

دانشگاه ولی عصر (عج) رفسنجان

نام پژوه : کاربرد آمار در یادگیری ماشین

نام و نام خانوادگی دانشجو : محمد حسین زارعی

استاد راهنمای: دکتر عباس مهدوی

۱۴۰۴ پاییز

فهرست مطالب

۱	مقدمه	۱
۲	انواع یادگیری	۲
۴	۱.۲ رگرسیون در یادگیری ناظارت شده	۱.۲
۴	۱.۱.۲ رگرسیون خطی ساده	۱.۱.۲
۶	۲.۱.۲ رگرسیون خطی چند متغیره	۲.۱.۲
۸	۳.۱.۲ گرادیان(gradeint)	۳.۱.۲
۱۰	۴.۱.۲ رگرسیون لجستیک	۴.۱.۲
۱۲	۵.۱.۲ نظمدهی (Regularization)	۵.۱.۲
۱۲	۶.۱.۲ رگرسیون لاسو	۶.۱.۲
۱۳	۷.۱.۲ رگرسیون ریج	۷.۱.۲
۱۴	۲.۲ یادگیری بدون ناظارت	۲.۲
۱۴	K-means	۱.۲.۲

یادگیری ماشین شاخه‌ای از هوش مصنوعی است که به ماشین‌ها امکان می‌دهد از داده‌ها یاد بگیرند و رفتارهای هوشمندانه‌ای مشابه انسان‌ها بروز دهنده، بدون اینکه به صورت صریح برنامه‌ریزی شوند. این فناوری به ماشین‌ها اجازه می‌دهد با استفاده از الگوریتم‌های یادگیری، الگوها و روابط موجود در داده‌ها را شناسایی کرده و بر اساس آن‌ها تصمیم‌گیری کنند یا به محیط اطراف خود تأثیر بگذارند. در یک نگاه ساده، یادگیری ماشین می‌تواند به دو صورت عمل کند: در روش سنتی، برنامه‌نویسان با استفاده از عملیات منطقی و ریاضی، برنامه‌ای مشخص را برای ماشین طراحی می‌کنند. برای مثال، در مدیریت موجودی کارخانه، برنامه‌نویس با دریافت فرمول مقدار اقتصادی سفارش، برنامه نویسی می‌کند. که با تحلیل داده‌های موجودی و پارامترهای محیطی، مقدار بهینه سفارش را محاسبه و اعلام می‌کند. اما در رویکرد مدرن یادگیری ماشین، به جای ارائه برنامه‌های از پیش تعیین شده، ماشین با قرار گرفتن در معرض داده‌ها و بهره‌گیری از الگوریتم‌های یادگیری، به تدریج مدل‌های مورد نیاز خود را شکل می‌دهد. این فرایند مشابه یادگیری انسان‌ها، مانند کودکی است که به تدریج زبان را می‌آموزد و توانایی صحبت کردن را کسب می‌کند.

یادگیری ماشین (Machine Learning) در سال‌های اخیر پیشرفت‌های چشمگیری داشته و در حوزه‌های مختلف کاربردهای پیشرفته‌ای پیدا کرده است. در ادامه، به برخی از مثال‌های پیشرفته و نوآورانه یادگیری ماشین اشاره می‌کنم که نشان‌دهنده قدرت و تنوع این فناوری هستند:

۱. تشخیص و درمان پزشکی پیشرفته: - تشخیص سرطان با دقت بالا: الگوریتم‌های یادگیری عمیق (Deep Learning) مانند شبکه‌های کانولوشنی (CNN) برای تحلیل تصاویر پزشکی (CT، MRI، و ماموگرافی) استفاده می‌شوند. برای مثال، مدل‌های AI مانند Google Health's DeepMind در تشخیص سرطان پستان از تصاویر ماموگرافی، گاهی دقت بیشتری نسبت به رادیولوژیست‌های انسانی دارند. - **پیش‌بینی بیماری‌های ژنتیکی**: ابزارهایی مثل AlphaFold DeepMind ساختار پروتئین‌ها را با دقت بی‌سابقه‌ای پیش‌بینی می‌کنند که در توسعه داروهای جدید برای بیماری‌های پیچیده مانند آلزایمر یا سرطان نقش مهمی دارند.
۲. پردازش زبان طبیعی (NLP): مدل‌های زبانی پیشرفته: مدل‌هایی مانند GPT-4 یا LLaMA با تحلیل حجم عظیمی از داده‌های متنی، توانایی تولید متن‌های شبیه انسان، پاسخ به سوالات پیچیده، ترجمه زبان‌ها، و حتی نوشتمندی را دارند. این مدل‌ها در چتبات‌ها، دستیارهای مجازی، و تحلیل احساسات (Sentiment Analysis) کاربرد دارند. - ترجمه بلادرنگ: سیستم‌هایی مانند Google Translate با استفاده از یادگیری ماشین، ترجمه‌های صوتی و متنی را به صورت بلادرنگ با دقت بالا انجام می‌دهند.
۳. بینایی کامپیوتری: - تشخیص اشیا و چهره: الگوریتم‌های یادگیری عمیق در سیستم‌های امنیتی برای شناسایی چهره‌ها یا تشخیص اشیا در تصاویر و ویدئوها استفاده می‌شوند. مثلاً، سیستم‌های نظرارت شهری از این فناوری برای شناسایی رفتارهای مشکوک بهره می‌برند. - خودروهای خودران: شرکت‌هایی مثل Tesla و Waymo از یادگیری ماشین برای پردازش داده‌های LiDAR (دوربین‌ها) استفاده می‌کنند تا خودروها بتوانند موانع، علائم راهنمایی، و مسیرها را شناسایی کرده و رانندگی ایمن انجام دهند.
۴. یادگیری تقویتی در رباتیک: - ربات‌های خودآموز: الگوریتم‌های یادگیری تقویتی (Reinforcement Learning) به ربات‌ها امکان می‌دهند تا از طریق آزمون و خطا وظایف پیچیده‌ای مانند راه رفتن، گرفتن اشیا، یا حتی انجام جراحی را یاد بگیرند. برای مثال، ربات‌های Boston Dynamics از این فناوری برای حرکات پویا و تعادل استفاده می‌کنند. - **بازی‌های استراتژیک**: مدل‌هایی مثل DeepMind از AlphaGo با استفاده از یادگیری تقویتی، استراتژی‌های پیچیده‌ای را در بازی‌های تخته‌ای مانند Go توسعه داده‌اند که حتی قهرمانان انسانی را شکست داده‌اند.
۵. توصیه‌گرهای هوشمند: - سیستم‌های پیشنهاد‌دهنده: پلتفرم‌هایی مانند Netflix، Spotify، و Amazon از الگوریتم‌های یادگیری ماشین (مانند Collaborative Filtering و یادگیری عمیق) برای تحلیل رفتار کاربران و پیشنهاد محظوظ یا محصولات شخصی‌سازی شده استفاده می‌کنند. - تبلیغات هدفمند: شرکت‌های تبلیغاتی از یادگیری ماشین برای تحلیل داده‌های کاربران و ارائه تبلیغات متناسب با علائق آن‌ها بهره می‌برند.
۶. پیش‌بینی و تحلیل داده‌های کلان: - پیش‌بینی بازارهای مالی: الگوریتم‌های یادگیری ماشین برای تحلیل روندهای بازار، پیش‌بینی قیمت سهام، و مدیریت ریسک در مؤسسات مالی استفاده می‌شوند. - **مدیریت زنجیره تأمین**: شرکت‌هایی مانند Walmart از یادگیری ماشین برای بهینه‌سازی موجودی، پیش‌بینی تقاضا، و مدیریت لجستیک استفاده می‌کنند.
۷. کشف دارو و زیست‌فناوری: - طراحی مولکول‌های جدید: AI با شبیه‌سازی تعاملات شیمیایی، ترکیبات دارویی جدید را پیشنهاد می‌دهد. برای مثال، شرکت Insilico Medicine از یادگیری ماشین برای کشف داروهای جدید در زمان کوتاه‌تر استفاده کرده است. - تحلیل داده‌های ژنومی: الگوریتم‌ها برای شناسایی جهش‌های ژنتیکی و پیش‌بینی پاسخ بیماران به درمان‌های خاص استفاده می‌شوند.

۸. کشاورزی هوشمند: - کشاورزی ماشین با تحلیل داده‌های حسگرها، تصاویر ماهواره‌ای، و داده‌های آبوهایی، به کشاورزان کمک می‌کند تا زمان مناسب کاشت، آبیاری، و برداشت را تعیین کنند. مثلاً، سیستم‌های AI می‌توانند آفات یا بیماری‌های گیاهی را زودهنگام تشخیص دهند. - اتوماسیون مزارع: ربات‌های مجهز به AI برای کاشت، برداشت، و حتی جمع‌آوری داده‌های خاک استفاده می‌شوند.

۹. امنیت سایبری: - تشخیص تهدیدات: الگوریتم‌های یادگیری ماشین برای شناسایی حملات سایبری، بدافزارها، و رفتارهای غیرعادی در شبکه‌ها استفاده می‌شوند. مثلاً، سیستم‌های تشخیص نفوذ (IDS) از AI برای تحلیل ترافیک شبکه بهره می‌برند. - احراز هویت بیومتریک: AI در سیستم‌های تشخیص صدا یا اثر انگشت برای افزایش امنیت کاربرد دارد.

۱۰. هنر و خلاقیت: - تولید محتوا: ابزارهایی مانند DALL-E یا MidJourney با استفاده از یادگیری عمیق، تصاویر، موسیقی، یا حتی داستان‌هایی خلاقانه تولید می‌کنند که مشابه آثار انسانی هستند. - تحلیل شبکه‌ای هنری: AI می‌تواند آثار هنری را تحلیل کرده و سبک‌های جدید خلق کند یا حتی در مرمت آثار باستانی کمک کند.

ویژگی‌های پیشرفته این کاربردها: - یادگیری عمیق: استفاده از شبکه‌های عصبی عمیق با لایه‌های متعدد برای پردازش داده‌های پیچیده. - تحلیل داده‌های کلان: توانایی پردازش حجم عظیمی از داده‌ها در زمان کوتاه. - تطبیق‌پذیری: قابلیت یادگیری و بهبود مستمر با داده‌های جدید. - تعامل چندگزنه‌ای: ترکیب یادگیری ماشین با حسگرها، رباتیک، و اینترنت اشیا (IoT).

این مثال‌ها تنها بخشی از پتانسیل یادگیری ماشین را نشان می‌دهند. با پیشرفت الگوریتم‌ها و افزایش قدرت محاسباتی، انتظار می‌رود که کاربردهای پیشرفته‌تری در آینده ظهر کنند که زندگی بشر را بیش از پیش متحول کنند.

۲ انواع یادگیری

در یادگیری ماشین با دونوع یادگیری مواجه می‌شویم، که یک نوع آن یادگیری ناظارت‌شده (Supervised Learning) یکی از شاخه‌های اصلی یادگیری ماشین است که در آن مدل با استفاده از داده‌های برچسبدار آموزش می‌بیند. در این روش، داده‌های ورودی (ویژگی‌ها) همراه با خروجی‌های متناظر (برچسب‌ها) به مدل ارائه می‌شوند تا مدل بتواند رابطه بین ورودی‌ها و خروجی‌ها را یاد بگیرد و از آن برای پیش‌بینی خروجی‌های جدید استفاده کند. مفاهیم اصلی:

- داده‌های برچسب‌دار: داده‌های آموزشی شامل دو بخش هستند: ویژگی‌ها، برچسب
- (Features): متغیرهای ورودی که مدل از آن‌ها برای یادگیری استفاده می‌کند (مثل قد، وزن، یا دما).
- (Labels) برچسب : خروجی‌های مورد انتظار که مدل باید پیش‌بینی کند (مثل "ثبت" یا "منفی" در طبقه‌بندی، یا یک عدد در رگرسیون). انواع مسائل در یادگیری ناظارت‌شده:
- رگرسیون (Regression) : پیش‌بینی یک مقدار پیوسته، مثل پیش‌بینی قیمت خانه یا دمای هوا.
- طبقه‌بندی (Classification) : پیش‌بینی یک دسته یا کلاس، مثل تشخیص ایمیل به عنوان "هرزنامه" یا "غیرهرزنامه". مراحل یادگیری ناظارت‌شده:
- جمع‌آوری داده: تهیه مجموعه داده‌ای با ویژگی‌ها و برچسب‌های دقیق.
- آموزش مدل: استفاده از الگوریتم‌هایی مثل رگرسیون خطی، درخت تصمیم، یا شبکه‌های عصبی برای یادگیری رابطه بین ویژگی‌ها و برچسب‌ها.
- ارزیابی مدل: بررسی عملکرد مدل با معیارهایی مثل دقت (Accuracy)، خطای میانگین مربعات (MSE) یا ماتریس درهم‌ریختگی (Confusion Matrix)
- پیش‌بینی: استفاده از مدل آموزش‌دیده برای پیش‌بینی برچسب داده‌های جدید. لگوریتم‌های رایج: - رگرسیون خطی (Linear Regression) - رگرسیون لجستیک (Logistic Regression) - ماشین بردار پشتیبان (SVM) - درخت تصمیم (Decision Tree) - شبکه‌های عصبی (Neural Networks) - جنگل تصادفی (Random Forest)

زایا: - دقت بالا در صورتی که داده‌های برچسب‌دار کافی و باکیفیت وجود داشته باشد. - کاربرد گسترده در مسائل واقعی مثل تشخیص پژوهشکی، پیش‌بینی مالی، و پردازش زبان طبیعی. چالش‌ها: - نیاز به داده‌های برچسب‌دار که ممکن است جمع‌آوری آن‌ها زمان‌بر و پرهزینه باشد. - حساسیت به داده‌های پرت (Outliers) یا نویز. - خطر بیش‌برازش (Overfitting) اگر مدل بیش از حد به داده‌های آموزشی وابسته شود.

مثال عملی: فرض کنید یک مدل برای تشخیص تصاویر سگ و گربه بسازید: داده آموزشی مجموعه‌ای از تصاویر که هر کدام برچسب "سگ" یا "گربه" دارند. ویژگی‌ها: پیکسل‌های تصویر یا ویژگی‌های استخراج شده مثل رنگ و شکل. آموزش: مدل (مثل یک شبکه کانولوشنی) یاد می‌گیرد که الگوهای مربوط به سگ یا گربه را تشخیص دهد.

۱.۲ رگرسیون در یادگیری نظارت شده

رگرسیون یکی از تکنیک‌های اصلی در یادگیری نظارت شده (Supervised Learning) است که برای پیش‌بینی مقادیر پیوسته (Continuous) استفاده می‌شود. در یادگیری نظارت شده، مدل با استفاده از داده‌های ورودی (ویژگی‌ها) و خروجی‌های متناظر (برچسب‌ها) آموزش می‌بیند تا رابطه‌ای بین ورودی‌ها و خروجی‌ها را یاد بگیرد. در مسائل رگرسیون، خروجی یک مقدار عددی پیوسته است، مانند پیش‌بینی قیمت خانه، دمای هوا، یا میزان فروش. در رگرسیون، متغیر هدف (خروجی) مقداری پیوسته است، برخلاف مسائل طبقه‌بندی که خروجی گستته (مانند کلاس‌ها) است. هدف: یافتن تابعی که رابطه بین ویژگی‌های ورودی (X) و خروجی (y) را به بهترین شکل مدل کند.

تابع پیش‌بینی: مدل رگرسیون یک تابع (X)^f. تولید می‌کند که مقادیر پیش‌بینی شده (\hat{y}) را به ازای ورودی‌های جدید تخمین می‌زنند. خطای تفاوت بین مقدار واقعی (y) و مقدار پیش‌بینی شده (\hat{y}) به عنوان خطای محاسبه می‌شود. هدف کمینه کردن این خطای این است

۱.۱.۲ رگرسیون خطی ساده

رگرسیون خطی (Linear Regression) یکی از بنیادی‌ترین و پرکاربردترین الگوریتم‌های یادگیری ماشین است که برای پیش‌بینی مقادیر پیوسته استفاده می‌شود. در یادگیری نظارت شده، هدف یادگیری یک مدل است که رابطه‌ای بین ویژگی‌های ورودی (X) و خروجی (y) را با استفاده از داده‌های آموزشی (که شامل جفت‌های y , X هستند) مدل‌سازی کند. در رگرسیون خطی، فرض بر این است که رابطه بین ورودی‌ها و خروجی به صورت خطی است. در ادامه، به طور کامل به جنبه‌های مختلف رگرسیون خطی می‌پردازیم

کمینه سازی: تابع هزینه معیاری برای سنجش میزان خطای مدل در پیش‌بینی خروجی‌ها نسبت به مقادیر واقعی است. در رگرسیون خطی، تابع هزینه عموماً میانگین مربعات خطای (Mean Squared Error - MSE) است که تفاوت بین مقادیر واقعی (y_i) و پیش‌بینی شده (\hat{y}_i) را اندازه‌گیری می‌کند. دقت مدل را افزایش دهیم:

کمینه کردن تابع هزینه به معنای کاهش خطای پیش‌بینی است. این کار باعث می‌شود مدل پیش‌بینی‌هایی تولید کند که به مقادیر واقعی نزدیک‌تر باشند. پارامترهای بهینه را پیدا کنید. در رگرسیون خطی، پارامترها (w_0 و w_1) تعیین کننده خط پیش‌بینی هستند.

کمینه سازی تابع هزینه، مقادیر بهینه این پارامترها را پیدا می‌کند تا مدل بهترین تطابق را با داده‌ها داشته باشد.

تعییم مدل با کمینه سازی تابع هزینه روی داده‌های آموزشی، مدل می‌تواند الگوهای واقعی داده‌ها را یاد بگیرد و روی داده‌های جدید (داده‌های آزمون) عملکرد بهتری داشته باشد، به شرطی که از بیش‌برازش (Overfitting) جلوگیری شود.

معیاری برای مقایسه مدل‌ها: مقدار تابع هزینه می‌تواند به عنوان معیاری برای مقایسه عملکرد مدل‌های مختلف (مثلاً رگرسیون خطی در مقابل رگرسیون چندجمله‌ای) استفاده شود.

پایه‌ای برای الگوریتم‌های پیچیده‌تر: مفهوم کمینه سازی تابع هزینه در رگرسیون خطی، پایه‌ای برای یادگیری الگوریتم‌های پیچیده‌تر مانند شبکه‌های عصبی است که از روش‌های مشابه (مثلاً گرادیان نزولی) برای بهینه سازی استفاده می‌کنند.

اثبات کمینه سازی تابع هزینه در رگرسیون خطی

$$J(w_0, w_1) = \sum_{i=1}^n (y_i^{(i)} - \hat{y}^{(i)})^2 = \sum_{i=1}^n (y_i^{(i)} - (w_0 + w_1 x_i^{(i)}))^2 \quad (1)$$

که در آن y_i مقدار واقعی، $\hat{y}_i = w_0 + w_1 x_i$ مقدار پیش‌بینی شده می‌باشد.

کمینه سازی تابع هزینه برای کمینه کردن ($J(w_0, w_1)$)، مشتق‌های جزئی نسبت به w_0 و w_1 را محاسبه و صفر می‌کنیم.

مشتق نسبت به w_0

$$\frac{\partial J}{\partial w_0} = \frac{\partial}{\partial w_0} \left[\sum_{i=1}^n (y_i^{(i)} - (w_0 + w_1 x_i^{(i)}))^2 \right]$$

$$-2 \sum_{i=1}^n (y_i^{(i)} - w_0 - w_1 x_i^{(i)})^2 = 0$$

$$nw_0 + w_1 \sum_{i=1}^n x_i = \sum_{i=1}^n y_i$$

$$w_0 = \frac{\sum_{i=1}^n y_i - w_1 \sum_{i=1}^n x_i}{n} = \bar{y} - w_1 \bar{x}$$

که در آن $\bar{y} = \frac{\sum y_i}{n}$ و $\bar{x} = \frac{\sum x_i}{n}$ میانگین داده‌ها هستند.

w_1 مشتق نسبت به

$$\frac{\partial J}{\partial w_1} = \frac{\partial}{\partial w_1} \left[\sum_{i=1}^n (y_i^{(i)} - (w_0 + w_1 x_i^{(i)}))^2 \right]$$

$$= \sum_{i=1}^n 2(y_i^{(i)} - w_0 - w_1 x_i^{(i)}) (-x_i^{(i)}) = \sum_{i=1}^n x_i (y_i^{(i)} - w_0 - w_1 x_i^{(i)})$$

صرف کردن مشتق و جایگذاری $w_0 = \bar{y} - w_1 \bar{x}$

$$\sum_{i=1}^n x_i^{(i)} \left(y_i^{(i)} - (\bar{y} - w_1 \bar{x}) - w_1 x_i^{(i)} \right) = 0$$

$$\sum_{i=1}^n x_i^{(i)} (y_i^{(i)} - \bar{y} - w_1 (x_i^{(i)} - \bar{x})) = 0$$

$$\sum_{i=1}^n x_i^{(i)} (y_i^{(i)} - \bar{y}) - w_1 \sum_{i=1}^n x_i (x_i^{(i)} - \bar{x}) = 0$$

: w_1 برای حل

$$w_1 = \frac{\sum_{i=1}^n x_i^{(i)} (y_i^{(i)} - \bar{y})}{\sum_{i=1}^n x_i^{(i)} (x_i^{(i)} - \bar{x})}$$

برای ساده‌سازی، از فرم معادل استفاده می‌کنیم:

$$w_1 = \frac{\sum_{i=1}^n (x_i^{(i)} - \bar{x})(y_i^{(i)} - \bar{y})}{\sum_{i=1}^n (x_i^{(i)} - \bar{x})^2}$$

برآورده‌گرهای بهینه بنا براین، برآورده‌گرهای بهینه برای پارامترها عبارت‌اند از:

$$\hat{w}_1 = \frac{\sum_{i=1}^n (x_i^{(i)} - \bar{x})(y_i^{(i)} - \bar{y})}{\sum_{i=1}^n (x_i^{(i)} - \bar{x})^2} \quad (2)$$

$$\hat{w}_0 = \bar{y} - \hat{w}_1 \bar{x} \quad (3)$$

۲.۱.۲ رگرسیون خطی چند متغیره

رگرسیون خطی چندگانه (Multiple Linear Regression) یکی از روش‌های کلیدی در یادگیری نظرات شده است که برای پیش‌بینی یک متغیر وابسته (خروجی پیوسته) بر اساس چندین متغیر مستقل (ویژگی‌ها) استفاده می‌شود. این روش زمانی کاربرد دارد که بخواهیم رابطه‌ای خطی بین چندین متغیر ورودی و یک خروجی پیوسته را مدل کنیم. در یادگیری نظرات شده، داده‌های آموزشی شامل جفت‌های ورودی (ویژگی‌ها) و خروجی (برچسب‌ها) هستند، و رگرسیون خطی چندگانه با یادگیری این رابطه، پیش‌بینی‌های دقیقی برای داده‌های جدید ارائه می‌دهد. در ادامه، به کاربردهای این روش و فرضیات آن می‌پردازیم.

رگرسیون چندگانه به ما کمک می‌کند تا اثر هر متغیر مستقل بر خروجی را درک کنیم. ضرایب مدل نشان‌دهنده میزان تأثیر هر ویژگی هستند.

مثال: در بازاریابی، می‌توان تأثیر بودجه تبلیغات، نوع رسانه، و فصل فروش بر میزان فروش را بررسی کرد. کاربردهای گسترده در حوزه‌های مختلف:

اقتصاد: پیش‌بینی رشد اقتصادی یا تورم بر اساس متغیرهایی مانند نرخ بهره، تولید ناخالص داخلی، و بیکاری.

بیشکی: پیش‌بینی فشار خون بیمار بر اساس سن، وزن، سطح فعالیت بدنی، و رژیم غذایی.

علوم طبیعی: پیش‌بینی دمای هوا با استفاده از متغیرهایی مانند رطوبت، فشار جو، و سرعت باد.

بازاریابی و تجارت: پیش‌بینی فروش محصول بر اساس قیمت، تبلیغات، و رفتار مشتری.

مهندسی: پیش‌بینی مصرف انرژی یک ساختمان بر اساس اندازه، تعداد ساکنان، و نوع عایق‌بندی. پایه‌ای برای مدل‌های پیچیده‌تر رگرسیون خطی چندگانه به دلیل سادگی و تفسیرپذیری، پایه‌ای برای درک مدل‌های پیشرفته‌تر مانند رگرسیون منظم شده (Ridge، لاسو) یا شبکه‌های عصبی است.

این روش به عنوان یک معیار اولیه (Baseline) برای مقایسه با مدل‌های پیچیده‌تر استفاده می‌شود.

تصمیم‌گیری مبتنی بر داده:

با ارائه پیش‌بینی‌های دقیق، رگرسیون چندگانه به تصمیم‌گیری در حوزه‌هایی مانند مدیریت ریسک، برنامه‌ریزی مالی، و بهینه‌سازی منابع کمک می‌کند.

مثال: پیش‌بینی تقاضای محصول برای مدیریت موجودی در زنجیره تأمین.

تحلیل روابط پیچیده:

این روش امکان مدل‌سازی تعاملات بین متغیرهای مختلف را فراهم می‌کند، بهویژه زمانی که یک متغیر به تنها یکی نمی‌تواند خروجی را توضیح دهد.

مثال: پیش‌بینی عملکرد تحصیلی دانش‌آموز بر اساس ساعات مطالعه، کیفیت تدریس، و سطح استرس. کمینه‌سازی تابع هزینه در رگرسیون خطی چندگانه

رگرسیون خطی چندگانه رابطه بین متغیر وابسته (y) و چندین متغیر مستقل (x_1, x_2, \dots, x_n) را به صورت زیر مدل می‌کند:

$$y = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n \quad (4)$$

به صورت ماتریسی: $y = Xw$ که در آن:

$$X = [1, x_1, x_2, \dots, x_n] \quad \bullet \quad \text{ماتریس ویژگی‌ها (ستون اول برای } w_0 \text{ پر از ۱ است).}$$

$$w = [w_0, w_1, \dots, w_n]^T \quad \bullet \quad \text{بردار ضرایب.}$$

$$y: \text{بردار خروجی‌ها.} \quad \bullet$$

تابع هزینه تابع هزینه میانگین مربعات خطأ (MSE) به صورت زیر تعریف می‌شود:

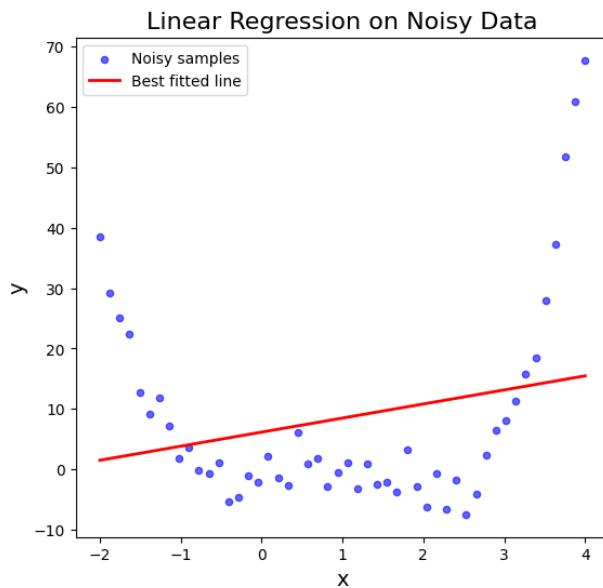
$$J(w) = \frac{1}{n} \sum_{i=1}^n (y_i - (w_0 + w_1x_{i1} + \dots + w_nx_{in}))^2 \quad (5)$$

یا به صورت ماتریسی:

$$J(w) = \frac{1}{n} (y - Xw)^T (y - Xw) \quad (6)$$

کمینه‌سازی رگرسیون چند جمله‌ای برای کمینه‌سازی ($J(w)$)، مشتق تابع هزینه نسبت به w محاسبه و صفر می‌شود. تابع هزینه را گسترش می‌دهیم:

$$J(w) = \frac{1}{n} [(y^T y - 2y^T Xw + w^T X^T Xw)]$$



شکل ۱: استفاده از رگرسیون خطی با که باعث نویز شده است.

مشتق نسبت به w :

$$\begin{aligned} \frac{\partial J}{\partial w} &= \frac{\partial}{\partial w} \left[\frac{1}{n} (y^T y - 2y^T Xw + w^T X^T Xw) \right] \\ &= \frac{1}{n} (-2X^T y + 2X^T Xw) = \frac{2}{n} (X^T Xw - X^T y) \end{aligned} \quad (7)$$

صفر کردن مشتق:

$$X^T Xw = X^T y \quad (8)$$

برآوردگر w :

$$\hat{w} = (X^T X)^{-1} X^T y \quad (9)$$

این برآوردگر بهینه، تابع هزینه را کمینه می‌کند، مشروط بر اینکه $X^T X$ معکوس پذیر باشد (یعنی ویژگی‌ها هم خطی کامل نداشته باشند).

تعمیم حالت خطی.؟؟ رگرسیون خطی ساده (Simple Linear Regression) برای مدل‌سازی رابطه بین یک متغیر مستقل (x) و یک متغیر وابسته (y) به کار می‌رود، اما در بسیاری از مسائل واقعی، خروجی (y) تحت تأثیر چندین متغیر مستقل است. در چنین مواردی، رگرسیون خطی چندگانه (Multiple Linear Regression) به دلیل توانایی در نظر گرفتن چندین متغیر مستقل به طور همزمان، انتخاب مناسب‌تری است. در ادامه، به دلایل اصلی استفاده از رگرسیون چندگانه به جای رگرسیون خطی ساده، استفاده خواهد شد.

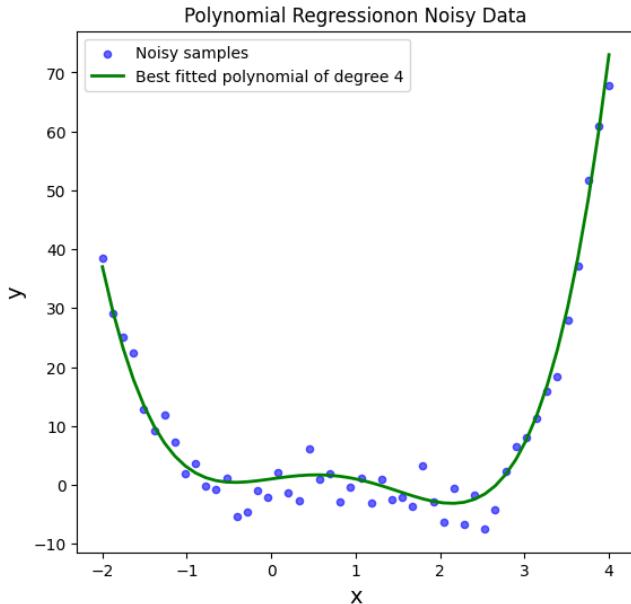
همان طوری که در بالا هم مشاهده می‌شود وقتی از رگرسیون خطی روی داده‌های چند گانه می‌شود باعث این می‌شود یک نویز یا خطایی به وجود آید حتی برای آن یک تخمین اولیه را نداشته باشیم. برای رفع این مشکل می‌توانیم از رگرسیون‌های چند گانه با درجه هایی مختلفی استفاده کنیم و این نویز را از بین ببریم. به تصویر عدی نگاه کنید.

همان طور که در تصویر بالا مشاهده می‌شود زمانی که درجه را زیاد می‌کنیم باعث می‌شود که نمودار به سمت چند جمله برود در نهایت خطایی مدل ما خیلی کاهش یابد و از overfit جلوگیری شود. نکته ای باید مد نظر داشته باشید شما می‌توانید از چند تبدیل های مختلفی استفاده کنید و بهترین بارزش را برای داده خود انجام دهید. در ادامه به برخی از این تبدیل‌ها اشاره خواهیم کرد.

انواع تبدیل‌ها

- تبدیل لگاریتمی: $y' = \log(y)$ یا $x'_i = \log(x_i)$. برای داده‌های با توزیع کج یا روابط نمایی.

- تبدیل چندجمله‌ای: $x'_i = x_i^2, x_i^3, \dots$. برای مدل‌سازی روابط غیرخطی.
- تبدیل ریشه مربع: $x'_i = \sqrt{x_i}$ یا $y' = \sqrt{y}$ برای کاهش کجی یا اثر مقادیر بزرگ.
- تبدیل معکوس: $\frac{1}{y}$. برای روابط معکوس یا مدیریت داده‌های پرت.



شکل ۲: استفاده نمودار غیر خطی.

از جمله تبدیل‌های هستند که می‌توانید برای برازش داده استفاده کنید از overfit و underfit شدن جلوگیری کنید.

۳.۱.۲ گرادیان (gradeint)

گرادیان نزولی (Gradient Descent) یکی از الگوریتم‌های بهینه‌سازی پرکاربرد در یادگیری ماشین است که برای کمینه‌سازی تابع هزینه (Cost Function) در مدل‌هایی مانند رگرسیون خطی، رگرسیون لجستیک، و شبکه‌های عصبی استفاده می‌شود. این روش بهویژه در مواردی که داده‌ها حجم زیادی دارند یا محاسبات تحلیلی (مانند روش حداقل مربعات) هزینه‌بر هستند، بسیار مفید است. در ادامه، توضیح جامعی درباره گرادیان نزولی، نحوه کار، انواع آن، کاربردها، و مزایا و معایب آن ارائه می‌شود.

گرادیان نزولی یک الگوریتم تکراری است که برای یافتن مقادیر بهینه پارامترهای مدل مانند (w_0, w_1, \dots, w_n) به گونه‌ای که تابع هزینه $J(w)$ کمینه شود، استفاده می‌شود. ایده اصلی این است که با حرکت در جهت مخالف گرادیان (مشتق تابع هزینه)، به سمت نقطه کمینه حرکت کنیم.

گرادیان: بردار مشتق‌های جزئی تابع هزینه نسبت به پارامترها، که جهت و شدت تغییرات تابع هزینه را نشان می‌دهد.

نزولی: حرکت در جهت مخالف گرادیان برای کاهش مقدار تابع هزینه.

شروع با مقادیر اولیه: مقادیر اولیه تصادفی یا صفر برای پارامترها (w) انتخاب می‌شود. محاسبه گرادیان: مشتق جزئی تابع هزینه نسبت به هر پارامتر محاسبه می‌شود:

$$\frac{\partial J}{\partial w_j}$$

به روزرسانی پارامترها: پارامترها در جهت مخالف گرادیان با گام مشخص (نرخ یادگیری، α) به روزرسانی می‌شوند:

$$w_j := w_j - \alpha \frac{\partial J}{\partial w_j}$$

تکرار: مراحل بالا تا زمانی تکرار می‌شود که تابع هزینه به مقدار کمینه برسد یا تغییرات آن ناچیز شود (همگرایی).

انواع گرادیان

گرادیان نزولی بر اساس نحوه استفاده از داده‌ها به سه نوع اصلی تقسیم می‌شود:

1. گرادیان نزولی دسته‌ای (Batch Gradient Descent): گرادیان با استفاده از کل داده‌های آموزشی محاسبه می‌شود. مزایا: همگرایی پایدارتر به سمت کمینه. معایب: برای داده‌های بزرگ، محاسبات سنگین و کند است.
 2. گرادیان نزولی تصادفی (Stochastic Gradient Descent - SGD): گرادیان برای هر نمونه داده به صورت جداگانه محاسبه و پارامترها به روزرسانی می‌شوند. مزایا: سریع‌تر برای داده‌های بزرگ، می‌تواند از کمینه‌های محلی فرار کند. معایب: نوسانات زیاد درتابع هزینه، ممکن است به کمینه دقیق نرسد.
 3. گرادیان نزولی مینی‌بچ (Mini-Batch Gradient Descent): داده‌ها به دسته‌های کوچک (mini-Batches) تقسیم می‌شوند و گرادیان برای هر دسته محاسبه می‌شود. مزایا: تعادل بین سرعت و پایداری، پرکاربرد در شبکه‌های عصبی. معایب: نیاز به تنظیم اندازه دسته.
- کاربرد گرادیان نزولی در رگرسیون خطی** در رگرسیون خطی،تابع هزینه معمولاً میانگین مربعات خطأ (MSE) است:

$$J(w) = \frac{1}{n} \sum_{i=1}^n (y_i - (w_0 + w_1 x_{i1} + \dots + w_n x_{in}))^2$$

گرادیان نزولی برای یافتن w_0, w_1, \dots, w_n استفاده می‌شود:
مشتق‌های جزئی:

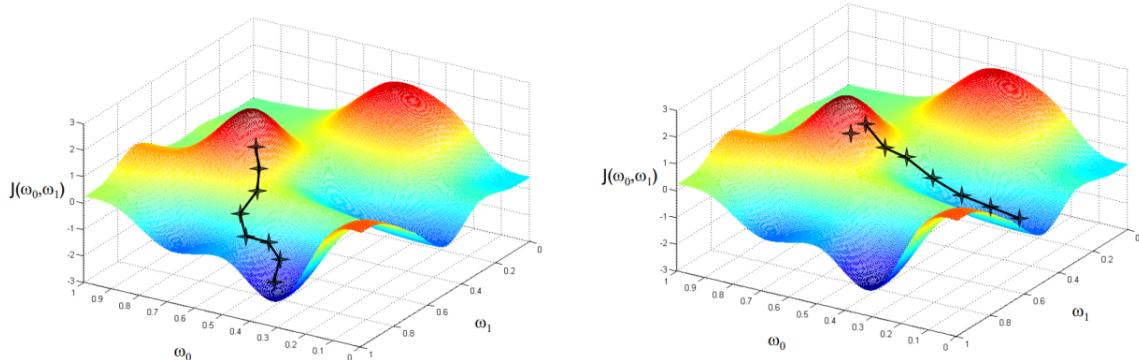
$$\frac{\partial J}{\partial w_0} = -\frac{2}{n} \sum_{i=1}^n (y_i - \hat{y}_i)$$

$$\frac{\partial J}{\partial w_j} = -\frac{2}{n} \sum_{i=1}^n (y_i - \hat{y}_i) x_{ij}, \quad j = 1, \dots, n$$

به روزرسانی:

$$w_j := w_j + \alpha \frac{2}{n} \sum_{i=1}^n (y_i - \hat{y}_i) x_{ij}$$

در این نمودار یک سطح سه بعدی است که محورهای آن به صورت زیر تعریف شده‌اند:



شکل ۳: گرادیان کاهشی

- محور x : θ_0 (پارامتر اول، مثلث (w_0) .

• محور y : θ_1 (پارامتر دوم، مثلًا w_1).

• محور هر نمودار یک سطح سه بعدی است که محورهای آن به صورت زیر تعریف شده‌اند:

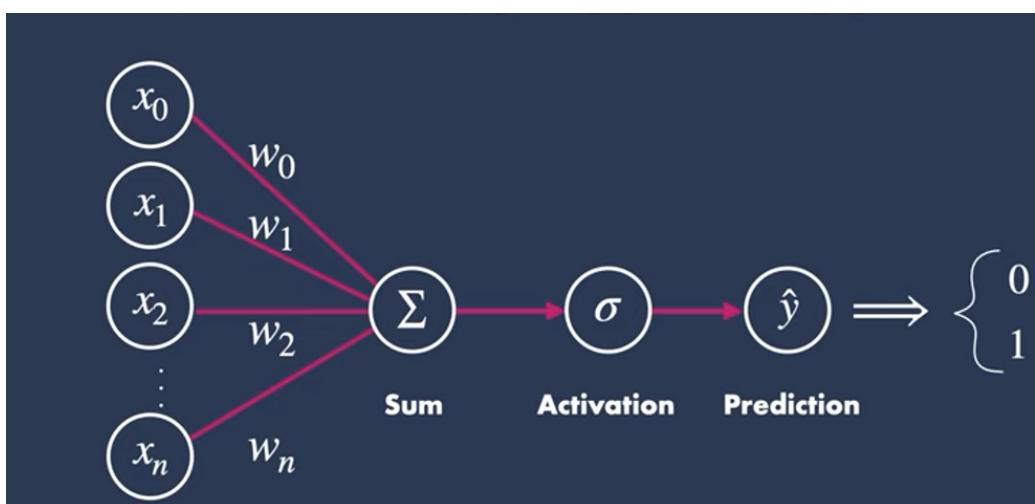
محور x : θ_0 (پارامتر اول، مثلًا w_0). محور y : θ_1 (پارامتر دوم، مثلًا w_1). محور z : $J(\theta_0, \theta_1)$. (مقدار تابع هزینه).
نمودار سمت چپ: مسیر گرادیان نزولی با نرخ یادگیری (α) بالاتر یا تعداد مراحل کمتر نشان داده شده است. مسیر دارای انحنا و نوسانات بیشتری است و به نظر می‌رسد هنوز به کمینه دقیق نرسیده است.
نمودار سمت راست: مسیر صاف‌تر و مستقیم‌تر است، که نشان‌دهنده نرخ یادگیری مناسب‌تر یا تعداد مراحل بیشتر است. این مسیر به طور واضح به سمت کمینه همگرا شده است.

۴.۱.۲ رگرسیون لجستیک

می‌توان تابع لجستیک را به این شکل تعریف کرد.

$$\hat{Y} = \sigma(z) = \sigma(x_0w_0 + x_1w_1 + \dots + x_nw_n)$$

تابع سیگومد به این شکل می‌باشد.



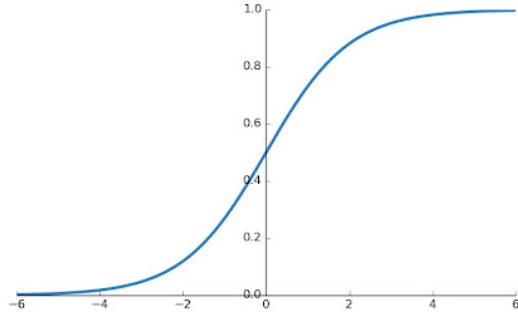
شکل ۴: تابع لجستیک در یادگیری ماشین

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

و همچنین در مورد مشقق پذیری

$$\begin{aligned}\lim_{z \rightarrow -\infty} \sigma(x) &= \frac{1}{1 + \infty} = 0 \\ \lim_{z \rightarrow \infty} \sigma(x) &= \frac{1}{1 + 0} = 1 \\ \lim_{z \rightarrow 0} \sigma(x) &= \frac{1}{1 + 1} = 0.5\end{aligned}$$

از آن جایی که $x < -\infty$ پس می‌توان نوشت $0 < \sigma(z) < 1$ می‌باشد. و نمودار به شکل زیر می‌باشد



شکل ۵: نمودار تابع لجسیتک

یکی دیگر از ویژگی های خوبی که این تابع دارد متفرق پذیر است.

$$\frac{\partial}{\partial z} \left(\frac{1}{(1 + e^{-z})} \right) = -\frac{e^{-z}}{(1 + e^{-z})^2} \frac{1}{1 + e^{-z}} = (\sigma(z))(1 - \sigma(z))$$

در یادگیری ماشین مانند داریم که عددی که با پیش بینی های که به ما باز گردانده می شوند یک عدد بین صفر و یک باشد که برای همین می توان از تابع لجسیتک به شکل نوشت:

$$p(y = 1 | x, w) = \sigma(w^T x)$$

$$p(y = 0 | x, w) = (1 - \sigma(w^T x))$$

به دست آوردن تابع هزینه در رگرسیون لجستیک

$$P(y_i = 1 | x_i; w) = \sigma(w^T x_i) = \frac{1}{1 + e^{-w^T x_i}}$$

برای $y_i \in \{0, 1\}$ ، توزیع برنولی را داریم:

$$P(y_i | x_i; w) = \sigma(w^T x_i)^{y_i} \cdot (1 - \sigma(w^T x_i))^{1-y_i}$$

با فرض مستقل بودن مشاهدات، تابع درستنمایی برابر است با:

$$L(w) = \prod_{i=1}^n \sigma(w^T x_i)^{y_i} \cdot (1 - \sigma(w^T x_i))^{1-y_i}$$

برای ساده سازی، لگاریتم تابع درستنمایی را می گیریم:

$$\ell(w) = \log L(w) = \sum_{i=1}^n [y_i \log \sigma(w^T x_i) + (1 - y_i) \log(1 - \sigma(w^T x_i))]$$

حال برای پیدا کردن بردار وزن بهینه w ، گرادیان تابع لگاریتم درستنمایی را محاسبه می کنیم:

$$\nabla_w \ell(w) = \sum_{i=1}^n (y_i - \sigma(w^T x_i)) x_i$$

این تابع شبیه به رگرسیون نیست که با استفاده از فرم بسته بتوان برای w پیدا کرد. پس با استفاده از گرایان مقدار بهینه را برای پارامتر پیدا خواهیم کرد.

برای بیشینه سازی تابع درستنمایی، از الگوریتم های عددی مانند گرادیان افزایشی (Gradient Ascent) یا گرادیان کاهشی (Gradient Descent) استفاده می شود. در عمل، ما عموماً گرادیان کاهشی را برای کمینه سازی تابع هزینه معادل استفاده می کنیم:

$$J(w) = -\ell(w) = -\sum_{i=1}^n [y_i \log \sigma(w^\top x_i) + (1-y_i) \log(1-\sigma(w^\top x_i))]$$

مشتق منفی آن نیز همان است:

$$\nabla_w J(w) = -\sum_{i=1}^n (y_i - \sigma(w^\top x_i)) x_i$$

۵.۱.۲ نظمدهی (Regularization)

یک تکینک مهم در ماشین لرنینگ است که باعث کاهش کمینه کردن و بهبود دقت می شود، با جلوگیری از بیش برآش (Overfitting) حذف داده های پرداز و باعث کاهش نویز می شود و کمک می کند مدل ساده تر شود و روی داده های جدید به خوبی عمل می کند. ابن عمل فقط با افزون یک ترم یا جریمه صورت خواهد گرفت.

انواع (Regularization) مدلی رگرسیونی که از تکینک نظمدهی L1 استفاده می کند، رگرسیون لاسو (LASSO) نام دارد که مخفف (Least Absolute Shrinkage and Selection Operator) است. در این روش، مقدار قدر مطلق ضرایب (ضرایب های مدل) به عنوان یک جریمه (penalty) بهتابع زیان (L) اضافه می شود. این جریمه می تواند باعث شود برخی ضرایب کاملاً به صفر برسند، که در نتیجه ویژگی های کم اهمیت حذف شده و تنها ویژگی های مهم باقی میمانند.

۶.۱.۲ رگرسیون لاسو

یک روش رگرسیونی مبتنی بر تکینک "کوچکسازی و انتخاب بر اساس قدر مطلق" است که در تحلیل رگرسیون برای انتخاب متغیرها و اعمال نظمدهی Regularization استفاده می شود. این روش به حذف ویژگی های غیرمرتبه داده کمک می کند و از بیش برآش Overfitting جلوگیری می کند. در نتیجه، ویژگی هایی که تأثیر ضعیفی دارند به خوبی مشخص می شوند، چرا که ضرایب متغیرهای کم اهمیت به سمت صفر می کنند. در حقیقت می توان گفت که رگرسیون لاسو یکی از روش هایی است که با افزودن یک جمله جریمه به تابع هزینه مدل خطی، از بیش برآش جلوگیری می کند. تابع هزینه در این روش به صورت زیر تعریف می شود:

$$J(w) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{i=1}^m |w_i|$$

و همچنین در فرمول بالا \hat{y}_i = مقدار مشاهده شده.

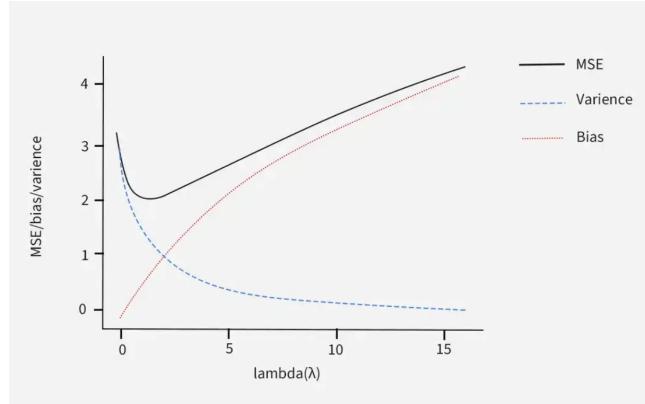
λ = مقدار پاسخ بیش بینی شده و در قسمت جمله جریمه (penalty prem) $\lambda \sum |w_i|$ این تابع مشتق پذیر نمی باشد.

w = ضرایب

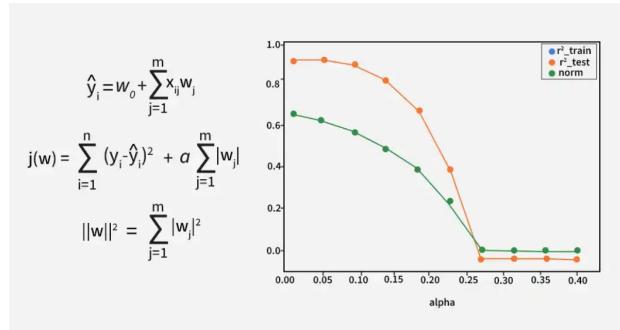
λ یک پارامتر تنظیم (uning parameter) است که شدت جریمه (penalty) را کنترل می کند. هرچه مقدار λ بیشتر شود، ضرایب بیشتری به سمت صفر رانده می شوند. و می توان به شکل زیر فرمول بالا را بازنویسی کرد:

$$J(w) = \min RSS + \lambda \sum_{i=1}^m |w_i|$$

تنظیم مقدار بهینه λ در ادامه دو معیار مهم در ماشین لرنینگ را مورد بررسی قرار خواهیم داد. $Bias$ = وقتی یک مدل بسیار ساده باشد و داده های ما پیچده باشند نقاط پرت زیاد می شوند در نهایت خطا افزایش می یابد. $Variance$ = در حقیقت وقتی مدل پیچیده باشد می تواند تنها روی داده های آموزش یا داده تست به خوبی فیت شود و در حالی در برگیرنده داده های آموزش نیست در نهایت می توان گفت که واریانس زیاد شده و اورفیت رخ داده است. $panlty$ = کاربردی که دارد باعث می شود ویژگی های که اهمیت کمی دارند حذف شوند. و این عمل باعث کاهش واریانس می شود و از اورفیت شدن جلوگیری می کند. افزایش مقدار λ باعث این می شود مدل بیش از حد ساده باشد و در نهایت باعث این می شود که underfit رخ دهد. برای رسیدن به تعادل درست بین Bias و Variance باید مقدار مناسب λ را با استفاده از cross-validation پیدا کرد.



شکل ۶: biase and varince in lasso



شکل ۷: cost function

در نمودار مربوط به Residual Sum of Squares، تابع هزینه به صورت ترکیبی از مجموع مربعات باقیماندها (RSS) و یک جریمه‌ی L_1 بر ضرایب β تعریف می‌شود. بخش اول یعنی L_1 ، اختلاف مربعی بین مقادیر واقعی و مقادیر پیش‌بینی شده را اندازه‌گیری می‌کند. بخش دوم، یعنی جریمه‌ی L_1 (با L_1 penalty)، مقدار قدر مطلق ضرایب را جریمه می‌کند و باعث می‌شود برخی از ضرایب به صفر نزدیک شوند یا دقیقاً صفر شوند. این امر باعث ساده‌تر شدن مدل می‌گردد. شدت اثر این جریمه توسط پارامتر λ (یعنی λ) کنترل می‌شود.

در نمودار، محور عمودی (محور y) مقدار تابع هزینه را نشان می‌دهد که Lasso Regression تلاش می‌کند آن را کمینه کند، و محور افقی (محور X) مقدار پارامتر λ را نمایش می‌دهد که شدت جریمه‌ی L_1 را مشخص می‌کند. منحنی‌ای که از رنگ سبز به نارنجی تغییر می‌کند، نشان‌دهنده‌ی تغییرات مقدار تابع هزینه نسبت به تغییرات λ است. با افزایش λ ، مقدار جریمه بیشتر شده و در نتیجه مقدار تابع هزینه نیز افزایش می‌یابد. این افزایش موجب می‌شود ضرایب بیشتری به سمت صفر رانده شوند و در نتیجه مدل ساده‌تر شود.

۷.۱.۲ رگرسیون ریج

رگرسیون ریج یک regularized cost function است از رگرسیون خطی. و با اضافه کردن یک تابع جریمه به کمک می‌کند یک به نقطه بهینه برسیم و هم‌چنین overfit کمتری رخ بدهد.

$$J(\theta) = MSE(\theta) + \alpha \sum_i^n \theta_i^2$$

نکته: در رگرسیو ریج به این مورد توجه کنید که ویژگی‌های کمی دارند کوچیک می‌شوند یا نزیدک به صفر. نکته: از این ترم $\alpha \sum_{i=1}^n \theta_i^2$ استفاده می‌شود.

نکته: برای این که معیار عمل کرد مدل را تشخیص دهیم. یعنی وقتی که مدل را استفاده از تست دیتا عملکرد آن را مورد سنجش قرار می‌دهیم از این ترم استفاده خواهیم $J(\theta) = MSE(\theta)$ این قسمت برای این که متوجه پذیر است و گرادیان آن راحت حساب می‌شود از آن استفاده خواهیم کرد.

توان منظمدهی (Regularization strength) با آلفا (α) مشخص می‌شود که یکی از hyperparameter و دو ویژگی دارد. اگر $\alpha = 0$ باشد، هیچ منظمدهی اعمال نمی‌شود و مدل به رگرسیون خطی استاندارد (OLS) تبدیل می‌شود.

۲.۲ یادگیری بدون ناظارت

تعریف... یادگیری بدون ناظارت (Unsupervised Learning) به تحلیل داده‌های بدون برچسب برای کشف الگوهای ساختارهای مخفی در داده‌ها اشاره دارد.

K-means ۱.۲.۲

۱. خوشبندی (Clustering)

- داده‌ها را بر اساس شباهت به گروههایی تقسیم می‌کند.
- نقاط داده در یک خوش به یکدیگر شبیه‌ترند تا به نقاط خوش‌های دیگر.
- شباهت بسته به نوع مسئله تعریف می‌شود (مثلاً رفتار خرید در تقسیم‌بندی بازار).

۲. کاهش ابعاد (Dimensionality Reduction):

- تعداد ویژگی‌های داده را کاهش می‌دهد، اما ویژگی‌های مهم و اطلاعاتی را حفظ می‌کند.

۳. تشخیص ناهنجاری (Anomaly Detection):

- شناسایی نقاط داده‌ای که به طور قابل توجهی از حالت عادی منحرف هستند (مثلاً تشخیص تقلب).

۴. مدل‌سازی مولد (Generative Modeling):

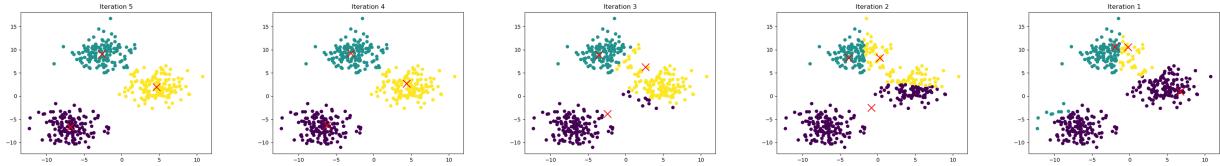
- یادگیری توزیع داده‌ها برای تولید نمونه‌های جدید و مشابه.

کاربردهای خوشبندی

- تقسیم‌بندی مشتریان (Customer Segmentation): در بازاریابی برای گروه‌بندی مشتریان بر اساس رفتار.
- تقسیم‌بندی تصویر و تشخیص اشیا (Image Segmentation and Object Detection): در بینایی کامپیوترا.
- تشخیص ناهنجاری: در امنیت سایبری و امور مالی.
- ژنتیک و بیوانفورماتیک: تحلیل داده‌های زیستی.
- تحلیل شبکه‌های اجتماعی و تشخیص جوامع: شناسایی گروه‌های مشابه در شبکه‌ها.

K-Means خوشبندی الگوریتم

- پرکاربردترین الگوریتم خوشبندی.
- داده‌ها را به K گروه مجزا بر اساس شباهت ویژگی‌ها تقسیم می‌کند.
- با تخصیص تکراری نقاط داده به نزدیک‌ترین مرکز (میانگین گروه) کار می‌کند و سپس مرکز را بر اساس عضویت‌های جدید گروه‌ها بازمحاسبه می‌کند.
- این فرآیند تا زمانی که تخصیص‌ها دیگر تغییر نکنند، تکرار می‌شود.
- هدف: نقاط داده در یک خوش باید به یکدیگر شبیه باشند.



شکل ۸: K-means

- در K-Means، این هدف به صورت تابع هزینه زیر بیان می‌شود:

$$J = \sum_{j=1}^K \sum_{x^{(i)} \in C_j} \|x^{(i)} - \mu_j\|^2$$

- انتخاب f و $\{\mu_1, \mu_2, \dots, \mu_K\}$ برای کمینه کردن تابع J .
- این مسئله NP-hard است. الگوریتم K-Means یک راه حل هیوریستیک است که تضمین نمی‌کند حتماً به راه حل بهینه برسد.

- ابتدا هر نمونه به نزدیکترین مرکز تخصیص داده می‌شود:

$$f(x) := \arg \min_j \|x - \mu_j\|^2$$

- تخصیص هر نمونه تا زمانی که مرکز نزدیکتری پیدا نشود، ثابت می‌ماند.
- هر بار که یک نمونه بازتخصیص می‌شود، مجموع فاصله بین نمونه‌ها و مرکز آن‌ها کاهش می‌یابد.
- تعداد تخصیص‌های ممکن نمونه به مرکز محدود است.
- الگوریتم زمانی خاتمه می‌یابد که هیچ نمونه‌ای مرکز تخصیص یافته خود را تغییر ندهد.
- در مرحله بهروزرسانی، با ثابت نگه داشتن $f(x)$ ، تابع J به صورت یک تابع درجه دوم از μ_j (مانند مجموع مربعات خطای مجموع فاصله) است و با مشتق‌گیری می‌توان آن را کمینه کرد:

$$\frac{\partial J}{\partial \mu_j} = 0 \implies \sum_{x^{(i)} \in C_j} -2(x^{(i)} - \mu_j) = 0$$

- این به این معناست که هر μ_j باید به عنوان میانگین خوش C_j بهروزرسانی شود:

$$\mu_j = \frac{\sum_{x^{(i)} \in C_j} x^{(i)}}{|C_j|}$$

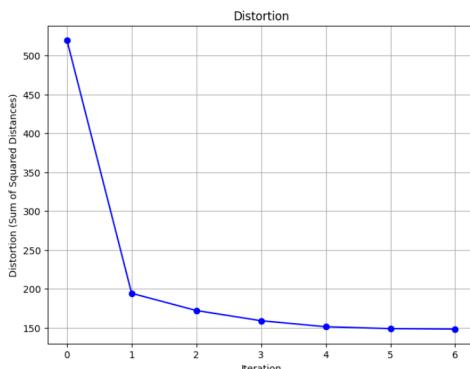
- برای هر خوش، میانگین نقاط آن خوش مقدار مجموع فاصله‌های مربعی را کمینه می‌کند.
- برای خوش C_j ، اگر μ'_j مرکز قبلی باشد، داریم:

$$\sum_{x^{(i)} \in C_j} \|x^{(i)} - \mu'_j\|^2 \geq \sum_{x^{(i)} \in C_j} \|x^{(i)} - \mu_j\|^2$$

یعنی انتخاب میانگین جدید هرگز مقدار خطای افزایش نمی‌دهد.

- در نتیجه، $J_{\text{new}} \leq J_{\text{old}}$ خواهد بود.

- از آنجایی که تابع هزینه J غیرمنفی است و تعداد تقسیم‌بندی‌های ممکن نیز محدود می‌باشد، دنباله مقادیر J نمی‌تواند تا بی‌نهایت کاهش یابد.
 - بنابراین الگوریتم K-Means در نهایت همگرا می‌شود.
 - ویژگی‌های همگرایی الگوریتم K-Means نخستین بار توسط مک‌کوئین (MacQueen) در سال ۱۹۶۷ بررسی شد.
- cost همان طوری که در تصویر مشاهده می‌شود در نهایت کمتر و کمتر می‌شود. و به یک نقطه صفر می‌رسد که بهترین جواب را پیدا خواهد کرد.



شکل ۹: نمودار تابع هزینه در k-means