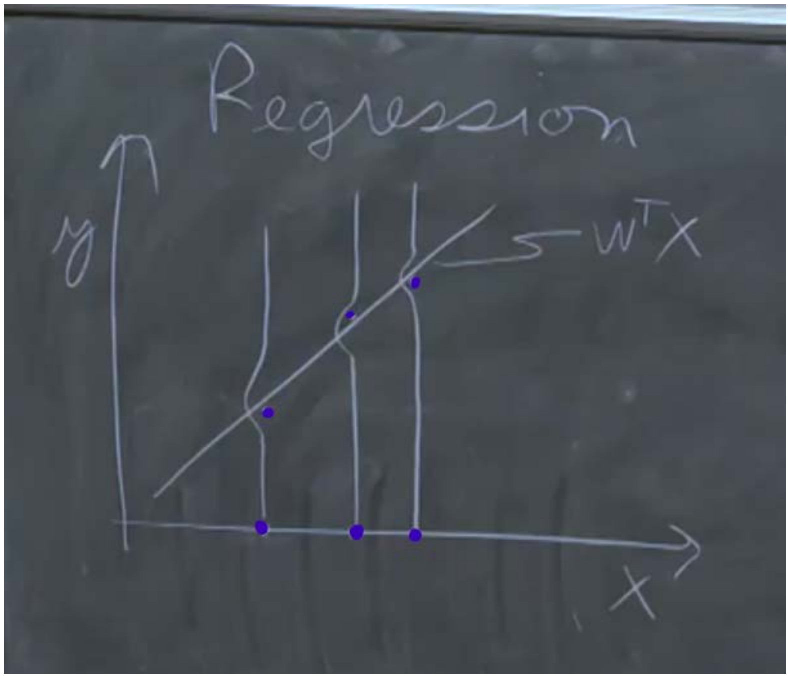
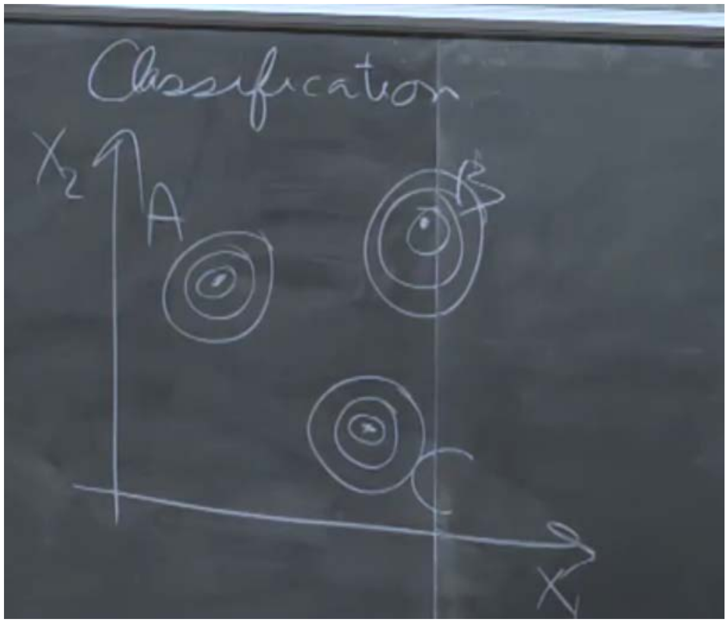
بسم الله الرحمن الرحيم

محاضرة ال Mixture of Gaussians .. ده برضو كلاسيفيكيشن يعني ... بس باستخدام الجاوسيان مكسشيرز ... وده model في family اسمها Generative models .. احنا اشفنا بالنسبه لل regression شفنا statistical linear regression model .. دلوقت هنعمل حاجه زي كدا لل classification .. تعال الاول نشوف يعني ايه Generative models .. في ال regressionاحنا كان هدفنا نقلل الدستانس بالنسبه للخط .. عندك شوية داتا وبتعمل فيتتنج كيرف .. ومن ناحية الأوبتيمازيشن .. هدفك تقلل المسافات .. في الريجريشن .. القصه كلها ان ممكن ن susupect that the underlying function is linear و هي ديه التروو و ديه اسمها wTx .. و الإنبوتس هي الإكس أكسيس و الأوتبوتس هي الواي أكسيس .. لو الفانكشن كانت صحيحه .. يبقا الخط هيعدي علي النقط كلها . بس الحقيقه ان الداتا اللي احنا لاحظناها اتقاست من شوية سنسورز و السنسورز مش جامده جداً يعني وبتضيف للنويز .. فالنويز ديه ممكن يبقا ليها جاوسين ديستربيوشن .. فلو عندك شوية داتا بوينت إكس .. حتي لو انا عارف ال true underlying function .. اللي انا would likely observe هو إن الاوتبوت sampled from a Gaussian distribution .. فاللي اترسم هو Gaussian علي ال vertical line و فعلياً انت تقدر تجيب measurement at the line ... فانا لو عاوز أجينريت شوية داتا .. ه sample from the Gaussian .. وهجيب نقطه .. مش هتبقا علي نفس الخط .. بس هتبقا قريبه من الخط ... فهنا نقدر نقول ان ده generative model عشان انا اقدر استخدم الموديل عشان ا simulate ال creation of data set .. طب ايه الهدف من حاجه زي كدا .. في أبلكيشنز كتيره هدفها حاجه زي كدا انها ت generate data .. وهنا بتحتاج generative model .. فهم يقدرو يجينيريتوا داتا زي اللي عندك في الترينتج سيت .. بس ففي الصوره عندنا هي بس حاجه بسيطه بتوضح الفكره يعني



في حالة الكلاسيفكيشن بقا .. عندك مثلاً 3 كلاسيز .. A, B and C ... في الحاله ديه .. استخدم أول أكسيس عشان ت represent x1 اللي هو أول كومبوننت .. و الإكس 2 علي الفيرتيكال لاين .. خد بالك ان الأكسيس مختلفه .. هنا الأكسيس انبوت انبوت .. الافتراضيه اللي عندي ان لو عندي كلاسيفيكيشن .. لما بقيس الإنبوت هو مش هيبقاaccurate .. بس بعد كدا انا هتوقع ان كل داتا بوينت في كلاس سي مثلا .. centered عند النقطه بتاعت C .. بس ممكن يكون في Gaussian distribution اللي ممكن ي correspond to noise in the measurements of the input .. وبعدين كنتيجه لكدا عندك داتا بت follow Gaussian distribution centered بالطريقه اللي الدكتور راسمها كدا ..



بناءً علي الفكره ان الداتا بتاعتنا بتبقا في مناطق مختلفه .. ولكل كلاس في ديستريبيوشن .. انت تقدر تبني a probabilistic model .. اللي هو تقدر تقول فيه .. لما مكنتش ببص علي أي داتا بوينت انا مجرد بخمّن ايه هو الكلاس بتاع الداتا بوينت اللي جايالي ديه فساعتها بيبقا عندي prior distribution وهنسميه probability of c و ال c هنا هي الكلاس يعني بس ديه prior يعني قبل ما نبص علي أي داتا بوينت .. أول ما هيكون عندك داتا بوينت .. ايه احتمالية انك تختار نقطه من الكلاس C فهيبقا عندك class conditional distribution اقدر استخدمو عشان أgenerate points from that class ..

انت كدا تقدر تحسب ال Posterior باستخدام Bayes theorem .. وده ال prediction for the class of this data point .. في السلايد ديه لازم تاخد بالك .. ان ده مش Bayesian learning احنا بنستخدم bayes theorem عشان نinfere و نestimate ايه هي البروبابلتي بتاعت الكلاس ل given data point .. انما لما كنا بنتكلم علي Bayesian learning .. ده كان في ال context بتاع اننا عندنا شوية بارمترز بتاعت الموديل وبعدين بحاول اني أ estimate البارمترز بتاعت الموديل باستخدام bayes theorem ... فهنا احنا متكلمناش خالص علي البارمترز ... كل اللي عملناه اننا اتكلمنا علي ال inference عشان تحسب ال posterior probability of a class ..

Probabilistic Generative Model

- $\Pr(C)$: prior probability of class C
- $\Pr(\boldsymbol{x}|C)$: class conditional distribution of $\boldsymbol{x}$

- Classification: compute posterior $\Pr(C|\boldsymbol{x})$ according to Bayes' theorem

$$\Pr(C|\boldsymbol{x}) = \frac{\Pr(\boldsymbol{x}|C)\Pr(C)}{\sum_C \Pr(\boldsymbol{x}|C)\Pr(C)}$$
$$= kPr(\boldsymbol{x}|C)\Pr(C)$$

طيب في الموديل ده هنحط شوية assumptions .. عندنا finite number of classes وهي حاجه categorical .. فبالتالي ال natural type of prior نقدر نستخدمو هو ال multinomial distribution .. وده distribution over finite number of possible outcomes that are categorical .. زي مثلا .. عندك coin اقلب الكوين .. تلاقي head or tail .. ده هيديك bernolli distribution .. انما لو عملت generalization هتلاقي عندك dice .. فتقدر تستخدم ال multinomial distribution ... هنا هنستخدم الرمز pi_k ... عشان نdenote the probability ofclass k وده مجرد رقم ما بين الصفر والواحد ..

هنفترض كمان ان عندنا داتا .. والداتا ديه ليها d dimensions .. فدلوقت انت عاوز يبقا عندك a class conditional distribution يقلنا :

For a given class, where the data is likely to be?

وهنا هنحط الافتراض ان ال class conditional distribution is a Gaussian distribution .. تمام .. فانا عندي الكلاس بتاعي .. P(x|Class c)

وده اللي احنا رسمناه علي البورد .. احنا كمان هنفترض ان كل Gaussian distribution هيبقا ليه نفس ال covariance matrix اللي هي capital sigma .. تحت الفرضيه ديه .. هنقول العلاقه اللي مكتوبه ديه .. والافتراض ده مهم جداً .. خليك فاكره ..

Assumptions

- In classification, the number of classes is finite, so a natural prior $\Pr(C)$ is the multinomial
$$\Pr(C = c_k) = \pi_k$$
- When $\boldsymbol{x} \in \Re^d$, then it is often OK to assume that $\Pr(\boldsymbol{x}|C)$ is Gaussian.
- Furthermore, assume that the same covariance matrix $\boldsymbol{\Sigma}$ is used for each class.
$$\Pr(\boldsymbol{x}|c_k) \propto e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_k})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_k})}$$

دلوقت بعد ما عرفنا ال prior و ال likelihood .. فال prior is a multinomial و ال likelihood is a Gaussian distribution .. نقدر نحسب ال posterior to do inference... وده مش Bayesian learning ... ده inference to estimate the property of each class given a data point ..

Posterior Distribution

$$\Pr(c_k|\boldsymbol{x}) = \frac{\pi_k e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_k})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_k})}}{\sum_k \pi_k e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_k})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_k})}}$$

$$= \frac{\pi_k e^{-\frac{1}{2}\left(\cancel{x^T\Sigma^{-1}x}-2\mu_k^T\Sigma^{-1}x+\mu_k^T\Sigma^{-1}\mu_k\right)}}{\sum_k \pi_k e^{-\frac{1}{2}\left(\cancel{x^T\Sigma^{-1}x}-2\mu_k^T\Sigma^{-1}x+\mu_k^T\Sigma^{-1}u_k\right)}}$$

Consider two classes c_k and c_j

$$= \frac{1}{1+\frac{\pi_j e^{\mu_j^T\Sigma^{-1}x-\frac{1}{2}\mu_j^T\Sigma^{-1}\mu_j}}{\pi_k e^{\mu_k^T\Sigma^{-1}x-\frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k}}}$$

Posterior Distribution

$$= \frac{1}{1+e^{-\left(\mu_k^T-\mu_j^T\right)\Sigma^{-1}x+\frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k-\frac{1}{2}\mu_j^T\Sigma^{-1}\mu_j-\ln\frac{\pi_k}{\pi_j}}}$$

$$= \frac{1}{1+e^{-(w^T x+w_0)}}$$

where $\boldsymbol{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_j)$

and $w_0 = -\frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \frac{1}{2}\boldsymbol{\mu}_j^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j + \ln\frac{\pi_k}{\pi_j}$

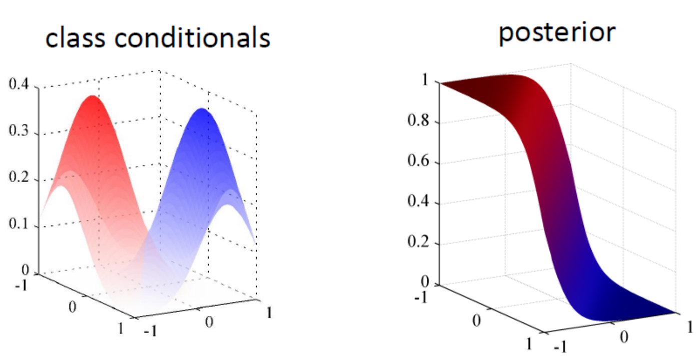
كدا بقي عندنا ال posterior في فورم كويسه .. هل حد عارف ايه الفانكشن ديه .. الفانكشن ديه مستخدمه كتير في النويرال نتورك .. هوب ديه السجمويد فانكشن ..

Logistic Sigmoid

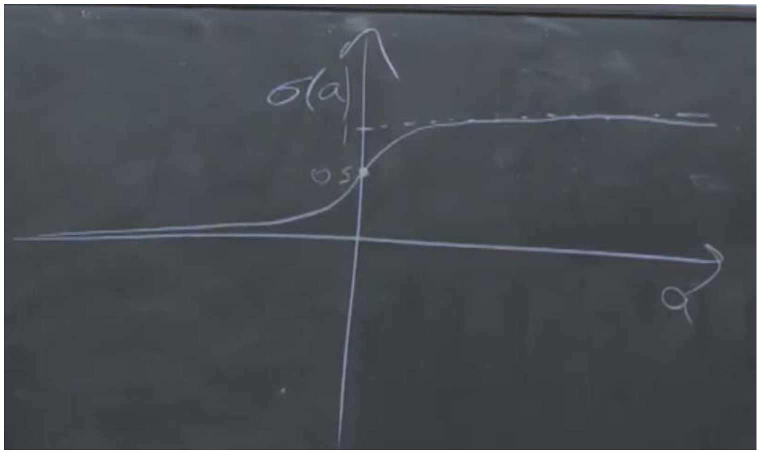
- Let $\sigma(a) = \frac{1}{1+e^{-a}}$
  → Logistic sigmoid
- Then $\Pr(c_k|\boldsymbol{x}) = \sigma(\boldsymbol{w}^T\boldsymbol{x} + w_0)$
- Picture:

الفانكشن ديه مهمه ... هنرسمها ..

Logistic Sigmoid

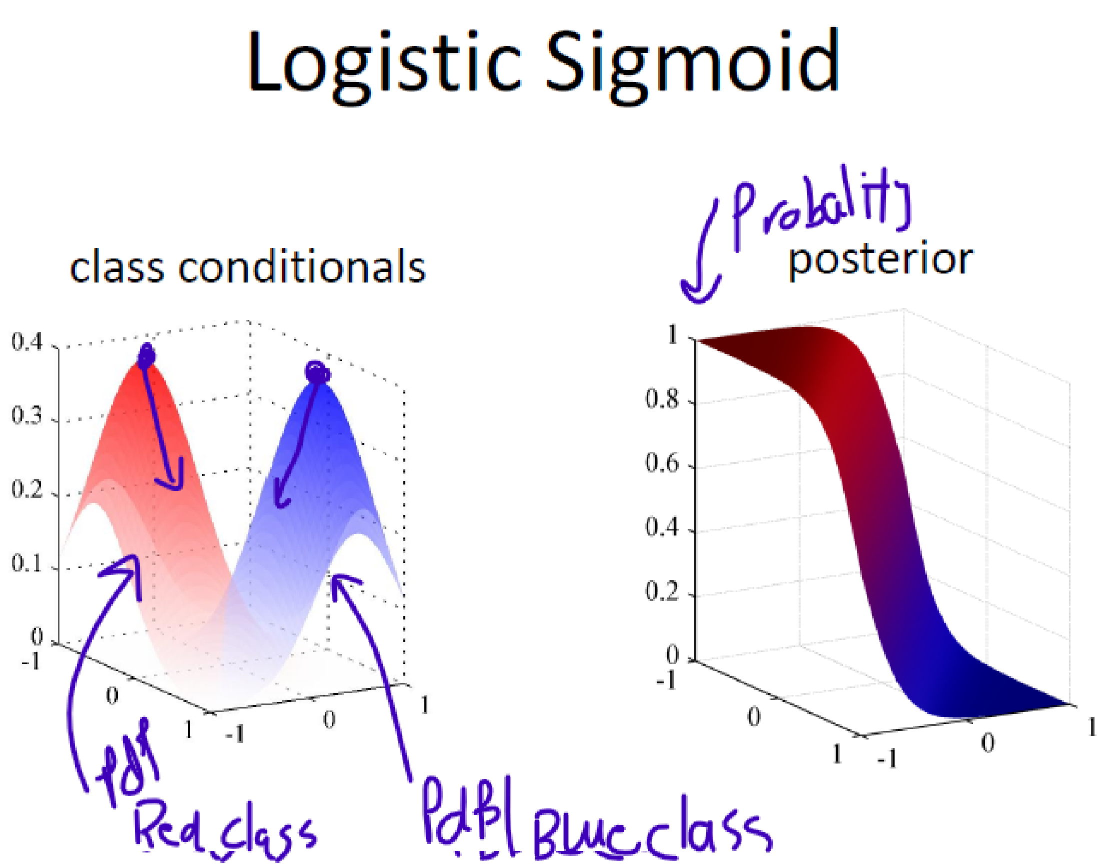


الإنبوت هو ريل نامبر .. والأوتبوت ما بين الصفر والواحد



وعشان كدا هي مشهوره في ال Neural networks .. هنا بتحشر الأوتبوت ما بين الصفر والواحد عشان تقدر تتخيلو علي انو probability ..

طيب تعال نشوف الصوره اللي في لسلايدز الي فيها الرسومات الملونه .. انت عندك اول كيرف في كيرف أحمر .. ده أول كلاس .. و ده ال Gaussian distribution للداتا اللي جايه منو .. وفي كلاس تاني هو الازرق ده ... لو هتيجي تحسب ال posterior .. هتلاقي ان ليها احتمالية اعلي للون الأحمر مثلاً يعني ..



دلوقت احنا نقدر نعمل prediction .. لما بنحسب ال posterior يبقا عندنا distribution over the classes و بالتالي اقدر اعمل بريديكشن .. في شوية أبلكيشنز احنا مش بنعوز نرجع ايه هي ال probability for every class بس اللي احنا عاوزينو ... اهو هو ده الكلاس اللي انا قدرت اعمله prediction حتي لو انا مش واثق ف نفسي .. وحالياً لو انت عاوز تختار كلاس ... الطبيعي انك تقول اختار الكلاس اللي ليه بروبابلتي اعلي من ال0.5 ..

طيب دلوقت احنا لما هنعمل predictions زي كدا .. احنا بنحتاج نعرف ال boundary ما بين ال 2 كلاسيز .... عند ال K = 1 .. عندك الدايجرام اللي اسمو voronoi diagram اللي بيديك الباونديرز لما بتستخدم إيقليدين ديستنس .. لما بتستخدم ال mixture of Gaussian .. الكلاس باوندري .. هو ان الباوندري محدد لما بتكون احتمالية كل كلاس قد بعض .. فهتبقا عند ال 0.5 .. بعد شوية simplifications .. بنوصل ان عندنا linear separator ....

Prediction

$$best\ class = argmax_k \Pr(c_k|\boldsymbol{x})$$
$$= \begin{cases} c_1 & \sigma(\boldsymbol{w}^T\boldsymbol{x} + w_0) \geq 0.5 \\ c_2 & \text{otherwise} \end{cases}$$

Class boundary: $\sigma\left(\boldsymbol{w}_k^T\bar{\boldsymbol{x}}\right) = 0.5$

$$\Longrightarrow \frac{1}{1+e^{-\left(\boldsymbol{w}_k^T\bar{\boldsymbol{x}}\right)}} = 0.5$$
$$\Longrightarrow \boldsymbol{w}_k^T\bar{\boldsymbol{x}} = 0$$

$\therefore$ **linear separator**

لما بتكون السجمويد ب 0.5 .. لو بصيت علي الإكس أكسيس .. هنا عندها الإنبوت بتاع السجمويد بيبقا بصفر .. وبكدا انت فهمت انت ازاي وصلت للإكسبرشن اللي هو wTx = 0 .. عشان ال activation = 0 .... وده linear separator ... عشان هو linear expression ... طيب لو عندنا أكتر من 2 كلاسز .. هنعمل نفس الحسابات .. وهنفترض ان كل كلاس ليه class conditional distribution اللي هو Gaussian و ليهم نفس ال covariance matrix اللي هي Capital sigma ... فبدل ما بنقسم البسط علي المقام هنسيب البسط والمقام زي ما هم .. ونحاول ن simplify طريقه الكتابه نفسها .. فديه هي ال softmax ..

Multi-class Problems

- Consider Gaussian conditional distributions with identical Σ

$$\Pr(c_k|\boldsymbol{x}) = \frac{\Pr(c_k)\Pr(\boldsymbol{x}|c_k)}{\sum_j \Pr(c_j)\Pr(\boldsymbol{x}|c_j)}$$
$$= \frac{\pi_k e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)}}{\sum_j \pi_j e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)}}$$
$$= \frac{\pi_k e^{-\frac{1}{2}\left(-2\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}+\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k\right)}}{\sum_j \pi_j e^{-\frac{1}{2}\left(-2\boldsymbol{\mu}_j^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}+\boldsymbol{\mu}_j^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j\right)}}$$
$$= \frac{e^{\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}-\frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{u}_k+\ln\pi_k}}{\sum_j e^{\boldsymbol{\mu}_j^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}-\frac{1}{2}\boldsymbol{\mu}_j^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j+\ln\pi_j}} = \frac{e^{\boldsymbol{w}_k^T\bar{\boldsymbol{x}}}}{\sum_j e^{\boldsymbol{w}_j^T\bar{\boldsymbol{x}}}} \Longrightarrow \text{softmax}$$

where $\boldsymbol{w}_k^T = (-\frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln\pi_k,\ \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1})$

السوفت ماكس هو الجينيريلايزيشن بتاع السجمويد ... طب هو ليه اسمو softmax ...

Softmax

- When there are several classes, the posterior is a **softmax** (generalization of the sigmoid)
- Softmax distribution: $\Pr(c_k|\boldsymbol{x}) = \frac{e^{f_k(x)}}{\sum_j e^{f_j(x)}}$
- Argmax distribution:

$$\Pr(c_k|\boldsymbol{x}) = \begin{cases} 1 & \text{if } k = argmax_j\, f_j(x) \\ 0 & \text{otherwise} \end{cases}$$

$$= \lim_{base \to \infty} \frac{base^{f_k(x)}}{\sum_j base^{f_j(x)}}$$

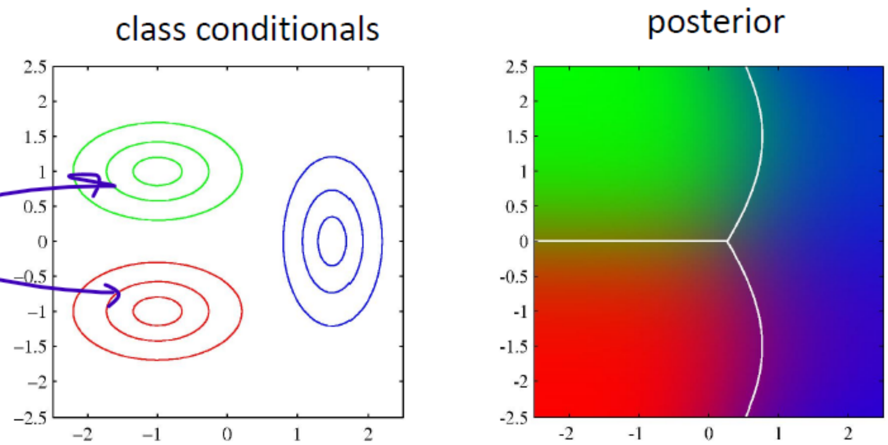
$$\approx \frac{e^{f_k(x)}}{\sum_j e^{f_j(x)}} \quad \text{(softmax approximation)}$$

تعال نقول انا بحسب posterior و عاوز أ associate probability 1 for the class that has the heighest function and zero otherwise

فهو هيدي واحد للي ليه أعلي قيمه .. وصفر للباقي .. ومن هنا جيه الماكسمم .. انو بياخد ال argmax .. لو عاوز أ soften الكلام ده .. من هنا هو ال softmax هو عباره عن approximation .. لإنو ممكن تكتب ال argmax بطريقه مختلفه يعني ... روح للمالانهايه .. و خد البيز و حط الفانكنش لكل كلاس ك exponent و اقسم علي ال base مرفوع للفانكشن بتاعت كل كلاس و اجمعهم كلهم ..... فلما البيز تروح للإننتي .. هتقسم علي 1 .. وهتاخد البسط بس ... فهنا انت عاوز ببييز فاينايت .. فعشان كدا البييز هنا هو ال exponent

Softmax

same covariance Matrix



المهم الدكتور بيقولك ان الشكل بتاع الجاوسين بيتحدد عن طريق ال covariance matrix ... الباوندري هنا شكلو زي الخوخه .. الباوندري linear separator ما بين الاتنين اللي ليهم نفس الجاوسين covariance matrix ... انما لما مكنش في نفس الكوفارينس ماتركس .. انت بقا عندك non-linear boundary ...

الخطوه الجايه ... احنا شفنا ازاي نعمل ال inference .. هنحسب ال posterior .. بس البارمترز بتاعت الموديل بتيجي منين .. فهدفنا هنا هو البارمترز استميشن

Parameter Estimation

- Where do $\Pr(c_k)$ and $\Pr(\boldsymbol{x}|c_k)$ come from?
- Parameters: π, $\boldsymbol{\mu_1}$, $\boldsymbol{\mu_2}$, $\boldsymbol{\Sigma}$

same Σ

$$\Pr(c_1) = \pi, \qquad \Pr(\boldsymbol{x}|c_1) = k_{\boldsymbol{\Sigma}}\, e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_1})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_1})}$$

$$\Pr(c_2) = 1-\pi, \qquad \Pr(\boldsymbol{x}|c_2) = k_{\boldsymbol{\Sigma}}\, e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_2})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_2})}$$

where $k_{\boldsymbol{\Sigma}}$ is the normalization constant that depends on $\boldsymbol{\Sigma}$

- Estimate parameters by
  - **Maximum likelihood**
  - Maximum a posteriori
  - Bayesian learning

بالنسبه للرننج بقا نقدر نستخدم ال Max Likelihood ... هنا هدفنا نلاقي البارمترز بتاعت الموديل اللي تماكسيمايز ال likelihood بتاع الداتا .. فاحنا هنبدأ بالlikelihood فانكشن بتاعت الداتا سيت ...

Maximum Likelihood Solution

- Likelihood:

class #1 class #2

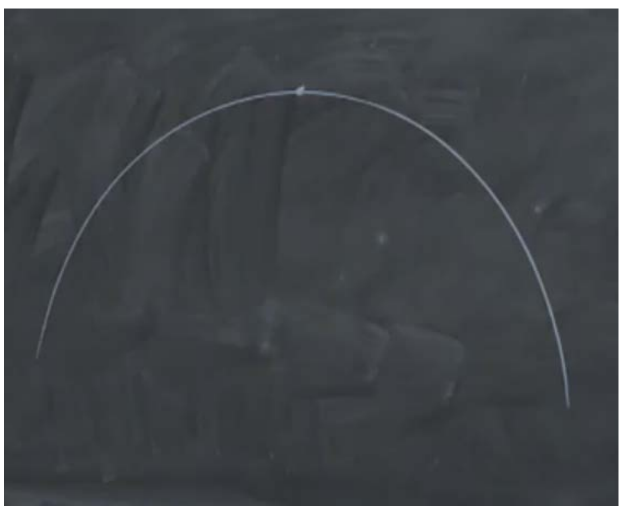
$y_n \in \{0,1\}$ عشان الرياضه تبقا اسهل

$$L(\mathbf{X}, \mathbf{y}) = \Pr(\boldsymbol{X}, \boldsymbol{y} | \pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_n [\pi N(\boldsymbol{x}_n | \boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{y_n} [(1-\pi) N(\boldsymbol{x}_n | \boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-y_n}$$

- ML hypothesis: optimization problem

$$< \pi^*, \boldsymbol{\mu}_1^*, \boldsymbol{\mu}_2^*, \boldsymbol{\Sigma}^* > = argmax_{\pi, \boldsymbol{\mu}_2, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}} \sum_n y_n \left[\ln \pi + \ln k_{\boldsymbol{\Sigma}} - \frac{1}{2} (\boldsymbol{x}_n - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_1) \right] + \underbrace{(1-y_n)} \left[\ln(1-\pi) + \ln k_{\boldsymbol{\Sigma}} - \frac{1}{2} (\boldsymbol{x}_n - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_2) \right]$$

الأوبجكتيف هنا بتاع ال maximizing log-likelihood بي correspond to a curve ليه الشكل اللي الدكتور رسمه ده .. حاجه concave function التوب بتاعها هو الماكسيمم .. فاشتق وساوي بالصفر وهيجيلك النقطه اللي انت عاوزها للماكسمم



Maximum Likelihood Solution

- Set derivative to 0

$$0 = \frac{\partial \ln L(\boldsymbol{X},\boldsymbol{y})}{\partial \pi}$$

$$\Rightarrow 0 = \sum_n y_n \left[\frac{1}{\pi}\right] + (1 - y_n)\left[-\frac{1}{1-\pi}\right]$$

$$\Rightarrow 0 = \sum_n y_n(1 - \pi) + (1 - y_n)(-\pi)$$

$$\Rightarrow \sum_n y_n = \pi\left(\sum_n y_n + \sum_n (1 - y_n)\right)$$

$$\Rightarrow \sum_n y_n = \pi N \quad \text{(where } N \text{ is the \# of training points)}$$

sum data points belong to class

$$\boxed{\therefore \frac{\sum_n y_n}{N} = \pi}$$

p(class)

This is sample mean (Empirical mean)

الإكسبرشن الأخير ده .. هو انك تجمع الداتا بوينتس اللي نفس الكلاس بتاعك و تقسم علي عددهم ...

Maximum Likelihood Solution

μ_1 N

$$0 = \partial \ln L(\boldsymbol{X},\boldsymbol{y}) / \partial \boldsymbol{\mu}_1$$

$$\Rightarrow 0 = \sum_n y_n\left[-\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_1)\right]$$

$$\Rightarrow \sum_n y_n \boldsymbol{x}_n = \sum_n y_n \boldsymbol{\mu}_1$$

$$\Rightarrow \sum_n y_n \boldsymbol{x}_n = N_1 \boldsymbol{\mu}_1$$

Empirical Mean

$$\boxed{\therefore \frac{\sum_n y_n \boldsymbol{x}_n}{N_1} = \boldsymbol{\mu}_1} \quad \text{Similarly: } \boxed{\frac{\sum_n (1-y_n)\boldsymbol{x}_n}{N_2} = \boldsymbol{\mu}_2}$$

where N_1 is the # of data points in class 1

N_2 is the # of data points in class 2