Machine Learning
Week 6 Lecture: - Classification and Evaluation

Debasis Ganguly
Debasis.Ganguly@glasgow.ac.uk

School of Computing Science
University of Glasgow

November 5, 2025

ML course so far

- Supervised learning
  - Regression
    - Minimised loss (least squares)
    - Maximised likelihood
    - Bayesian approach
  - **Classification** (This and the next 3 weeks)
    - Logistic Regression, Evaluation (Week 6)
    - Softmax Regression, Naive Bayes, K-NN (Week 7)
    - Bayesian approaches for Classification (Week 8)
    - Support Vector Machines (Week 9)
- **Unsupervised learning**
  - Clustering (Week 10)

A word about the rest of the course

A bit more Dense!

- More topics covered (but there're similarities with what you have learned in the first 5 weeks).
- Make sure that you use the lab sessions and the open office hours to your advantage.

A bit more Mathy!

- It's okay if you don't understand every detail (**you don't have to do any math in the exam**!)
- Some concepts are actually easier to understand with notations.

Anonymous feedback: `https://tinyurl.com/5eynv5cb`

Classification vs. Regression

Similarity with regression

- Learn a map $\theta : \mathbf{x} \mapsto y$.
- Inputs: $\mathbf{x}$ feature vector representation of an instance x.
  - E.g., if x represents housing data then $(backyard\_area, postcode, \#bedrooms, \ldots)$.

Differences with regression

- Given an $\mathbf{x}$ the **task is to predict** a y, where $y \in \{0, 1, \ldots, k-1\}$ – a **categorical variable** of k possible values (also abbv. as $y \in \mathbb{Z}_k$).
  - For a given house with values of backyard_area, postcode etc., one may predict the house-price range as $\{low, medium, high\}$ $(k = 3)$.
- Recall for regression: $y \in \mathbb{R}$.

Regression vs. Classification

Predicting continuous values by regression

- **Input**: a **training set** of labelled data: $\mathcal{T} = \{(\mathbf{x}^{(i)}, y^{(i)})\}$, where each $\mathbf{x}^{(i)} \in \mathbb{R}^d$ (d dimensional feature vectors or embeddings) and each $y^{(i)} \in \mathbb{R}$ (a continuous value).
- **Output**: A model vector $h_\theta(\mathbf{x}) = \theta^T\mathbf{x}$ or (in this course, we will interchangeably use both θ and $\mathbf{w}$ to denote parameter vectors).
- **Predictions**: $\hat{y}(\mathbf{x}_{new}) = h_\theta(\mathbf{x}_{new}) = \theta^T\mathbf{x}_{new}$

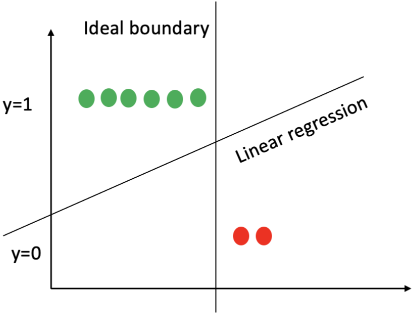
What needs to change?

- Obviously, the output $\hat{y}$ is now **constrained** (for now, let's say between 0 and 1).
- So, $\theta^T\mathbf{x}$ **doesn't work**. Why?
  - Because $\theta^T\mathbf{x}$ is a line in the data space ($\notin \{0, 1\}$).

Regression vs. Classification

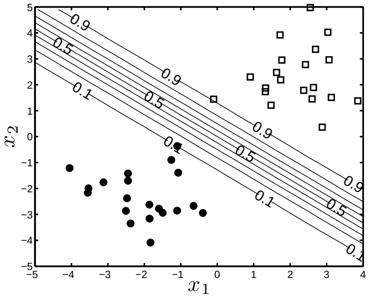
Does the naive solution work?

- Wait! But can't we just fit a line by **linear regression** and then treat that line as a threshold (decision boundary)?
  - $\arg\min_\theta J(\theta) = (y - \underbrace{\theta^T \mathbf{x}}_{\text{linear activation}})^2$?
  - $\hat{y} = 1$ if $\theta^T \mathbf{x} > 0$?.



- A closely fitting line isn't a good decision boundary!

What can we do about it (Intuitively)?



- Interpretation of the line $\theta^T\mathbf{x}$: Needs to change from a "**good fit**" to a **"decision boundary"**.

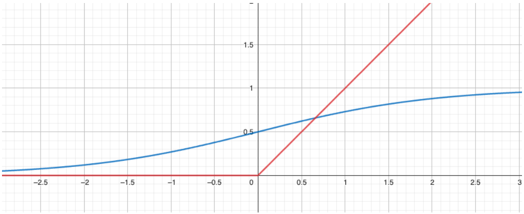
- Points just on the line – maximum uncertainty or $P(y|\mathbf{x}) = 0.5$.
- Points just above the line $\hat{y} = 1$ and $P(y|\mathbf{x}) \to 1$ with increasing distances from the boundary.
- Predictions then become $g(\theta^T\mathbf{x})$, where g is (**has to be a non-linear**) function such that:
  - $g(0) = 0.5$
  - $g(z) \to 1$ as $z \to \infty$
  - $g(z) \to 0$ as $z \to -\infty$

Activation Functions

Non-linearity

- Have seen this before?
- In the context of deep learning (stacked layers of parameter vectors)

- The class of functions $g(z) \in [0,1]$ with $g(0) = 0.5$ are called activation functions.
  - Sigmoid: $g(z) = \frac{1}{1+exp(-z)}$
  - Relu: $g(z) = \max(0, z)$



Model thus changes to: $h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$

What's the loss function to minimize?

Mean Square Loss (MLS): $L(\theta) = \arg\min_{\theta}(y - g(\theta^T \mathbf{x}))^2$

Menti-Quiz (Code: 8867-8596)

Why is mean square loss suitable for classification?

- Okay, but our *desired interpretation* of $g(\theta^T \mathbf{x})$ is that it's a **probability**.
- Isn't it then weird to "**differences with probabilities**" in $(y - g(\theta^T \mathbf{x}))^2$?

Cross-Entropy Loss

Bernoulli Distribution

$$P(y|\mathbf{x};\theta) = \begin{cases} g(\theta^T\mathbf{x}), & \text{if } y = 1 \\ 1 - g(\theta^T\mathbf{x}), & \text{if } y = 0 \end{cases}$$

Cross-Entropy Likelihood:
$P(y|\mathbf{x};\theta) = g(\theta^T\mathbf{x})^y(1 - g(\theta^T\mathbf{x}))^{(1-y)}$

- Important: $y \in \{0, 1\}$; $g(\theta^T\mathbf{x}) \in (0, 1)$.

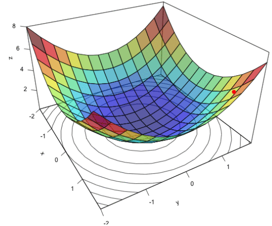
Menti-Quiz (Code: 8867-8596)

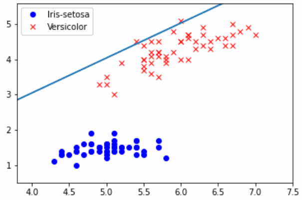
When is cross-entropy likelihood maximized and when is it minimized? Hint: Work out the 4 possible cases — **target=1 vs. high activation**, and so on.

Loss/Objective function

- $\arg\max_\theta \log P(y|\mathbf{x};\theta)$ is a valid objective function.
- And so is: $\arg\min_\theta - \log P(y|\mathbf{x};\theta)$ (called **negative log likelihood** or the **cross-entropy loss**).

Visualizing the parameter space





- A **line** (boundary) in the **data space** $\equiv$ A **point** in the **parameter space**.
- Example: For a 2D data space (x_1, x_2), the parameter vector is three dimensional $\theta = (\theta_0, \theta_1, \theta_2)$.
- $y = 1$ for a point $\mathbf{x} = (x_1, x_2)$ if $\theta_0 + \theta_1 x_1 + \theta_2 x_2 > 0$

Gradient Descent

- Finding way through the parameter space requires computing **gradients** $\rightarrow$ direction we should walk!
- For linear regression, we needed to compute gradients for z, where $z = \theta \cdot \mathbf{x}$ (that was easy! because z is linear!).
- For logistic regression, now we need to compute gradients of $g(z)$, where g is an activation function like the **sigmoid**.

Chain rule of derivative

$$\frac{d}{dx} g(f(x)) = \underbrace{\frac{d}{dy} g(y)}_{\text{gradient of the sigmoid}} \underbrace{\frac{d}{dx} f(x)}_{\text{gradient of the i/p to the sigmoid}}$$

Gradient of the Sigmoid

All we need to know for the computation

$$\frac{d}{dx}(x) = 1 \qquad \frac{d}{dx}(e^x) = e^x$$
$$\frac{d}{dx}(x^n) = nx^{n-1} \qquad \frac{d}{dx}(\log x) = \frac{1}{x}$$

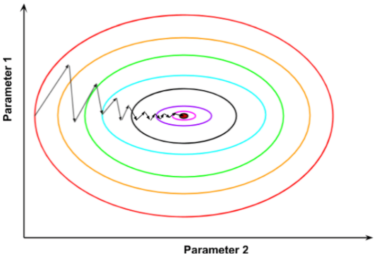
$$\begin{aligned}
g'(z) &= \frac{d}{dz}\frac{1}{1+e^{-z}} \\
&= \boxed{\frac{-1}{(1+e^{-z})^2}}\frac{d}{dz}e^{-z}, \quad \because \frac{d}{dx}\frac{1}{x} = -\frac{1}{x^2} \\
&= \frac{-1}{(1+e^{-z})^2}\boxed{e^{-z}}\frac{d}{dz}(-z), \quad \because \frac{d}{dx}e^x = e^x \\
&= (\frac{1}{1+e^{-z}})(\frac{e^{-z}}{1+e^{-z}}) = \boxed{g(z)\ (1-g(z))}
\end{aligned}$$

Logistic Regression Gradient Updates

Loss function

$$l(\theta) = \log L(\theta) = y \log g(\theta^T \mathbf{x}) + (1 - y)(1 - \log g(\theta^T \mathbf{x}))$$

- Partial derivative wrt one component of the parameter vector.
- Because we need to take a step in a particular direction at a time.



Logistic Regression Gradient Updates

Loss function

$$l(\theta) = \log L(\theta) = y \log g(\theta^T \mathbf{x}) + (1-y)(1 - \log g(\theta^T \mathbf{x}))$$

$$\frac{\partial}{\partial \theta_j} l(\theta) = \left(y \frac{1}{g(\theta^T x)} - (1-y)\frac{1}{1-g(\theta^T x)}\right) \underbrace{\frac{\partial}{\partial \theta_j} g(\theta^T x)}_{\text{we know this!}}$$

Logistic Regression Gradient Updates

Loss function

$$l(\theta) = \log L(\theta) = y \log g(\theta^T \mathbf{x}) + (1-y)(1 - \log g(\theta^T \mathbf{x}))$$

$$\frac{\partial}{\partial \theta_j} l(\theta) = (y \frac{1}{g(\theta^T x)} - (1-y)\frac{1}{1-g(\theta^T x)}) \underbrace{\frac{\partial}{\partial \theta_j} g(\theta^T x)}_{\text{we know this!}}$$

$$(y \frac{1}{g(\theta^T x)} - (1-y)\frac{1}{1-g(\theta^T x)}) \underbrace{g(\theta^T x)(1-g(\theta^T x))}_{\text{substituted!}} \frac{\partial}{\partial \theta_j} \theta^T x$$

Logistic Regression Gradient Updates

- Looks familiar? $\frac{\partial}{\partial \theta_j} \theta^T x$

Menti-Quiz (Code: 8867-8596)

Find out: $\frac{\partial}{\partial \theta_j} \theta^T x$

Logistic Regression Gradient Updates

- Looks familiar? $\frac{\partial}{\partial \theta_j} \theta^T x$

Menti-Quiz (Code: 8867-8596)

Find out: $\frac{\partial}{\partial \theta_j} \theta^T x$

Finally,

$$\frac{\partial}{\partial \theta_j} l(\theta) = (y(1 - g(\theta^T x)) - (1 - y)g(\theta^T x))\, \boxed{x_j}$$

$$= (y - \underbrace{g(\theta^T x)}_{\text{same as linear regression!}})x_j$$

Gradient Descent Algorithm

Negative log-likelihood

$$\frac{\partial}{\partial \theta_j} l(\theta) = (y - h_\theta(x))x_j$$

- The form of this derivative is **identical** to that of the linear regression with square loss.
- However, these are **not the same algorithm**. Because,
  - $h_\theta(\mathbf{x}) = \theta^T \mathbf{x}$ for linear regression
  - $h_\theta(\mathbf{x}) = g(\theta^T \mathbf{x})$ for logistic regression

Generalised Linear Models (Not taught)

In fact, these are same because both linear regression and logistic regression belong to the same family of models.

Logistic Regression Algorithm

Recap

- Activation: $h_\theta(\mathbf{x}) = g(\theta \cdot \mathbf{x}) = 1/(1 + \exp(-\theta \cdot \mathbf{x}))$
- Objective function: $y \log h_\theta(\mathbf{x}) + (1 - y) \log(1 - h_\theta(\mathbf{x}))$

Stochastic Gradient Descent Algorithm

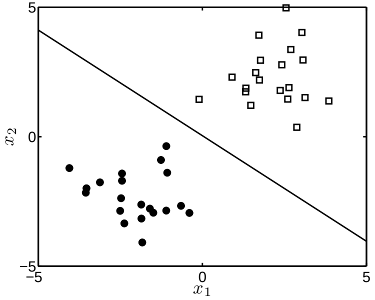
- For each training point $\mathbf{x}^{(i)}$ (one training instance at a time):
  - For each parameter vector component j (one parameter dimension at a time):
    - $\theta_j \leftarrow \theta_j + (y^{(i)} - h_\theta(\mathbf{x}^{(i)}))\mathbf{x}_j^{(i)}$

Menti-Quiz (Code: 8867-8596)

Think about ways of improving and generalising the algorithm (you may work in pairs before submitting a solution).

Decision boundary

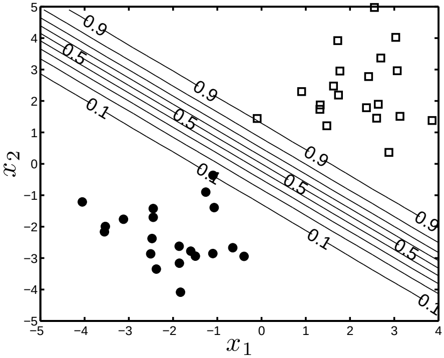
- Once we have θ, we can classify new examples.



Line corresponding to $P(y_{\text{new}} = 1 | \mathbf{x}_{\text{new}}; \theta) = 0.5$

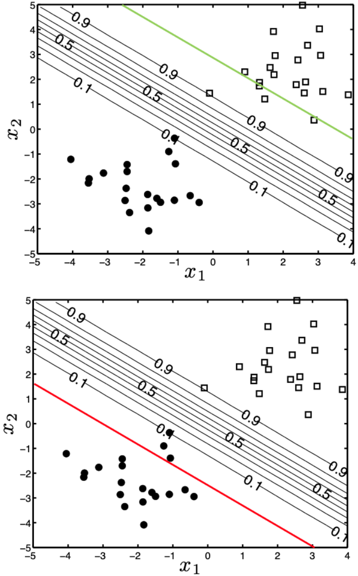
- $\theta^T \mathbf{x}_{new} = 0 \implies \exp(\theta^T \mathbf{x}_{new}) = 1$
- $\frac{1}{1+\exp(-\theta^T \mathbf{x}_{new})} = \frac{1}{2}$
- The classifier is the **most uncertain** (**least confident**) along the boundary.

Contours of the posterior probabilities



- Contours of $P(y_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \theta)$.
- Do they look sensible?

Model Calibration (with thresholds)

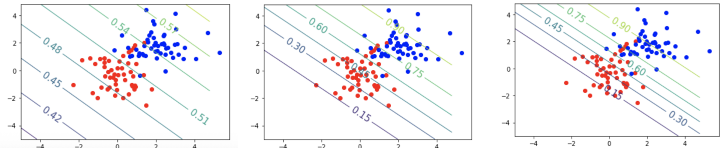


- An easy way to change model predictions to be more conservative towards predicting a particular class.
- Confidence threshold set to a value $\tau \in [0, 1]$.
- Example: Predict $y = 1$ only if $P(y|x_{new}) > \tau$.
- Such calibrations are needed for critical tasks, such as cancer prediction, e.g., predict "`not cancer`" only if confidence > 0.95.

Model Calibration (with temperature)

Generalized Sigmoid

$$g(z) = \frac{1}{1+\exp(-\alpha\theta\cdot\mathbf{x})}$$



Menti-Quiz (Code: 8867-8596)

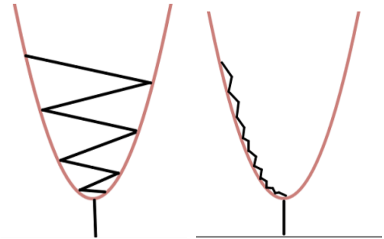
Match each plot with a temperature ($\alpha \in \{1, 10, 50\}$)

- Higher the temperature:
  - Steeper is the activation
  - Narrower are the contour bands (iso-p lines)

Model Calibration (learning rate)

Generalized Gradient Descent

$\theta_j \leftarrow \theta_j + \boxed{\nu}(y - \theta \cdot \mathbf{x})x_j$

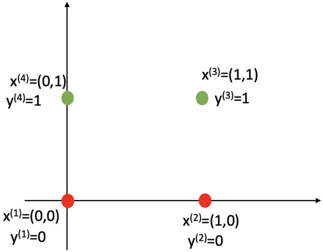


Menti-Quiz (Code: 8867-8596)

Which one (left or right) has a higher ν?

- Higher the learning rate (ν), bigger the steps:
  - Faster convergence.
  - More Oscillations

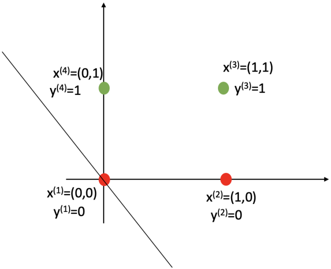
Logistic Regression Working Example



- Let's take $\theta = (0, 0.1, 0.1)$ (the first term being the bias term). $\theta \in \mathbb{R}^3$.
- Each $\mathbf{x}$ needs to be prepended with a '1', e.g., we work with $\mathbf{x}^{(1)} = (1, 0, 0)$.

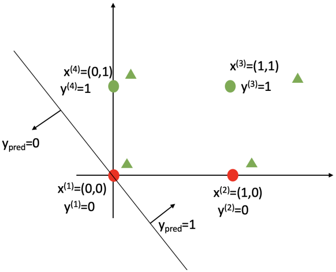
Logistic Regression Working Example

- How do we plot the decision boundary?
  - Remember at the boundary: $\theta \cdot \mathbf{x} = 0$.
- Visually plot the line $\theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$.
  - Substituting: $0 + 0.1x_1 + 0.1x_2 = 0 \implies x_2 = -x_1$.
- Boundary as shown below.



Logistic Regression Working Example

- Is our boundary good?
  - Let's denote our predictions with triangles.
  - How many misclassifications? 2.
- Now let's see how we can modify the boundary to do better.



Logistic Regression Working Example

Gradient Update

$$\theta_j \leftarrow \theta_j + (y^{(i)} - sigmoid(\theta \cdot \mathbf{x}^{(i)}))x_j^{(i)}.$$

- **Select a point** and **a component**.
  - Let's take the point $\mathbf{x}^{(2)}$. (Note that this is a point for which the current classifier makes a mistake!).
  - And take the second component, i.e., we update θ_1.

Logistic Regression Working Example

Gradient Update

$\theta_j \leftarrow \theta_j + (y^{(i)} - sigmoid(\theta \cdot \mathbf{x}^{(i)}))x_j^{(i)}$.

- **Select a point** and **a component**.
  - Let's take the point $\mathbf{x}^{(2)}$. (Note that this is a point for which the current classifier makes a mistake!).
  - And take the second component, i.e., we update θ_1.
- Compute:
  - $\theta \cdot \mathbf{x}^{(2)} = (0, 0.1, 0.1) \cdot (1, 1, 0) = 0 \times 1 + 0.1 \times 1 + 0.1 \times 0 = \boxed{0.1}$.

Logistic Regression Working Example

Gradient Update

$$\theta_j \leftarrow \theta_j + (y^{(i)} - sigmoid(\theta \cdot \mathbf{x}^{(i)}))x_j^{(i)}.$$

- **Select a point** and **a component**.
  - Let's take the point $\mathbf{x}^{(2)}$. (Note that this is a point for which the current classifier makes a mistake!).
  - And take the second component, i.e., we update θ_1.
- Compute:
  - $\theta \cdot \mathbf{x}^{(2)} = (0, 0.1, 0.1) \cdot (1, 1, 0) = 0 \times 1 + 0.1 \times 1 + 0.1 \times 0 = \boxed{0.1}$.
  - $sigmoid(\theta \cdot \mathbf{x}^{(2)}) = 1/(1 + \exp(-\boxed{0.1})) = \exp(0.1)/(1 + \exp(0.1)) = 1.1/2.1 = 0.52$.

Logistic Regression Working Example

Gradient Update

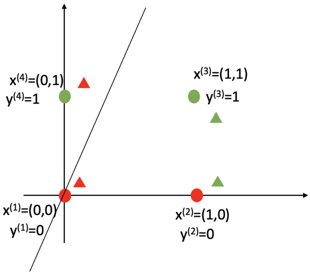
$\theta_j \leftarrow \theta_j + (y^{(i)} - sigmoid(\theta \cdot \mathbf{x}^{(i)}))x_j^{(i)}$.

- **Select a point** and **a component**.
  - Let's take the point $\mathbf{x}^{(2)}$. (Note that this is a point for which the current classifier makes a mistake!).
  - And take the second component, i.e., we update θ_1.
- Compute:
  - $\theta \cdot \mathbf{x}^{(2)} = (0, 0.1, 0.1) \cdot (1, 1, 0) = 0 \times 1 + 0.1 \times 1 + 0.1 \times 0 = \boxed{0.1}$.
  - $sigmoid(\theta \cdot \mathbf{x}^{(2)}) = 1/(1 + \exp(-\boxed{0.1})) = \exp(0.1)/(1 + \exp(0.1)) = 1.1/2.1 = 0.52$.
- Now modify θ_1.
  - $\theta_1 \leftarrow \boxed{0.1} + (0 - \boxed{0.52}) \times 1$.
- New parameter vector: $\theta = (0, -0.42, 0.1)$.

Logistic Regression Working Example

- How does the new boundary look like?
  - New parameter vector: $\theta = (0, -0.42, 0.1)$.
  - Put it in the slope intercept form.

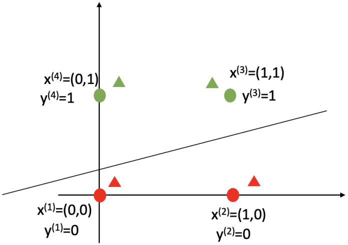
$$-0.42x_1 + 0.1x_2 = 0, \text{ i.e., } x_2 = \frac{0.42}{0.1}x_1 = 4.2x_1$$



- #Misclassifications: still 2; but we're making progress towards the ideal boundary.

Logistic Regression Working Example

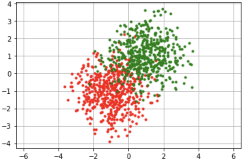
- Homework: Work out on paper one more update.
- If you run the updates for an adequate number of times (say 5 times), what do you expect the decision boundary to be?



- If the classes are linearly separable logistic regression is guaranteed to converge to the perfect classifier.

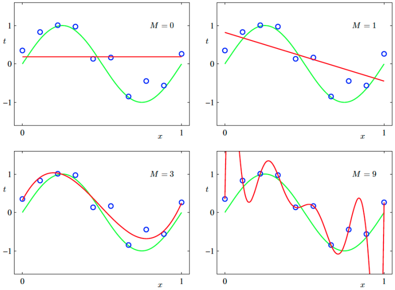
What do we do for linearly inseparable classes?

- Logistic regression algorithm **can yield non-linear decision boundaries**.



- Use a feature map function of higher order features.
  - $\mathbf{x} = (x_1, \ldots, x_d)$
  - Higher order feature map examples:
    - $\phi_1(\mathbf{x}) = (x_1, \ldots, x_d, x_1^2, \ldots, x_d^2)$
    - $\phi_2(\mathbf{x}) = (x_1, \ldots, x_d, x_1 x_2, \ldots, x_{d-1} x_d)$
  - Apply logistic regression on $\phi(\mathbf{x})$ instead of on $\mathbf{x}$.
- This means that decision boundary $\theta_1 x_1 + \theta_{d+1} x_1^2 + \ldots$ has now non-linear terms.

Problems of using a feature map



- The same problem as in linear regression - **Overfitting**.
- **Solution**: Encourage sparse solutions - higher order features but you don't want the boundary to depend on too many of them.
  - Modify objective function with an added term for the norm of the parameter vector
  - Penalize those θs whose norm is high.

- Minimise: $J(\theta) = -(h_\theta(\mathbf{x}) \log y + (1 - h_\theta(\mathbf{x})) \log(1 - y)) + \boxed{||\theta||}$

Classifier Performance Evaluation

- How do we know how accurate are our predictions?
  - Which algorithm? Regularization or without
  - What model calibration?
- Need performance indicators.

Classifier Performance Evaluation

- How do we know how accurate are our predictions?
  - Which algorithm? Regularization or without
  - What model calibration?
- Need performance indicators.
- We'll cover:
  - Accuracy
  - Precision/Recall
  - Confusion matrix and Precision-recall curves

Accuracy

- How many correct classifications in total.
- Consider a set of predictions $\hat{y}_1, \ldots, \hat{y}_N$ and a set of true labels $y_1^{,} \ldots, y_N$.
- Mean accuracy is defined as:

$$A = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\hat{y}_i = y_i)$$

- $\mathbb{I}(A)$ is 1 if A is true and 0 otherwise

Accuracy

- How many correct classifications in total.
- Consider a set of predictions $\hat{y}_1, \ldots, \hat{y}_N$ and a set of true labels $y_1, \ldots, y_N$.
- Mean accuracy is defined as:

$$A = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\hat{y}_i = y_i)$$

- $\mathbb{I}(A)$ is 1 if A is true and 0 otherwise
- Advantages:
  - Can do binary or multi-class classification.
  - Simple to compute.
  - Single value.

Accuracy

Disadvantage: Doesn't take into account **class imbalance**.

- We're building a classifier to detect a rare disease.
- Assume only 1% of population is diseased.
- Diseased: $y = 1$
- Healthy: $y = 0$
- What if we always predict healthy? ($y = 0$)
- Accuracy 99%
- But classifier is rubbish!

Precision and Recall

- Need to define 4 quantities. The numbers of:

Precision and Recall

- Need to define 4 quantities. The numbers of:
- **True positives (TP)** – the number of objects with $y = 1$ that are classified as $\hat{y} = 1$ (diseased people diagnosed as diseased).

Precision and Recall

- Need to define 4 quantities. The numbers of:
- True positives (TP) – the number of objects with $y = 1$ that are classified as $\hat{y} = 1$ (diseased people diagnosed as diseased).
- **True negatives (TN)** – the number of objects with $y = 0$ that are classified as $\hat{y} = 0$ (healthy people diagnosed as healthy).

Precision and Recall

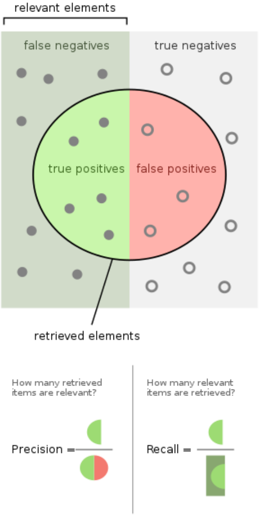
- Need to define 4 quantities. The numbers of:
- True positives (TP) – the number of objects with $y = 1$ that are classified as $\hat{y} = 1$ (diseased people diagnosed as diseased).
- True negatives (TN) – the number of objects with $y = 0$ that are classified as $\hat{y} = 0$ (healthy people diagnosed as healthy).
- **False positives (FP)** – the number of objects with $y = 0$ that are classified as $\hat{y} = 1$ (healthy people diagnosed as diseased).

Precision and Recall

- Need to define 4 quantities. The numbers of:
- True positives (TP) – the number of objects with $y = 1$ that are classified as $\hat{y} = 1$ (diseased people diagnosed as diseased).
- True negatives (TN) – the number of objects with $y = 0$ that are classified as $\hat{y} = 0$ (healthy people diagnosed as healthy).
- False positives (FP) – the number of objects with $y = 0$ that are classified as $\hat{y} = 1$ (healthy people diagnosed as diseased).
- **False negatives (FN)** – the number of objects with $y = 1$ that are classified as $\hat{y} = 0$ (diseased people diagnosed as healthy).

Visual representation of precision and recall

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN}$$



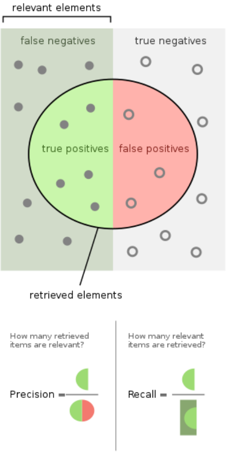
- Effective for class imbalanced problems.
- Two different performance measures used for specific types of tasks - **precision oriented** or **recall oriented**.
- Trade-off between the two. Why?

Menti-Quiz (Code: 8867-8596)

Think an example each for a precision and a recall oriented task.

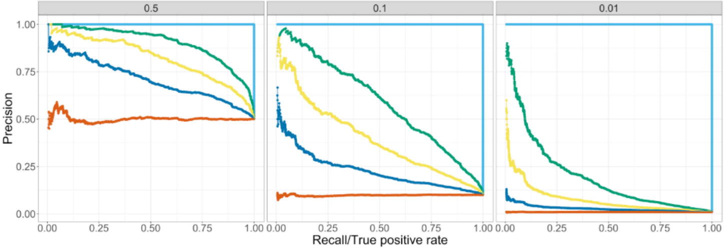
- Detecting cancer cells from images is a recall-oriented task. Why?
- Whether to turn an AI-driven car is a precision-oriented task. Why?

Precision-Recall Trade-off



- Useful to analyze the effects of model calibration.
- Default threshold in logistic regression is 0.5.
- However, we could use any threshold we like.
- Threshold$\uparrow$:
  - Precision$\uparrow$, Recall$\downarrow$.
- Threshold$\downarrow$:
  - Precision$\downarrow$, Recall$\uparrow$.

Precision-Recall Curves



- With an increase from low to high recall, the precision should stay as high as possible for an ideal classifier.
- Each error brings precision down.

A word about the practical coursework (25%)

- Will be released early next week (or sooner!) — this **won't be just a coding exercise**!.
- Task: Breast Cancer Detection from tabular data.
- **Task-1**: **Ensemble Model**
  - Teach yourself about model-ensembles or mixture of experts (we will not cover this in lectures).
  - Implement an ensemble of logistic regression trained over different partitions of the data.
- **Task-2**: Explore model calibration to **maximise recall**.
  - Write a report **explaining** and **motivating** your design choices and hyper-parameters.