Discriminative classifiers: Logistic Regression, SVMs

CISC 5800

Professor Daniel Leeds

Maximum A Posteriori: a quick review

- Likelihood: $P(D|\theta) = P(D|p) = p^{|H|}(1-p)^{|T|}$
- Prior: $P(\theta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha,\beta)} = P(p) = \frac{p^{\alpha-1}(1-p)^{\beta-1}}{B(\alpha,\beta)}$
- Posterior Likelihood x prior = $P(D|\theta)P(\theta)$

Choose α and β to give the prior belief of Heads bias $p \in [0,1]$

Higher α: Heads more likely
Higher β: Tails more likely

- MAP estimate:

$$\underset{\theta}{\operatorname{argmax}} \log P(D|\theta) + \log P(\theta)$$

$$\underset{p}{\operatorname{argmax}} \log P(D|p) + \log P(p)$$

$$p = \frac{|H| + (\alpha - 1)}{|H| + (\alpha - 1) + |T| + (\beta - 1)}$$

2

Estimate each P(X$_i$|Y) through **MAP**

Incorporating prior for each class β_j

$$P(X_i = x_k | Y = y_j) = \frac{\#D(X_i = x_k \wedge Y = y_j) + (\beta_j - 1)}{\#D(Y = y_j) + \sum_m(\beta_m - 1)}$$

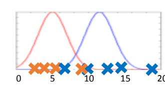
$$P(Y = y_j) = \frac{\#D(Y = y_j) + (\beta_j - 1)}{|D| + \sum_m(\beta_m - 1)}$$

$(\beta_j - 1)$ – "frequency" of class j
$\sum_m(\beta_m - 1)$ – "frequencies" of all classes

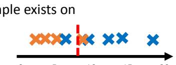
Note: both X and Y can take on multiple values (binary **and** beyond)

3

Classification strategy: generative vs. discriminative



- Generative, e.g., Bayes/Naïve Bayes:
  - Identify probability distribution for each class
  - Determine class with maximum probability for data example
- Discriminative, e.g., Logistic Regression:
  - Identify boundary between classes
  - Determine which side of boundary new data example exists on



4

Linear algebra: data features

- Vector – list of numbers: each number describes a data **feature**
- Matrix – list of lists of numbers: features for each data point

	Document 1	Document 2	Document 3
Wolf	12	8	0
Lion	16	10	2
Monkey	14	11	1
Broker	0	1	14
Analyst	1	0	10
Dividend	1	1	12
⋮	⋮	⋮	⋮

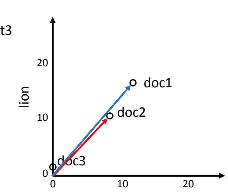
# of word occurrences

5

Feature space

- Each data feature defines a dimension in space

	Document1	Document2	Document3
Wolf	12	8	0
Lion	16	10	2
Monkey	14	11	1
Broker	0	1	14
Analyst	1	0	10
Dividend	1	1	12
⋮	⋮	⋮	⋮

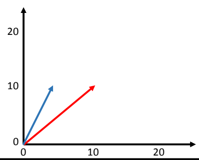


6

The dot product

The dot product compares two vectors:

- $\boldsymbol{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$ $\quad \boldsymbol{a} \cdot \boldsymbol{b} = \sum_{i=1}^{n} a_i b_i \;\; = \boldsymbol{a}^T \boldsymbol{b}$



$$\begin{bmatrix} 5 \\ 10 \end{bmatrix} \cdot \begin{bmatrix} 10 \\ 10 \end{bmatrix} = \mathbf{5 \times 10 + 10 \times 10}$$

$$= \mathbf{50 + 100 = 150}$$

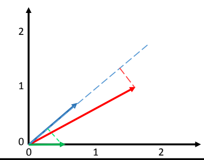
7

The dot product, continued $\quad \boldsymbol{a} \cdot \boldsymbol{b} = \sum_{i=1}^{n} a_i b_i$

Magnitude of a vector is the sum of the squares of the elements

$$|\boldsymbol{a}| = \sqrt{\sum_i a_i^2}$$

If $\boldsymbol{a}$ has unit magnitude, $\boldsymbol{a} \cdot \boldsymbol{b}$ is the "projection" of $\boldsymbol{b}$ onto $\boldsymbol{a}$



$$\begin{bmatrix} 0.71 \\ 0.71 \end{bmatrix} \cdot \begin{bmatrix} 1.5 \\ 1 \end{bmatrix} = .71 \times 1.5 + .71 \times 1$$

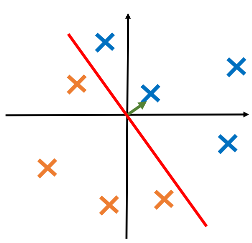
$$\approx 1.07 + .71 = 1.78$$

$$\begin{bmatrix} 0.71 \\ 0.71 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} = .71 \times 0 + .71 \times 0.5$$

$$\approx 0 + .35 = 0.35$$

8

Separating boundary, defined by w

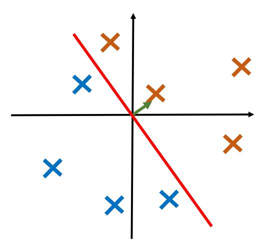


- Separating **hyperplane** splits **class 0** and **class 1**
- Plane is defined by line **w** perpendicular to plan
- Is data point **x** in class 0 or class 1? $\mathbf{w}^T\mathbf{x} > 0$ class 0
  $\mathbf{w}^T\mathbf{x} < 0$ class 1

9

Separating boundary, defined by w

More typically

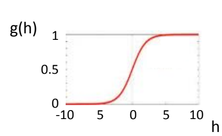


- Separating **hyperplane** splits **class 0** and **class 1**
- Plane is defined by line **w** perpendicular to plan
- Is data point **x** in class 0 or class 1? $\mathbf{w}^T\mathbf{x} > 0$ class **1**
  $\mathbf{w}^T\mathbf{x} < 0$ class **0**

10

From real-number projection to 0/1 label

- Binary classification: 0 is class A, 1 is class B
- Sigmoid function stands in for p(x|y)



- Sigmoid: $g(h) = \frac{1}{1+e^{-h}}$
- $p(y=0|x;\theta) = 1 - g(w^T x) = \frac{e^{-w^T x}}{1+e^{-w^T x}}$
- $p(y=1|x;\theta) = g(w^T x) = \frac{1}{1+e^{-w^T x}}$

$$\boldsymbol{w^T x} = \sum_j \boldsymbol{w_j x_j}$$

11

Learning parameters for classification

- Similar to MLE for Bayes classifier
- "Likelihood" for data points $y^1, \ldots, y^n$ (different from Bayesian likelihood)
  - If y^i in class A, $y^i = 0$, multiply $(1-g(x^i;w))$
  - If y^i in class B, $y^i=1$, multiply $(g(x^i;w))$

$$\underset{w}{\operatorname{argmax}}\, L(y|x;w) = \prod_i \left(1 - g(x^i;w)\right)^{(1-y^i)} g(x^i;w)^{y^i}$$

$$LL(y|x;w) = \sum_i (1-y^i)\log\left(1-g(x^i;w)\right) + y^i \log\left(g(x^i;w)\right)$$

$$LL(y|x;w) = \sum_i y^i \log\frac{g(x^i;w)}{1-g(x^i;w)} + \log\left(1-g(x^i;w)\right)$$

12

Learning parameters for classification

$$g(h) = \frac{1}{1+e^{-h}}$$

$$LL(y|x;w) = \sum_i y^i \log \frac{g(x^i;w)}{1-g(x^i;w)} + \log\left(1-g(x^i;w)\right)$$

$$LL(y|x;w) = \sum_i y^i \log \frac{\frac{1}{1+e^{-w^Tx^i}}}{1-\frac{1}{1+e^{-w^Tx^i}}} + \log\left(\frac{e^{-w^Tx^i}}{1+e^{-w^Tx^i}}\right)$$

$$LL(y|x;w) = \sum_i y^i \log \frac{1}{1+e^{-w^Tx^i}-1} + \log\left(\frac{e^{-w^Tx^i}}{1+e^{-w^Tx^i}}\right)$$

$$LL(y|x;w) = \sum_i y^i w^T x^i - w^T x^i - \log\left(1+e^{-w^Tx^i}\right)$$

13

Learning parameters for classification

$$w^Tx = \sum_j w_j x_j$$

$$g'(h) = \frac{e^{-h}}{(1+e^{-h})^2}$$

$$LL(y|x;w) = \sum_i y^i w^T x^i - w^T x^i + \log\left(g(w^Tx^i)\right)$$

$$\frac{\partial}{\partial w_j} LL(y|x;w) = \sum_i y^i x_j^i - x_j^i + \frac{x_j^i e^{-w^Tx^i}}{1+e^{-w^Tx^i}}$$

$$\frac{\partial}{\partial w_j} LL(y|x;w) = \sum_i x_j^i\left(y^i - \left(1-\left(1-g(w^Tx^i)\right)\right)\right)$$

$$\frac{\partial}{\partial w_j} LL(y|x;w) = \sum_i x_j^i\left(y^i - g(w^Tx^i)\right)$$

14

Iterative gradient ascent

y^i – true data label

$g(w^Tx^i)$ – computed data label

- Begin with initial guessed weights w
- For each data point (y^i,x^i), update each weight w_j

$$w_j \leftarrow w_j + \varepsilon x_j^i\left(y^i - g(w^Tx^i)\right)$$

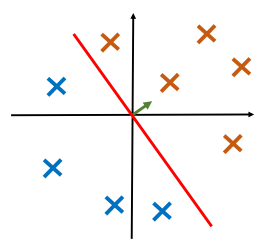
- Choose ε so change is not too big or too small – "**step size**"

Intuition

- $x_j^i\left(y^i - g(w^Tx^i)\right)$
  - If y^i=1 and $g(w^Tx^i)$=0, and x^i_j>0, make w_j larger and push w^Tx^i to be larger
  - If y^i=0 and $g(w^Tx^i)$=1, and x^i_j>0, make w_j smaller and push w^Tx^i to be smaller

15

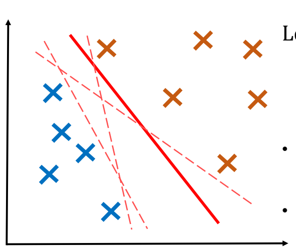
Separating boundary, defined by w



- Separating **hyperplane** splits **class 0** and **class 1**
- Plane is defined by line **w** perpendicular to plan
- Is data point **x** in class 0 or class 1? $\mathbf{w}^T\mathbf{x}$ > 0 class **1**
  $\mathbf{w}^T\mathbf{x}$ < 0 class **0**

16

But, where do we place the boundary?



Logistic regression:

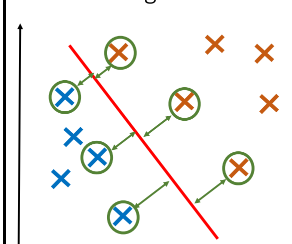
$LL(y|x;w)$:

$$\sum_i (y^i-1)\boldsymbol{w}^T\boldsymbol{x}^i - \log\left(1+e^{-\boldsymbol{w}^T\boldsymbol{x}^i}\right)$$

- Each data point x^i considered for boundary w
- Outlier data pulls boundary towards it

17

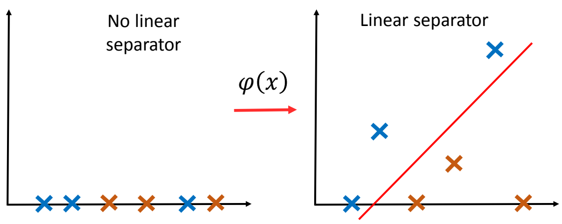
Max margin classifiers



- Focus on boundary points
- Find largest margin between boundary points on both sides
- Works well in practice
- We can call the boundary points **"support vectors"**

18

Classifying with additional dimensions



19

Mapping function(s)

- Map from low-dimensional space $x = (x_1, x_2)$ to higher dimensional space $\varphi(x) = (x_1, x_2, x_1^2, x_2^2, x_1 x_2)$
- N data points guaranteed to be separable in space of N-1 dimensions or more

$$\boldsymbol{w} = \sum_i \boldsymbol{\alpha_i \varphi(x_i) y_i}$$

Classifying x_j:

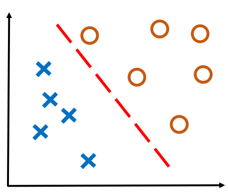
$$\sum_i \alpha_i y_i \varphi^T(x_i)\varphi(x_j) + b$$

20

Discriminative classifiers

Find a separator to minimize classification error

- Logistic Regression
- Support Vector Machines



21

Logistic Regression review

Logistic function
$g(h) = \frac{1}{1 + e^{-h}}$

- $p(y = 0|x;\theta) = 1 - g(w^T x) = \frac{e^{-w^T x}}{1+e^{-w^T x}}$
- $p(y = 1|x;\theta) = g(w^T x) = \frac{1}{1+e^{-w^T x}}$
- Maximize likelihood:
  - $\underset{w}{\operatorname{argmax}} L(y|x;w) = \prod_i \left(1 - g(x^i;w)\right)^{(1-y^i)} g(x^i;w)^{y^i}$
  - Likelihood is $P(D|\theta)$: $D = \{(x^i, y^i)\}, \theta = w$
  - Update w : $\frac{\partial}{\partial w_j} LL(y|x;w) = \sum_i x_j^i \left(y^i - g(w^T x^i)\right)$

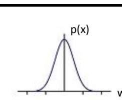
22

MAP for discriminative classifier

- MLE: P(y=1|x;w) ~ g(w^Tx), P(y=0|x;w) ~ 1-g(w^Tx)
- MAP: P(y=1,w|x) ∝ P(y=1|x;w) P(w) ~ g(w^Tx) ???
  (different from Bayesian posterior)
- P(w) priors
  - L2 regularization – minimize all weights
  - L1 regularization – minimize number of non-zero weights

23

MAP – L2 regularization



- P(y=1,w|x) ∝ P(y=1|x;w) P(w):

$$L(y,w|x) = \prod_i \left(1 - g(x^i;w)\right)^{(1-y^i)} g(x^i;w)^{y^i} \quad \prod_j e^{-\frac{w_j^2}{2\lambda}}$$

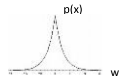
$$LL(y,w|x) = \sum_i y^i w^T x^i - w^T x^i + \log\left(g(w^T x^i)\right) - \sum_j \frac{w_j^2}{2\lambda}$$

$$\frac{\partial}{\partial w_j} LL(y,w|x) = \sum_i x_j^i \left(y^i - g(w^T x^i)\right) - \frac{w_j}{\lambda}$$

Prevent w^Tx from getting too large

24

MAP – L1 regularization



- P(y=1,w|x) ∝ P(y=1|x;w) P(w):

$$L(y,w|x) = \prod_i \left(1-g(x^i;w)\right)^{(1-y^i)} g(x^i;w)^{y^i} \prod_j e^{-\frac{|w_j|}{\lambda}}$$

$$LL(y,w|x) = \sum_i y^i w^T x^i - w^T x^i + \log\left(g(w^T x^i)\right) - \sum_j \frac{|w_j|}{\lambda}$$

$$\frac{\partial}{\partial w_j} LL(y,w|x) = \sum_i x_j^i\left(y^i - g(w^T x^i)\right) - \frac{\text{sign}(w_j)}{\lambda}$$

Force most dimensions to 0

25

Parameters for learning

$$w_j \leftarrow w_j + \varepsilon\left[x_j^i\left(y^i - g(w^T x^i)\right) - \frac{w_j}{\lambda N}\right]$$

- Regularization: selecting λ influences the strength of your bias
- Gradient ascent: selecting ε influences the effect of individual data points in learning
- Bayesian: selecting β_j indicates the strength of the class prior beliefs
- $\lambda, \varepsilon, \beta_j$ are parameters controlling our learning

26

Multi-class logistic regression: class probability

Recall binary class:

- $p(y=0|x;\theta) = 1 - g(w^T x) = \frac{e^{-w^T x}}{1+e^{-w^T x}}$
- $p(y=1|x;\theta) = g(w^T x) = \frac{1}{1+e^{-w^T x}}$

Multi-class – m classes:

- $p(y=j|x;\theta) = \frac{1}{e^{-w_j^T x} + \sum_{k=1}^{m-1} \frac{e^{-w_j^T x}}{e^{-w_k^T x}}}$
- $p(y=m|x;\theta) = 1 - \sum_{j=1}^{m-1} p(y=j|x;\theta)$

27

Multi-class logistic regression: likelihood

Recall binary class:

- $L(y|x;\theta) = \prod_i p(y^i=0|x^i;\theta)^{(1-i)} p(y^i=1|x^i;\theta)^i$
- $\frac{\partial}{\partial w_j} LL(y|x;w) = \sum_i x_j^i\left(y^i - g(w^T x^i)\right)$

$$\boldsymbol{\delta(a)} = \begin{cases} \mathbf{1} & \textbf{if } \boldsymbol{a=0} \\ \mathbf{0} & \textbf{otherwise} \end{cases}$$

Multi-class:

- $L(y|x;\theta) = \prod_{i,k} p(y^i=k|x^i;\theta)^{\delta(y^i-k)}$
- $\frac{\partial}{\partial w_j} LL(y|x;w) = \sum_{i,k} x_j^i\left(\delta(y^i-k) - p(y^i=k|x^i;\theta)\right)$

28

Multi-class logistic regression: update rule

Recall binary class:

- $w_j \leftarrow w_j + \varepsilon x_j^i\left(y^i - g(w^T x^i)\right)$

Multi-class:

$$\boldsymbol{\delta(a)} = \begin{cases} \mathbf{1} & \textbf{if } \boldsymbol{a=0} \\ \mathbf{0} & \textbf{otherwise} \end{cases}$$

- $w_{j,k} \leftarrow w_{j,k} + \varepsilon x_j^i\left(\delta(y^i-k) - p(y^i=k|x^i;\theta)\right)$

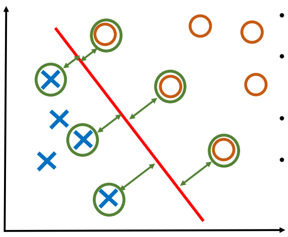
29

Logistic regression: how many parameters?

- N features
- M classes
- Learn w_k for each of M-1 classes: N (M-1) parameters
- Actually: $w^T x = \sum_j w_j x_j$
- Would be better to allow offset from origin: $w^T x + b$: N+1 parameters per class
- Total (N+1) (M-1) parameters

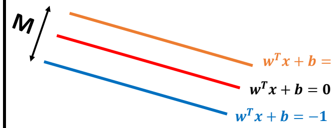
30

Max margin classifiers



- Focus on boundary points
- Find largest margin between boundary points on both sides
- Works well in practice
- We can call the boundary points **"support vectors"**

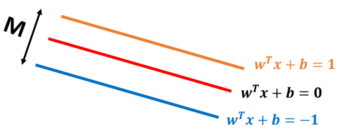
Maximum margin definitions



Classify as +1 if $w^Tx+b\geq 1$
Classify as -1 if $w^Tx+b\leq -1$
Undefined if $-1\leq w^Tx+b\leq 1$

- M is the margin width
- x⁺ is a +1 point closest to boundary, x⁻ is a -1 point closest to boundary
- $x^+ = \lambda w + x^-$
- $|x^+ - x^-| = M$

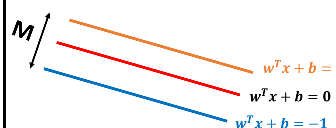
λ derivation



- $w^Tx^- + b = -1$
- $w^Tx^+ + b = +1$
- $x^+ = \lambda w + x^-$

- $w^Tx^+ + b = +1$
- $w^T(\lambda w + x^-) + b = +1$
- $\lambda w^T w + w^T x^- + b = +1$
- $\lambda w^T w - 1 - b + b = +1$
- $\lambda = \frac{2}{w^T w}$

M derivation



- $w^Tx^- + b = -1$
- $w^Tx^+ + b = +1$
- $x^+ = \lambda w + x^-$
- $|x^+ - x^-| = M$

- $M = |\lambda w + x^- - x^-| = |\lambda w| = \lambda |w|$
- $M = \lambda\sqrt{w^T w}$
- $M = \frac{2}{w^T w}\sqrt{w^T w} = \frac{2}{\sqrt{w^T w}}$

maximize M minimize $w^T w$

Support vector machine (SVM) optimization

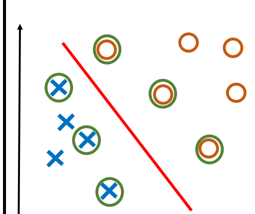
- $\text{argmax}_{w,b}\ M = \frac{2}{\sqrt{w^T w}}$

$\text{argmin}_{w,b}\ w^T w$
subject to
$w^Tx+b\geq 1$ for x in class +1
$w^Tx+b\leq -1$ for x in class -1

Optimization with constraints: $\frac{\partial}{\partial w_j} f(w_j) = 0$ with Lagrange multipliers.

- Gradient descent
- Matrix calculus

Alternate SVM formulation



$w = \sum_i \alpha^i\ x^i y^i$

Support vectors x^i have $\alpha^i > 0$

y_i are the data labels +1 or -1

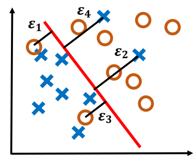
To classify sample x^j, compute:

$w^T x^j + b = \sum_i \alpha^i\ y^i (x^i)^T x^j + b$

$\alpha^i \geq 0\ \forall i$ $\sum_i \alpha^i\ y^i = 0$

Support vector machine (SVM) optimization
with slack variables

What if data not linearly separable?



$\text{argmin}_{w,b}\ \boldsymbol{w}^T\boldsymbol{w} + C\sum_i \varepsilon_i$

subject to

$\boldsymbol{w}^T\boldsymbol{x} + b \geq 1 - \varepsilon_i$ for x in class 1

$\boldsymbol{w}^T\boldsymbol{x} + b \leq -1 + \varepsilon_i$ for x in class -1

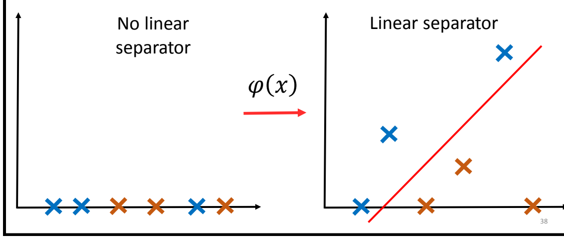
$\varepsilon_i \geq 0 \quad \forall i$

Each error ε_i is penalized based on distance from separator

37

Classifying with additional dimensions

Note: More dimensions makes it easier to separate N training points: training error minimized, may risk over-fit



38

Quadratic mapping function

$w^T x^j + b = \sum_i \alpha^i y^i (x^i)^T x^j + b$

- x_1, x_2, x_3, x_4 -> $x_1, x_2, x_3, x_4, x_1^2, x_2^2, x_3^2, x_4^2, x_1x_2, x_1x_3, x_1x_4, \ldots, x_2x_4, x_3x_4$
- N features -> $N + N + \frac{N\times(N-1)}{2} \approx N^2$ features
- N^2 values to learn for w in higher-dimensional space

v with N elements operating in quadratic space

- Or, observe: $(v^T x + 1)^2 = v_1^2 x_1^2 + \cdots + v_N^2 x_N^2$
  $+ v_1 v_2 x_1 x_2 + \cdots + v_{N-1} v_N x_{N-1} x_N$
  $+ v_1 x_1 + \cdots + v_N x_N$

39

Kernels

$w^T x^j + b = \sum_i \alpha^i y^i (x^i)^T x^j + b$

Classifying $\boldsymbol{x}^j$:

$$\sum_i \alpha_i y_i \varphi(\boldsymbol{x}^i)^T \varphi(\boldsymbol{x}^j) + b$$

Kernel trick:

- Estimate high-dimensional dot product with function
- $K(\boldsymbol{x}^i, \boldsymbol{x}^j) = \varphi(\boldsymbol{x}^i)^T \varphi(\boldsymbol{x}^j)$
- E.g., $K(\boldsymbol{x}^i, \boldsymbol{x}^j) = \exp\left(-\frac{(x^i - x^j)^2}{2\sigma^2}\right)$

40

The power of SVM (+kernels)

- Boundary defined by a few support vectors
  - Caused by: maximizing margin
  - Causes: less overfitting
  - Similar to: regularization
- Kernels keep number of learned parameters in check

41

Multi-class SVMs

- Learn boundary for class k vs all other classes
- Find boundary that gives highest margin for data point x^i

42

Benefits of generative methods

- $P(D|\theta)$ and $P(\theta|D)$ can generate non-linear boundary
- E.g.: Gaussians with multiple variances