner(s).

The Unicode Consortium is not liable for errors or omissions in this file or the standard itself. Information on characters added to the Unicode Standard since the publication of the most recent version of the Unicode Standard, as well as on characters currently being considered for addition to the Unicode Standard can be found on the Unicode web site.

See http://www.unicode.org/pending/pending.html and http://www.unicode.org/alloc/Pipeline.html.

0600 Arabic 06FF



Subtending marks

0600 ARABIC NUMBER SIGN
0601 ARABIC SIGN SANAH
0602 ARABIC FOOTNOTE MARKER
0603 ARABIC SIGN SAFHA

Radix symbols

0606 ARABIC-INDIC CUBE ROOT
→ 221B ∛ cube root
0607 ARABIC-INDIC FOURTH ROOT
→ 221C ∜ fourth root

Letterlike symbol

0608 ARABIC RAY

Punctuation

0609 ARABIC-INDIC PER MILLE SIGN
→ 2030 ‰ per mille sign
060A ARABIC-INDIC PER TEN THOUSAND SIGN
→ 2031 ‱ per ten thousand sign

Currency sign

060B ARABIC SIGN AFGHANI

Punctuation

060C ARABIC COMMA
• also used with Thaana and Syriac in modern text
→ 002C , comma
060D ARABIC DATE SEPARATOR

Poetic marks

060E ARABIC POETIC VERSE SIGN
060F ARABIC SIGN MISRA

Honorifics

0610 ARABIC SIGN SALLALLAHOU ALAYHE WASSALLAM
• represents sallallahu alayhe wasallam "may God's peace and blessings be upon him"
0611 ARABIC SIGN ALAYHE ASSALLAM
• represents alayhe assalam "upon him be peace"
0612 ARABIC SIGN RAHMATULLAH ALAYHE
• represents rahmatullah alayhe "may God have mercy upon him"
0613 ARABIC SIGN RADI ALLAHOU ANHU
• represents radi allahu 'anhu "may God be pleased with him"
0614 ARABIC SIGN TAKHALLUS
• sign placed over the name or nom-de-plume of a poet, or in some writings used to mark all proper names

Koranic annotation sign

0615 ARABIC SMALL HIGH TAH
• marks a recommended pause position in some Korans published in Iran and Pakistan
• should not be confused with the small TAH sign used as a diacritic for some letters such as 0679 ٹ

Extended Arabic mark

0616 ARABIC SMALL HIGH LIGATURE ALEF WITH LAM WITH YEH
• early Persian

Koranic annotation signs

0617 ARABIC SMALL HIGH ZAIN
0618 ARABIC SMALL FATHA
• should not be confused with 064E ◌َ FATHA
0619 ARABIC SMALL DAMMA
• should not be confused with 064F ◌ُ DAMMA
061A ARABIC SMALL KASRA
• should not be confused with 0650 ◌ِ KASRA

Punctuation

061B ARABIC SEMICOLON
• also used with Thaana and Syriac in modern text
→ 003B ; semicolon
061C <reserved>
061D <reserved>
061E ARABIC TRIPLE DOT PUNCTUATION MARK
061F ARABIC QUESTION MARK
• also used with Thaana and Syriac in modern text
→ 003F ? question mark
→ 2E2E ⸮ reversed question mark

Addition for Kashmiri

0620 ARABIC LETTER KASHMIRI YEH

Based on ISO 8859-6

0621 ARABIC LETTER HAMZA
→ 02BE ʾ modifier letter right half ring
0622 ARABIC LETTER ALEF WITH MADDA ABOVE
= 0627 ا 0653 ◌ٓ
0623 ARABIC LETTER ALEF WITH HAMZA ABOVE
= 0627 ا 0654 ◌ٔ
0624 ARABIC LETTER WAW WITH HAMZA ABOVE
= 0648 و 0654 ◌ٔ
0625 ARABIC LETTER ALEF WITH HAMZA BELOW
= 0627 ا 0655 ◌ٕ
0626 ARABIC LETTER YEH WITH HAMZA ABOVE
= 064A ي 0654 ◌ٔ
0627 ARABIC LETTER ALEF
0628 ARABIC LETTER BEH
0629 ARABIC LETTER TEH MARBUTA
062A ARABIC LETTER TEH
062B ARABIC LETTER THEH
062C ARABIC LETTER JEEM
062D ARABIC LETTER HAH
062E ARABIC LETTER KHAH
062F ARABIC LETTER DAL
0630 ARABIC LETTER THAL
0631 ARABIC LETTER REH
0632 ARABIC LETTER ZAIN
0633 ARABIC LETTER SEEN
0634 ARABIC LETTER SHEEN
0635 ARABIC LETTER SAD
0636 ARABIC LETTER DAD
0637 ARABIC LETTER TAH
0638 ARABIC LETTER ZAH
0639 ARABIC LETTER AIN
→ 01B9 ʹ latin small letter ezh reversed
→ 02BF ʿ modifier letter left half ring
063A ARABIC LETTER GHAIN

Additions for early Persian and Azerbaijani

063B ARABIC LETTER KEHEH WITH TWO DOTS ABOVE
063C ARABIC LETTER KEHEH WITH THREE DOTS BELOW

063D ۽ ARABIC LETTER FARSI YEH WITH INVERTED V
• Azerbaijani
063E ۾ ARABIC LETTER FARSI YEH WITH TWO DOTS ABOVE
063F ۿ ARABIC LETTER FARSI YEH WITH THREE DOTS ABOVE

Based on ISO 8859-6

0640 ـ ARABIC TATWEEL
= kashida
• inserted to stretch characters
• also used with Syriac
0641 ف ARABIC LETTER FEH
0642 ق ARABIC LETTER QAF
0643 ك ARABIC LETTER KAF
0644 ل ARABIC LETTER LAM
0645 م ARABIC LETTER MEEM
0646 ن ARABIC LETTER NOON
0647 ه ARABIC LETTER HEH
0648 و ARABIC LETTER WAW
0649 ى ARABIC LETTER ALEF MAKSURA
• represents YEH-shaped letter with no dots in any positional form
064A ي ARABIC LETTER YEH

Points from ISO 8859-6

064B ً ARABIC FATHATAN
064C ٌ ARABIC DAMMATAN
064D ٍ ARABIC KASRATAN
064E َ ARABIC FATHA
064F ُ ARABIC DAMMA
0650 ِ ARABIC KASRA
0651 ّ ARABIC SHADDA
0652 ْ ARABIC SUKUN
• marks absence of a vowel after the base consonant
• used in some Korans to mark a long vowel as ignored
• can have a variety of shapes, including a circular one and a shape that looks like '٥'
→ 06E1 ٰ arabic small high dotless head of khah

Combining maddah and hamza

0653 ٓ ARABIC MADDAH ABOVE
0654 ٔ ARABIC HAMZA ABOVE
0655 ٕ ARABIC HAMZA BELOW

Other combining marks

0656 ٖ ARABIC SUBSCRIPT ALEF
0657 ٗ ARABIC INVERTED DAMMA
= ulta pesh
• Kashmiri, Urdu
0658 ٘ ARABIC MARK NOON GHUNNA
• Baluchi
• indicates nasalization in Urdu
0659 ٙ ARABIC ZWARAKAY
• Pashto
065A ٚ ARABIC VOWEL SIGN SMALL V ABOVE
• African languages
065B ٛ ARABIC VOWEL SIGN INVERTED SMALL V ABOVE
• African languages
065C ٜ ARABIC VOWEL SIGN DOT BELOW
• African languages
065D ٝ ARABIC REVERSED DAMMA
• Ormuri, African languages
065E ٞ ARABIC FATHA WITH TWO DOTS
• Kalami
065F ٟ ARABIC WAVY HAMZA BELOW
• Kashmiri

Arabic-Indic digits

These digits are used with Arabic proper; for languages of Iran, Pakistan, and India, see the Eastern Arabic-Indic digits at 06F0..06F9.

0660 ٠ ARABIC-INDIC DIGIT ZERO
0661 ١ ARABIC-INDIC DIGIT ONE
0662 ٢ ARABIC-INDIC DIGIT TWO
0663 ٣ ARABIC-INDIC DIGIT THREE
0664 ٤ ARABIC-INDIC DIGIT FOUR
0665 ٥ ARABIC-INDIC DIGIT FIVE
0666 ٦ ARABIC-INDIC DIGIT SIX
0667 ٧ ARABIC-INDIC DIGIT SEVEN
0668 ٨ ARABIC-INDIC DIGIT EIGHT
0669 ٩ ARABIC-INDIC DIGIT NINE

Punctuation

066A ٪ ARABIC PERCENT SIGN
→ 0025 % percent sign
066B ٫ ARABIC DECIMAL SEPARATOR
066C ٬ ARABIC THOUSANDS SEPARATOR
→ 0027 ' apostrophe
→ 2019 ' right single quotation mark
066D ٭ ARABIC FIVE POINTED STAR
• appearance rather variable
→ 002A * asterisk

Archaic letters

066E ٮ ARABIC LETTER DOTLESS BEH
066F ٯ ARABIC LETTER DOTLESS QAF

Point

0670 ٰ ARABIC LETTER SUPERSCRIPT ALEF
• actually a vowel sign, despite the name

Extended Arabic letters

0671 ٱ ARABIC LETTER ALEF WASLA
• Koranic Arabic
0672 ٲ ARABIC LETTER ALEF WITH WAVY HAMZA ABOVE
• Baluchi, Kashmiri
0673 ٳ ARABIC LETTER ALEF WITH WAVY HAMZA BELOW
• Kashmiri
0674 ٴ ARABIC LETTER HIGH HAMZA
• Kazakh
• forms digraphs
0675 ٵ ARABIC LETTER HIGH HAMZA ALEF
• Kazakh
≈ 0627 ا 0674 ٴ
0676 ٶ ARABIC LETTER HIGH HAMZA WAW
• Kazakh
≈ 0648 و 0674 ٴ
0677 ٷ ARABIC LETTER U WITH HAMZA ABOVE
• Kazakh
≈ 06C7 ۇ 0674 ٴ
0678 ٸ ARABIC LETTER HIGH HAMZA YEH
• Kazakh
≈ 064A ي 0674 ٴ

0679 ٹ ARABIC LETTER TTEH
• Urdu

067A ٺ ARABIC LETTER TTEHEH
• Sindhi

067B ٻ ARABIC LETTER BEEH
• Sindhi

067C پ ARABIC LETTER TEH WITH RING
• Pashto

067D ٽ ARABIC LETTER TEH WITH THREE DOTS ABOVE DOWNWARDS
• Sindhi

067E پ ARABIC LETTER PEH
• Persian, Urdu, ...

067F ٿ ARABIC LETTER TEHEH
• Sindhi

0680 ڀ ARABIC LETTER BEHEH
• Sindhi

0681 ځ ARABIC LETTER HAH WITH HAMZA ABOVE
• Pashto letter "dze"

0682 ڂ ARABIC LETTER HAH WITH TWO DOTS VERTICAL ABOVE
• not used in modern Pashto

0683 ڃ ARABIC LETTER NYEH
• Sindhi

0684 ڄ ARABIC LETTER DYEH
• Sindhi

0685 څ ARABIC LETTER HAH WITH THREE DOTS ABOVE
• Pashto, Khwarazmian

0686 چ ARABIC LETTER TCHEH
• Persian, Urdu, ...

0687 ڇ ARABIC LETTER TCHEHEH
• Sindhi

0688 ڈ ARABIC LETTER DDAL
• Urdu

0689 ډ ARABIC LETTER DAL WITH RING
• Pashto

068A ڊ ARABIC LETTER DAL WITH DOT BELOW
• Sindhi, early Persian

068B ڋ ARABIC LETTER DAL WITH DOT BELOW AND SMALL TAH
• Lahnda

068C ڌ ARABIC LETTER DAHAL
• Sindhi

068D ڍ ARABIC LETTER DDAHAL
• Sindhi

068E ڎ ARABIC LETTER DUL
• older shape for DUL, now obsolete in Sindhi
• Burushaski

068F ڏ ARABIC LETTER DAL WITH THREE DOTS ABOVE DOWNWARDS
• Sindhi
• current shape used for DUL

0690 ڐ ARABIC LETTER DAL WITH FOUR DOTS ABOVE
• old Urdu, not in current use

0691 ڑ ARABIC LETTER RREH
• Urdu

0692 ڒ ARABIC LETTER REH WITH SMALL V
• Kurdish

0693 ړ ARABIC LETTER REH WITH RING
• Pashto

0694 ڔ ARABIC LETTER REH WITH DOT BELOW
• Kurdish, early Persian

0695 ڕ ARABIC LETTER REH WITH SMALL V BELOW
• Kurdish

0696 ږ ARABIC LETTER REH WITH DOT BELOW AND DOT ABOVE
• Pashto

0697 ڗ ARABIC LETTER REH WITH TWO DOTS ABOVE
• Dargwa

0698 ژ ARABIC LETTER JEH
• Persian, Urdu, ...

0699 ڙ ARABIC LETTER REH WITH FOUR DOTS ABOVE
• Sindhi

069A ښ ARABIC LETTER SEEN WITH DOT BELOW AND DOT ABOVE
• Pashto

069B ڛ ARABIC LETTER SEEN WITH THREE DOTS BELOW
• early Persian

069C ڜ ARABIC LETTER SEEN WITH THREE DOTS BELOW AND THREE DOTS ABOVE
• Moroccan Arabic

069D ڝ ARABIC LETTER SAD WITH TWO DOTS BELOW
• Turkic

069E ڞ ARABIC LETTER SAD WITH THREE DOTS ABOVE
• Berber, Burushaski

069F ڟ ARABIC LETTER TAH WITH THREE DOTS ABOVE
• old Hausa

06A0 ڠ ARABIC LETTER AIN WITH THREE DOTS ABOVE
• old Malay

06A1 ڡ ARABIC LETTER DOTLESS FEH
• Adighe

06A2 ڢ ARABIC LETTER FEH WITH DOT MOVED BELOW
• Maghrib Arabic

06A3 ڣ ARABIC LETTER FEH WITH DOT BELOW
• Ingush

06A4 ڤ ARABIC LETTER VEH
• Middle Eastern Arabic for foreign words
• Kurdish, Khwarazmian, early Persian

06A5 ڥ ARABIC LETTER FEH WITH THREE DOTS BELOW
• North African Arabic for foreign words

06A6 ڦ ARABIC LETTER PEHEH
• Sindhi

06A7 ڧ ARABIC LETTER QAF WITH DOT ABOVE
• Maghrib Arabic

06A8 ڨ ARABIC LETTER QAF WITH THREE DOTS ABOVE
• Tunisian Arabic

06A9 ک ARABIC LETTER KEHEH
• Persian, Urdu, ...

06AA ڪ ARABIC LETTER SWASH KAF

06AB ګ ARABIC LETTER KAF WITH RING
• Pashto
• may appear like an Arabic KAF (0643 ك) with a ring below the base

06AC ڬ ARABIC LETTER KAF WITH DOT ABOVE
• old Malay

06AD ڭ ARABIC LETTER NG
• Uighur, Kazakh, old Malay, early Persian, ...

06AE ڮ ARABIC LETTER KAF WITH THREE DOTS BELOW
• Berber, early Persian

06AF گ ARABIC LETTER GAF
• Persian, Urdu, ...

06B0 ڰ ARABIC LETTER GAF WITH RING
• Lahnda

06B1 ڱ ARABIC LETTER NGOEH
• Sindhi
06B2 ڲ ARABIC LETTER GAF WITH TWO DOTS BELOW
• not used in Sindhi
06B3 ڳ ARABIC LETTER GUEH
• Sindhi
06B4 ڴ ARABIC LETTER GAF WITH THREE DOTS ABOVE
• not used in Sindhi
06B5 ڵ ARABIC LETTER LAM WITH SMALL V
• Kurdish
06B6 ڶ ARABIC LETTER LAM WITH DOT ABOVE
• Kurdish
06B7 ڷ ARABIC LETTER LAM WITH THREE DOTS ABOVE
• Kurdish
06B8 ڸ ARABIC LETTER LAM WITH THREE DOTS BELOW
06B9 ڹ ARABIC LETTER NOON WITH DOT BELOW
06BA ں ARABIC LETTER NOON GHUNNA
• Urdu
06BB ڻ ARABIC LETTER RNOON
• Sindhi
06BC ڼ ARABIC LETTER NOON WITH RING
• Pashto
06BD ڽ ARABIC LETTER NOON WITH THREE DOTS ABOVE
• old Malay
06BE ھ ARABIC LETTER HEH DOACHASHMEE
• Urdu
• forms aspirate digraphs
06BF ڿ ARABIC LETTER TCHEH WITH DOT ABOVE
06C0 ۀ ARABIC LETTER HEH WITH YEH ABOVE
= arabic letter hamzah on ha (1.0)
= izafet
• Urdu
• actually a ligature, not an independent letter
≡ 06D5 ہ 0654 ◌ٔ
06C1 ہ ARABIC LETTER HEH GOAL
• Urdu
06C2 ۂ ARABIC LETTER HEH GOAL WITH HAMZA ABOVE
• Urdu
• actually a ligature, not an independent letter
≡ 06C1 ہ 0654 ◌ٔ
06C3 ۃ ARABIC LETTER TEH MARBUTA GOAL
• Urdu
06C4 ۄ ARABIC LETTER WAW WITH RING
• Kashmiri
06C5 ۅ ARABIC LETTER KIRGHIZ OE
• Kirghiz
06C6 ۆ ARABIC LETTER OE
• Uighur, Kurdish, Kazakh, Azerbaijani
06C7 ۇ ARABIC LETTER U
• Kirghiz, Azerbaijani
06C8 ۈ ARABIC LETTER YU
• Uighur
06C9 ۉ ARABIC LETTER KIRGHIZ YU
• Kazakh, Kirghiz
06CA ۊ ARABIC LETTER WAW WITH TWO DOTS ABOVE
• Kurdish
06CB ۋ ARABIC LETTER VE
• Uighur, Kazakh

06CC ی ARABIC LETTER FARSI YEH
• Arabic, Persian, Urdu, Kashmiri, ...
• initial and medial forms of this letter have dots
→ 0649 ى arabic letter alef maksura
→ 064A ي arabic letter yeh
06CD ۍ ARABIC LETTER YEH WITH TAIL
• Pashto, Sindhi
06CE ێ ARABIC LETTER YEH WITH SMALL V
• Kurdish
06CF ۏ ARABIC LETTER WAW WITH DOT ABOVE
06D0 ې ARABIC LETTER E
• Pashto, Uighur
• used as the letter bbeh in Sindhi
06D1 ۑ ARABIC LETTER YEH WITH THREE DOTS BELOW
• old Malay
06D2 ے ARABIC LETTER YEH BARREE
• Urdu
06D3 ۓ ARABIC LETTER YEH BARREE WITH HAMZA ABOVE
• Urdu
• actually a ligature, not an independent letter
≡ 06D2 ے 0654 ◌ٔ

Punctuation

06D4 ۔ ARABIC FULL STOP
• Urdu

Extended Arabic letter

06D5 ە ARABIC LETTER AE
• Uighur, Kazakh, Kirghiz

Koranic annotation signs

06D6 ◌ۖ ARABIC SMALL HIGH LIGATURE SAD WITH LAM WITH ALEF MAKSURA
06D7 ◌ۗ ARABIC SMALL HIGH LIGATURE QAF WITH LAM WITH ALEF MAKSURA
06D8 ◌ۘ ARABIC SMALL HIGH MEEM INITIAL FORM
06D9 ◌ۙ ARABIC SMALL HIGH LAM ALEF
06DA ◌ۚ ARABIC SMALL HIGH JEEM
06DB ◌ۛ ARABIC SMALL HIGH THREE DOTS
06DC ◌ۜ ARABIC SMALL HIGH SEEN
06DD ۝ ARABIC END OF AYAH
06DE ۞ ARABIC START OF RUB EL HIZB
06DF ◌۟ ARABIC SMALL HIGH ROUNDED ZERO
• smaller than the typical circular shape used for 0652 ◌ْ
06E0 ◌ۢ ARABIC SMALL HIGH UPRIGHT RECTANGULAR ZERO
06E1 ◌ۡ ARABIC SMALL HIGH DOTLESS HEAD OF KHAH
= Arabic jazm
• presentation form of 0652 ◌ْ, using font technology to select the variant is preferred
• used in some Korans to mark absence of a vowel
→ 0652 ◌ْ arabic sukun
06E2 ◌ۣ ARABIC SMALL HIGH MEEM ISOLATED FORM
06E3 ◌ۤ ARABIC SMALL LOW SEEN
06E4 ◌ۥ ARABIC SMALL HIGH MADDA
06E5 ◌ۦ ARABIC SMALL WAW
06E6 ◌ۧ ARABIC SMALL YEH
06E7 ◌ۨ ARABIC SMALL HIGH YEH
06E8 ◌ۨ ARABIC SMALL HIGH NOON
06E9 ۩ ARABIC PLACE OF SAJDAH
• there is a range of acceptable glyphs for this character

