

هنتكلم انهارده عن ال SGD .. وال SGD هي نقطه حلوه هنتكلم منها علي الجورنم مهتمين بيه دأ .. معظم كلامنا لحد دلوقت هو انك تفكر في الالجورنم بتاعت الماشين ليرننج عشان يعني هم بيوضحولك شوية high-level ideas .. هنا الدكتور عاوزنا نهتم بشكل أساسي بالالجورنم ده عشان هو fundamentally important لمعظم ال recent progress بتاع الماشين ليرننج و خصوصاً في الديب ليرننج .. وهو كمان بيديلك key-idea و الفكره ديه بيعاد استخداماها across a whole family of optimization algorithms .. فالالجورنم ده مهم جداً نيقا عارفينو هو وشوية الجورنمز تنبين .. احنا انهارده هنروح ناحية ال Binary logistic regression ... الدكتور بيتكلم علي الامتحان ... اتمني اننا نلاقيه اونلاين ونحلو فعلاً يعني عشان نشوف هل احنا جامدين جداً ولا احنا وحشين جداً ولا احنا فين من الدنيا اللي احنا فيها ديه ... الدكتور بيقول الاتحان شبه ال HWs و في برضو trial exam ع النت الدكتور بيقول ممكن تجيب معاك ورقه صغيره كذا "برشام" .. تملأها كلها زي منت عاوز بس متبقاش ورقه كبيره ... المهم ازاى نستعد للميدترم إكزام .. هيرفعو شوية امتحانات وبص علي الامتحانات .. الدكتور برضو بيدي نصايح للامتحانات .. حل الأسئلة السهلة الاول .. الأسهله .. هه ... وديه المواضيع اللي هتيجي في الامتحان ..

Topics for Midterm 1

- Foundations
 - Probability, Linear Algebra, Geometry, Calculus
 - Optimization
- Important Concepts
 - Overfitting
 - Experimental Design
- Classification
 - Decision Tree
 - KNN
 - Perceptron
- Regression
 - Linear Regression
 - KNN Regression

كان في مثال في المحاضره

Sample Questions

5.2 Constructing decision trees

Consider the problem of predicting whether the university will be closed on a particular day. We will assume that the factors which decide this are whether there is a snowstorm, whether it is a weekend or an official holiday. Suppose we have the training examples described in the Table 5.2.

Snowstorm	Holiday	Weekend	Closed
T	T	F	F
T	T	F	T
F	T	F	F
T	T	F	F
F	F	F	F
F	F	F	T
T	F	F	T
F	F	F	T

Table 1: Training examples for decision tree

- [2 points] What would be the effect of the Weekend attribute on the decision tree if it were made the root? Explain in terms of information gain.
- [8 points] If we cannot make Weekend the root node, which attribute should be made the root node of the decision tree? Explain your reasoning and show your calculations. (You may use $\log_2 0.75 = -0.4$ and $\log_2 0.25 = -2$)

هنا أول سؤال اللي هو ايه ال effect بتاع لو اخترنا ال Weekend ك root attribute في ال decision tree بتاعتنا .. هو اللي يحصل انها هتودي كلو ناحية ال False فهو technically انت معملتش اي improvement لأي حاجه خالص سواء كان ال gini gain او mutual information ...

طب سؤال ثاني .. لو احنا مش هنعرف نخط ال Weekend هي الروت .. هنستخدم ايه ... في الحاله ديه جرب بقا ال gini gain و ال error rate و ال mutual information و احسب حساباتك و قول انهي الاحسن ..

السؤال اللي بعدو ... عندك شوية داتا سبت .. وبتقول بلا ن apply ال KNN باستخدام ال Euclidean distance .. وهي binary classification problem و انت بتحاول ت classify read square or blue diamond .. بعدين تقول أول سؤال .. يقابلك هو .. what is the DB? ... هتقول مثلاً لو احنا قلنا عند ال $k=1$.. هتعرف ترسمها؟ ..

اجابتي... هو كل مثال هيبقا اقرب واحد لنفسو .. فالديسجين باوندري هتبقا فاصله ما بين كل مثال واللي جمبو ..

الدكتور جاب وقال ... هي ال DB هتقولك how every possible point would be classified .. فممك مثلاً تفكر في ال DB كإنو بي shade in .. all the parts be classified red if we trained at $k = 1NN$

Sample Questions

4 K-NN [12 pts]

Now we will apply K-Nearest Neighbors using Euclidean distance to a binary classification task. We assign the class of the test point to be the class of the majority of the k nearest neighbors. A point can be its own neighbor.

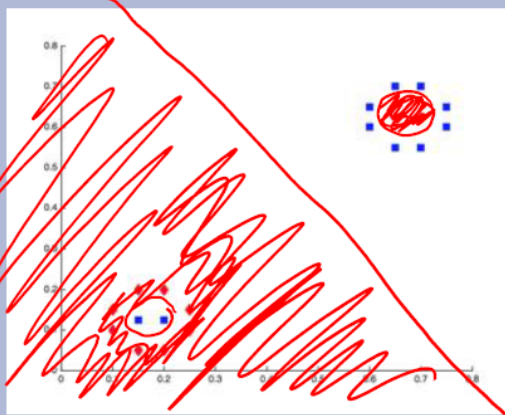


Figure 5

2. what is the DB? $k=1$

N-fold C.V.

3. [2 pts] What value of k minimizes leave-one-out cross-validation error for the dataset shown in Figure 5? What is the resulting error?

11

فهتلاقي ان في خط هيفصل في النص كذا .. وبعدين يلون دايره حمرا عند اللي فوق علي اليمين .. و هيلون كلو أحمر ماعدا الدايره الزرقا عند اللي تحت علي الشمال .. وهي ديه ال DB ..

السؤال اللي بعدو ... ال leave-one-out هو ال N-fold cross validation ... وده بيقول أنه في قيمه لل K اللي هتقلل ال N-fold cross validation error للداتا سبت اللي عندنا .. فبالتالي ده سؤال مهم جداً .. عشان ال N-fold cross validation error بيتحسب بشكل مختلف عن التريننج إيرور ليه . لما بتيجي تحسب ال training error .. هتسأل السؤال ده .. what is the points nearest neighbor .. لما بتعمل evaluation for training error ... اقرب نقطه هي نفسها ... هه ... بس في ال N-Fold-cross-validation error ... انت بتقول إن عدد ال folds بيساوي عدد ال training examples وبالتالي .. كل فولد جواه إكزامبل واحد .. فكأنك بتقول . خلاص انا هاخذ النقطه الزرقا ديه من الداتا سبت .. واحطها علي جمب .. واعمل تريننج علي النقط اللي فاضله .. وده هيديلك decision boundary .. وبعدين تقوم واخذ النقطه الزرقا اللي حطيتها علي جمب ديه وتقوم عاملها evaluation علي ال DB اللي طلعتك .. وتقول هل انا كذا صح ولا ايه ..

السؤال اللي بعدو عن البرسيبيترون .. اللي احنا مش عارفينو هنا هو المارجن ايه اصلاً .. ومش عارفين كمان ال max L2 norm هيبقا عامل ازاي فبالتالي انت معندكش المعلومات الكافيه اللي تخليك تقول يا آه يا لا .. فهو الاجابه هتبقا لا لأن انت معندكش معلومات كفايه

Sample Questions

4.1 True or False

Answer each of the following questions with T or F and provide a one line justification.

- (a) [2 pts.] Consider two datasets $D^{(1)}$ and $D^{(2)}$ where $D^{(1)} = \{(x_1^{(1)}, y_1^{(1)}), \dots, (x_n^{(1)}, y_n^{(1)})\}$ and $D^{(2)} = \{(x_1^{(2)}, y_1^{(2)}), \dots, (x_m^{(2)}, y_m^{(2)})\}$ such that $x_i^{(1)} \in \mathbb{R}^{d_1}$, $x_i^{(2)} \in \mathbb{R}^{d_2}$. Suppose $d_1 > d_2$ and $n > m$. Then the maximum number of mistakes a perceptron algorithm will make is higher on dataset $D^{(1)}$ than on dataset $D^{(2)}$.

False

$$K \leq \left(\frac{R}{\gamma}\right)^2$$

اللي بعدو سؤال عن اللينير ريجريشن .. عندك داتا سيت و عندك الخط ده بعد التريننج ... السؤال هنا ايه اللي هيجصل لل learned linear regression line و احنا بن manipulate the data set ..

Sample Questions

3.1 Linear regression

Consider the dataset S plotted in Fig. 1 along with its associated regression line. For each of the altered data sets S^{new} plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line					

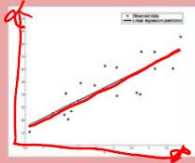


Figure 1: An observed data set and its associated regression line.

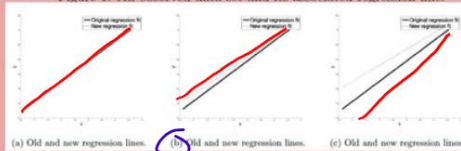
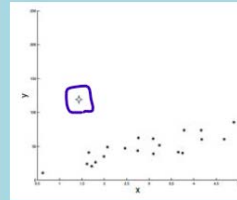


Figure 2: New regression lines for altered data sets S^{new} .

Dataset



(a) Adding one outlier to the original data set.

13

Sample Questions

3.1 Linear regression

Consider the dataset S plotted in Fig. 1 along with its associated regression line. For each of the altered data sets S^{new} plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line					

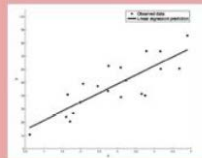


Figure 1: An observed data set and its associated regression line.

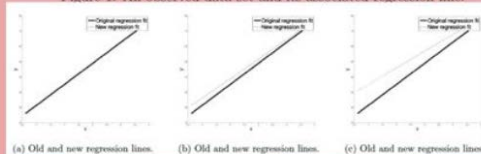
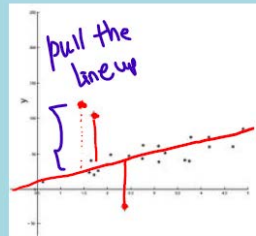


Figure 2: New regression lines for altered data sets S^{new} .

Dataset



(c) Adding three outliers to the original data set. Two on one side and one on the other side.

14

الدكتور في كل الأمثلة فيه احنا بنبص علي اننا عندنا outlier بطبيعتها بتحاول تشد الخط ناحيتها .. و الخط ده بيحاول اصلا انو يقلل ال residuals .. فلما هتبص من الناحية فيه .. هتفهم ايه اللي بيحصل .

Sample Questions

3.1 Linear regression

Consider the dataset S plotted in Fig. 1 along with its associated regression line. For each of the altered data sets S^{new} plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line					

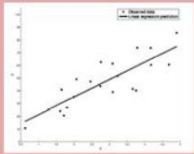


Figure 1: An observed data set and its associated regression line.

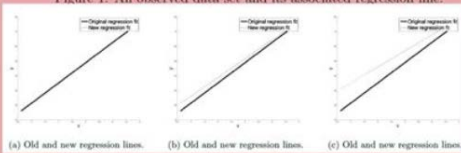
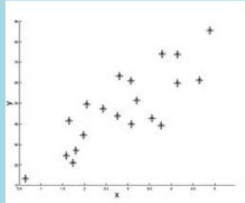


Figure 2: New regression lines for altered data sets S^{new} .

Dataset



(d) Duplicating the original data set.

stay the same

15

Sample Questions

3.1 Linear regression

Consider the dataset S plotted in Fig. 1 along with its associated regression line. For each of the altered data sets S^{new} plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line					

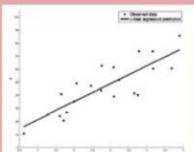


Figure 1: An observed data set and its associated regression line.

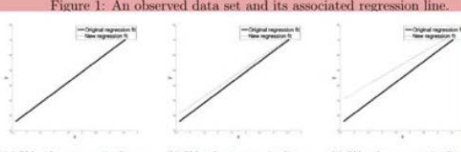
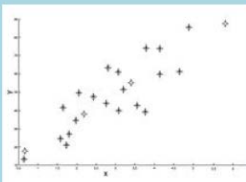


Figure 2: New regression lines for altered data sets S^{new} .

Dataset



(e) Duplicating the original data set and adding four points that lie on the trajectory of the original regression line.

same

16

سؤال ثاني .. احنا ليه بنتكلم علي ال linearly separable data في البرسيبترون .. هل مش هيشغل علي حاجه non linear separable .. الاجابه لا ممكن تضيف فينتشرز او ان الواحد يستخدم kernels تعمل مابينج ل higher dimensional features الداتا هناك بتبقا linearly separable ..

يلا نتكلم بقا علي ال optimization method رقم 3 ...

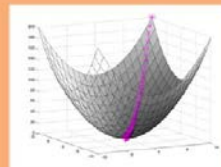
Gradient Descent

Algorithm 1 Gradient Descent

```

1: procedure GD( $\mathcal{D}$ ,  $\theta^{(0)}$ )
2:    $\theta \leftarrow \theta^{(0)}$ 
3:   while not converged do
4:      $\theta \leftarrow \theta - \gamma \nabla_{\theta} J(\theta)$ 
5:   return  $\theta$ 

```



ال SGD .. انت كنت عارف ان ال gradient descent انت بت optimize some objective function .. اللي هي $J(\theta)$... فانت كنت بت initialize الثيتا لقيمه ما .. وبعدين تفضل تلف تلف بخطوات معينه ناحية الثيتا .. اللي هي الجاما اللي هي الليرننج ريت اللي هي الستيب سايز .. بالنسبه لل SGD انت هتغير بس انك هتضيف سطر علي الكود ... انت هت uniformly sample اكراميل واحد بس .. بشكل عام هنفترض ان ال $G(\theta)$ متعرفه كال sum of a bunch of little j_i functions .. اللي ال SGD هتعملو انها هت randomly pick one of the terms في ال summation و بعدين تحسب الجريدينت بتاعه وتاخذ خطوه عكس الجريدينت ده ... لو جينا نبص من ناحية ال machine learning setting .. في ال mean squared error انت بتاخذ السامشن بتاع الإيروزر لشوية تريننج إكزامبلز فبالتالي من واحد لحد N هو N training examples .. لما بن sample a single i فده كإنك بتقول انا هختار training example وبعدين ت evaluate الجريدينت للإكزامبل ده بس ..

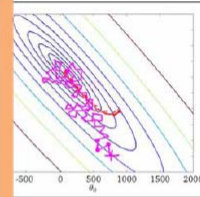
Stochastic Gradient Descent (SGD)

Algorithm 2 Stochastic Gradient Descent (SGD)

```

1: procedure SGD( $\mathcal{D}, \theta^{(0)}$ )
2:    $\theta \leftarrow \theta^{(0)}$ 
3:   while not converged do
4:      $i \sim \text{Uniform}(\{1, 2, \dots, N\})$ 
5:      $\theta \leftarrow \theta - \gamma \nabla_{\theta} J^{(i)}(\theta)$ 
6:   return  $\theta$ 

```



We need a per-example objective:

$$\text{Let } J(\theta) = \sum_{i=1}^N J^{(i)}(\theta)$$

29

الدكتور قال اننا in practice لما هنجي ن SGD implement .. هتخط for loop معاك .. وديه هتأخذ شوية ارقام من واحد لحد N .. وبعدين randomly reorder them وبعدين iterate through that random order of the examples ... السبب لحاجه زي كذا .. ان ده فعلياً كإنك بتعمل sampling without replacement .. وفي ال practice ده مفيد عشان sampling without replacement بيضمنلك انك هتشوف كل تريننج اكزامبل من التريننج إكزامبلز ع الاقل مره واحده قبل ما تشوف أي تريننج اكزامبل من التريننج إكزامبلز مره ثانيه .. فالدكتور ببسأل .. ايه هو الفرق ما بين ال sampling with/without replacement ... الصوره الجايه هي ال with replacement الصوره اللي فوقك علي طول اللي مفهانش ال for loop هي الي without replacement

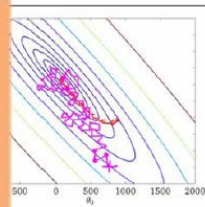
Stochastic Gradient Descent (SGD)

Algorithm 2 Stochastic Gradient Descent (SGD)

```

1: procedure SGD( $\mathcal{D}, \theta^{(0)}$ )
2:    $\theta \leftarrow \theta^{(0)}$ 
3:   while not converged do
4:     for  $i \in \text{shuffle}(\{1, 2, \dots, N\})$  do
5:        $\theta \leftarrow \theta - \gamma \nabla_{\theta} J^{(i)}(\theta)$ 
6:   return  $\theta$ 

```



In practice, it is common to implement SGD using sampling **without** replacement (i.e. $\text{shuffle}(\{1, 2, \dots, N\})$), even though most of the theory is for sampling **with** replacement (i.e. $\text{Uniform}(\{1, 2, \dots, N\})$).

We need a per-example objective:

$$\text{Let } J(\theta) = \sum_{i=1}^N J^{(i)}(\theta)$$

لو هنتكلم علي ال with replacement .. فانت هتقول عندك big bag of n numbers و انت بتدخّل إيدك و بتشد رقم و تطلع .. وليكن رقم 7 .. فتقوم واخذ 7 step opposite the direction of gradient .. وبعدين تحطو تاني في الشنطه و بعدين تحط إيدك تاني و تشد رقم تاني .. وبعدين تشوف طلع رقم 13 .. حطو تاني وبعدين شد رقم تاني .. هوب طلعلك رقم 7 تاني عادي ... وهكذا

انما في ال without replacement .. انت بتتكلم علي نفس الشنطه اللي فيها شوية ارقام .. وتمد إيدك تطلع برقم 7 .. تاخذ الجريدنييت في عكس الاتجاه وبعدين ترمي رقم 7 في الشارع برا الشنطه خالص .. وبعدين تمد إيدك تاني .. عمرك ما هيچيلك رقم 7 تاني .. وديه اللي احنا بنعملو في ال practice ..

لو جينا نفكر في ال 2 الجورنمز اللي عندنا عشان يتحركو من نقطة ما . إلي نقطة minimum .. اللي gradient descent هتعملو انها هتقعد و تحسب ال full gradiet of objective function .. واول ما هتحسبها هوب روح اعمل ال update وخذ خطوة ناحية المنييم . وبعدين تحسبها تاني و تاخذ خطوه تانيه وهكذا وهكذا .. بس خد بالك ان الحسبه الواحده لل gradient decsent بتحرق وقت كثير ..

لو جينا نبص علي ال SGD هتقولك بص بص بص .. بص علي الحلاوه؟ .. بيني ركز احنا في محاضره .. هتقولك بص بص خد مثال وليكن مثلاً مثال رقم 7 .. انا بحب رقم 7 .. وخذ ال direction of the gradient for example 7 .. direction of the gradient for example 7 .. خدنا step في الاتجاه الغلط .. ال SGD تقولك . يا بيه ولا يهيمك يا بيه .. خد مثال 13 .. و 13 ده هيوديك في الاتجاه اللي انت عاوزو .. متزعش يا بشمهندس .. و هتتوالي الارقام زي مثال 12 .. تقوم مودياك في الناحيه اللي انت عاوزها فتتبسط أكثر و تفتكر العجله دارت و هيبه بقا هنوصل .. و فضلت ماشي مثال 1 و مثال 11 و مثال 7 تاني عادي .. فزي منت شايف ال SGD هي مش محتاجه انها تقعد وقت طويل أوي تحسب كميات كبيره جداً تاخذ علي اساسها القرار .. انما ال SGD هي بتحسب كميات من ال updates الصغيره واحده واحده وبسرعه جداً .. فحتي لو هي غلطت في واحده .. الباقي هيصححها .. فهتوصل إن شاء الله

Convergence Curves

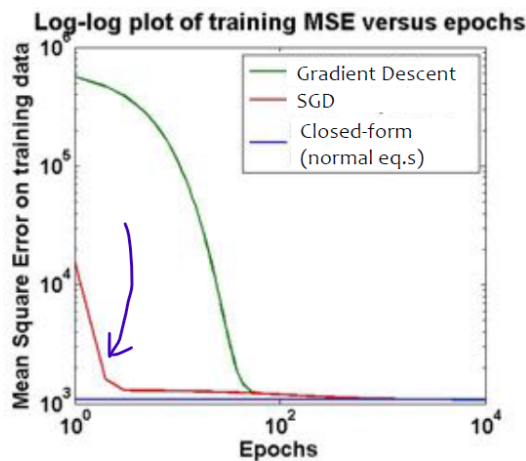


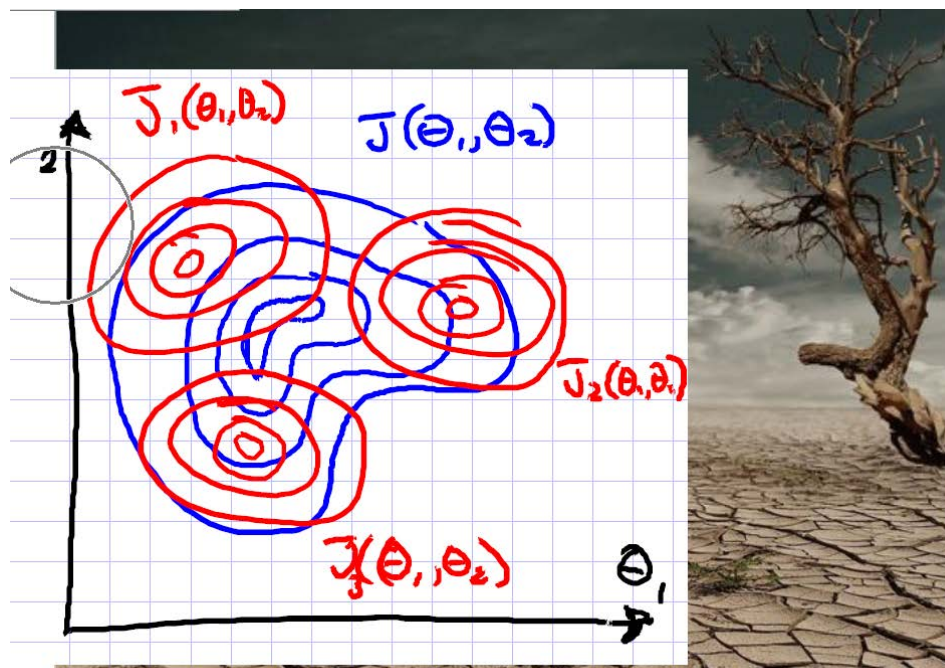
Figure adapted from Eric P. Xing

- Def: an epoch is a single pass through the training data
- 1. For GD, only **one update** per epoch
- 2. For SGD, **N updates** per epoch
 $N = (\# \text{ train examples})$
- SGD reduces MSE much more rapidly than GD
- For GD / SGD, training MSE is initially large due to uninformed initialization

واحد لكو مثال

عشان نفهم هي ليه ال SGD بتشتغل .. تعال نتخيل انك بتشتغل في cellphone company و ديه اسمها بطيخه .. والبطيخه عاوزه بيقا ليها coverage في كل ناحيه و بيقا coverage جامد ... فانت موظف في الشركه ديه .. و بتسأل هل تسمعني .. و بتتأكد يعني ان ال coverage مظبوطه في كل ناحيه .. ففي مرة من المرات .. رحت علي مكان صحرا .. وهوب العربيه بتاعتك باظت ... تفتح موبايلك وتتصل ... الموبايل قرب يفصل خلاص فاضل 50 ثانيه .. و انت محتاج تعمل phone call .. بس لو حصل مشكله هوب دابل كيك .. حرفياً انت هتفرقع في الصحرا محدش هيسمعك ... فعشان حظك حلو ... انت معاك جهاز قادر يقولك الاتجاه اللي فيه ال cell phone tower وليه سجنال قويه .. اللي تقدر تعمله انك تشغل الجهاز ده لكل tower من التاورز اللي موجوده وتسأل .. اروح فين عشان اجيب احسن signal منو ... الموبايل بتاعك بيقدر ي combine cell phone tower signals وبيعمل ده additively .. فانت تقدر تفكر في ال 3 سجنالز اللي جايبين من ال 3 cell phone towers .. وال $J(\theta)$ في الصوره بتعبر عن ال sum of the signal from all 3 towers و انت هدفك انك ت maximize او انك ت minimize ال sum of those signals .. فأنهي الجورنم نستخدمو .. انت متقدرش تحسب ال J علي طول بشكل مباشر كذا .. عشان انت معندكش ال .. ability to .. فانت بتشتغل جهازك وتسألو فين الاتجاه اللي هي most increase my ignal to cell phone tower 1 ... فيقولك .. روح فوق في الاتجاه ده ... بس وانت ماشي تقوم مشغل الجهاز تاني .. وتسألو عن ال cell tower رقم 2 .. فيقولك روح هناك في الاتجاه ده .. فتروح فعلاً .. وبعد ما توصل تقوم مشغل الجهاز تاني .. وتسألو علي cell phone tower 3 .. فيقولك روح هناك .. فتروح فعلاً .. وبعدين تقوم مشغل جهازك تاني .. تسألو علي cell tower 2 .. يقولك روح هناك ... فتروح برضو .. وبعدين بعد ما توصل تسألو علي أول سيلل تاور .. فيقولك روح هنا .. فتروح برضو .. وبعدين تفضل تروح و تروح و تروح .. وواحد واحد فعلاً هتلاقيك وانت مش واخذ بالك ... بتوصل لل

Minimum أو ال maximum بتاع الأوبتيمم بتاع ال J function .. اللي هيديك ال strongest sum of all signals .. فالدكتور ببسال سؤال .. هو
 ازاي الحوار ده بيشتغل .. ليه بيشتغل يعني .. ازاي وانا بتحرك ناحية each of these little minima هينتهي بيك الحال انك توصل لل minimum of
 both of the sum of all 3 .. هنجاب عن طريق اننا نشوف ايه اللي بيحصل في ال expectation ..
 تعال الاول نحط ال problem setting الاول .. هنعول ان ال $J(\theta)$ هي الأفريدج بتاع شوية فانكشنز $J_i(\theta)$.. فلو حسبت الدريفاثيف ... هتلاقي
 الصورة الجايه



Expectations of Gradients

$$\frac{dJ(\theta)}{d\theta_j} = \frac{1}{N} \sum_{i=1}^N \frac{d}{d\theta_j} (J_i(\theta))$$

$$\nabla J(\theta) = \begin{bmatrix} \vdots \\ \text{Jth} \\ \vdots \end{bmatrix} = \frac{1}{N} \sum_{i=1}^N \nabla J_i(\theta)$$

Recall: for any discrete r.v. X

$$E_X[f(X)] \triangleq \sum_{x} P(X=x) f(x)$$

sum of all values
* $\nabla J(\theta)$

Q: What is the expected value of a randomly chosen $\nabla J_i(\theta)$?

Let $I \sim \text{Uniform}(\{1, \dots, N\})$
 $\Rightarrow P(I=i) = \frac{1}{N}$ if $i \in \{1, \dots, N\}$

$$E_I[\nabla J_I(\theta)] = \sum_{i=1}^N P(I=i) \nabla J_i(\theta)$$

Random variable

$$= \frac{1}{N} \sum_{i=1}^N \nabla J_i(\theta)$$

definition of gradient

$$= \nabla J(\theta)$$

لو جينا نبص علي ال definition of gradient .. هتلاقي معناه ان expectation you are moving in the direction of the gradient of the full objective function when you randomly sample these little gradients of J_i
 .. SGD بيشتغل .. وده هو السبب اللي عشانو ال GD and SGD convergence analysis للاتنين الجورنمز ممكن نفكر في convergence analysis للاتنين الجورنمز .. الكونفرجنز هو الموقف اللي بيبقا الاختلاف ما بين ال
 بالنسبة للسرعه ...

$J(\theta)$ و $J(\theta^*)$.. الثيتا ستار هي قيم الثيتا اللي فعلاً بتقلل ال J وانت هنا مش عارف ايه هي الثيتا ستار لحد ما هتحل ال optimization problem .. هنا احنا بنفكر من ناحية ال theoretical perspective .. انت عندك some convex function J .. الكونفيرجنس هيبقا هو قيمة ال J اللي بتجيبك دلوقت بتبقا قريبه للمنييم .. الإبسيلون ديه بتبقا رقم صغير زي مثلاً 10 اس سالب 8 .. هنا انت شايف ان ال SGD . هيكونفرج في عدد من الخطوات $O(1/\epsilon)$.. بص عالصوره

Convergence of Optimizers

Convergence Analysis:

Def: Convergence is when $|J(\theta) - J(\theta^*)| < \epsilon$ true value min

Methods	Steps to Converge	Computation per iteration
Newton's Method	$O(\ln \ln 1/\epsilon)$	$\nabla^2 J(\theta) \leftarrow O(NM^2)$
GD	$O(\ln 1/\epsilon)$	$\nabla J(\theta) \leftarrow O(NM)$
SGD	$O(1/\epsilon)$	$\nabla J_i(\theta) \leftarrow O(M)$

not correct (pointing to Newton's Method)

$\epsilon = 10^{-8} \rightarrow \frac{1}{\epsilon} = 10^8$ "almost sure" convergence lots of constants and conditions way less computation

Takeaway: SGD has much slower asymptotic convergence, but is often faster in practice.

35

فهتلاقى ان عدد الخطوات بتاعت ال GD اقل من ال SGD ... حاجه تانيه هنفكر فيها هو عدد الحسابات اللي بتتعمل لكل ايتيريشن .. ال GD بتبص علي N examples ولكل واحد ليه M features انما ال SGD هو مثال واحد ب M فيتشرز . فده هيقولك ان SGD ليه slower asymptotic convergence بس علي ارض الواقع هو اسرع ... وده اللي شغناه في الرسمه ما بين ال SGD and GD ..

Linear Regression as Function Approximation

$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$
where $\mathbf{x} \in \mathbb{R}^M$ and $y \in \mathbb{R}$

1. Assume \mathcal{D} generated as:

$$\mathbf{x}^{(i)} \sim p^*(\cdot) \\ y^{(i)} = h^*(\mathbf{x}^{(i)})$$

2. Choose hypothesis space, \mathcal{H} :
all linear functions in M -dimensional space

$$\mathcal{H} = \{h_{\theta} : h_{\theta}(\mathbf{x}) = \theta^T \mathbf{x}, \theta \in \mathbb{R}^M\}$$

3. Choose an objective function:
mean squared error (MSE)

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N e_i^2 \\ = \frac{1}{N} \sum_{i=1}^N (y^{(i)} - h_{\theta}(\mathbf{x}^{(i)}))^2 \\ = \frac{1}{N} \sum_{i=1}^N (y^{(i)} - \theta^T \mathbf{x}^{(i)})^2$$

4. Solve the unconstrained optimization problem via favorite method:

- gradient descent
- closed form
- stochastic gradient descent
- ...

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} J(\theta)$$

5. Test time: given a new \mathbf{x} , make prediction \hat{y}

$$\hat{y} = h_{\hat{\theta}}(\mathbf{x}) = \hat{\theta}^T \mathbf{x}$$

37

Gradient Calculation for Linear Regression

Derivative of $J^{(i)}(\theta)$:

$$\begin{aligned} \frac{d}{d\theta_k} J^{(i)}(\theta) &= \frac{d}{d\theta_k} \frac{1}{2} (\theta^T \mathbf{x}^{(i)} - y^{(i)})^2 \\ &= \frac{1}{2} \frac{d}{d\theta_k} (\theta^T \mathbf{x}^{(i)} - y^{(i)})^2 \\ &= (\theta^T \mathbf{x}^{(i)} - y^{(i)}) \frac{d}{d\theta_k} (\theta^T \mathbf{x}^{(i)} - y^{(i)}) \\ &= (\theta^T \mathbf{x}^{(i)} - y^{(i)}) \frac{d}{d\theta_k} \left(\sum_{j=1}^K \theta_j x_j^{(i)} - y^{(i)} \right) \\ &= (\theta^T \mathbf{x}^{(i)} - y^{(i)}) x_k^{(i)} \end{aligned}$$

Gradient of $J^{(i)}(\theta)$

[used by SGD]

$$\begin{aligned} \nabla_{\theta} J^{(i)}(\theta) &= \begin{bmatrix} \frac{d}{d\theta_1} J^{(i)}(\theta) \\ \frac{d}{d\theta_2} J^{(i)}(\theta) \\ \vdots \\ \frac{d}{d\theta_M} J^{(i)}(\theta) \end{bmatrix} = \begin{bmatrix} (\theta^T \mathbf{x}^{(i)} - y^{(i)}) x_1^{(i)} \\ (\theta^T \mathbf{x}^{(i)} - y^{(i)}) x_2^{(i)} \\ \vdots \\ (\theta^T \mathbf{x}^{(i)} - y^{(i)}) x_N^{(i)} \end{bmatrix} \\ &= (\theta^T \mathbf{x}^{(i)} - y^{(i)}) \mathbf{x}^{(i)} \end{aligned}$$

Derivative of $J(\theta)$:

$$\begin{aligned} \frac{d}{d\theta_k} J(\theta) &= \sum_{i=1}^N \frac{d}{d\theta_k} J^{(i)}(\theta) \\ &= \sum_{i=1}^N (\theta^T \mathbf{x}^{(i)} - y^{(i)}) x_k^{(i)} \end{aligned}$$

Gradient of $J(\theta)$

[used by Gradient Descent]

$$\begin{aligned} \nabla_{\theta} J(\theta) &= \begin{bmatrix} \frac{d}{d\theta_1} J(\theta) \\ \frac{d}{d\theta_2} J(\theta) \\ \vdots \\ \frac{d}{d\theta_M} J(\theta) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N (\theta^T \mathbf{x}^{(i)} - y^{(i)}) x_1^{(i)} \\ \sum_{i=1}^N (\theta^T \mathbf{x}^{(i)} - y^{(i)}) x_2^{(i)} \\ \vdots \\ \sum_{i=1}^N (\theta^T \mathbf{x}^{(i)} - y^{(i)}) x_N^{(i)} \end{bmatrix} \\ &= \sum_{i=1}^N (\theta^T \mathbf{x}^{(i)} - y^{(i)}) \mathbf{x}^{(i)} \end{aligned}$$

39

SGD for Linear Regression

SGD applied to Linear Regression is called the “Least Mean Squares” algorithm

Algorithm 1 Least Mean Squares (LMS)

1: **procedure** LMS($\mathcal{D}, \theta^{(0)}$)

2: $\theta \leftarrow \theta^{(0)}$

3: **while** not converged **do**

4: **for** $i \in \text{shuffle}(\{1, 2, \dots, N\})$ **do** *pick random example*

5: $\mathbf{g} \leftarrow (\theta^T \mathbf{x}^{(i)} - y^{(i)}) \mathbf{x}^{(i)}$ *Compute gradient*

6: $\theta \leftarrow \theta - \gamma \mathbf{g}$ *Update parameters*

7: **return** θ

▷ Initialize parameters

▷ Compute gradient

▷ Update parameters

GD for Linear Regression

Gradient Descent for Linear Regression repeatedly takes steps opposite the gradient of the objective function

Algorithm 1 GD for Linear Regression

```

1: procedure GDLR( $\mathcal{D}, \theta^{(0)}$ )
2:    $\theta \leftarrow \theta^{(0)}$  ▷ Initialize parameters
3:   while not converged do
4:      $\mathbf{g} \leftarrow \sum_{i=1}^N (\theta^T \mathbf{x}^{(i)} - y^{(i)}) \mathbf{x}^{(i)}$  ▷ Compute gradient
5:      $\theta \leftarrow \theta - \gamma \mathbf{g}$  ▷ Update parameters
6:   return  $\theta$ 

```

دلوقت هنتكلم علي ال probabilistic learning .. هنا هنعوز ن paint a contrast ما بين اللي احنا كنا بنعملو و اللي احنا هنعملو .. قيل كذا انت افترضت ان الأوتبوت كان generated using some deterministic target function .. ده اللي هو اننا افترضنا ان ال X_i كانت sampled من some unknown p^* .. بس بعد كذا ال Y_i كانوا computed deterministically from some function C^* .. واللي حاولنا نعملو اننا حاولنا نتعلم ال hypothesis h .. اللي بت best approximate C^* .. في ال probabilistic learning احنا هنعمل حاجه مختلفه شويه .. احنا هنفترض تمام ان الإكسبات جايين من نفس المكان .. بس الواي $P^*(y|x)$ sampled from unknown probability distribution .. فالهدف اننا نتعلم ال $p(y|x)$ اللي بت best approximate $p^*(y|x)$... تعال نفكر في مثال بس ... لو حاولنا نحل مشكلة ال robotic farming .. ده هيبقا جامد جداً ... وهيبقا ليك massive impact علي الزراعة ..

Probabilistic Learning

Function Approximation

Previously, we assumed that our output was generated using a **deterministic target function**:

$$\mathbf{x}^{(i)} \sim p^*(\cdot)$$

$$y^{(i)} = c^*(\mathbf{x}^{(i)})$$

Our goal was to learn a hypothesis $h(\mathbf{x})$ that best approximates $c^*(\mathbf{x})$

Probabilistic Learning

Today, we assume that our output is **sampled** from a conditional **probability distribution**:

$$\mathbf{x}^{(i)} \sim p^*(\cdot)$$

$$y^{(i)} \sim p^*(\cdot | \mathbf{x}^{(i)})$$

Our goal is to learn a probability distribution $p(y|x)$ that best approximates $p^*(y|x)$

47

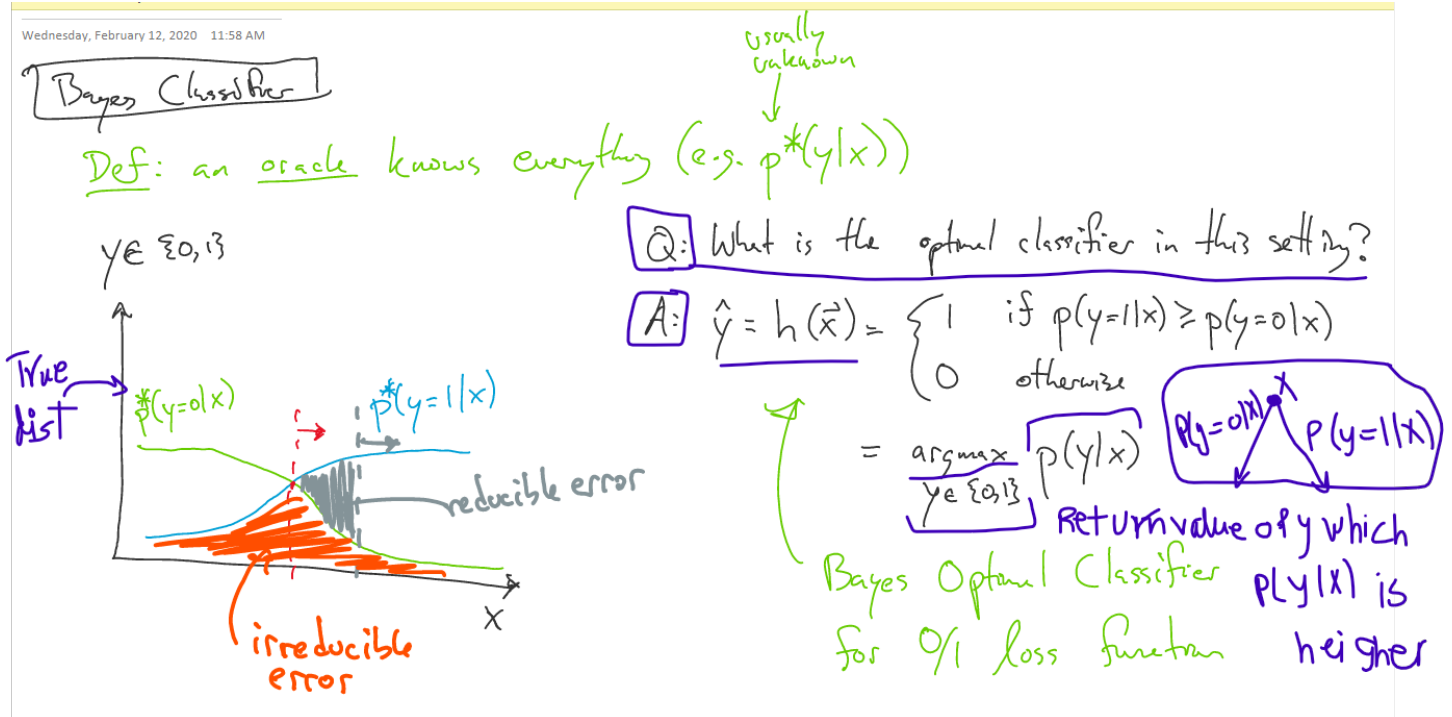
Robotic Farming

	Deterministic	Probabilistic
Classification (binary output)	Is this a picture of a wheat kernel?	Is this plant drought resistant?
Regression (continuous output)	How many wheat kernels are in this picture?	What will the yield of this plant be?



تعال نشوف الصورة اللي فاتت ... في سؤال ممكن تسألو اللي هو هل الصورة فيها wheat kernel .. ده يا آه يا لا .. بس انا مش عارف انا مش مزارع .. هات مزارع يقولك ... لو سألنا سؤال ريجريشن ... عندنا كام wheat kernels ... طيب منين نجيب ال probabilistic notion .. ممكن نسأل هل ال plat drought resistant ... ده سؤال صعب اجابته .. عشان حتي لو عندك زرع ممكن ت survive a drought وليها احتمالية عاليه انها ت survive a drought هيكون عندك شوية زرع فعلاً هيعدي المرحلة ديه و في مش هيعدي .. فحتي لو هم في نفس ال species انت ممكن ت give a probability distribution over whether a plant of that species is going to survive a drought

بس انت متقدرش تقول للزرع كلو .. يعني مثلاً ... حد سألك علي زرع واحد من كل الزرع اللي قدامك .. فانت ممكن تقولو هو بشكل عام احنا عندنا ال probability distribution for our species انما للزرع ديه بالذات مقدرش اقول .. فديه هي ال probabilistic questions .. فاللي احنا عاوزين نعملو عشان نفهم احسن ال setting الجديد ده ... هو ان نفكر بطريقه مختلفه شويه الي هو اننا هنفترض اننا بنحاول نلاقي optimal classifier بس الكلاسيفاير نفسو ... هستخدم شوية معلومات اكثر من المتاح.. ففي ال setting ده بنفكر في ال bayes classifier .. هنعرف حاجه اسمها ال oracle .. ديه حاجه عارفه كل حاجه .. يعني مثلاً .. $p^*(y|x)$.. ديه حاجه مبتقاش معروفه .. بس الأوراكل عارفها .. هم عارفين كل حاجه ... الصورة فيها كل الكلام اللي عاوزو



هو هيبص علي الإكس ويحدد هل ارجع صفر ولا واحد .. فهتلاقي عندك نقطه هيحصل عندها السويتش .. فهتلاقي ال DB هو الخط الأحمر الفيرتكال ده .. و كل حاجه علي اليمين هتبقا واي ب واحد .. علي الشمال هتبقا واي بصفر .. طيب احنا ليه منستخدمش ال DB اللي بالرمادي .. ايه اللي احنا بنكرهه في ال DB .. فحتي لما كان عندك ال $y=1$ ليها بروبابلتي اعلي .. انت هتقول انها صفر .. وده غلط .. فانت عندك مساحه كذا بتبص عليها اسمها ال reducible error .. و المساحه اللي عندك باللون الأحمر هي ال irreducible error .. فده بقي probabilistic classifier .. وهو ده ال Bayes optimal classifier .. وهو كلاسيفايير عشان مفيش stochasticity بس ده للصفر والواحد للووس فانكشن .. فخد بالك ان باختلاف ال loss functions هيبقا عندك different notions of optimal .. تعال ناخذ مثال .. بيتكلم علي ال public health setting اللي هو انك بتحاول ت predict whether or not a patient has some form of a heart disease .. فهتعمل تست بس التست هيديلك احتماليه ان عندهم مرض في القلب .. وانت وظيفتك كدكتور انك تقرر هل تقول للمريض انو عندو heart disease ولا لا .. في الموقف ده .. ايه اللي هيحصل لو انت استخدمت ال Bayes optimal classifier for 0 1 loss function .. قول ان التست جيه و احتماليه ان في مرض في القلب أعلي من انو مفيش .. فتقوم قايل للمريض انت عندك مرض في القلب .. تخيل نفسك المريض بقا .. والتست جيه برضو .. و في احتماليه 45 % عندك مرض في القلب و 55 % معندكش فالدكتور يقولك قوم يا واد اجري وانتظ انت سليم فانت كذا عندك مشكله في الدكتور لإنو بيعرض حياتك للخطر .. بيقدر علي ال 5 % الدكتور الجامد جداً .. فاللي انت تتمني انو يحصل انو يستخدم different loss function بتقول ان ال false negative more costly عشان انت بتقول للمريض انو كويس .. ومتغيرش حاجه ف حياتك بس هو اصلاً احتمال ان التست غلط وانت محتاج تتدخل ... طيب ده لو انت عارف ال true probability distribution فهتتعرف تعمل منو ال classifier .. طب هتجيب منين ال true prob. Distr. ... الاجابه مفيش .. متقدرش .. عشان كل الي معانا هو مجرد sample access distribution .. فبالتالي احنا هنستخدم ال principle بتاع ال max likelihood estimation عشان نقدر نلاقي ال probability distribution اللي بتبقا اقرب حاجه .. فاحنا هنختار ال parameters الي تخلي الداتا اقرب حاجه ..

Maximum Likelihood Estimation

The principle of Maximum likelihood estimator (MLE):

Choose parameters that make the data "most likely".

Assumptions: Data generated iid from distribution $p^*(x|\theta^*)$ and comes from a family of distn parameterized $\theta \in \Theta$

Formally:

$$\begin{aligned}\theta_{MLE} &= \underset{\theta \in \Theta}{\operatorname{argmax}} p(D|\theta) \\ &= \underset{\theta \in \Theta}{\operatorname{argmax}} \log p(D|\theta) \\ &= \underset{\theta \in \Theta}{\operatorname{argmax}} \ell(\theta)\end{aligned}$$

where $\ell(\theta) \triangleq \log p(D|\theta)$
'log-likelihood'

usually a continuous optimization

set of possible parameters

$= \prod_{i=1}^N p(x^{(i)})$ since log is monotonic

treat as function of θ where D is constant

50

تعال ناخذ مثال ..

\bar{X} : public health

Stawman Example

Suppose we have dataset

y	x_1	x_2
1	0.5	9
0	3	4
1	2	1
1	1	-3

تعال نخط الألوآزم .. هنسميه bad idea#1 .. نخط شوية افتراضات .. اول واحد .. هو ايه هو ال inductive bias بتاعنا .. اننا نتجاهل تماماً ال x ..
تاني افتراض هنخطو ان الواي sampled من bernolli(phi) distribution ..

Bad Idea #1: "Bernoulli Classifier"

- ① Assumption: Ignore \bar{x}

- ② Model: $y \sim \text{Bernoulli}(\theta)$

$$p(y|\vec{x}) = \begin{cases} \emptyset & \text{if } y=1 \\ 1-\emptyset & \text{if } y=0 \end{cases}$$

- ③ ~~Likelihood~~ Conditional log-likelihood

h/c D is iid

Condition on θ \times

$$\begin{aligned} \ell(\theta) &= \log p(D|\theta) = \log \left[\prod_{i=1}^N p(y^{(i)}|x^{(i)}, \theta) \right] && \text{identical} \\ &= \sum_{i=1}^N \log p(y^{(i)}|x^{(i)}) && \text{4 terms = 4 examples} \\ &= \log \theta + \log(1-\theta) + \log \theta + \log \theta = 3\log \theta + \log(1-\theta) \end{aligned}$$

- ④ $\phi_{MLE} = \underset{\phi \in [0,1]}{\operatorname{argmax}} \ell(\phi) = \boxed{3/4}$ some value of ϕ that maximize $L(\phi)$

④ $\phi_{MLE} = \underset{\phi \in [0,1]}{\operatorname{argmax}} \ell(\phi) = 3/4$
 ↗ constrained optimization

- ⑤ Bayes Classifier

$$\hat{y} = h_{\theta_{MLE}}(\vec{x}) = \underset{y \in \{0,1\}}{\operatorname{argmax}} P(y|\vec{x}, \phi_{MLE}) = 1$$

Majority Vote