אלגוריתמי למידה גנרטיביים
Generative Learning Algorithms

- בעיית הסווג: נניח שרוצים להבחין בין כלבים (y=1) לחתולים (y=0) בהתבסס על מספר תכונות של החיה.
- אלגוריתמים כמו רגרסיה לוגיסטית או פרספטרון – ניסיון להתאים ישר מפריד או על-מישור, להפרדה בין כלבים לחתולים – כלומר לימוד ההסתברות p(y|x) (ההסתברות האפוסטריורית של המחלקה בהינתן התכונות) באופן ישיר
- או לחילופין ניסיון ללמוד באופן ישיר את המיפוי ממרחב הקלט X לקבוצת התגיות {0,1}.
- אלגוריתמים כאלה נקראים אלגוריתמי למידה מבחינים (Discriminative Learning Algorithms)


אלגוריתמי למידה גנרטיביים
Generative Learning Algorithms

- דרך אלטרנטיבית: בניית מודל על-פי התכונות.
- מתבוננים בתכונות החיה (כלבים), ובונים מודל, לאחר מכן באופן דומה בונים מודל נפרד עבור חתולים.
- לבסוף עבור חיה חדשה, בוחנים את ההתאמה עבור כל אחד מהמודלים, ומסווגים לפי הדמיון הרב יותר.

אלגוריתמי למידה גנרטיביים
Generative Learning Algorithms

- מנסים למדל את p(x|y) ואת p(y)
- התפלגות תכונה עבור "כלב" p(x|y=1), ועבור "חתול" p(x|y=0).
- לאחר קבלת מודל של ההסתברויות האפריוריות p(y) ושל ההסתברויות המותנות (או הסבירויות likelihoods) מקבלים את ההסתברות האפוסטריורית – p(y|x)

$$p(y|x)=\frac{p(x|y)\cdot p(y)}{p(x)}$$

כאשר:

$$p(x)=p(x|y=1)\cdot p(y=1)+p(x|y=0)\cdot p(y=0)$$

(נוסחת ההסתברות השלמה).

אלגוריתמי למידה גנרטיביים
Generative Learning Algorithms

$$\arg\max_y p(y \mid x) = \arg\max_y \frac{p(x \mid y) \cdot p(y)}{p(x)} =$$

$$= \arg\max_y p(x \mid y) \cdot p(y)$$

(אין צורך להשתמש במכנה לצורך חיזוי)

אלגוריתמי למידה גנרטיביים
Generative Learning Algorithms

- אלגוריתם הלמידה הגנרטיבי הראשון:

Gaussian Discriminant Analysis (GDA)

- ההסתברות המותנית (הסבירות) מתפלגת בהתאם להתפלגות גאוסית מרובה:

Multivariate Gaussian (normal) distribution

P(x|y)~N(μ,Σ)

$$p\left(x;\mu,\Sigma\right)=\frac{1}{\left(2\pi\right)^{(n/2)}\left|\Sigma\right|^{1/2}}e^{-\frac{1}{2}\left(x-\mu\right)^{T}\Sigma^{-1}\left(x-\mu\right)}$$

Multivariate Gaussian Distribution

ההתפלגות הגאוסית הרב-מימדית מוגדרת על-ידי :

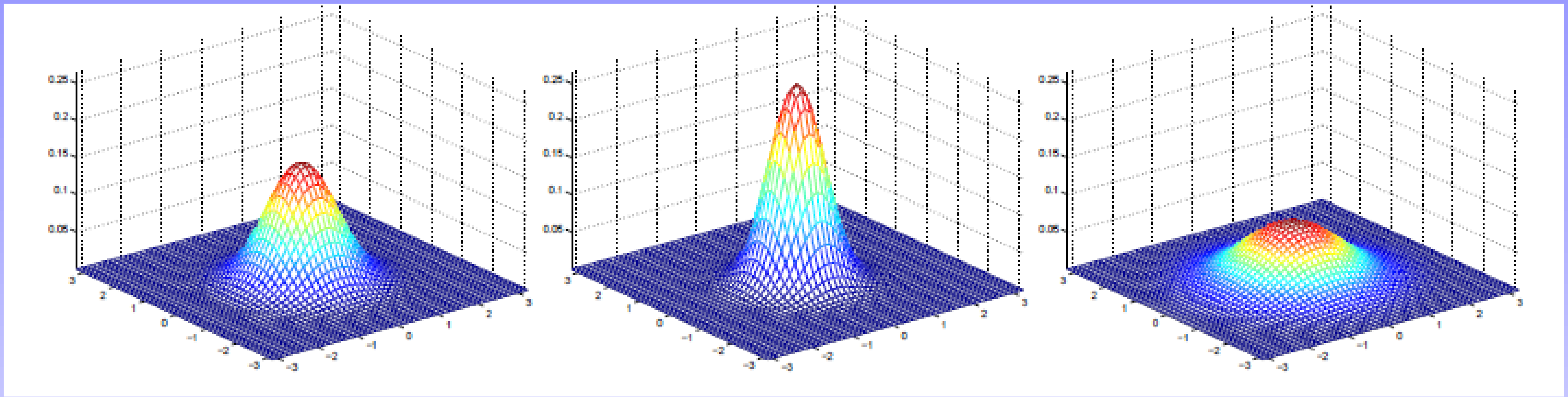
$$p(x) = \frac{1}{(2\pi)^{l/2}|S|^{1/2}} \exp\left(-\frac{1}{2}(x-m)^T S^{-1}(x-m)\right)$$

כאשר m=E(x) הוא וקטור התוחלות, והמטריצה S היא מטריצת הקווריאנס המוגדרת על-ידי :

$$S = E[(x-m)(x-m)^T]$$

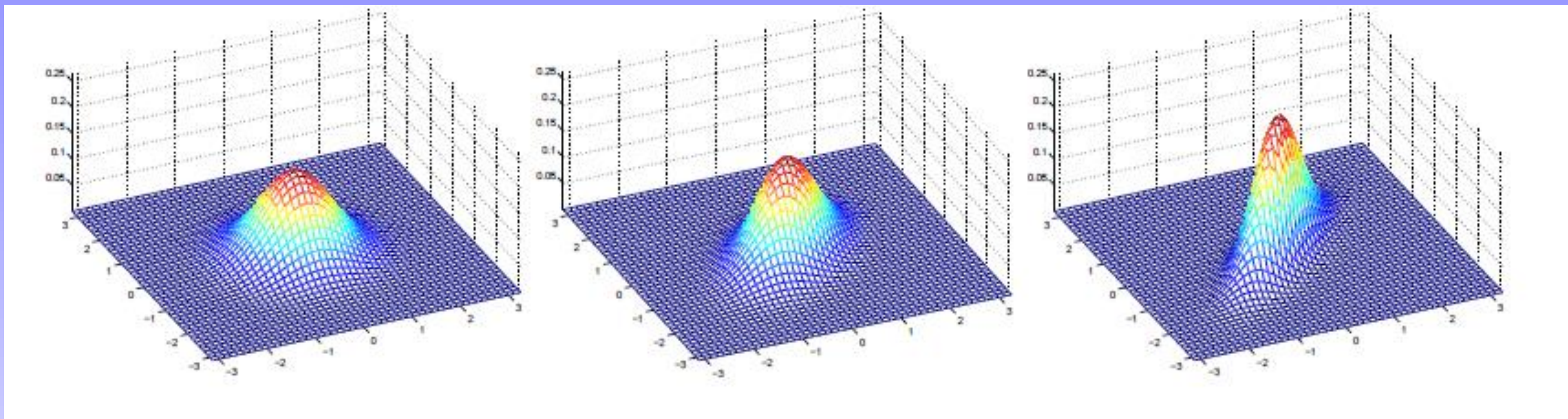
וכאשר $|S|$ היא הדטרמיננטה של מטריצת הקווריאנס.

Multivariate Gaussian Distribution



The left-most figure shows a Gaussian with mean zero (that is, the 2x1 zero-vector) and covariance matrix $\Sigma = I$ (the 2x2 identity matrix). A Gaussian with zero mean and identity covariance is also called the **standard normal distribution**. The middle figure shows the density of a Gaussian with zero mean and $\Sigma = 0.6I$; and in the rightmost figure shows one with , $\Sigma = 2I$. We see that as Σ becomes larger, the Gaussian becomes more "spread-out," and as it becomes smaller, the distribution becomes more "compressed."

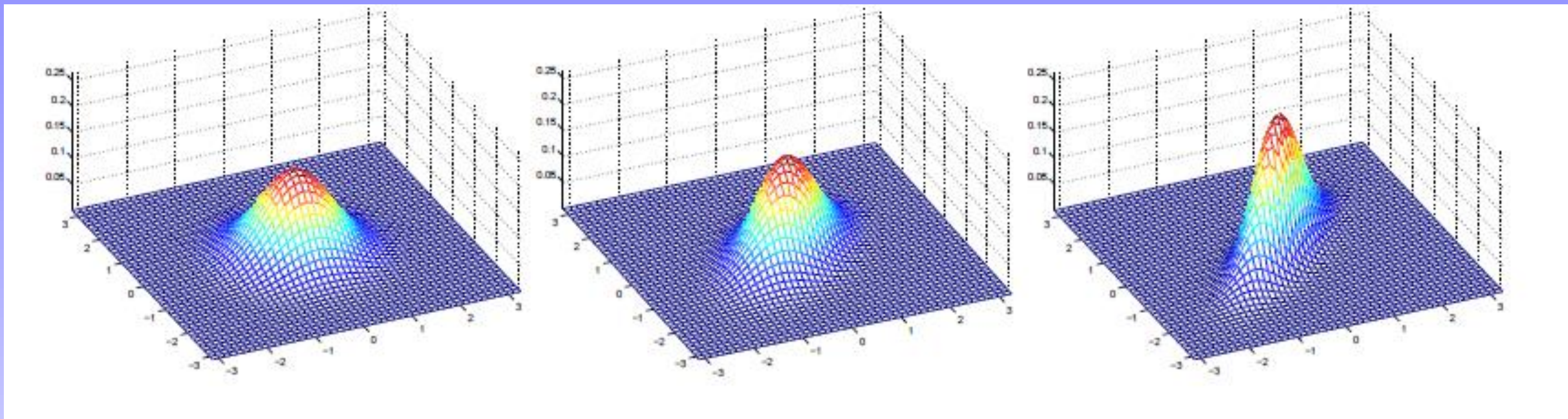
Multivariate Gaussian Distribution



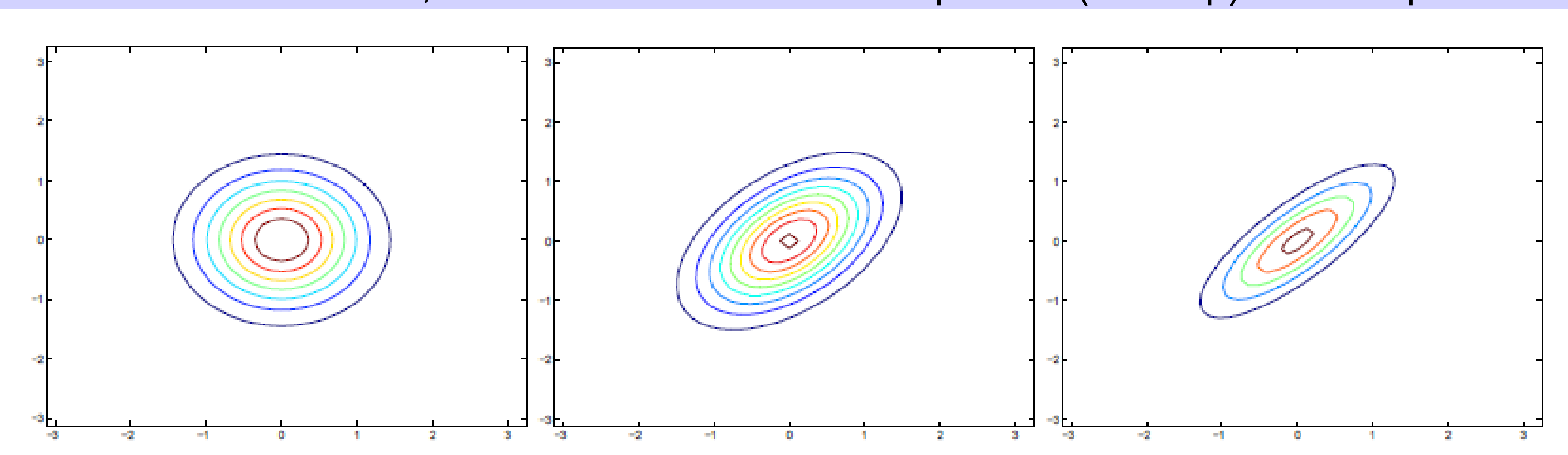
The figures above show Gaussians with mean 0, and with covariance matrices respectively

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \ \ \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}; \ \ .\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}.$$

Multivariate Gaussian Distribution

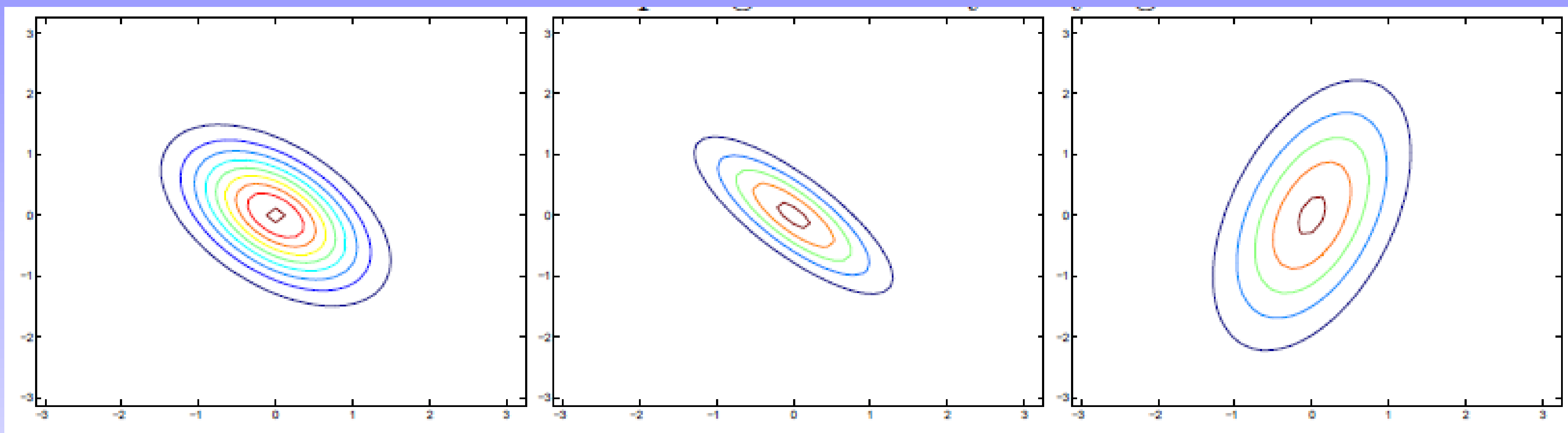


למטה: עקומי מתאר (קונטורים) של פונקציות הצפיפות המופיעות למעלה, בהתאמה.



Multivariate Gaussian Distribution

על-ידי שינוי של מטריצת הקווריאנס, נשנה את פונקציות הצפיפות:

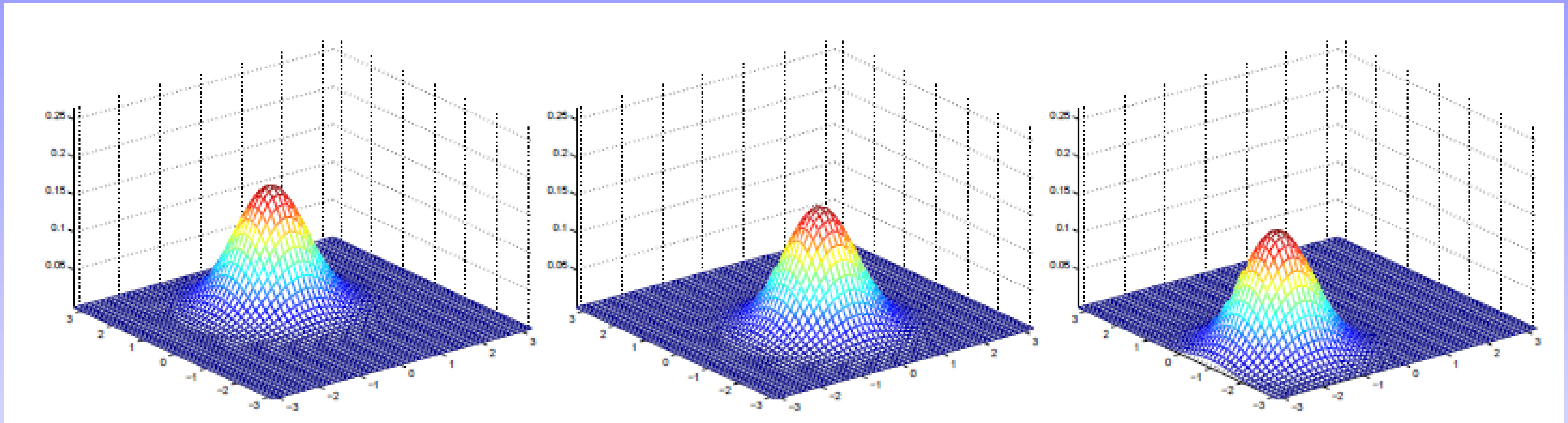


The plots above used, respectively,

$$\Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}; \quad \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}; \quad .\Sigma = \begin{bmatrix} 3 & 0.8 \\ 0.8 & 1 \end{bmatrix}.$$

Multivariate Gaussian Distribution

נקבע את מטריצת הקווריאנס, ונשנה את וקטורי התוחלת (הממוצע):



The figures above were generated using $\Sigma = I$, and respectively

$$\mu = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \ \mu = \begin{bmatrix} -0.5 \\ 0 \end{bmatrix}; \ \mu = \begin{bmatrix} -1 \\ -1.5 \end{bmatrix}.$$

מודל ה- Gaussian Discriminant Analysis (GDA)

עבור וקטור תכונות קלט שהן משתנים אקראיים רציפים, אפשר להשתמש במודל ה- GDA

הסבירות מתפלגת עם התפלגות נורמלית מרובה: $P(x|y) \sim N(\mu,\Sigma)$

המודל:

$$\begin{aligned} y &\sim \mathrm{Bernoulli}(\phi) \\ x|y=0 &\sim \mathcal{N}(\mu_0,\Sigma) \\ x|y=1 &\sim \mathcal{N}(\mu_1,\Sigma) \end{aligned}$$

עם ההתפלגויות הבאות:

$$\begin{aligned} p(y) &= \phi^y(1-\phi)^{1-y} \\ p(x|y=0) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right) \\ p(x|y=1) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right) \end{aligned}$$

מודל ה- Multivariate Gaussian Distribution

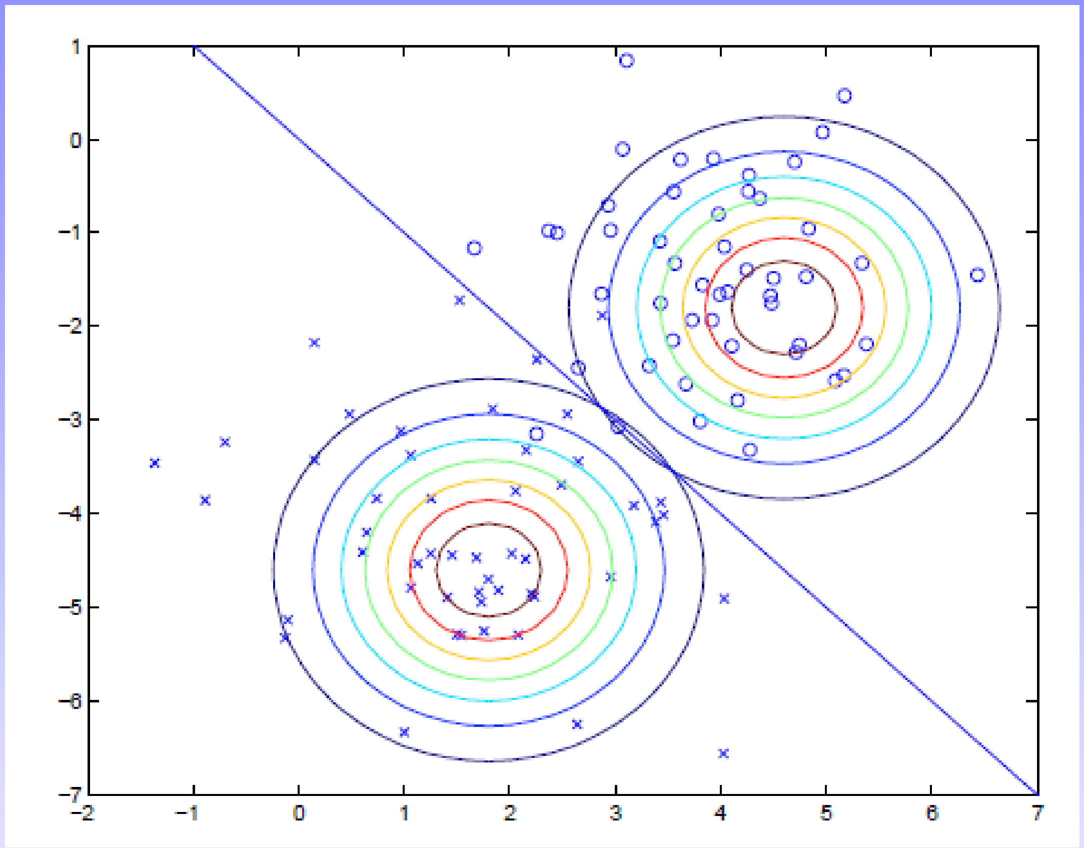
לוג הסבירות (log-likelihood) של ה- data

$$
\begin{aligned}
\ell(\phi, \mu_0, \mu_1, \Sigma) &= \log \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma) \\
&= \log \prod_{i=1}^{m} p(x^{(i)} | y^{(i)}; \mu_0, \mu_1, \Sigma) p(y^{(i)}; \phi).
\end{aligned}
$$

נביא למקסימום את לוג הסבירות ביחס לפרמטרים, ונמצא את משערך ה- ML

$$
\begin{aligned}
\phi &= \frac{1}{m} \sum_{i=1}^{m} 1\{y^{(i)} = 1\} \\
\mu_0 &= \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 0\} x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}} \\
\mu_1 &= \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}} \\
\Sigma &= \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T.
\end{aligned}
$$

מודל ה- Multivariate Gaussian Distribution



- רואים את שתי קבוצות האימון ואת עקומי המתאר של שתי ההתפלגויות שהותאמו לנתונים עבור כל אחת מהמחלקות.
- רואים כי מטריצת הקווריאנס היא זהה וכי שני וקטורי הממוצעים שונים.
- הישר העובר בין שתי ההתפלגויות – מחלק את המישור לשניים – מצד אחד החיזוי הוא y=1 ומצד שני החיזוי הוא y=0.

השוואה בין GDA לרגרסיה לוגיסטית

- אפשר להראות כי אם מסתכלים על הגודל
- כפונקציה של x אזי:

$$p\left(y=1 \mid x; \phi, \mu_0, \mu_1, \Sigma\right)$$

$$p\left(y=1 \mid x; \phi, \mu_0, \mu_1, \Sigma\right) = \frac{1}{1+e^{\left(-\theta^T x\right)}}$$

(מגדירים מחדש את x באגף ימין על-ידי הוספת $x^i_0=1$)

השוואה בין GDA לרגרסיה לוגיסטית

- איזה מודל עדיף?
- לשני המודלים תוצאות שונות עבור אותה קבוצת נתונים.
- אם p(x|y) מתפלג גאוסיינית (עם Σ משותפת לשתי המחלקות) אז בהכרח p(y|x) מהצורה של רגרסיה לוגיסטית.
- ההפך לא בהכרח נכון.
- לפיכך ל- GDA הנחות מודל חזקות יותר על הנתונים מאשר רגרסיה לוגיסטית.
- לכן אם ההנחות מתקיימות – המודל מתאים יותר לנתונים.
- באופן ספיציפי, אם p(x|y) מתפלג גאוסית עם Σ משותפת מודל יעיל אסימפטוטית – באופן לא פורמלי עבור נתוני אימון רבים לא קיים אלגוריתם טוב יותר במובן של שערוך p(y|x)
- לכן GDA מודל טוב יותר מאשר רגרסיה לוגיסטית, בדרך כלל גם עבור מדגמים קטנים.
- עבור הנחות חלשות יותר לרגרסיה לוגיסטית רובסטיות גבוהה יותר, פחות רגישה להנחות מודל.
- אם הנתונים אכן לא מתפלגים גאוסית, אזי בגבול של הרבה מאוד נקודות אימון רגרסיה לוגיסטית כמעט תמיד תהיה טובה יותר מ- GDA, ולכן משתמשים בה יותר מ- GDA.

Naïve Bayes

- הנחה – ה- x_i הם משתנים אקראיים בדידים.
- נבנה מסנן לסינון ספאם ב- e-mail .
- דוגמא לקבוצת בעיות רחבה יותר – סווג טקסט (text classification).
- קבוצת האימון: קבוצת e-mails מתוייגת:

  spam / non-spam
- כל אימייל מיוצג על-ידי וקטור תכונות שארכו כמספר המלים במילון.

Naïve Bayes

- באופן ספיציפי, אם ה- e-mail מכיל את המילה ה- i במילון, נקבע: $x_i = 1$
- אחרת: $x_i = 0$
  - לדוגמא: הווקטור:
  - משמש לייצוג e-mail המכיל את המילה “a” ואת המילה "buy", אך לא את המלים האחרות.

$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ \\ \end{bmatrix} \begin{array}{ll} \ldots & \text{a} \\ \ldots & \text{aardvark} \\ \ldots & \text{aardwolf} \\ & \\ \ldots & \text{buy} \\ & \\ \ldots & \text{zygmurgy} \end{array}$$

צבוען הרעמה

שנבוב

Naïve Bayes

- באופן מעשי מסתכלים רק על קבוצת האימון ומקודדים את המלים הנמצאות שם.
- היתרונות: 1) צמצום מספר המלים, 2) הכללת מלים שאינן נמצאות במילון (אך כן ב- e-mails).
- לעתים מוציאים מהמילון את המלים הנפוצות מאוד הקיימות בכל מסמך כמו "and", "of", "the", ולא משמשות כאינדיקטור האם המייל הוא ספאם או לא ספאם
- מלים אלה נקראות "content free" words
- קבוצת המלים המקודדות ב- feature vector נקראות מילון.
- הממד של x שווה לגודל המילון.

Naïve Bayes

- עתה נבנה מודל גנרטיבי: מודל של p(x|y).
- הבעייה – אם יש 100000מלים אזי x הוא וקטור 100000 - ממדי של אפסים ואחדים, ואם ממדלים את x באופן מפורש עם התפלגות מולטינומית כל התוצאות האפשריות, אז מגיעים לוקטור פרמטרים של $2^{1000000}$ וזה כמובן יותר מדי.

Naïve Bayes Assumption

- הנחה – Naïve Bayes Assumption – ה-x_i הם בלתי תלויים בהינתן y.
- האלגוריתם נקרא Naïve Bayes Classifier.
- לדוגמא: נניח כי y=1, אז המילה ה- 2087 (נניח buy) לא משפיעה על המילה ה- 39831 (נניח price).
- כלומר:

$$p\left(x_{2087} \mid y\right) = p\left(x_{2087} \mid y, x_{39831}\right)$$

- זוהי אי-תלות מותנה ולא אי-תלות סטטיסטית, כלומר אי אפשר לכתוב:

$$p\left(x_{2087}\right) = p\left(x_{2087} \mid x_{39831}\right)$$

Naïve Bayes Classifier

- לפיכך:

$$p\left(x_1, x_2, \ldots, x_{100000}, | \, y\right) =$$

$$= p\left(x_1 \,|\, y\right) \cdot p\left(x_2 \,|\, y, x_1\right) \cdot p\left(x_3 \,|\, y, x_1, x_2\right) \ldots$$

$$\cdot\, p\left(x_{100000} \,|\, y, x_1, x_2 \ldots, x_{99999}\right)$$

$$= p\left(x_1 \,|\, y\right) \cdot p\left(x_2 \,|\, y\right) \cdot p\left(x_3 \,|\, y\right) \cdot \ldots \cdot p\left(x_{100000} \,|\, y\right) =$$

$$\prod_{i=1}^{n} p\left(x_i \,|\, y\right)$$

- השוויון הראשון נובע מחוקי ההסתברות, והשוויון השני מהנחת ה- NB.
- למרות שההנחה חזקה באופן קיצוני, האלגוריתם עובד היטב במגוון של בעיות.

Naïve Bayes Classifier:
Model Parametrization

- יש צורך לשערך את הפרמטרים הבאים:

$$\phi_{i|y=1} = p\left(x_i = 1 \mid y = 1\right)$$

$$\phi_{i|y=0} = p\left(x_i = 1 \mid y = 0\right)$$

$$\phi_y = p\left(y = 1\right)$$

Naïve Bayes Classifier: Model Parametrization

- נתונה קבוצת האימון:

$$\left\{x^{(i)}, y^{(i)};\ i = 1, 2, \ldots, m\right\}$$

- הסבירות המשותפת של הנתונים:

$$L\left(\phi_y, \phi_{i|y=0}, \phi_{i|y=1}\right) = \prod_{i=1}^{m} p\left(x^{(i)}, y^{(i)}\right)$$

- הסבר: זהו מודל הסתברותי של התצפיות.
- רוצים לשערך את הפרמטרים באמצעות MLE

Naïve Bayes Classifier: Model Parametrization

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \cap y^{(i)} = 1\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}$$

מנת ה-spams בהם הופיעה
המילה ה- j מתוך סה"כ
הספאמים.

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \cap y^{(i)} = 0\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\}}$$

מנת ה- e-mails (לא ספאם)
בהם הופיעה המילה ה- j
מתוך סה"כ ה- e-mails.

$$\phi_y = \frac{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}{m}$$

מנת ה- spams מתוך סה"כ
קבוצת האימון

Naïve Bayes Classifier: Prediction

כדי לבצע חיזוי עבור מייל חדש עם וקטור תכונות x:

$$p(y=1|x)=\frac{p(x|y=1)\cdot p(y=1)}{p(x)}=$$

$$=\frac{\prod_{i=1}^{n}p(x_i|y=1)\cdot p(y=1)}{\prod_{i=1}^{n}p(x_i|y=1)\cdot p(y=1)+\prod_{i=1}^{n}p(x_i|y=0)\cdot p(y=0)}$$

באותו אופן מחשבים:

$$p(y=0|x)=1-p(y=1|x)$$

ובוחרים את המחלקה עם ההסתברות האפוסטריורית הגבוהה ביותר.

Naïve Bayes Classifier:

הכללה: לכל משתנה יותר משני מצבים, כלומר:

$$x_i \in \{1, 2, \ldots k_i\}$$

ההתפלגות של $p(x_i|y)$ תהיה מולטינומית במקום התפלגות ברנולי.

Naïve Bayes Classifier: Discretization

הערה: אם תכונות הקלט הן רציפות, עדיין אפשר לבצע דיסקרטיזציה, כלומר להפוך אותן למספר קטן של ערכים בדידים, ולהפעיל את אלגוריתם NB.

לדוגמא: עבור בעיית שטחי הבתים:

שטח הבית (מ"ר)	<50	50-100	100-150	150-200	>200
xi	1	2	3	4	5

כלומר עבור בית עם שטח של 125 מ"ר הערך של התכונה המתאימה xi יהיה 3.

מסקנה: כאשר אין אפשרות למדל את התכונות המקוריות הרציפות באמצעות התפלגות גאוסית מרובה, דיסקרטיזציה של התכונות ושימוש ב- NB במקום ב- GDA יביאו בדרך כלל למסווג טוב יותר.

Naïve Bayes Classifier: Discretization

- מסקנה:

כאשר אין אפשרות למדל את התכונות המקוריות הרציפות באמצעות התפלגות גאוסית מרובה, דיסקרטיזציה של התכונות ושימוש ב- NB במקום ב- GDA יביאו בדרך כלל למסווג טוב יותר.

Laplace smoothing

- למודל הנוכחי נבצע שיפור פשוט המונע בעייה שכיחה.
  נציג את הבעייה, ולאחר מכן איך לתקן אותה.
- לדוגמא: נניח שנתקלים ב- e-mails במילה כמו IEEE, המתקבלת בפעם הראשונה ולא נמצאת בדוגמאות האימון.
- נניח כי זוהי המילה ה- 15000 במילון, ולכן המסנן של ה- NB יעריך את שערוכי ה- ML הבאים:

$$\phi_{15000|y=1} = \frac{\sum_{i=1}^{m} 1\left\{x_{15000}^{(i)} = 1 \cap y^{(i)} = 1\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}} = 0$$

$$\phi_{15000|y=0} = \frac{\sum_{i=1}^{m} 1\left\{x_{15000}^{(i)} = 1 \cap y^{(i)} = 0\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}} = 0$$

- הסבר: מאחר והמילה IEEE לא נראתה אף פעם, המסווג "חושב" שעבור כל אחת מהאפשרויות מייל/ספאם ההסתברות היא 0.

Laplace smoothing

- חישוב ההסתברות האפוסטריורית:

$$p\left(y=1 \mid x\right)=\frac{p\left(x \mid y=1\right)\cdot p\left(y=1\right)}{p\left(x\right)}=$$

$$=\frac{\prod_{i=1}^{n} p\left(x_i \mid y=1\right)\cdot p\left(y=1\right)}{\prod_{i=1}^{n} p\left(x_i \mid y=1\right)\cdot p\left(y=1\right)+\prod_{i=1}^{n} p\left(x_i \mid y=0\right)\cdot p\left(y=0\right)}=\frac{0}{0}$$

וזאת מאחר וכל אחד מהביטויים:

$$\prod_{i=1}^{n} p\left(x_i \mid y=1\right)$$

מכיל ביטוי:
בתוך המכפלה.

$$p\left(x_{15000} \mid y=1\right)=0$$

Laplace smoothing

- האלגוריתם מקבל אם כן תוצאה של 0/0 ולא יודע איך לבצע חיזוי.
- בהסתכלות רחבה יותר: באופן סטטיסטי לשערך הסתברות של מאורע כלשהו כ- 0 רק בשל כך שלא ראינו אותו בקבוצת האימון זה רעיון גרוע.

Laplace smoothing

- נניח שרוצים לשערך את הממוצע של משתנה אקראי מולטינומי z המקבל ערכים מתוך {1,2,…k}.
- אפשר לבצע פרמטריזציה של המשתנה המולטינומי עם

$$\phi_i = p\left(z = i\right)$$

- אם נתונות m תצפיות בת"ס

$$\left\{ z^{(1)}, z^{(2)}, \dots, z^{(m)} \right\}$$

- שערוך ה-ML נתונים על-ידי:

$$\phi_j = \frac{\sum_{i=1}^{m} 1\left\{ z^{(i)} = j \right\}}{m}$$

Laplace smoothing

- כפי שראינו קודם אם משתמשים בשערוכי ML חלק מההסתברויות עשוי להיות 0. כדי להימנע ממצב זה -

משתמשים בהחלקת Laplace: (Laplace smoothing)

$$\phi_j = \frac{\sum_{i=1}^{m} 1\left\{ z^{(i)} = j \right\} + 1}{m + k}$$

כלומר מוסיפים 1 למונה ו- k למכנה.

Laplace smoothing

תרגיל: הראו כי עדיין $\sum_{j=1}^{k} \phi_j = 1$

$$\phi_j = \frac{\sum_{i=1}^{m} 1\left\{z^{(i)} = j\right\} + 1}{m + k}$$

- זוהי כמובן תכונה רצויה כי סכום ההסתברויות צריך להיות 1,
- בנוסף כמובן נמנע המצב של הסתברויות ששוות ל- 0.
- בתנאים מסויימים (די חזקים) אפשר להראות שהחלקת לפלאס יוצרת משערך אופטימלי של ה- Φ_j's.

Naïve Bayes Classifier: Model Parametrization

$$\phi_{j|y=1} = \frac{\left(\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \cap y^{(i)} = 1\right\}\right) + 1}{\left(\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}\right) + 2}$$

מנת ה-spams בהם הופיעה המילה ה- j מתוך סה"כ ה-spams.

$$\phi_{j|y=0} = \frac{\left(\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \cap y^{(i)} = 0\right\}\right) + 1}{\left(\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\}\right) + 2}$$

מנת ה- e-mails (לא ספאם) בהם הופיעה המילה ה- j מתוך סה"כ ה- e-mails.

$$\phi_y = \frac{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}{m}$$

מנת ה- spams מתוך סה"כ קבוצת האימון

כאן אין צורך בהחלקת לפלאס כי יש בדר"כ יש מספיק דוגמאות של spams ו- non spams כך שהפרקציות של כל אחד מהם לא מביאות לשערוך הסתברות של אפס.