

دانشکده ریاضی و علوم کامپیوتر

تمرین اول درس یادگیری ماشین

عنوان: رگرسیون خطی

موعد تحویل:

# Linear Regression .1 (رگرسیون خطی)

در این تمرین شما رگرسیون خطی چند متغیره[[1]](#footnote-1) را پیاده سازی خواهید کرد. این تمرین طولانی به نظر می رسد، اما در عمل شما تنها حدود 10 خط کد را پیاده سازی می کنید. این تمرین برای آشنایی شما با مکانیزم کتابخانه scikit-learn و پیاده سازی الگوریتم های یادگیری ماشین در پایتون طراحی شده است.

## Files and API (فایل‎ها و توابع)

در این تمرین یک سری فایل و توابع پایتون در اختیار شما قرار دارد:

:test\_linreg\_univariate.py اسکریپتی برای تست کردن رگرسیون خطی تک متغیره[[2]](#footnote-2)

* plotData1D: رسم نمودار نقطه‎ای(پراکندگی)[[3]](#footnote-3) دیتای یک بعدی
* :plotRegLine1d بوسیله تابع plotData1D نمودار نقطه‎ای دیتا را رسم می‎کند و بوسیله پارامترهای یادگرفته شده توسط الگوریتم، خط رگرشن را در همان نمودار رسم می‎کند.
* :visualizeObjective رسم نموار surface و contour تابع هزینه[[4]](#footnote-4) (شما این تابع را تغییر نمی‎دهید)

:test\_linreg\_multivariate.py اسکریپتی برای تست کردن رگرسیون خطی متغیره

:linreg.py اسکریپی شامل کدهای رگرسیون خطی

:LinearRegression کلاس رگرسیون خطی چند متغیره

* \_\_init\_\_: کانستراکتور کلاس
* :fit تابعی برای یادگیری[[5]](#footnote-5) مدل رگرسیون خطی چند متغیره
* predict: تابعی برای پیش بینی ورودی جدید توسط مدل یاد گرفته شده
* computeCost: محاسبه مقدار تابع هزینه
* gradientDescent: بهینه سازی پارامترهای مدل توسط الگوریتم گرادیان کاهشی[[6]](#footnote-6)

## Data Sets (مجموعه داداه‎ها واقع در فولدر (data

* univariateData.dat: مجموعه داده‎ها برای مسئله رگرسیون تک متغیره
* multivariateData.dat: مجموعه داده‎ها برای مسئله رگرسیون چند متغیره

## Visualizing the Data (بصرسازی داده‎ها)

بصرسازی داده‎ها بینش ارزشمندی از مسئله به ما ارائه می دهد، اما اغلب به عنوان بخشی از فرآیند یادگیری ماشین نادیده گرفته می شود. ما با رسم مجموعه داده‎های تک متغیره با استفاده از نمودار پراکندگی 2 بعدی شروع خواهیم کرد. با این حال ما معمولاً با مجموعه داده‎های چند بعدی مواجه هستیم. هنگامی که از دو بعد فراتر می رویم، تجسم بسیار دشوارتر می شود. در چنین مواردی یا باید هر بعد را به طور جداگانه رسم کنیم، یا از تکنیک های کاهش ابعاد (مانند PCA) برای کاهش تعداد ویژگی‎ها[[7]](#footnote-7) استفاده کنیم. بعداً در کلاس در مورد چنین تکنیک هایی صحبت خواهیم کرد.

با اجرای دستورات زیر در مفسر پایتون از داخل دایرکتوری hw1 می توانید داده‎های تک متغیره را در متغیرهای X و y بصورت ماتریس بارگذاری کنید:

import numpy as np

filePath = “data/univariateData.dat”

file = open(filePath, ‘r’)

allData = np.loadtxt(file, delimiter=’,’)

X = np.matrix(allData[:,:-1])

y = np.matrix((allData[:,-1])).T

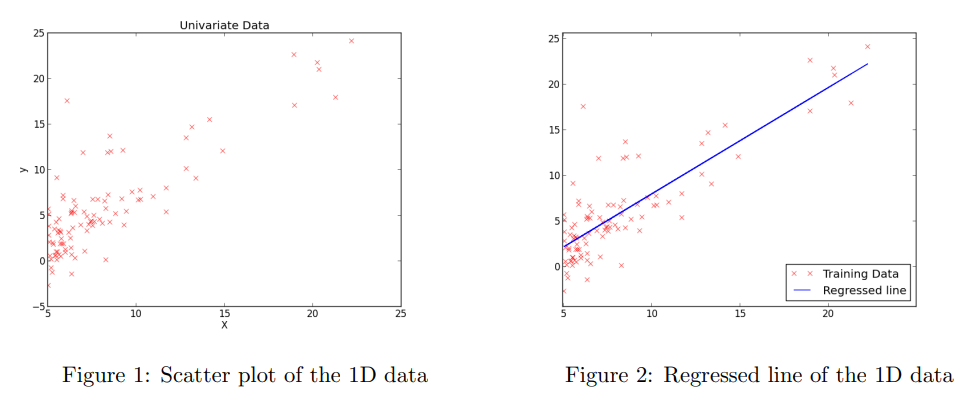
# get the number of instances (n) and number of features (d)

n, d = X.shape

سپس بوسیله‎ی تابع plotData1D نمودار پراکندگی را رسم کنید، خروجی شما باید مانند Figure 1 باشد:

from test\_linreg\_univariate import plotData1D

plotData1D(X, y)



## Implementation (پیاده سازی)

با تکمیل کلاس LinearRegression، رگرسیون خطی چند متغیره را از طریق گرادیان کاهشی اجرا کنید. اسم توابع (APIها) را ابدا تغییر ندهید. قسمت‎هایی از کد که باید تغییر دهید با کامنت‎ "TODO" مشخص شده اند.

الگوریتم رگرسیون خطی بعد از یادگیری پارامترهای مدل به کمک دیتا، آن‎ها را بر روی بردار[[8]](#footnote-8) θ ذخیره می‎کند[[9]](#footnote-9). در این تمرین از گرادیان کاهشی برای یافتن جواب بهینه استفاده می کنیم. دقت کنید که تابع هزینه رگرسیون خطی L2، محدب است، بنابراین الگوریتم گرادیان کاهشی مینیمم مطلق را پیدا می کند.

تابع در رگرسیون خطی تک متغیره (ماتریس x تنها یک ستون دارد( به این شکل است، که همان عرض از مبدا[[10]](#footnote-10) است. برای هندل کردن عرض از مبدا در قالب معادله‎ی (3) می توانیم یک ویژگی جدید به تمام سطرهای داده با مقدار 1 اضافه کنیم. در واقع را به عنوان ضریب در نظر می‎گیریم. برای اضافه کردن عرض از مبدا به کل مجموعه داده‎ها می توانیم ستونی از یک ها را به ماتریس X اضافه کنیم:

X = np.c\_[np.ones((n, 1)), X]

*الگوریتم گرادیان کاهشی برای یافتن مقدار مینیمم تابع فضای* θهای ممکن را جست و جو می‎کند. حلقه for اولیه گرادیان کاهشی برای شما پیاده سازی شده است. شما فقط باید معادله را به روز رسانی کنید. در هر مرحله[[11]](#footnote-11) از گرادیان کاهشی باید به کمک معادله زیر **بصورت همزمان** همه پارامترها را به روز رسانی کنید.

در هر مرحله از گرادیان کاهشی با به روز رسانی θ، به مقدار مینیمم تابع نزدیک‎تر می‎شویم. متغیر نرخ یادگیری[[12]](#footnote-12) است که آن را بسیار کوچک در نظر می‎گیریم (مثلا ). همیشه مقدار اولیه θ را یک مقدار کوچک تصادفی در حوالی صفر انتخاب می‎کنیم (می توان از توزیع نرمال با واریانسی کوچک و میانگین صفر استفاده کرد).

در نهایت تابع هزینه‎ (معادله‎ی 2) را در قالب تابع computeCost در کلاس LinearRegression پیاده سازی می‎کنیم.

**مشکلات متداول:**

* در هر مرحله از گرادیان کاهشی پارامترهای θ را بصورت همزمان آپدیت کنید. یعنی درهر مرحله بعد از به روز رسانی یکی از اعضای بردار θ، بردار θ را بصورت جزیی به روز رسانی نکنید و مقدار را محاسبه کنید. در پایان حلقه بردار θ را کاملا به روز رسانی کنید.
* به یاد داشته باشید در هر مرحله از گرادیان کاهشی شما تنها فضای θهای ممکن را جست و جو می‎کنید و نباید X و y را تغییر دهید.

**تست کردن کد نهایی:**

یک راه ساده برای تست کردن کد چاپ کردن مقدار در هر مرحله است. اگر مقدار آن با گذر زمان در هر مرحله بصورت یکنواخت در حال کاهش است یعنی کد شما درست کار می‎کند.

بعد از اتمام پیاده‎سازی کد، مدل خود را بر روی داده‎های univariateData آموزش دهید و سپس تابع plotRegLine1D در فایل test\_linreg\_univariate.py را اجرا کنید.

from test\_linreg\_univariate import plotRegLine1D

from linreg import LinearRegression

X = np.c\_[np.ones((n,1)), X] # if you didn’t this step already

lr\_model = LinearRegression(alpha = 0.01, n\_iter = 1500)

lr\_model.fit(X, y)

plotRegLine1D(lr\_model, X, y)

خروجی باید یک نمودار مانند Figure 2 باشد.

## Understanding Gradient Descent

در این قسمت نیازی به پیاده سازی کد ندارید. برای درک بهتر کد پیاده سازی شده در مرحله قبل، تابع هزینه و مسیر رسیدن به مینیمم آن که توسط الگوریتم گرادیان کاهشی انتخاب شده را رسم می‎کنیم.

برای مجموعه داده‌های تک متغیره می‌توانیم تغییر تابع هدف را در فضای و به‌عنوان یک نمودار سطحی[[13]](#footnote-13) و یک نمودار کانتور[[14]](#footnote-14) ترسیم کنیم تا شکل محدب و نزولی آن را نشان دهیم. خط آبی در شکل 3 مسیر طی شده توسط نزول گرادیان را نشان می دهد و نقاط سرخابی نقاط را در هر تکرار نشان می دهد.

**موعد تحویل:**

**معیار ارزیابی شما:**

**گزارش کار:**

1. نکته مهم در گزارش نویسی روشن بودن پاسخ ها می‎باشد، اگر فرضی برای حل سوال استفاده میکنید حتما آن را ذکر کنید، اگر جواب نهایی عددی است به صورت واضح آن را بیان کنید.
2. هرگونه شباهت در گزارش به منزله تقلب میباشد و کل نمره تمرین صفر می‎باشد.

1. multivariate linear regression [↑](#footnote-ref-1)
2. univariate linear regression [↑](#footnote-ref-2)
3. scatter plot [↑](#footnote-ref-3)
4. cost function (objective function) [↑](#footnote-ref-4)
5. train [↑](#footnote-ref-5)
6. gradient descent [↑](#footnote-ref-6)
7. features [↑](#footnote-ref-7)
8. vector [↑](#footnote-ref-8)
9. fit [↑](#footnote-ref-9)
10. bias term [↑](#footnote-ref-10)
11. iteration [↑](#footnote-ref-11)
12. learning rate [↑](#footnote-ref-12)
13. Surface plot [↑](#footnote-ref-13)
14. Contour plot [↑](#footnote-ref-14)