

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
۹	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
۱۳	۶.۱.۲ رگرسیون لاسو	۶.۱.۲
۱۴	۷.۱.۲ رگرسیون ریج	۷.۱.۲
۱۴	۲.۲ یادگیری بدون ناظارت	۲.۲
۱۴	K-means	۱.۲.۲

۱ مقدمه

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، کدی می‌نویسد 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 DeepMind AlphaFold 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 متنهای تولید متن‌های شبیه انسان، پاسخ به سوالات پیچیده، ترجمه زبان‌ها، و حتی نوشتگری کرد 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 ویدئوهای استفاده می‌شوند. مثلاً، سیستم‌های نظارت شهری از این فناوری برای شناسایی رفتارهای مشکوک بهره می‌برند. - خودروهای خودران: شرکت‌هایی مثل Tesla و Waymo از یادگیری ماشین برای پردازش داده‌های حسگرها (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 AlphaGo یا DeepMind از 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 محظوظات 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 مؤسسه‌های 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 ژئومی: 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 شبکه‌های هنری: AI می‌تواند آثار هنری را تحلیل کرده و شبکه‌های 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

یادگیری ماشین با دونوع یادگیری نظارت شده (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 برچسب‌داد: 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 نظارت شده:
- 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 نظارت شده:
- 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)
- یش‌بینی: استفاده از مدل آموزش‌دیده برای پیش‌بینی برچسب داده‌های جدید. لگوریتم‌های رایج: - رگرسیون خطی-ear Regression - رگرسیون لجستیک (Logistic Regression) - ماشین بردار پشتیبان (SVM) - درخت تصمیم (Decision Tree) - شبکه‌های عصبی (Neural Networks) - جنگل تصادفی (Random Forest)

زايا: - دقت بالا در صورتی که داده‌های برچسب‌دار کافی و باکیفیت وجود داشته باشد. - کاربرد گسترده در مسائل واقعی مثل تشخیص پزشکی، پیش‌بینی مالی، و پردازش زبان طبیعی. چالش‌ها: - نیاز به داده‌های برچسب‌دار که ممکن است جمع‌آوری آن‌ها زمان بر

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 می‌خواهید 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 ناظارت شده

گرسیون یکی از تکنیک‌های اصلی در یادگیری ناظارت شده (Supervised Learning) است که برای پیش‌بینی مقادیر پیوسته (Continous) استفاده می‌شود. در یادگیری ناظارت 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. مفاهیم کلیدی رگرسیون داده‌های پیوسته: در رگرسیون، متغیر هدف (خروجی) مقداری پیوسته است، برخلاف مسائل طبقه‌بندی که خروجی گسسته (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 می‌زند. خطای تفاوت: بین مقدار واقعی (y) و مقدار پیش‌بینی شده (\hat{y}) به عنوان خطای محاسبه می‌شود. هدف کمینه کردن این خطای تفاوت 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 ناظارت شده، هدف یادگیری یک مدل است که رابطه‌ای بین ویژگی‌های ورودی (X) و خروجی (y) را با استفاده از داده‌های آموزشی (که شامل جفت‌های (X_i, y_i) 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 معمولاً 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 کنیم:

در رگرسیون خطی، پارامترها (مانند w_0 و w_1) تعیین‌کننده خط پیش‌بینی هستند. کمینه‌سازی تابع هزینه، مقادیر بهینه این پارامترها را پیدا می‌کند تا مدل بهترین تطابق را با داده‌ها داشته باشد. تعمیم‌پذیری مدل را بهبود دهیم:

با کمینه‌سازی 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 کمینه‌سازی تابع هزینه در رگرسیون خطی، پایه‌ای برای یادگیری الگوریتم‌های پیچیده‌تر مانند شبکه‌های عصبی است که از روش‌های مشابه (مثل گرادیان نزولی) برای بهینه‌سازی استفاده می‌کنند.

اثبات کمینه‌سازی تابع هزینه در رگرسیون خطی

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

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

کمینه‌سازی 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

$$\begin{aligned}\frac{\partial J}{\partial w_0} &= \frac{\partial}{\partial w_0} \left[\sum_{i=1}^n (y_i^{(i)} - (w_0 + w_1 x_i^{(i)}))^2 \right] \\ &- 2 \sum_{i=1}^n (y_i^{(i)} - w_0 - w_1 x_i^{(i)})^2 = 0 \\ n w_0 + w_1 \sum_{i=1}^n x_i &= \sum_{i=1}^n y_i \\ w_0 &= \frac{\sum_{i=1}^n y_i - w_1 \sum_{i=1}^n x_i}{n} = \bar{y} - w_1 \bar{x} \\ \text{که در آن } \bar{y} &= \frac{\sum y_i}{n} \text{ و } \bar{x} = \frac{\sum x_i}{n}\end{aligned}$$

مشتق نسبت به w_1

$$\begin{aligned}\frac{\partial J}{\partial w_1} &= \frac{\partial}{\partial w_1} \left[\sum_{i=1}^n (y_i^{(i)} - (w_0 + w_1 x_i^{(i)}))^2 \right] \\ &= \sum_{i=1}^n 2(y_i^{(i)} - w_0 - w_1 x_i^{(i)})(-x_i^{(i)}) = \sum_{i=1}^n x_i(y_i^{(i)} - w_0 - w_1 x_i^{(i)}) \\ w_0 &= \bar{y} - w_1 \bar{x} \text{ صفر کردن مشتق و جایگذاری}\end{aligned}$$

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

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

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

حل برای w_1

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

برای سادهسازی، از فرم معادل استفاده می‌کنیم:

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

برآوردهای بهینه بنابراین، برآوردهای بهینه برای پارامترها عبارتند از:

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

$$w_0 = \bar{y} - w_1 \bar{x} \quad (3)$$

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. کمینه‌سازیتابع 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:

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

$w = [w_0, w_1, \dots, w_n]^T$: بردار ضرایب.

y : بردار خروجی‌ها.

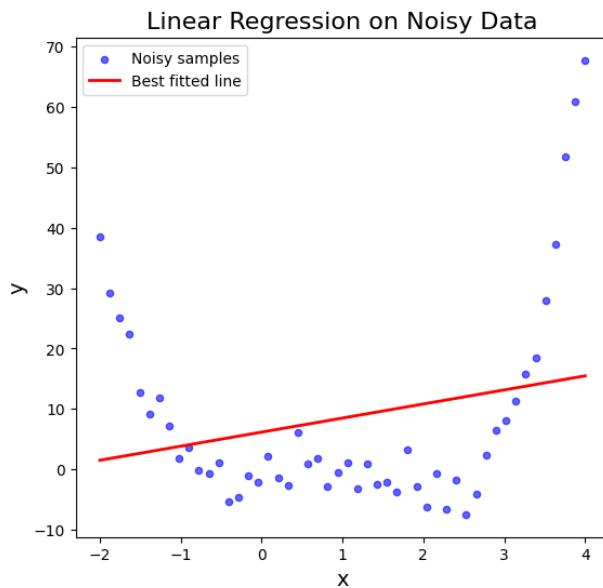
تابع هزینه تابع هزینه میانگین مربعات خطأ (MSE) به صورت زیر تعریف می‌شود:

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

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:



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

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

مشتق نسبت به w :

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

صفر کردن مشتق:

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

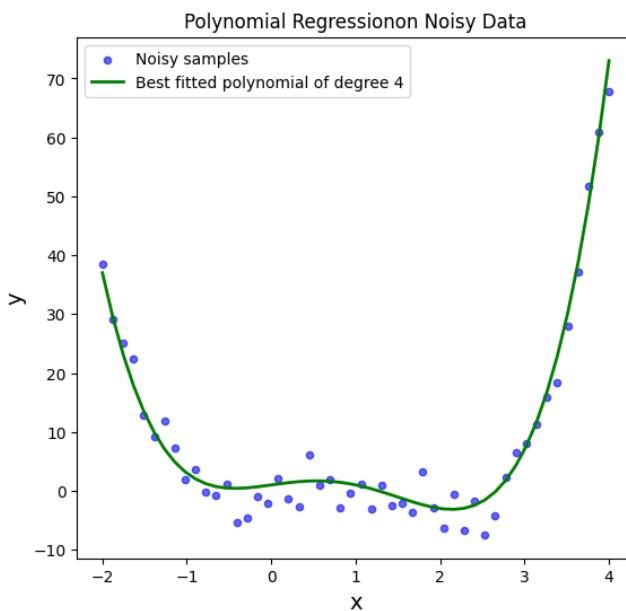
حل برای w :

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

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

تعمیم حالت خطی.
؟؟

گرسیون خطی ساده (Simple Linear Regression) برای مدل سازی رابطه بین یک متغیر مستقل (x) و یک متغیر وابسته (y) به کار می رود، 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 مناسب تری 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 می شود وقتی از رگرسیون خطی روی داده های چند گانه می شود یک نویز یا خطایی به وجود آید حتی برای آن یک تخمین اولیه را نداشته باشیم. برای رفع این مشکل می توانیم از رگرسیون های چند گانه با درجه هایی مختلفی استفاده کنیم و این نویز را از بین ببریم. به تصویر عدی نگاه کنید.



شکل ۲: استفاده نمودار غیر خطی.

همان طور که در تصویر بالا مشاهده می شود زمانی که درجه را زیاد می کنیم باعث می شود 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 می توانید از چند تبدیل های مختلفی استفاده کنید و بهترین برآرزنش را برای داده خود انجام دهید. در ادامه به برخی از از تبدیل ها اشاره خواهیم کرد.

انواع تبدیل ها

- تبدیل لگاریتمی: $y' = \log(y)$ یا $x'_i = \log(x_i)$. برای داده های با توزیع کج یا روابط نمایی.
- تبدیل چندجمله ای: $x'_i = x_i^2, x_i^3, \dots$. برای مدل سازی روابط غیرخطی.
- تبدیل ریشه مربع: $y' = \sqrt{y}$ یا $x'_i = \sqrt{x_i}$. برای کاهش کجی یا اثر مقادیر بزرگ.
- تبدیل معکوس: $y' = \frac{1}{y}$. برای روابط معکوس یا مدیریت داده های پرت.

از جمله تبدیل های هستند که می توانید برای برآش داده استفاده کنید از underfit و overfit شدن جلوگیری کنید. **گرادیان میخواهیم** یک محبت جدید را بگوییم که به گرایان کاهشی می گویند. و یک قسمت خیلی مهم می باشد و برای حل کردنی سییاری از محبت های ماشین لرنینگ نیز به کار میروند. در ادامه به آن خواهیم پرداخت. الگوریتم گرادیان به صورت خلاصه میتوان گفت که یک دارایی یک شبیب می باشد. که اگر یک یک توب را از آن رها کنیم میتوفت ته داره جایی که کاست فاکشن بهینه می شود برای این کار ما گرادیان هر نقطه را پیدا می کنیم که در نهایت به اون مینیمم مقدار بررسیم. گرادیان یعنی شبیب باید در راستایی عکس آن گرایان حرکت کنیم. و ما باید در راستایی عکس شبیب حرکت کنیم و آن مقدار را مینم کنیم. ما می توانیم نسبیت به پارامتر های خود مشتق بگیریم آن نقطه را مینیمم کنیم. به مثال زیر توجه کنید.

$$j(W_0, \dots, W_n) \rightarrow \nabla j_{(w_0, \dots, w_n)} = \begin{bmatrix} \frac{\partial J}{\partial w_0} \\ \vdots \\ \frac{\partial J}{\partial w_n} \end{bmatrix}$$

می توان گفت که این بردار به این صورت می گوید یک بار در راستایی D پارامتری که داری مشتق بگیر. که یک بردار دی بعدی به ما می دهد که وقتی این بردار را بر عکس کنیم یعنی در جهت عکس این گرادیان حرکت کنیم و نقطه مینیمم را پیدا خواهیم کرد.
مثال ۲:

$$j(w_0, w_2) = w_0^2 + 2w_1^2 \rightarrow \nabla j_{(w_0, w_1)} = \begin{bmatrix} \frac{\partial J}{\partial w_0} \\ \vdots \\ \frac{\partial J}{\partial w_n} \end{bmatrix} = \begin{bmatrix} 2w_0 \\ \vdots \\ 4w_1 \end{bmatrix}$$

پیدا کردن جواب بهینه در کاهشی رخی از وقت های وقتی کاست فاکشن خود را تعریف می کنیم می توان گفت که یک فرم تچه مانندی را به دنبال دارد و برخی از جواب های آن جواب های بهینه ای هستند که اگر به پارامتر های که نمودار تعریف شده نگاه کنیم مشاهده خواهد کرد و این کاست فاکشن بهینه ای هست که دارای نقطه بهینه می باشد. و ما می توانیم این تصور را داشته باشیم رخی از کاست فاکشن های دارایی نقطه بهینه هستند که خوب ندیده باشند. در تصویر مشاهده می شود که یک سری دره های هستند که به آن دره لوكال گفته می شود. اما در آن جایی بیشترین مقدار را به خود دارد می توان گفت که لوكال جهانی گفته می شود. و ته یک لوكال محلی هم نیز گرادیان در آن نقطه هم صفر می باشد. اگر به یک جام توجه کنید زمانی که یک تیله را از بالا به سمت پایین را می کنیم در یک مقدار ثابت قرار می گیرد که به ان نقطه می شود. یک نقطه بهینه برای ما که این شکل ها برای ما چالشی ایجاد نمی کند اما رنگنگ گریت آن را خیلی بد انتخاب نکرده باشیم این چالشی برای ما ندارد.

۳.۱.۲ گرادیان (gradeint)

گرادیان نزولی (Gradient Descent) یکی از الگوریتم های بهینه سازی پر کاربرد در یادگیری ماشین است که برای کمینه سازیتابع هزینه (Cost Function) در مدل هایی مانند رگرسیون خطی، رگرسیون لجستیک، و شبکه های عصبی استفاده می شود. این روش بهویژه در مواردی که داده ها حجم زیادی دارند یا محاسبات تحلیلی (مانند روش حداقل مربعات) هزینه بر هستند، بسیار مفید است. در ادامه، توضیح جامعی درباره گرادیان نزولی، نحوه کار، انواع آن، کاربردها، و مزایا و معایب آن ارائه می شود. گرادیان نزولی یک الگوریتم تکراری است که برای یافتن مقادیر بهینه پارامتر های مدل (مانند w_0, w_1, \dots, w_n) به گونه ای که تابع هزینه $J(w)$ کمینه شود، استفاده می شود. ایده اصلی این است که با حرکت در جهت مخالف گرادیان (مشتق تابع هزینه)، به سمت نقطه کمینه حرکت کنیم.

گرادیان: بردار مشتق های جزئی تابع هزینه نسبت به پارامتر ها، که جهت و شدت تغییرات تابع هزینه را نشان می دهد. نزولی: حرکت در جهت مخالف گرادیان برای کاهش مقدار تابع هزینه. شروع با مقادیر اولیه: مقادیر اولیه تصادفی یا صفر برای پارامتر ها (w) انتخاب می شود. محاسبه گرادیان: مشتق جزئی تابع هزینه نسبت به هر پارامتر محاسبه می شود:

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

به روزرسانی پارامتر ها: پارامتر ها در جهت مخالف گرادیان با گام مشخص (نرخ یادگیری، α) به روزرسانی می شوند:

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

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

انواع گرادیان

گرادیان نزولی بر اساس نحوه استفاده از داده ها به سه نوع اصلی تقسیم می شود:
۱. گرادیان نزولی دسته ای (Batch Gradient Descent): گرادیان با استفاده از کل داده های آموزشی محاسبه می شود. مزایا: همگرایی پایدار تر به سمت کمینه. معایب: برای داده های بزرگ، محاسبات سنگین و کند است.

۲. گرادیان نزولی تصادفی (Stochastic Gradient Descent - SGD): گرادیان برای هر نمونه داده به صورت جداگانه محاسبه و پارامتر ها به روزرسانی می شوند. مزایا: سریع تر برای داده های بزرگ، می تواند از کمینه های محلی فرار کند. معایب: نوسانات زیاد در تابع هزینه، ممکن است به کمینه دقیق نرسد.

۳. گرادیان نزولی مینی بچ (Mini-Batch Gradient Descent): داده ها به دسته های کوچک (mini-Batches) تقسیم می شوند و گرادیان برای هر دسته محاسبه می شود. مزایا: تعادل بین سرعت و پایداری، پر کاربرد در شبکه های عصبی. معایب: نیاز به تنظیم اندازه دسته.

کاربرد گرادیان نزولی در رگرسیون خطی در رگرسیون خطی، تابع هزینه معمولاً میانگین مربعات خطأ (MSE) است:

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

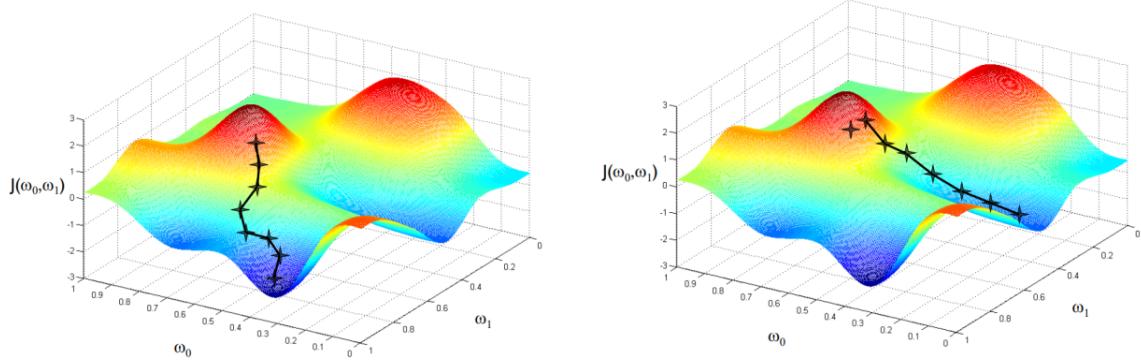
گرادیان نزولی برای یافتن w_n, w_1, \dots, w_0 استفاده می شود:
مشتق های جزئی:

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

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

به روزرسانی:

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



شکل ۳: گرادیان کاهشی

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

- محور x : پارامتر اول، مثلاً (w_0) .
- محور y : پارامتر دوم، مثلاً (w_1) .
- محور هر نمودار یک سطح ۲D است که محورهای آن به صورت زیر تعریف شده‌اند:

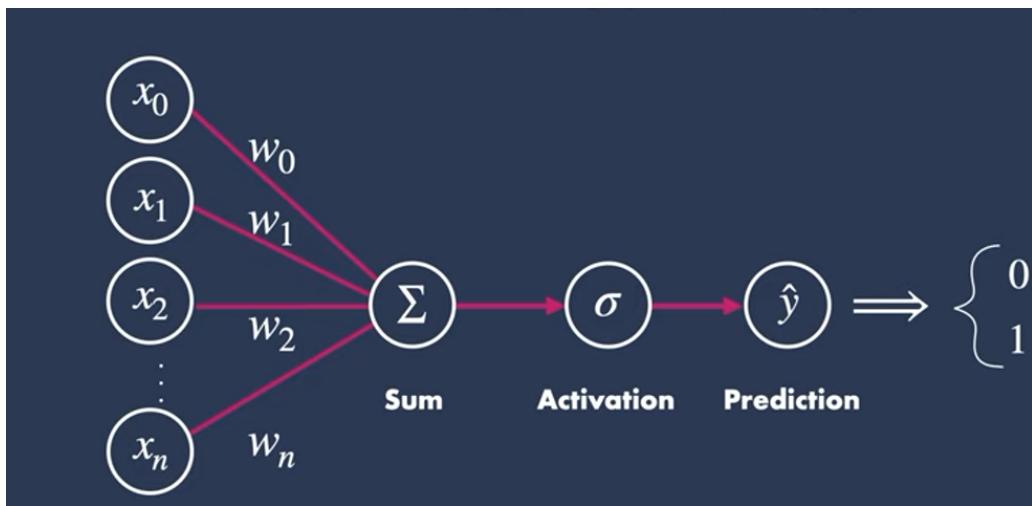
محور x : θ_0 (پارامتر اول، مثلاً w_0). محور y : θ_1 (پارامتر دوم، مثلاً w_1). محور z : $J(\theta_0, \theta_1)$ (مقدار تابع هزینه). نمودار سمت چپ: مسیر گردایان نزولی با نرخ یادگیری (α) بالاتر یا تعداد مراحل کمتر نشان داده شده است. مسیر دارای انحنای و نوسانات بیشتری است و به نظر می‌رسد هنوز به کمینه دقیق 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

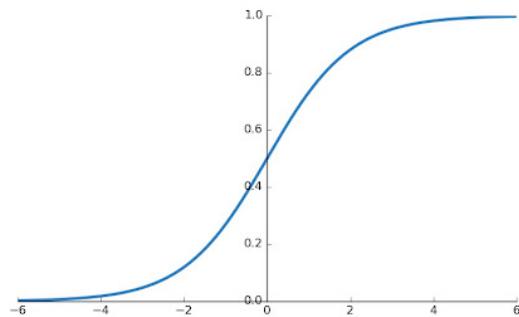
می‌توان تابع لجستیک را به این شکل تعریف کرد.

$$\hat{Y} = \sigma(X) = \sigma(x_0 w_0 + x_1 w_1 + \dots + x_n w_n)$$

تابع سیگومد به این شکل می‌باشد.



شکل ۴: تابع لجسيتك در يادگيري ماشين



شکل ۵: نمودار تابع لجسيتك

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

و همچنین در مورد مشتق پذيری

$$h \lim_{x \rightarrow -\infty} \sigma(x) = \frac{1}{1 + \infty} = 0$$

$$\lim_{x \rightarrow \infty} \sigma(x) = \frac{1}{1 + 0} = 1$$

$$\lim_{x \rightarrow 0} \sigma(x) = \frac{1}{1 + 1} = 0.5$$

از آن جايی که $x < -\infty$ پس می توان نوشت $\sigma(x) < 1$ می باشد. و يك نمودار به اين شكل می باشد

يکي دیگر از ویژگی های خوبی که اين تابع دارد مشتق پذير است.

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

در يادگيري ماشين ما نياز داريم که عددی که يا پيش بينی های که به ما باز گرداننده می شوند يك عدد بين صفر و يك باشد که برای

همین می توان از تابع لجسیتیک به این 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 ساده‌سازی، لگاریتم تابع درستنمایی را می‌گیریم:

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

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

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

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 بیشینه‌سازی تابع درستنمایی، از الگوریتم‌های عددی مانند گرادیان افزایشی (Gradient Ascent) یا گرادیان کاهشی (Gradient Descent) استفاده می‌شود. در عمل، ما معمولاً گرادیان کاهشی را برای کمینه‌سازی تابع هزینه معادل استفاده می‌کنیم:

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

مشتق منفی آن نیز همان است:

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

۵.۱.۲ نظمدهی (Regularization)

یک تکنیک مهم در ماشین لرنینگ است که باعث کاهش کمینه کردن و بهبود دقیقی شود، با جلوگیری از بیش‌بازش (Overfitting). حذف داده‌های پرداز و باعث کاهش نویز می‌شود و کمک می‌کند مدل ساده‌تر شود و روی داده‌های جدید به خوبی عمل می‌کند. این عمل فقط با افزون یک ترم یا جریمه صورت خواهد گرفت. انواع (Regularization)

مدلی رگرسیونی که از تکنیک نظمدهی L1 استفاده می‌کند، رگرسیون لاسو (Least Absolute Shrinkage and Selection Operator) نام دارد که مخفف (LASSO) است. در این روش، مقدار قدر مطلق ضرایب (ضرایب‌های مدل) به عنوان یک جریمه (penalty) به تابع زیان (Loss) اضافه می‌شود. این جریمه می‌تواند باعث شود برخی ضرایب کاملاً به صفر برسند، که در نتیجه ویژگی‌های کم‌اهمیت حذف شده و تنها ویژگی‌های مهم باقی می‌مانند.

۶.۱.۲ رگرسیون لاسو

یک روش رگرسیونی مبتنی بر تکنیک "کوچکسازی و انتخاب بر اساس قدر مطلق است که در تحلیل رگرسیون برای انتخاب متغیرها و اعمال نظمدهی Regularization استفاده می‌شود. این روش به حذف ویژگی‌های غیرمربوط داده کمک می‌کند و از بیش‌برازش Overfitting جلوگیری می‌کند. در نتیجه، ویژگی‌هایی که تأثیر ضعیفی دارند به خوبی مشخص می‌شوند، چرا که ضرایب متغیرهای کم‌اهمیت به سمت صفر میل می‌کنند. در حقیقت می‌توان گفت که رگرسیون لاسو یکی از روش‌هایی است که با افزودن یک جمله‌ی جریمه بهتابع هزینه مدل خطی، از بیش‌برازش جلوگیری می‌کند.تابع هزینه در این روش به صورت زیر تعریف می‌شود:

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

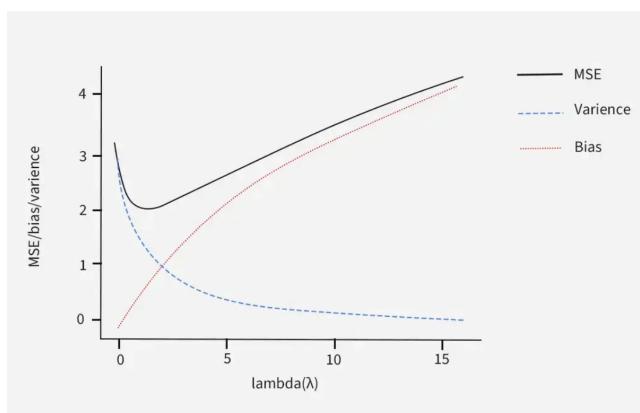
و همچنین در فرمول بالا y = مقدار مشاهده شده.

\hat{y} = مقدار پاسخ بیش‌بینی شده
و در قسمت جمله جریمه (penalty prem) λ این تابع مشتق پذیر نمی‌باشد.
 w = ضرایب

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

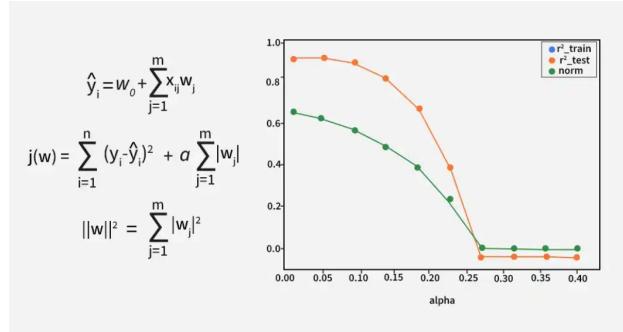
$$J(w) = \min RSS + \lambda \sum_{i=1}^m |w_i|$$

تنظیم مقدار بهینه لاندا در ادامه دو معیار مهم در ماشین لرینینگ را مورد بررسی قرار خواهیم داد.



شکل ۶: biase and varince in lasso

= وقتی یک مدل بسیار ساده باشد و داده‌های ما پیچیده باشند نقاط پرت زیاد می‌شوند در نهایت خط افزایش می‌یابد.
variance = در حقیقت وقتی مدل پیچیده باشد می‌تواند تنها روی داده‌های آموخته یا داده تست به خوبی فیت شود و در حالی در برگیرنده داده‌های آموخته نیست در نهایت می‌توان گفت که واریانس زیاد شده و اورفیت رخ داده است.
panlty = کاربردی که دارد باعث می‌شود ویژگی‌هایی که اهمیت کمی دارند حذف شوند. و این عمل باعث کاهش واریانس می‌شود و از اورفیت شدن جلوگیری می‌کند. افزایش مقدار λ باعث این می‌شود مدل بیش از حد ساده باشد و در نهایت باعث این می‌شود 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.



شكل ۷ cost function :

در نمودار مربوط به Residual Sum of Squares (RSS) یا L_2 بر ضرایب β تعریف می‌شود. بخش اول یعنی RSS اختلاف مربعی بین مقادیر واقعی و مقادیر پیش‌بینی شده را اندازه‌گیری می‌کند. بخش دوم، یعنی جریمه‌ی L_1 (یا 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 lambda) 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 است از رگرسیون خطی. و با اضافه کردن یک تابع جریمه به cost function کمک می‌کند یک به نقطه بهینه برسیم و هم چنین overfit کمتری رخ بدهد.

$$J(\theta) = MSE(\theta) + \alpha \sum_{i=1}^n \theta_i^2$$

نکته: در رگرسیو ریج به این مورد توجه کنید که ویژگی‌های که اهیمت 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 $MSE(\theta) = j(\theta)$ این قسمت برای این که متوجه پذیر است و گرادیان آن راحت حساب می‌شود از آن استفاده خواهیم کرد.

توان منظم دهنده (Regularization strength) با آلفا (α) می‌شود. که یکی از hyperparameter می‌باشد که که دو ویژگی دارد:

$$\begin{cases} 1 & \text{اگر} \\ 0 & \text{اگر} \end{cases}$$

۲.۲ یادگیری بدون نظارت

تعریف... یادگیری بدون نظارت (Unsupervised Learning) به تحلیل داده‌های بدون برچسب برای کشف الگوهای یا 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 خوشه به یکدیگر شبیه‌ترند تا به نقاط خوشه‌های 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)

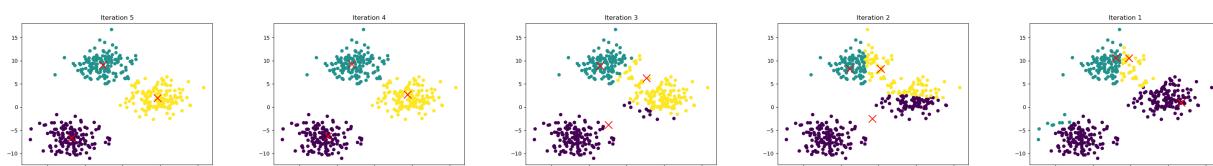
- یادگیری توزیع داده‌ها برای تولید نمونه‌های جدید و مشابه.

کاربردهای خوشه‌بندی

- تقسیم‌بندی مشتریان (Customer Segmentation): در بازاریابی برای گروه‌بندی مشتریان بر اساس رفتار.
- تقسیم‌بندی تصویر و تشخیص اشیا (Image Segmentation and Object Detection): در بینایی کامپیوتری.
- 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 الگوریتم خوشه‌بندی

- پرکاربردترین الگوریتم خوشه‌بندی.
- داده‌ها را به \mathbb{K} گروه مجزا بر اساس شاهت 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.



شکل ۸: K-means

- هدف: نقاط داده در یک خوشه باید به یکدیگر شبیه باشند.

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

- ابتدا هر نمونه به نزدیکترین مرکز تخصیص داده می شود:

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

- تخصیص هر نمونه تا زمانی که مرکز نزدیکتری پیدا نشود، ثابت می ماند.
- هر بار که یک نمونه باز تخصیص می شود، مجموع فاصله بین نمونه ها و مرکز آن ها کاهش می یابد.
- تعداد تخصیص های ممکن نمونه به مرکز محدود است.
- الگوریتم زمانی خاتمه می یابد که هیچ نمونه ای مرکز تخصیص یافته خود را تغییر ندهد.
- در مرحله بروزرسانی، با ثابت نگه داشتن $f(x)$ ، تابع J به صورت یک تابع درجه دوم از μ (مانند مجموع مربعات خط) است و با مشتق گیری می توان 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 بروزرسانی شود:

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

- برای هر خوش، میانگین نقاط آن خوش مقدار مجموع فاصله های مربعی را کمینه می کند.

- برای خوش 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 خطرا افزایش نمی دهد.

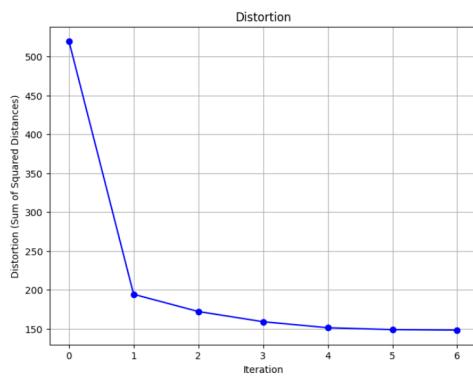
- در نتیجه، $J_{\text{new}} \leq J_{\text{old}}$ خواهد بود.

- از آنجایی که تابع هزینه J غیر منفی است و تعداد تقسیم بندی های ممکن نیز محدود می باشد، دنباله مقادیر J نمی تواند تا بی نهایت کاهش یابد.

- بنابراین الگوریتم K-Means در نهایت همگرا می شود.

- ویژگی های همگرایی الگوریتم K-Means نخستین بار توسط مک کوئین (MacQueen) در سال ۱۹۶۷ بررسی شد.

همان طوری که در تصویر مشاهده می شود در نهایت کمتر و کمتر می شود. و به یک نقطه صفر می رسد که بهترین جواب را پیدا خواهد کرد.



شکل ۹: نمودار تابع هزینه در k-means