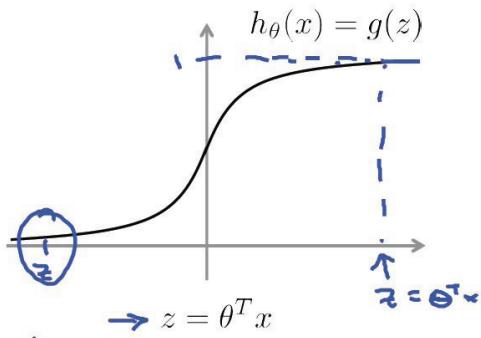


الأسبوع السابع

Support Vector Machines

- هذا هو التكنيك الأخير ، في الـ Supervised Learning
- في بعض الامور ، مقارنة بالـ NN و logistic regression من الادوات القوية اللي بتقدر تحل مشاكل بفعالية
- وعشان نفسهم الـ SVM كويس ، تعالى نفترك الـ logistic regression والسيجمويد

$$\rightarrow h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$



If $y = 1$, we want $h_{\theta}(x) \approx 1$ $\theta^T x \gg 0$
If $y = 0$, we want $h_{\theta}(x) \approx 0$ $\theta^T x \ll 0$

- قيمة الـ Z اللي هي ثيتا ترانزبوس في اكس ، كل ما تزيد ، كل ما الاس للاكسبونينيشيال يقل ، وبالتالي قيمة h تزيد تدريجيا لما توصل 1 ، وده اللي بابن في الرسم على الطرف اليمين
- قيمة الـ Z كل ما تقل ، كل ما الاس للاكسبونينيشيال يزيد ، وبالتالي قيمة h تقل تدريجيا لما توصل 0 ، وده اللي بابن في الرسم على الطرف اليسير

- طيب تعالى نبص على المعادلة نفسها :

Cost of example: $-(y \log h_{\theta}(x) + (1 - y) \log(1 - h_{\theta}(x)))$

- ديه هي المعادلة للـ L اللي ممكن نعرض فيها كدة

$$= -y \log \frac{1}{1 + e^{-\theta^T x}} - (1 - y) \log \left(1 - \frac{1}{1 + e^{-\theta^T x}}\right)$$

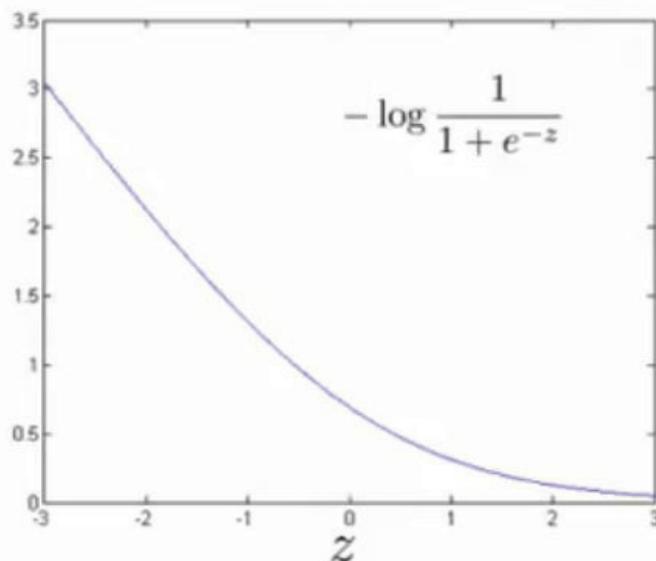
- وهنا ه يكون فيه حالتين ، إما y تساوي 1 او 0

- في حالة y تساوي 1 محور اكس او قيمة z (ثيتا ترانزبوز في اكس) ومحور واي هو قيمة $\theta^T x$
- في حالة y تساوي 1 ، ساعتها هنمكفي الجزء الاول من المعادلة اللي هو ده ، لأن الباقي هيكون بصفر

$$-y \log \frac{1}{1 + e^{-\theta^T x}}$$

○ وقتها الرسم هيكون كدة :

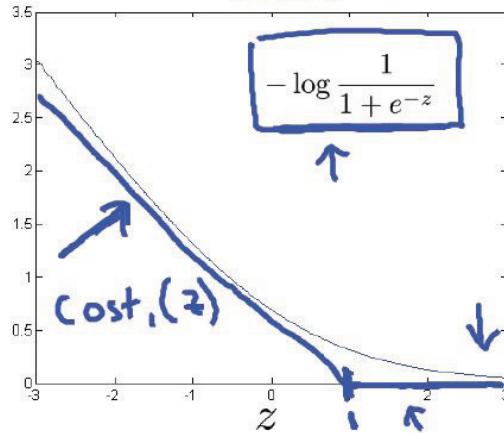
If $y = 1$ (want $\theta^T x \gg 0$):



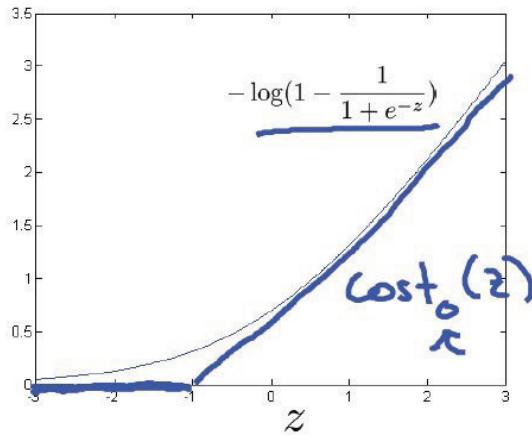
■ كل ما تزيد قيمة z كل ما قيمة $\frac{1}{1 + e^{-\theta^T x}}$ هتنزد لـ 1 ، اللي اللوج بتاعها بصفر

■ وكل ما تقل قيمة z كل ما قيمة $\frac{1}{1 + e^{-\theta^T x}}$ هتنزد (لما تقرب لصفر) فاللوج بتاعها هيكون بالسالب ، نضربه في سالب في القانون يبقى رقم موجب ، يبقى مرسم زي ما شاييفين

● لو عايزين نرسم الجراف ده خط مستقيم ، ممكن نعمل زي كدة :



- هنا هنلاقي ان بدل الا curve , عايزين نعمل خط مستقيم , لسبب معين
- عشان كدة جينا عند رقم 1 بالتحديد , وقلنا اللي اكتر منه هيكون قيمته بصفر , واللي اقل منه هيطلع خط مستقيم تقريبا يوازي الا curve
- الخط ده يمثل ما يسمى الا SVM ويسهل حسابه كمعادلات منه
- ذلك بالتوازي لو y تساوي صفر , الجراف هيكون العكس , وهنعمل خط الا SVM ليه كدة , واللي هيبدأ من نقطة سالب 1



- فالخط الأول يسمى $\text{cost}_1(\theta^T x^{(i)})$ عشان بيقيس قيمة الكوست لما y واحد بينما الخط الثاني اسمه $\text{cost}_0(\theta^T x^{(i)})$ عشان هنا y تساوي صفر
- دلوقي نفترق المعادلة ديه , الخاصة بال $\text{logistic regression}$

$$\min_{\theta} \frac{1}{m} \left[\sum_{i=1}^m y^{(i)} \left(-\log h_{\theta}(x^{(i)}) \right) + (1 - y^{(i)}) \left(-\log(1 - h_{\theta}(x^{(i)})) \right) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

- ويمكن نعرض قيمة $- \log h_\theta(x^{(i)})$ بـ $\text{cost}_1(\theta^T x^{(i)})$ وقيمة $- \log(1 - h_\theta(x^{(i)}))$ بـ $\text{cost}_0(\theta^T x^{(i)})$

احنا عايزين نجيب قيمة ثيتا اللي هتقل الدلالة لاقصي قدر ، ممكن نحذف $\frac{1}{m}$ من الطرفين لأن هو ثابت ، واحدنا عايزين نعمل تقليل

دلوقي هيكون الجزء الاول ده $\sum_{i=1}^m [y^{(i)}\text{cost}_1(\theta^T x^{(i)}) + (1 - y^{(i)})\text{cost}_0(\theta^T x^{(i)})]$ زائد لما في مجموع ثيتا سكوير

يعني متغير زائد ثابت في ثابت ، هنعكس العمليه ، بحيث يكون ثابت في متغير زائد ثابت ، وده مسموح فيه عشان هعمل تقليل

فهذا المدا ، و اضيف ثابت جديد للقيمة الاولى اسه C

يعني هيكون $C A + B$ بدل $A + B$

ساعتها المعادلة تبقي :

$$\min_{\theta} C \sum_{i=1}^m [y^{(i)}\text{cost}_1(\theta^T x^{(i)}) + (1 - y^{(i)})\text{cost}_0(\theta^T x^{(i)})] + \frac{1}{2} \sum_{j=1}^n \theta_j^2$$

فالنسبة للمقارنة بين الشكلين دول :

$$\begin{array}{c} A + \lambda B \\ C A + B \end{array}$$

الشكل الاول هو المعتاد للـ Logistic regression بينما الثاني للـ SVM

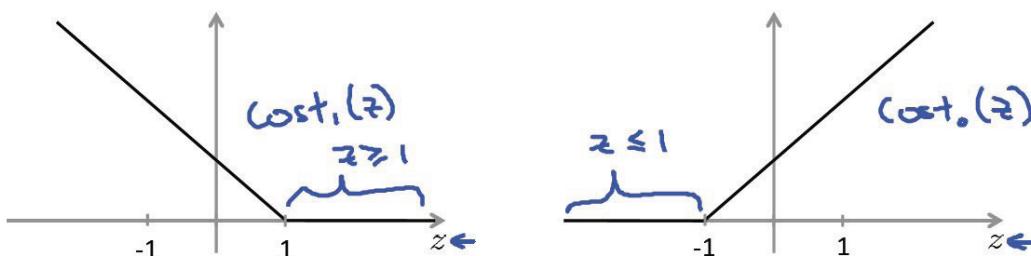
في الاول لما المدا تزيد ، قيمة B (الثبات) بتقل و الدالة A هترزيد

في الثاني لما الدالة C تزيد ، الدالة A هتقل بينما الدالة B هترزيد

قيم الدالة C والمدا بيتسابو عكسيا مع بعض

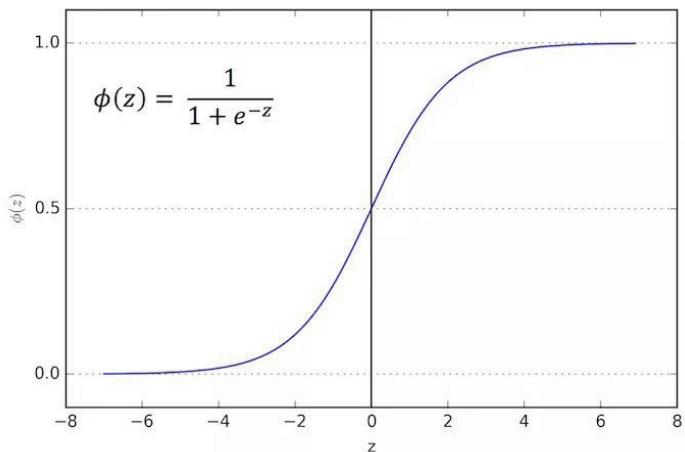
آلية مصفوفة الدعم ● Support vector machine

تعالي نبص في الصورة :



- هناقي ان عشان قيمة ϕ ل نقل محتاج تكون قيمة ϕ اللي هي $(\theta^T x^{(i)})$ تكون كالتالي :
 - لو y تساوي 1 , لازم ϕ تزيد عن 1 ■
 - لو y تساوي 0 , لازم ϕ تقل عن 1 ■

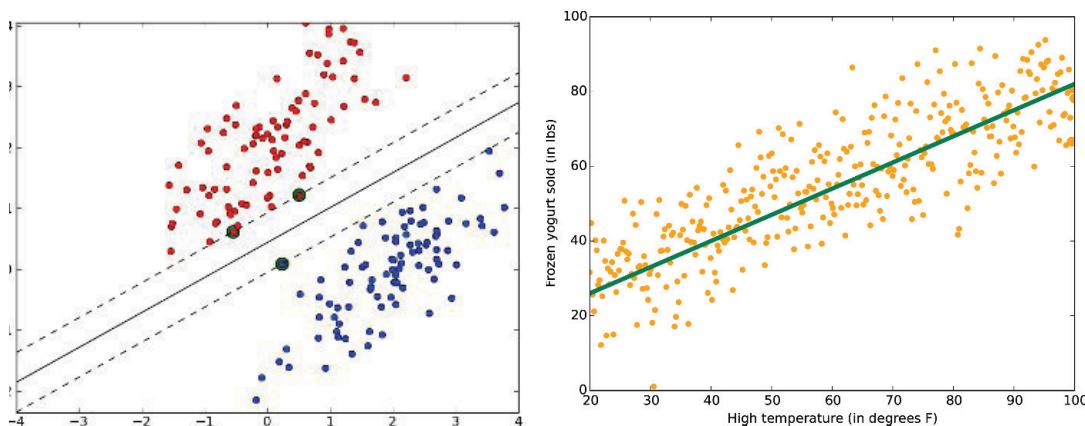
- و خد بالك ده يختلف شوية عن مفهوم ϕ اللي كان كدة :



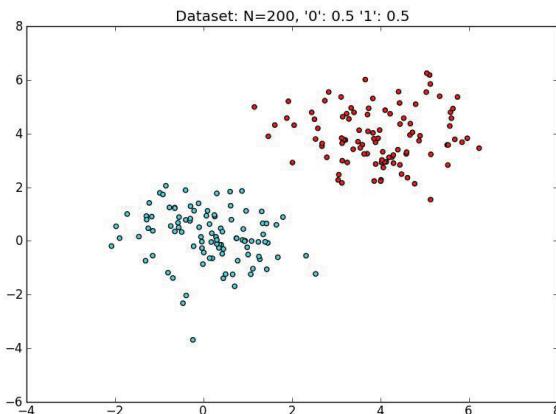
- هنا كنا بنقول طالما ϕ زادت عن صفر بقى بوحد , اقل من صفر بقى بصغر
- فهنا الحدود بتاعتني بقت اصعب شوية , رقم 1 بدل صفر في حالو ϕ بوحد , وسالب واحد بدل صفر في حاله ϕ بصغر

- If $y = 1$, we want $\theta^T x \geq 1$ (not just ≥ 0)
- If $y = 0$, we want $\theta^T x \leq -1$ (not just < 0)

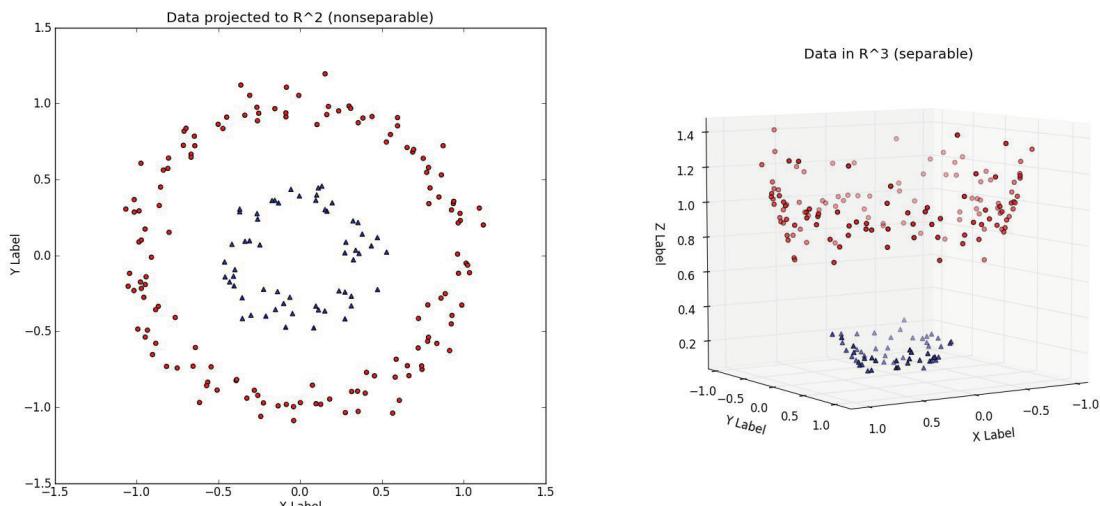
- زيادة الحدود ديه (من صفر لوحد , ومن سفر لسالب واحد) بتعمل فرق في الرسم من الشكل اليمين للشمال :



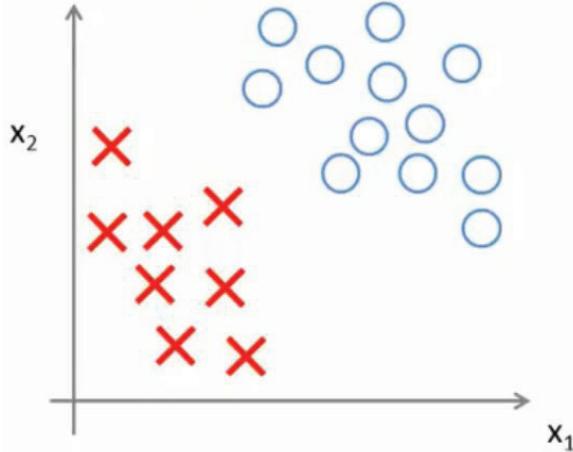
- يعني بدل ما كنا بنتعامل ان اي رقم للـ Z اكبر من صفر يبقى تبع ده ، واي رقم اصغر من صفر يبقى تبع ده ، عملنا ازاحة لكل النقط بعيد شوية عن الخط الفاصل من هنا و من هنا
- يعني البرنامج بيرسم خط بين النوعين ، بحيث يكون المسافة بين هذا الخط الفاصل ، والخط الذي يمس اخر نقاط المجموعة الاولى ، يساوي نفس المسافة بين الخط المنصف و الخط الذي يمس اخر نقاط المجموعة الثانية
- لاحظ ان التقسيم ممكن يكون خططي زي ده



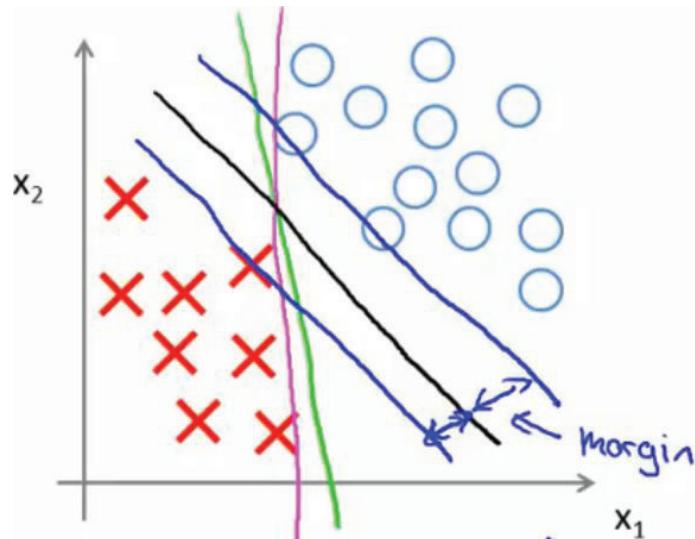
○ او غير خططي زي ده



- و خد بالك ان آلية رسم الخط مش اي حاجة و خلاص
- فلو النقط هي كدة مثلا :



- فممكن نعمل خط مائل يمين او شمال , زي الاخضر او البني
- بس هنلاقي ان الخط الاسود هو افضلهم , لانه عنده القدرة انه يتوقع و يقسم اي نقط مستقبلية قريبة من ده او ده , بينما الاخضر او البني مش هيعرف يقسم اي نقط مستقبلية صح
- و من علامات الخط السليم (الاسود مثلا) ان المسافات بينه وبين اقرب نقط من هنا , واقرب نقط من هنا تبقى اقصى ما يكون , يعني بيفصل بينهم بقدر الامكان
- المسافة بين الخط الازرق و الاسود بيسماها margin يعني هامش , عشان كدة الـ SVM بيسموها تكنيك الـ large margin classifier

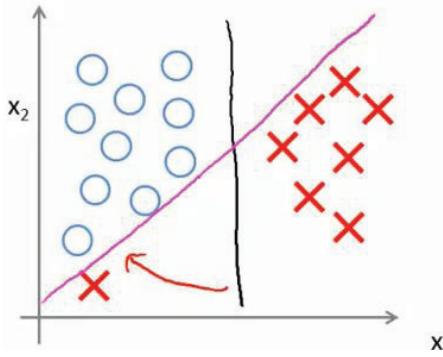


- طيب عايزين نعرف تأثير قيمة الـ C في الموضوع ده
- او لا عايزين نفكير ان C تساي تقريبا 1 علي لمدا من المعادلة ديه :

$$\min_{\theta} C \sum_{i=1}^m \left[y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{j=1}^n \theta_j^2$$

- اللي زي الصيغة ديه

$$C A + B$$
 - شايفين ان C مضروبة في مجموع الوايات في سيمجويد الثيتا و الاكس , فيكون فيه حالتين
 - الحالة الأولى تكون C كبيرة جدا , يعني مثلا 100000
 - ساعتها قيمة A المضروبة في C هتقل جدا , لما توصل لقرب الصفر , ويبقى الاعتماد على قيمة B
 - اللي هو مجموع مربع الثيتات
 - هنا بيكون تأثير الثيتا الكبيرة , هو نفس تأثير الل마다 الصغيرة (ديه مقلوب ديه) الل마다 الصغيرة بتزود C
 - OF يعني الدالة بتكون حساسة جدا لاي تغيرات تتعمل من اي قيم جديدة
 - الحالة الثانية تكون C صغيرة
 - ساعتها قيمة A المضروبة في C هتنزد وهنقول كتير قيمة B اللي هو مجموع مربع الثيتات
 - هنا هيحصل العكس , هو نفس تأثير الل마다 الكبيرة واللي بتزود OF و $Bias$ يعني الدالة بتكون حساسيتها اقل
 - وتطبيق ده على الرسم بيكون كدة
 - لو عندي بيانات زي كدة
-
- فلو C كبيرة (الدالة حساسة و عندنا OF) ساعتها لو فيه نقطة زيادة زونها احمر تحت علي اليمين , الخط الفاصل الاسود , هيتتحول فورا للخط البني , لأن الدالة حساسة لاي تغير بينما لو C قليلة , فهتمون حساسيتها اقل , فلو زادت النقطة الشاذة , الخط الاسود هيفضل زي ما هو , او هيميل حاجة بسيطة , و يضحي بنقطة واحدة خارجة , افضل ما يعمل خط مائل كتير , ينفع هنا و يأذينا مع اي بيانات جديدة

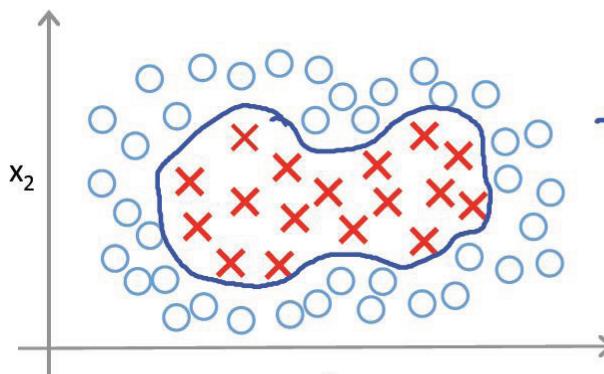


• نتعرف على الكرنيل Kernels

- الكرنيل هي دالة تشابه ، اللي انت بتعطيها مدخلين مع بعض (صورة و كلام مثلا ، او صوت و كلام) ، الدالة بتحددلك مقدار التشابه بينهم وبين بعض
- فانت في مرحلة التدريب ، بتدي الكرنيل ، عدد كبير من الصور و الكلام المرتبط بيها ، فيتعلم و يعمل الخوارزم بينهم
- و بيتم استخدام الكرنيل ، لانه اسهل بكثير ما اتعامل مع عشرات الـ features واللي هتدوخي في حسابها
- يعني الكرنيل هي ببساطة دالة ، فيها حسابات معقدة ، وانا باستدعها بسهولة عشان اديها المدخلات و هي تحسبلي المخرجات

• طب ايه تطبيق الكرنيل في الـ ML :

- لو عندي decision boundary لوي



- ساعتها هعمل دالة فيها اكسات كتيرة مضروبة في بعض زي دي :

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \dots \geq 0$$

- عملت اكسات كتيرة كدة ، عشان تبقى curvy بالقدر الكافي انها تلف حولين النقط
- و هافتفرض ان قيمة الواي الواحد لو كل المعادلة دي اكبر من صفر ، وبصفر لو هي اقل من صفر
- تعالي نشيل قيم اكسات و نحط مكانها افهات كدة

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \dots$$

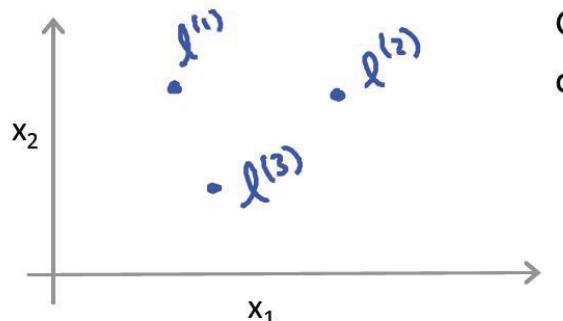
○ بحث

$$f_1 = x_1, f_2 = x_2, f_3 = x_1 x_2, f_4 = x_1^2, \dots$$

○ السؤال بقى ، مفيش قيم بديلة للإفهات (معاملات ثيتات) عشان تكون اسهل شوية من الاكسات المضروبة في بعض ديه ، عشان نتجنب البطئ الرهيب

● يعني هنا مثلا :

Kernel



○ يعني هنا مثلا :

■ هنفترض L_1, L_2, L_3 هي نقط في مجال اكس 1 و 2

■ هنقول إن الإفهات هي عبارة عن مقدار التشابه بين الـ L & X بدالة كرنيل

■ ودالة كرنيل هنا هتكون عبارة عن اكسبونينيشيال لسالب مربع النورم الفرق بين اكس و ال 1 ، علي

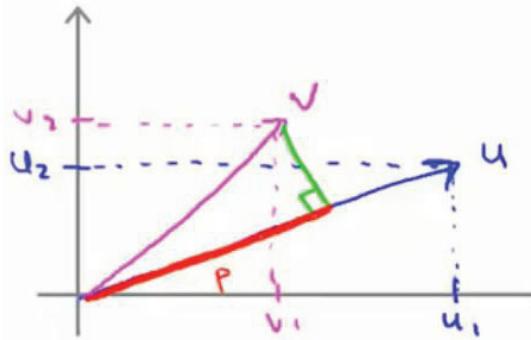
ضعف السيجما تربيع

$$\begin{aligned} f_1 &= \text{similarity}(x, l^{(1)}) = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right) \\ f_2 &= \text{similarity}(x, l^{(2)}) = \exp\left(-\frac{\|x - l^{(2)}\|^2}{2\sigma^2}\right) \\ f_3 &= \text{similarity}(x, l^{(3)}) = \exp(\dots) \end{aligned}$$

\uparrow kernel (Gaussian kernels) $k(x, l^{(1)})$

● عشان نفهم ايه معنى النورم :

○ لو عندي متوجهين ، U & V و عايز ارسمهم



- هنفرض ان الخط الازرق هو U والبني هو الاحمر

$$\sqrt{u_1^2 + u_2^2}$$

يكون

قيمة

الـ

norm

الـ

v

فيثاغورث

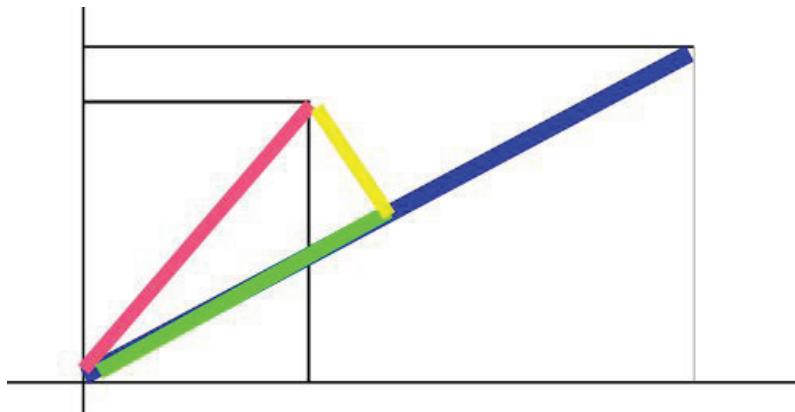
يعني

- قيمة نورم v بنفس الطريقة

- اسقاط الفيكتور v على الفيكتور U هو الخط الاحمر اللي اسمه p

- قيمة $p \cdot \|U\|$ هتساوي بالضبط حاصل ضرب U ترانزبوس في v , يعني $U_1v_1 + U_2v_2$

- عشان نتأكد : تعالى نشوف الرسم ده



- نقول ان الخط الاورق هو U الاكس بتاعه 15 , والواي 8 , بيبقى الخط الازرق 17

- والخط الاحمر هو v اكس 5 و واي 12 , بيبقى الخط نفسه 13

- قيمة زاوية U هي $\tan^{-1} \frac{8}{15}$, يعني 28 درجة

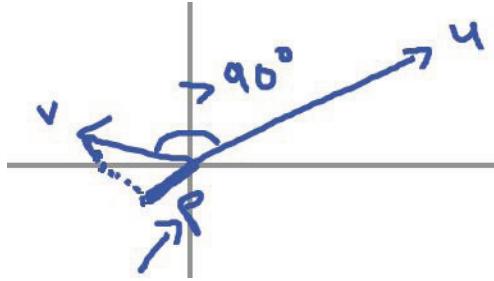
- قيمة زاوية v هي $\tan^{-1} \frac{12}{5}$, يعني 67 درجة

- الزاوية بينهم (قصد الخط الاصفر) هتكون 39

- الخط الاخضر : اللي هو p هو 13 في $\cos 39$, هيساوي 10

- 10 في 17 هيساوي 170 , اللي هو قيمة 5 في 15 في 8 في 12

- و خد بالك , ان قيمة p ممكن تكون سالب , وده في حالة ان الزاوية بين المتجهين اكبر من 90 درجة , هتبقى كدة



- و متنساش ان القيمة اللي جوة الاكسوبونيشيال ، اسمها Gaussian Kernel
 - بينما الدالة دلوقتي بقت كرنيل ، اللي بيجيب العلاقة بين الاكس و الال
-

- طيب عرفنا ايه هو الكرنيل ، هيبي قيمته كام في القانون المستخدم $\|x - l^{(1)}\|^2$
- ه تكون قيمته كالتالي :

$$f_1(x, L^1) = \sum (X_j - L^1_j)^2$$

- يعني مجموع مربعات الفرق بين الاكسات و الالات المناظرة

- وهذا يكون فيه حالتين :

- الحالة الأولى ان قيم X قريبة من قيم L

■ وقتها هيكون الفرق بينهم تقربيا صفر ، واكسوبونيشيال الصفر تساوي تقربيا 1

- الحالة الثانية ان قيم X بعيدة عن قيم L

■ وقتها هيكون الفرق بينهم كبير ، واكسوبونيشيال سالب الرقم الكبير تساوي تقربيا 0

- مع مراعاة ان فيه قيمتين L اللي هي اكس 1 و 2 ، أما L فهي ليها قيم داخلية ، يعني L^1 فيها قيم داخلية هي (L_1^1, L_1^2, L_1^3) وهكذا

- ففي المعادلة فوق عشان اجي f_1 هنضرب اكس 1 في L_1^1 و اكس 2 في L_1^2 وهكذا

- خد بالك ان L معروفة ، بينما الاكسات مجهرولة ، يعني انا جرب كل الاكسات و برسم بيها

- طيب ماذا عن السيجما ، هنشوفها حالا
-

- تعالى نشوف الكلام عملي :

- لو عندي مثل فيه اتنين اكس و اتنين ال ، زي كدة

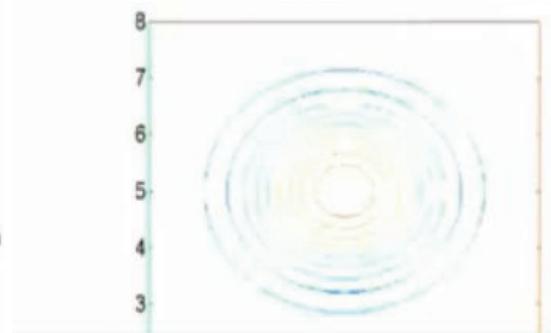
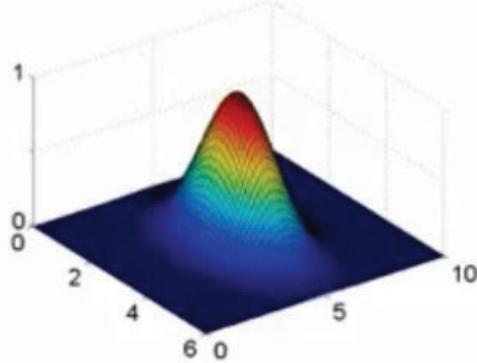
Example:

$$l^{(1)} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}, \quad f_1 = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right)$$

- هنبدا نعرض ، بس محتاجين قيم لسيجما ، فنعمل الحالة الاولى ان سيجما تربع بتساوي 1

• ساعتها الرسم هيكون كدة

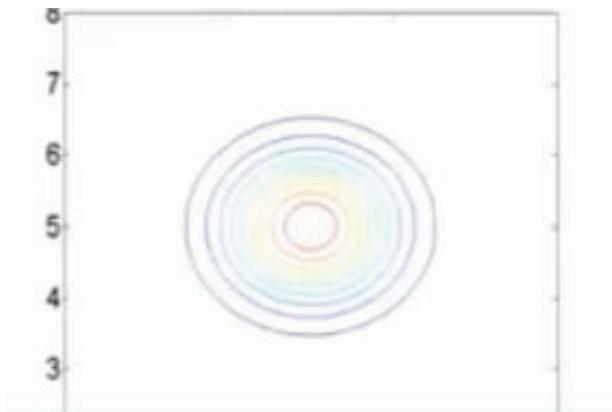
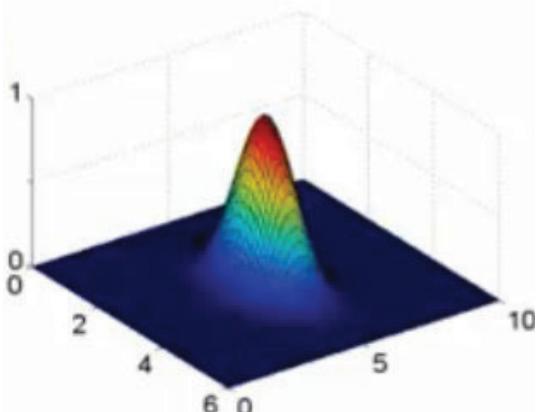
$$\sigma^2 = 1$$



- الشكل اليسير بيديني رسم ثلاثي الابعاد ، لكل قيم اكس 1 و اكس 2 (المحاور الافقية) مع اف (الرأسي) بحيث كل نقطة علي الشكل المجمد ، هي قيمة اف ، مع قيمتي اكستين
- الرسم اليمين نفس القيم ، بس شكل ثاني الابعاد ، بحيث كل دائرة بتعملني قيمة معينة للاف (مثلا الدائرة الثالثة قيمتها 0.4) و ده لاي قيمة اكس 1 و اكس 2 تقع عليها
- هنلاحظ (في الشكل اليسير)، ان اعلي قيمة لاف ، لما كان $X_1 = 5$ و $X_1 = 3$
- ده لأن قيمة الالتين تساوي 3 و 5 ، فلما تساوي الاكستين مع الالتين ، الفروق بقت بصفر ، والاكسوبونينشيل بقى بـ 1 زي ما شفنا من شوية
- وكل ما تبعد الاكستين عن الالتين كل ما تقل قيمة اف لما بتتوصل لصفر

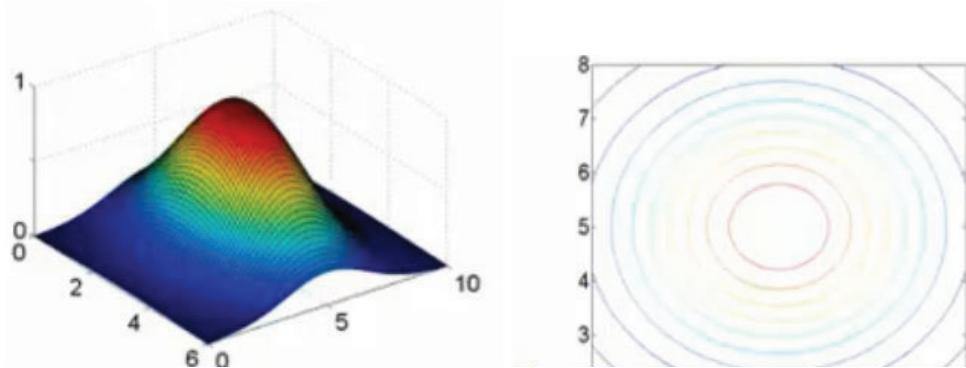
• طيب في حالة سيجما تربع تساوي نص

$$\sigma^2 = 0.5$$



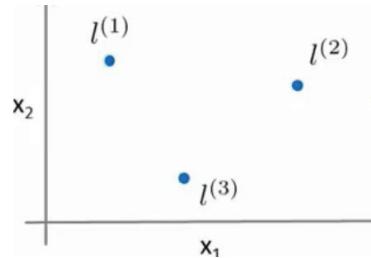
- لما السيجما تربع تقل قيمة الكسر $\frac{\|x - l^{(1)}\|^2}{2\sigma^2}$ هتزيد ، يتضرب في سالب هيبيقي أقل بكثير ، نعمله اكسبو ، هيقل قيمته ، وده للجميع ، فتلقي ان الجراف كله نزل تحت ، والدوائر اللي على اليمين صغرت

- طيب في حالة سيجما تربع تساوي رقم اكبر من 1



- لما كبرنا قيمة سيجما ، هنلاقي الكسر قيمته هتقل لكسور عشرية ، اضرب في سالب هتقرب الارقام السالبة للصفر ، فلما اعملها اكسبو ، هتزيد القيم و تقرب من -1 ، فالجراف كله هيتفخ لفوق
- فبشكل عام لاحظنا ان ، كل ما تقرب الاكتين من الالتين ، كل ما الإف تقرب من 1 ، وكل ما تبعد تقرب اف من الصفر ، عشان كدة اسم الكرنيل هو similarity لأنه بيقيس مدى تشابه القيم الاولى مع الثانية

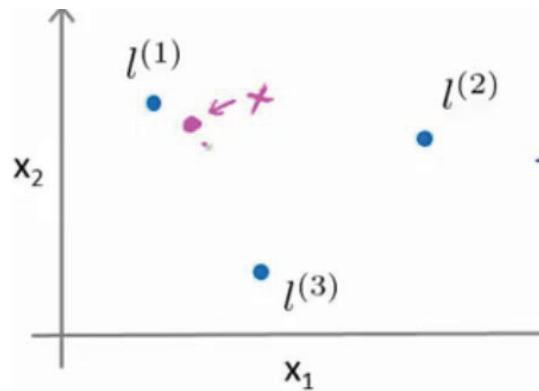
- عايزين نشوف بقى ازاي بيتتم عمل التقسيمات classifications
- نفرض عندي مجموعة من النقاط زي كدة :



- عندي 3 نقط إل 1 و 2 و 3 ، وعمل ليهم قيم ثباتات مع الإفات كدة $\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$

○ نفرض ان قيم ثباتات كالتالي :
 $\theta_0 = -0.5 , \theta_1 = 1 , \theta_2 = 1 , \theta_3 = 0$

- دلوقتي لقيت النقطة الحمراء ديه :

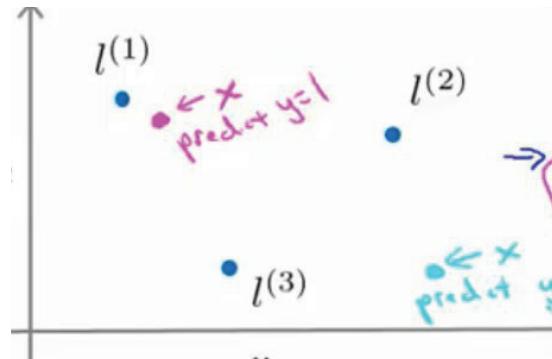


- عشان اكس قريبة من ال 1 ، ه تكون اف 1 تساوي 1 تقريبا ، بينما الافتين الثانيين بصفرين
- دلوقتي نعرض

$$\begin{aligned}
 & f_1 \approx 1, \quad f_2 \approx 0, \quad f_3 \approx 0. \\
 & \Theta_0 + \Theta_1 \cdot 1 + \Theta_2 \cdot 0 + \Theta_3 \cdot 0 \\
 & = -0.5 + 1 = 0.5 > 0
 \end{aligned}$$

- هنلاقي ان القيمة النهائية ه تكون نص ، يعني الواي النهائي ه تكون بـ 1 ، وده منطقي لأن اكس قريبة من ال 1
- ولو فيه برضه نقطة جنب ال 2 ، ه تكون قيمة الواي النهائي بـ 1

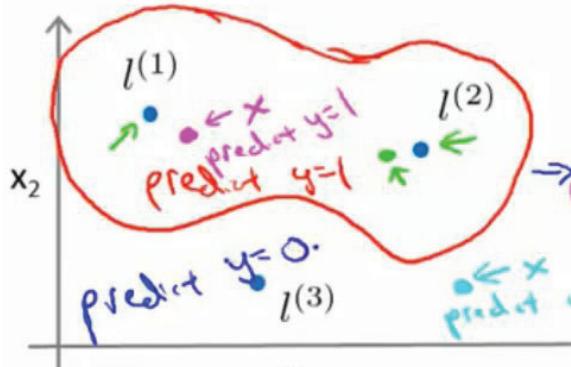
● تعالى نشوف النقطة الخضرا بعيدة عن كل الالات



- نيجي نعرض :

$$\begin{aligned}
 & f_1, f_2, f_3 \approx 0 \\
 & \Theta_0 + \Theta_1 \cdot 1 + \dots \approx -0.5 < 0
 \end{aligned}$$

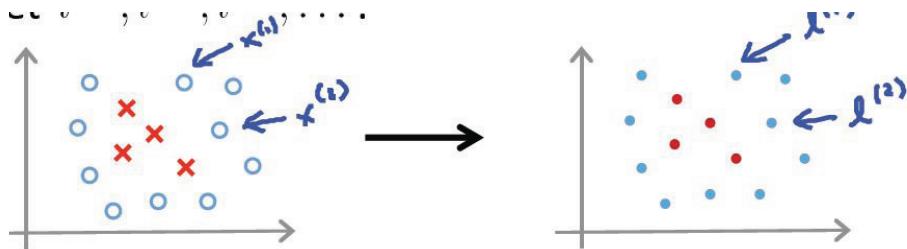
- هنلاقي ان القيمة النهائية بقت سالب نص ، و ده معناه ان الواي بصفر ، وده منطقي لأن النقطة ديه بعيدة عن كل القيم المتاحة
- في النهاية هتلاقي اوتوماتيك ان كل النقاط القريبة من ال 1 و ال 2 ، بقت بقيمة 1 ، وان كل النقاط بعيدة عنهم بقت بصفر



- يعني احنا كدة عملنا classification هايل ، و ممكن يتعمل خط يفصل بين قيم 1 و 0 ، اللي هي بعيدة و قريبة من النقط المطلوبة

• ماذا عن النقاط L

- عرفنا من المرة اللي فاتت ، اني اعتمادا علي وجود نقاط L اللي بتكون معطاة ، باقدر احدد مدي وجود احد نقاط الاختبار قريبة مني ولا بعيدة ولا ايه ، لغاية لما اوصل اني اوصل خط يجمع النقاط المطلوبة مني مع بعض ، وتفصلها عن النقاط بعيدة عنني
- طيب ماذا عن نقاط L نفسها ، هل هي معطاة ؟ ؟
- الحقيقة لا ، نقاط L هي نفسها نقاط الاختبار ، يعني انا بحوال كل نقطة معطاة عندي (الرسم اليسار) لنقاط L (الرسم اليمين) ، بحيث كل واحدة فيهم تعتبر guide في حد ذاتها



● نيجي هنا بقى لخطوة المقارنات :

- احنا بيكون معطي عندنا مدخلات و مخرجات : اكسات و وايات بالشكل ده :
- $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$
- خد بالك ان الاكس ممكن تكون فيكتور ، يعني جوة كل امس فيه اكس 1 و اكس 2
- زي ما قلنا هنعمل مساواة الالات مع الاكسات
- $$l^{(1)} = x^{(1)}, l^{(2)} = x^{(2)}, \dots, l^{(m)} = x^{(m)}$$
- دلوقتي هنعمل دوال التشابه الإفات ، هنقول ان دالة F_1 هي التشابه بين كل الاكسات ، و بين L يعني كدة

$$\begin{aligned} f_1^{(i)} &= \overline{\sin(x^{(i)}, L^{(i)})} \\ f_2^{(i)} &= \sin(L^{(i)}) \\ &\vdots \\ f_m^{(i)} &= \sin(x^{(i)}, L^{(m)}) \end{aligned}$$

- كل F فيهم تساوي افات صغيرة جواها
- معناه ايه :

ان الد F_1 نفسها هي فيكتور ، تساوي افات صغيرة F_1^1 ، F_1^2 ، F_1^3 ، F_1^4 ، كل افية فيهم هتكون كالتالي :

- $F_1^1 = \text{similarity}(X^1, L^1)$
- $F_1^2 = \text{similarity}(X^2, L^1)$
- $F_1^3 = \text{similarity}(X^3, L^1)$
-

وكان F_1 معناها مجموع الافات ، مدي تشابه الإكسات (قيم اكس 1 ، اكس 2 ، وهكذا) من إل 1 وكذلك ، عشان اجيب F^5 مثلا ، بجيبي قيمة L^5 واقارنها بكل الإكسات من واحد لآخر والإكسات وهي ماشية ، هتخبط في نفس الد L بتاعتتها (X^5, L^5) مثلا ، فديه ه تكون بوحد اكيد وبالتالي القيمة النهائية للـ F_1 ، ه تكون فيكتور الافات ، اللي من واحد لـ m اللي هو عدد العناصر والقيمة النهائية للـ F هي محصلة الإفات الكبيرة F_1, F_2, F_3, F_4, F_5 ، متتساش اننا بنضيف F_0 دايما الي بتكون قيمتها بوحد ، فعدد الفيكتور $m+1$

• وبالتالي المعادلة النهائية ه تكون :

- حاصل ضرب الثيتات في الافات اللي هو ده

$$\theta_0 f_0 + \theta_1 f_1 + \dots + \theta_m f_m$$

- ممكن نلخصه يبقى كدة

$$\theta^T f \geq 0$$

- ساعتها المعادلة النهائية اللي كانت كدة :

$$\min_{\theta} C \sum_{i=1}^m \left[y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{j=1}^n \theta_j^2$$

- هتبقي كدة :

$$\min_{\theta} C \sum_{i=1}^m y^{(i)} \text{cost}_1(\theta^T f^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T f^{(i)}) + \frac{1}{2} \sum_{j=1}^n \theta_j^2$$

○ لاحظ ان تم استبدال $\theta^T f^{(i)}$ بـ $\theta^T x^{(i)}$

$$\cdot \frac{1}{2} \sum_{j=1}^n \theta_j^2$$

● طيب تعالى نتكلم شوية عن القيمة الأخيرة :

- لاحظ اننا بنجمع الثيتات من 1 , مش من صفر , لأن ثيتا صفر اللي هي بتساوي 1 مش بنعملها regularization
- محصلة ضرب الثيتات تربيع , معناها تربيع حاصل ضرب (ثيتا 1 في ثيتا 1) زائد تربيع حاصل ضرب (ثيتا 2 في ثيتا 2) وهكذا لآخر واحدة

○ فممكن نعملها بالصيغة $\Theta^T \Theta$ والتي هتدينا ماتريكس 1 في 1 , بنفس القيم , طبعا مع تجاهل قيمة ثيتا صفر اللي بتساوي 1

○ لكن هنضطر اننا نضرب في معامل M (يختلف عن سمول m اللي هي عدد العينة) لأن المعامل ده بيقدر يضبط الارقام لما يكون عدد العينة كبير , فوق العشر الاف مثلا , فه تكون عاملة كدة $\Theta^T M \Theta$

● ماذا عن المعامل C

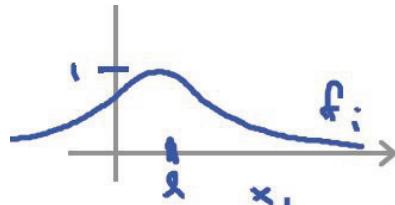
○ احنا فاكرين قبل كدة قلنا ان زيادة الـ C بتزود الـ OF و الـ Bias و بتقلل الـ Variance يعني الدالة بتكون حساسة اكتر

○ كمان تقليل الـ C بتزود الـ UF و الـ Bias و تقلل الـ Variance يعني بتقلل الحساسية

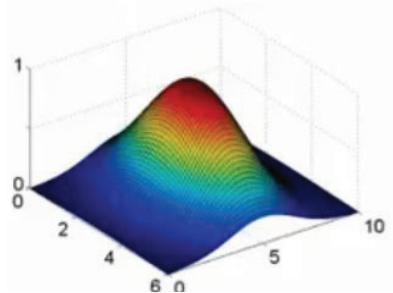
$$\frac{\|x - l^{(1)}\|^2}{2\sigma^2}$$

● طيب ماذا عن السيجما اللي موجودة هنا

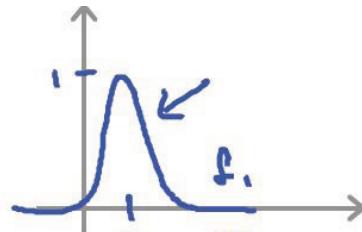
○ السيجما تربيع الكبيرة , بتعمل عكس الـ C يعني بتزود الـ UF و الـ Bias و تقلل الـ Variance , و هتكون زي الرسمة دي



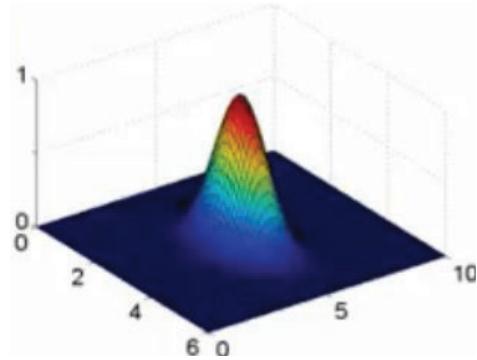
○ بحيث نشوف ان قيمة X لما تساوي L هي واحد , لكن اللي قبلها و اللي بعدها ماشية بانحراف كبير , وتتنوع اقل (ناعمة شوية) والتي تفكروا بيها



- بينما السيجما تربيع الصغيرة ، بتقليل الـ UF و الـ Bias وتزداد الـ Variance ،



- وبالتالي ، هناقي ان برضه قيمة X لما تساوي L هي بوحد لكن اللي قبلها و بعدها قيم اقل ، فالقفز عنيف شوية ، وده اللي يعكس الـ OF و التنويع العالي . واللي تفكربنا بديه



-
- استخدام الـ SVM
 - لأن هناك عشرات المبرمجين و الباحثين قاموا ببرمجة مكتبات للـ SVM فلا ينصح باعادة كتابة كود لحسابها ، بل باستخدام المكتبات فقط
 - من افضل المكتبات لها هي : **liblinear & libsvm** :
 - لكن حتى مع استخدام مكتبات جاهزة ، يظل فيه دور للمبرمج في تحديد حاجات زكي :
 - تحديد قيمة الـ C
 - تحديد العوامل اللي في دالة الكرنيل
 - و هنا بيكون فيه اكتر من اختيار ، بناء علي نوع البيانات عندك :

○ أولاً الـ Linear Kernel

- و المقصود بيه ان فعلياً مش هيتم استخدام اي كرنيل ، لأن الكرنيل الخطى ، معناها مفيش اي كريفات ، مفيش اي اكسات مضروبة في بعض ، يعني Linear Classification عادي من غير اي كرنيل ، وده ابسط انواع الـ SVM و اضعفها
- يعني ساعتها $\theta^T x > 0$ يعني حاصل ضرب الثيتات في الاقسات بشكل مباشر ، وده مفيهوش اي افات او الات امتي بيستخدم ده ؟ في حالة عندي عدد features كتير (n) وعدد بيانات قليل (m) ، وقتها انت لو استخدمت كرنيل كبير ، الخوارزم هي عمل OF و ديه هتبقي مشكلة ، لأن عدد البيانات قليل بالفعل ، فالافضل استخدم كرنيل خطى ، هيكون مناسب اكتر

○ ثانياً الـ Gaussian Kernel

- $f_i = \exp\left(-\frac{\|x - l^{(i)}\|^2}{2\sigma^2}\right)$ اللي هو :
- و هنا لازم نفترض ان قيمة السيجما هتفرق معانا كتير ، لو مربعها كبير، هيزيد الـ UF و الـ Bias ، ويقل الـ Bias ، ولو مربعها قليل هيزيد الـ OF و الـ Variance ، ويقل الـ Variance طيب امتي باستخدمنها
- غالباً لما بيكون عندي عدد features قليل (n) وعدد بيانات كبير (m) ، فبيكون عايز دالة مرنة شوية بحيث تقدر تغطي لي كل النقط المطلوبة وبيكون شكلها كدة :

```
function f = kernel(x1, x2)
    f = exp( $\frac{-\|x1 - x2\|^2}{2\sigma^2}$ )
return
```

- وكان مطلوب اني اكتب كود ، ياخذ بيانات اكس 1 واكس 2 ، واحدد قيمة السيجما تربيع ، وادخلهم في دالة gaussian وهي هتجibli الـ f

○ طيب خد بالك من حاجة شديدة الأهمية ، و هي الـ scaling

- عشان نفهمها تعالي شوف هنا :
- لو ها عمل كرنيل جوسيان في الفروق بين L & X كدة :

$$\|x - l\|^2$$

■ هيقي الفاك بتاعه كدة

$$(x_1 - l_1)^2 + (x_2 - l_2)^2 + \dots + (x_n - l_n)^2$$

■ احظ ان X_1 و X_2 هي الـ features

■ فلو عندنا X_1 هو مساحة البيت و X_2 عدد الغرف مثلا , فممكن X_1 يبقى قيمته 1500 قدم مربع , بينما

X_2 تساوي 3

■ لأن L ثابتة , فمثلاً تساوي 20 , فهتلاقى $L - X_1$ رقم كبير بينما $L - X_2$ رقم صغير

■ فلما اربع الارقام ديه , واجمعها , هلاقى ان تأثير قيم عدد غرف البيت في الكرنيل قلت جدا , وكان مساحة البيت بقت هي العامل المؤثر

■ وطبعاً ده ينطبق على باقي الحاجات , فلو فيه قيمة كبيرة و باقي صغير , فكأن الكبيرة بس هي اللي بتاثر و الباقي ملوش لازمة

■ عشان كدة لازم اعمل scaling اللي هو اعادة ضبط القيم عشان تكون مطبوبة مع بعض

- طيب حاجة مهمة , مش اي كرنيل هبقى متاكد انه هيعملني كلاسيفيكاشن مطبوبط , لأن اي كرنيل قبل ما استخدمه لازم يكون متبع ما يسمى نظرية ميرسر Mercer's Theory

- صحيح ان اشهر اتنين كرنيل (Linear & Gaussian) بيتبعدو ميرسير , لكن مش هما لوحدهم , فيه كرنيلات تانية بتتبعها , مش مشهورة زي مثلا :

■ كرنيل Polynomial Kernel

● واللي بتكون صيغتها كدة :

$$(X^T L + C)^D$$

- يعني اكس ترانسبوز , مضروبة في الـ L , زائد ثابت , وكله مرافق للأس D

- فممكن تكون كدة $(x^T l + 1)^3$ أو كدة $(x^T l + 1)^4$ أو كدة $(x^T l + 1)^5$

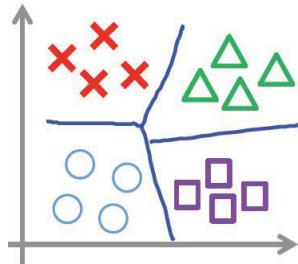
$$(x^T l + 1)^{④}$$

- ومع إنه مش مشهور , لكن بيستخدم احياناً لحل بعض المشاكل

■ أنواع تانية من الكرنيل اللي نادراً ما تستخدم زي

(String Kernel , Chi-Square , Histogram Intersection)

- طيب ماذَا في حالة وجود اكتر من تقسيمة , يعني مش هقسم الحاجات لنوعين بس , لكن لثلاث او اربع انواع



- بالفعل اغلب دوال الكرنيل ، بيكون فيها امكانية ، انك تقسم اكتر من قسم ، وانت بتختار العدد ساعتها ها عمل موضوع الـ SVM اكتر من مرة لغاية لما اعرف اقسامهم كلهم بشكل سليم ، وده هيكون عن طريق تكنيك one vs all method اللي هو باعمل كذا ثيتا ، كل ثيتا فيهم بتختار الـ y بقاعدتها وتجيبها وتجاهلباقي

$y = i$ from the rest, for $i = 1, 2, \dots, K$, get $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(K)}$
 Pick class i with largest $(\theta^{(i)})^T x$

- فهنا مثلا ، هتتيجي ثيتا 1 ، تختار $y=1$ ، وتسيبباقي ، وتحتيجي ثيتا 2 تجيب $y=2$ وتسيبباقي و هكذا

● أخيرا ، نختم بمقارنة بين استخدام الـ SVM ، والـ Logistic Regression

- بشكل عام ، لو كان عدد الـ features اللي هي n اكبر من عدد العوامل m (يعني مثلا n بعشر الاف ، و m من 10 لـ 1000) نستخدم **linear kernel svm** او حتى **logistic regression** بس بنظام
 - لو n صغيرة (الغاية الف) ، و m متوسطة (الغاية 10 الاف) نستخدم **SVM with Gaussian**
 - لو n صغيرة (الغاية الف) ، و m كبيرة (فوق الـ 50000) ساعتها اعمل **features** جديدة و نستخدم **logistic regression** او **linear kernel svm** بس بنظام
-