কৃত্রিম বুদ্ধিমত্তা - Artificial intelligence　　মেশিন লার্নিং - Machine learning

মেশিন লার্নিং (২): লিনিয়ার রিগ্রেশন, মডেল রিপ্রেজেন্টেশন ও কস্ট (Cost) ফাংশন

Sharif Hasan • August 4, 2020 সর্বশেষ আপডেট May 4, 2021 4 1,710 পড়তে 7 মিনিট লাগতে পারে

এটা মেশিন লার্নিং সিরিজের দ্বিতীয় লেখা। আগের লেখটি এখান থেকে পড়তে পারেন। যদিও আগের লেখায় বলেছিলাম লিনিয়ার রিগ্রেশন নিয়ে লিখবো তারপর ভাবলাম মডেল রিপ্রেজেন্টেশনটাও একটি গুরুত্বপূর্ণ টপিক। তাই এটাকেও কভার করার চেষ্টা করেছি।

লিনিয়ার রিগ্রেশন কি?

লিনিয়ার রিগ্রেশন হলো পরিসংখ্যানের একটি পদ্ধতি, যার মাধ্যমে আমরা কিছু নির্দিস্ট অধীন চলক (Dependent variable) এবং কিছু স্বাধীন চলক (Independent variable) এর সম্পর্ক মডেল করি। যদি আমাদের স্বাধীন চলক একটি হয় তবে তাকে বলা হয় সিম্পল লিনিয়ার রিগ্রেশন এবং যদি একাধিক স্বাধীন চলক থাকে তবে তাকে বলা হয় মাল্টিপল লিনিয়ার রিগ্রেশন। একইভাবে যখন একের অধিক স্বাধীন চলক এবং একের অধিক অধীন চলক থাকবে তখন তাকে মালটিভারিয়াট লিনিয়ার রিগ্রেশন বলে। এখানে আমাদের স্বাধীন চলক গুলো হলো আমাদের ইনপুট জা আমরা আমদের এলগরিদম কে দিবো। অপরদিকে অধীন চলকগুলো হলো আমাদের এলগরিদম এর আউটপুট। মেশিন লার্নিং এ লিনিয়ার রিগ্রেশন ব্যাবহার করে অবিচ্ছিন্ন কোনও কিছু প্রেডিকশন করি। যেমন, বাড়ির দাম, মার্কেট প্রাইস প্রেডিকশন ইত্যাদি।

মেশিন লার্নিং: লিনিয়ার রিগ্রেশন মডেল উপস্থাপন করা

মেশিন লার্নিং এলগরিদম গুলোর প্রথম কাজ হলো মডেল তৈরি করা। লিনিয়ার রিগ্রেশন এর মডেল মূলত একটি ফাংশন $h(x)$। এই ফাংশনটিকে আমরা বলি হাইপোথিসিস যাকে কিছু ইনপুট দিলে সে সেই মডেল এর উপরে নির্ভর করে কিছু প্রেডিকশন করবে। আমরা আমাদের ইনপুট কে X দ্বারা প্রকাশ করবো এবং হাইপোথিসিস আউটপুটকে Y দ্বারা প্রকাশ করবো। এককথায় বলতে গেলে আমরা ইনপুট এবং অউটপুট এর মধ্যে প্যাটার্ন খুঁজে বের করবো।

বিষয়টা আরেকটু সহজ করতে পারি। ধরা যাক আমার একটা সিস্টেম বানাতে হবে। যেখানে আমি হলাম কোন হাউজিং কম্পানির একজন ইন্জিনিয়ার। আমাকে কম্পানি বিগত ৩/৪ বছরের ডাটা দিয়েছে, যেখানে আমাকে বলা হয়েছে বিভিন্ন বাড়ির বিভিন্ন বৈশিষ্টের ওপর ভিত্তি করে এর দাম কত। এখন আমাকে বলা হলো এমন একটা মেশিন

লার্নিং মডেল বানাতে যেখানে ক্লায়েন্ট কোন নির্দিষ্ট বৈশিষ্টের বাড়ির দাম চাইলে আমার মডেল তা প্রেডিকশন করতে পারে। এখানে আমার ডাটাসেটের প্রত্যেকটা বাড়ির যে বৈশিষ্ট তা হলো আমাদের একেকটি ইনপুট X, এবং এর বিপরীতে আমরা যা প্রেডিকশন করবো তা হলো আমাদের আউটপুট Y।

এখন কথা হলো হাইপোথিসিস ফাংশন টা কেমন? ফাংশনটির একটি উদাহরণ হিসেবে নিচের সমীকরনটি লক্ষ করি,

$h(X^{(i)}) = x_0 \times \theta_0 + x_1 \times \theta_1 + x_2 \times \theta_2 + \ldots\ldots + x_n \times \theta_n$ এখানে $x_0 = 1$

আমাদের হাইপোথিসিস ফাংশন লিনিয়ার বা নন লিনিয়ার দুইটাই হতে পারে। উপরের উদাহরণে ক্যাপিটাল $X^{(i)}$ হলো একটি রো ভেক্টর বা একটি Array ($X = [x_1, x_2, x_3, \ldots, x_n]$) যা আমাদের ডেটাসেটের i তম রো এর ডেটাকে উপস্থাপন করে। লক্ষণীয়, এখানে i কিন্তু পাওয়ার বা সূচক না। $\theta_k, k \epsilon N$ হলো কিছু ধ্রুবক এবং এগুলো হল আমাদের লার্নিং প্যারামিটার (একে অনেক জায়গায় **Weight** বলা হয়)। আমাদের লার্নিং এলগরিদম গুলো মূলত এই θ কে দরকারমত টিউন করে নেয় (এমন ভাবে মান প্রদান করে যাতে আমাদের হাইপোথিসিস এর কার্ভটি সবচেয়ে ভালোভাবে ডেটাসেটের মধ্যে ফিট হয়ে যায়) যাতে আমাদের আউটপুট যতটা সম্ভব নিখুঁত হয়। নিচের ছবিটা দেখি

ছবি সূত্র: https://www.ablebits.com/office-addins-blog/2018/08/01/linear-regression-analysis-excel/

উপরের ছবিতে লাল রেখাটি আমাদের হইপোথিসিস ফাংশন এর কার্ভ। নীল বিন্দুগুলো আমাদের ডাটাগুলোর প্লট। লাল রেখাটি এইমুহূর্তে আমাদের ডাটাগুলোর মধ্যে সবচেয়ে ভালোভাবে ফিট হয়েছে।

মেশিন লার্নিং মডেল – Coursera

উপরে যে ছবিটি দেয়া আছে তা লক্ষ করি। এখানে X আমাদের ইনপুট, যা ঘরে ক্ষেত্রফল কত তা প্রকাশ করে। এই মানটাকে আমাদের হাইপোথিসিস ফাংশন h এ দেয়া হয়েছে। এখানে h আগে থেকে আমাদের ট্রেইনিং সেট কে শিখে নিয়েছে (θ দরকার মতো সেট করে নিয়েছে) নির্দিষ্ট লার্নিং এলগরিদম দ্বারা। h ফাংশনটি দরকার মতো ক্যালকুলেশন করে আমাদের কে নির্দিষ্ট আউপুট Y প্রদান করবে।

আমরা প্রথমে এক চলকের লিনিয়ার রিগ্রেশন দেখবো। পরে একাধিক চলকের দিকে দেখা যাবে।

উদাহরণসরূপঃ সিম্পল লিনিয়ার রিগ্রেশনের (একটি ইনপুট x_1 এবং একটি টার্গেট আউটপুট y) জন্য আমাদের হাইপোথিসিস ফাংশন হবে,

$$y = h(x) = \theta_0 + \theta_1 \times x_1$$

θ_0, θ_1 হলো আমাদের লার্নিং প্যারামিটার , y হলো আমাদের টার্গেট আউটপুট এবং x_1 হলো আমাদের ইনপুট। বহুমাত্রিক সমস্যার ক্ষেত্রে আমাদের আরও ইনপুট থাকবে। এক্ষেত্রে আমাদের হাইপথেসিস আর একটি সরলরেখা থাকবে না, এটি একটি সমতল (Plane) বা অধি সমতল (Hyper plane) হয়ে যেতো।

> *উপরে যেই ফাংশনটি লেখা হয়েছে আপনি ইচ্ছা করলে একে একটি দ্বিমাত্রিক (x_1, y) গ্রাফ এ উপস্থাপন করতে পারবেন, যদি আমাদের ইনপুট দুইটি হতো তাহলে ফাংশনটিকে একটি ত্রিমাত্রিক (x_1, x_2, y) গ্রাফ এ উপস্থাপন করা যেতো। যদি আমাদের ইনপুট সংখ্যা ৪ হতো তবে কি হতো? চতুর্মাত্রিক গ্রাফে উপস্থাপন করতাম? করলে কি আসলে বুঝতে পারতাম?*

এখন পরবর্তি কথা হলো θ_0 টা আসলে কি? একে বলা হয় বায়াস (Bias) প্যারামিটার। এর ব্যাপারে পরের পর্বে কথা বলবো।

আসেন আমরা আমাদের হাইপথিসিসকে প্লট করার চেষ্টা করি। এজন্য আপাতত $\theta_0 = 1.5$ এবং $\theta_1 = 0$ ধরে নিই। এখন আমাদের হাইপথিসিস এর প্লট হবে নিচের মতো।

$h(x) = 1.5 + 0 \times x_1$ (ক্রেডিট: Midium)

যেহেতু আমাদের $\theta_1 = 0$ সেহেতু আমাদের হাইপথিসিস ফাংশনের কোনও ঢাল নেই এবং আমাদের রেখাটি x অক্ষের সমান্তরাল রেখা হয়েছে। আবার ধরা যাক $\theta_0 = 1, \theta_1 = 0.5$। তাহলে আমাদের হাইপথিসিসের প্লট নিচের মতো হবে,

$h(x) = 1 + 0.5 \times x$ (ক্রেডিট: Midium)

তো এখন আমাদের উদ্দেশ্য কি?

ধরা যাক এবার আমাদের ডাটাসেট এ কিছু ডাটা (x,y) দেয়া আছে নিচের মত করে।

$$\boldsymbol{x = 1, 1, 2, 3, 4, 3, 4, 6, 4}$$
$$\boldsymbol{y = 2, 1, 0.5, 1, 3, 3, 2, 5, 4}$$

এখানে প্রতিটা ফিচার $\boldsymbol{x}$ এর বিপরীতে আমাদের টার্গেট আউটপুট $\boldsymbol{y}$ দেয়া আছে। এই ডাটাগুলোকে যদি $\boldsymbol{x}$ এর বিপরীতে $\boldsymbol{y}$ কে প্লটকরা হয় তবে নিচের মতো বিন্দুগুলো দেখা যাবে,

ক্রেডিট: Midium

আমরা লিনিয়ার রিগ্রেশন ব্যাবহার করে এই ডাটাগুলোর মধ্যে আমাদের হাইপথিসিস $\boldsymbol{h(x)}$ কে আঁকার চেষ্টা করবো, যাতে আমাদের প্রেডিকশন সর্বাপেক্ষা নিখুঁত হয়।

গ্র্যাডিয়েন্ট ডিসেন্ট ও কস্ট ফাংশন

উপরে আমরা প্যারামিটার $\boldsymbol{\theta}$ এর কথা বলেছিলাম। $\boldsymbol{\theta}$ কেই আমরা দরকার মত মান প্রদান করব যাতে আমাদের প্রেডিকশন যতটা সম্ভব নিখুঁত হয়।

এখন আমাদের একটা ফাংশন দরকার যা আমাদের এই থেটার নির্দিষ্ট মান এর জন্য আমাদের হাইপোথিসিস এর এরর (Error) পরিমাপ করবে। এই ফাংশনটাকেই আমরা

বলি কস্ট ফাংশন।

সাধারণত রিগ্রেশন প্রবলেমগুলোতে কস্ট ফাংশন হিসেবে **Mean squared error** (MSE) কস্ট ফাংশন জনপ্রিয়। যা আমাদের আউটপুট এবং টার্গেট আউটপুট এর মধ্যে কতোটুকু এরর আছে তা পরিমাপ করে। এই এররকে বিবেচনায় নিয়ে আমরা আমাদের θ কে optimal পজিশন এ সেট করতে পারি যাতে আমাদের কস্ট বা এরর সর্বনিম্ন হয়।

আমাদের কস্ট ফাংশন,

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

আমরা যদি আমাদের কস্ট ফাংশনের $\hat{y}$ এর জায়গায় আমাদের হাইপথিসিস $h(x)$ বসাই এবং আমাদের হিসেবের সুবিধার জন্য কস্ট ফাংশনকে একটু সহজ করি তবে, আমাদের কস্ট ফাংশন হয়,

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h(X) - y^i)^2$$

এখানে,

- m হলো আমাদের ডাটাসেট এ কি পরিমাণ রো বা স্যাম্পল ডাটা আছে তার সংখ্যা।
- $J(\theta)$ আমাদের কস্ট ফাংশন, আমাদের টার্গেট হলো θ এর নির্দিষ্ট মানের জন্য $J(\theta)$ কে সর্বনিম্ন করা।
- y_i হলো আমাদের i তম স্যাম্পল এর টার্গেট আউটপুট। x_i হলো আমাদের i তম স্যাম্পল এর ইনপুট।
- $\hat{y}$ হলো আমাদের হাইপথিসিস যা প্রেডিকশন করেছে তার মান।

কস্ট ফাংশন বুঝার সুবিধার জন্য আমরা কিছু ডাটা নেই। X={1,2,3}, Y={1,2,3}। এখন এই ডাটা পয়েন্ট আঁকারে আমাদের গ্রাফ এ প্লট করি।

এখনকার জন্য আমরা আমাদের প্যারামিটার θ_1 এর জন্য কিছু মান নিবো ম্যানুয়ালি। আপাতত আমরা কাজ সহজ করার জন্য $\theta_0 = 0$ ধরে নিবো।

ধরি $\theta_1 = 1$, তাহলে আমাদের হাইপথিসিস ফাংশনটি যা দাঁড়াচ্ছে তা হলো, $h(x_i) = 0 + 1 \times x_i$, সুতরাং $J(1)$ হলে আমরা নিচের চিত্রের মতো লাল একটি সরলরেখা পাবো যা আমদের তিনটি বিন্দুকে চমৎকার ভাবে ছেদ করে গেছে। এখানে আমাদের কস্ট ফাংশনের মান 0। $\theta_1 = 0.5$, হলে আমরা হলুদ রেখাটির মতো একটি রেখা পাবো।

এখন,আমাদের হলুদ রেখাটির জন্য আমাদের এররের মান শুন্য হবে না। কারণ, আমরা পেয়েছি,

$$J(1) = \frac{1}{2 \times 3}((h(1)-1)^2 + (h(2)-2)^2 + (h(3)-3)^2) = 0, [h($$

এবং $J(0.5)$ এর জন্য,

$$J(0.5) = \frac{1}{2 \times 3}((h(1)-1)^2 + (h(2)-2)^2 + (h(3)-3)^2) = 0.58$$

এখন $J(1), J(0.5)$ কে আমাদের গ্রাফ এ প্লট করি, (ক্রস দিয়ে চিহ্নিত করা)।

Source: Machine learning – coursera

একই ভাবে $J(0), J(-0.5)$ এর জন্য আমাদের কস্ট হবে,

$$J(0) = \frac{1}{2 \times 3}((h(1) - 1)^2 + (h(2) - 2)^2 + (h(3) - 3)^2) = 2.3, h$$

$$J(-0.5) = \frac{1}{2 \times 3}((h(1) - 1)^2 + (h(2) - 2)^2 + (h(3) - 3)^2) = 0.$$

এখন $J(0), J(-.5)$ কেউ আমাদের গ্রাফ এ প্লট করে নেই।

Source: Machine learning – coursera

এখন আমরা আমাদের বিন্দুগুলিকে সুন্দর করে যুক্ত করে দিবো।

উপরের গ্রাফ থেকে আমরা দেখতে পাচ্ছি, আমাদের $\theta_0 = 0, \theta_1 = 1$ হলেই আমাদের কস্ট ফাংশনের মান সর্বনিম্ন হয়। সুতরাং $\theta_0 = 0, \theta_1 = 1$ হলো আমাদের প্যারামিটার এর জন্য অপটিমাম সমাধান।

নীল বক্ররেখাটা ঐটা আমাদের কস্ট ফাংশনের গ্রাফ। এখন কথা হচ্ছে যখন আমাদের আরও বড় সমস্যা দেয়া থাকবে তখন তো আমরা আর হাতে হাতে এই সমস্যা সমাধান করবো না। তখন কি উপায়? উত্তর হচ্ছে উপায় আছে, এবং তা হচ্ছে গ্র্যাডিয়েন্ট ডিসেন্ট এলগরিদম।

মেশিন লার্নিং: গ্র্যাডিয়েন্ট ডিসেন্ট

উপরে আমরা আমাদের প্যারামিটার এর জন্য বিভিন্ন মান ধরে নিয়ে আমাদের কস্ট ফাংশনকে মিনিমাইজ করেছি। তো আমরা তো আর আমাদের হাতে করে প্যারামিটারের মান বের করবো না। এর জন্য অবশ্যই ভালো উপায় থাকা উচিত। উত্তর হলো হ্যাঁ আছে, গ্র্যাডিয়েন্ট ডিসেন্ট এলগরিদম, মেশিন লার্নিং এর একটি গুরুত্বপূর্ণ লার্নিং এলগরিদম।

গ্র্যাডিয়েন্ট ডিসেন্ট তাই করে যা আমরা উপড়ে হাতে করে করলাম। একটু একটু করে আমাদের কস্ট ফাংশনের মান মিনিমাম এর দিকে নিয়ে যায় প্যারামিটারের মান একটু একটু করে পরিবর্তন করার মাধ্যমে। আমরা আমাদের প্যারামিটারের মান প্রথমে যেকোনো কিছু নিয়ে নিবো। তারপর আমরা আমাদের প্যারামিটার কে গ্র্যাডিয়েন্ট ডিসেন্ট এর মাধ্যমে আপডেট করবো যাতে আমাদের কস্ট ফাংশনের মান আস্তে আস্তে কমে যায়। আমাদের পারামিটের আপডেট করার মুল সূত্রটি এমন,

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

এখানে আমাদের α হলো লার্নিং রেট। এই প্যারামিটার নির্দেশ করে কতদ্রুত আমাদের এলগরিদম মিনিমাম এর দিকে ধাবিত হবে। অন্তরীকরণ অংশটা হলো θ_j এর সাপাক্ষে আমাদের আংশিক অন্তরীকরণ। যেখানে আমাদের কস্ট ফাংশন হলো
$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h(x) - \hat{y})^2$ এবং $h(x^{(i)}) = \theta_0 + \theta_1 \times x_1$

নিচের ছবিটা দেখি,

এখানের **ab** রেখাটি আমাদের কস্ট ফাংশন এর একটি স্পর্শক। কস্ট ফাংশন এবং ab রেখা যেই বিন্দুতে স্পর্শ করেছে θ_1 এর সাপেক্ষে সেই রেখার ঢাল,

$$\frac{\partial}{\partial \theta_1} J(\theta) = 1 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

এবং আমাদের কস্ট ফাংশনের মিনিমাম হলো $\theta_1 = 22$। সুতরাং আমাদের প্যারামিটার এর মান কমাতে হবে। চলুন দেখি কিভাবে গ্র্যাডিয়েন্ট ডিসেন্ট আমাদের প্যারামিটারের মান কমাতে পারে।

আমরা জানি, $\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$

সমীকরণ ১ থেকে পাই, $\frac{\partial}{\partial \theta_1} J(\theta) = 1$

সুতরাং আমাদের প্যারামিটার $\theta_1 = \theta_1 - 0.1 \times 1 = 32.9$ এখানে $\alpha = 0.1$ ধরে।

অতএব দেখা গেলো গ্র্যাডিয়েন্ট ডিসেন্ট কিভাবে আমাদের প্যারামিটারের মান একটু কমিয়ে দিলো। যদি আরও প্যারামিটার থাকে তবে আমরা একই পদ্ধতি অনুসরণ করে আমাদের কস্ট ফাংশনের মান মিনিমাম করতে পারবো।

আজকে আর লেখছি না। আগামী লেখায় আমরা কস্ট ফাংশন এবং গ্র্যাডিয়েন্ট ডিসেন্ট নিয়ে আরও আলোচনা করবো তখন আমাদের অন্তরীকরণ অংশটুকু সমাধান করা হবে।

আপাতত বিদায়। আর মেশিন লার্নিং,মডেল রিপ্রেজেন্টেশন, কস্ট ফাংশন নিয়ে যার যা কিছু কনফিউশন, কমেন্ট করতে ভুলবেন না।

লেখাটি কেমন লেগেছে আপনার?

রেটিং দিতে হার্টের উপর ক্লিক করুন।

গড় রেটিং 4.3 / 5. মোট ভোট: 12

#AI #Machine learning #মেশিন লার্নিং

4 টি মন্তব্য

guEsS

September 4, 2020 at 2:42 PM

Please bro, continue

Reply

guEsS

September 4, 2020 at 2:44 PM

Demanding next part and also implementation..

Reply

Pingback: কস্ট ফাংশনের অন্তরীকরণ এবং গ্র্যাডিয়েন্ট ডিসেন্ট - শরিফ হাসানের ব্লগ

Pingback: Face recognition: ছবি থেকে মুখ শনাক্তকরণ পদ্ধতি - শরিফ হাসান