

Machine Learning

Week 7 Lecture: Softmax Regression, Non-Parametric Classification, and Mixture of Experts

Multi-class
Classification

Multi-Label
Classification

K-NN classifier

Mixture of Experts

Debasis Ganguly

School of Computing Science
University of Glasgow

November 13, 2025

Multi-class Classification

Binary Classification



- ▶ Target values can only be binary, i.e. $y \in \{0, 1\}$.
- ▶ Common applications - sentiment classification, spam detection.
- ▶ Model:
$$h_{\theta}(\mathbf{x}) = 1/(1 + \exp(-\theta \mathbf{x}))$$

Multi-class Classification



- ▶ More general
- ▶ Target variables can be categorical, i.e., $y \in \{0, 1, \dots, k - 1\}$.
- ▶ Categories can be mapped to integers.
- ▶ More practical use-case than binary classification.
- ▶ Model:
$$h_{\theta}(\mathbf{x}) = 1/(1 + \exp(-\theta \mathbf{x}))?$$

Multi-class
Classification

Multi-Label
Classification

K-NN classifier

Mixture of Experts

Motivation

- ▶ Does $\underbrace{h_{\theta}(\mathbf{x}) = 1/(1 + \exp(-\theta \mathbf{x}))}_{\text{Sigmoid}}$ work for categorical values, i.e., when $y \in \{0, 1, \dots, k - 1\}$?
- ▶ We need to first verify (just as we did from linear regression \mapsto logistic regression) if slight modifications to existing solutions work.
- ▶ Recall from last week that:

Linear regression

- ▶ Target variables are unbounded.
- ▶ Objective function treats the parameter vector (line in the data space) as a **close fit**.

Logistic regression

- ▶ Target values in $\{0, 1\}$.
- ▶ Objective function has to treat the parameter vector as a **decision boundary**.

Logistic Regression for Multiple class prediction

- ▶ For k classes, draw $k - 1$ decision boundaries - **one for each class.**

- ▶ $\mathbf{x} \in C_k$ if $\mathbf{x} \notin C_i$ ($i = 1, \dots, k - 1$).

$$\text{▶ } h_{\theta}^{(i)}(\mathbf{x}) = 1/(1 + \exp(-\theta^{(i)}\mathbf{x}))$$

- ▶ Interpretation of i^{th} boundary:

▶ **Positive part of the hyper-plane**
 $(h_{\theta}^{(i)}(\mathbf{x}) > 0)$: Member of the i^{th} class C_i .

▶ **Negative part of the hyper-plane**: Not a member of the i^{th} class C_i .

- ▶ Leads to **ambiguous regions**.

▶ Green Region: Both C_1 and C_2 !

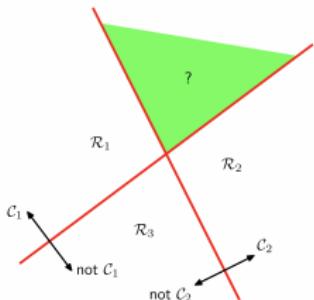


Figure: One vs. Rest

Logistic Regression for Multiple class prediction

- ▶ Learn a decision boundary for each pair to distinguish between C_i and C_j ($j \neq i$).
- ▶ How many boundaries?

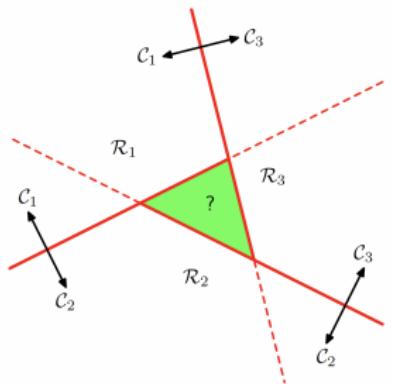


Figure: Pairwise

Logistic Regression for Multiple class prediction

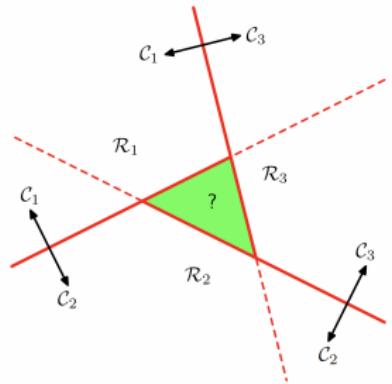
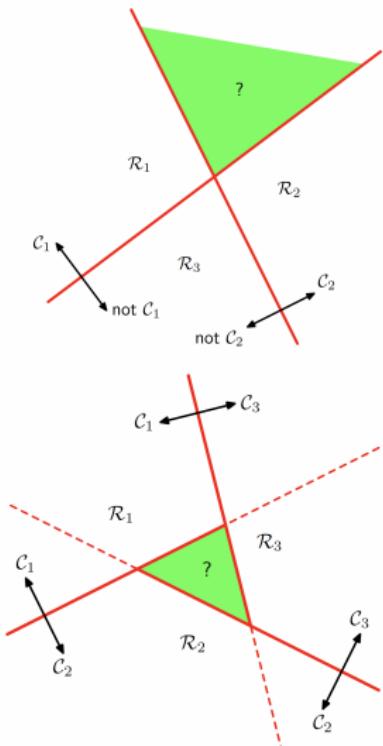


Figure: Pairwise

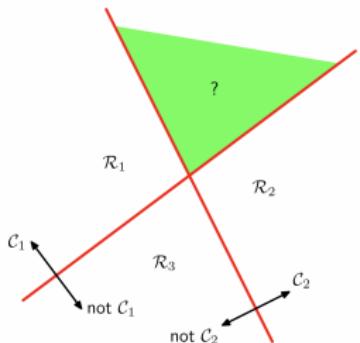
- ▶ Learn a decision boundary for each pair to distinguish between C_i and C_j ($j \neq i$).
- ▶ How many boundaries?
 - ▶ Number of unordered pairs (combinations): $k(k - 1)/2$.
- ▶ Leads to **ambiguous regions**.
 - ▶ Green Region: Both C_1 , C_2 and C_3 !
- ▶ Quadratic number of decision boundaries → Increased training time!

Can we resolve these ambiguities?



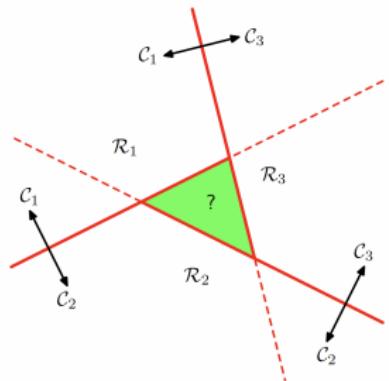
- ▶ Yes, we can by using the posteriors
- ▶ Recall: $P(y|\mathbf{x}; \theta) = h_\theta(\mathbf{x}) = 1/(1 + \exp(-\theta\mathbf{x}))$
 - ▶ Higher $P(y|\mathbf{x}; \theta) \rightarrow$ more confidence!
- ▶ Despite ambiguity of the green regions, we can make a **per-instance decision** to resolve ambiguity!
- ▶ For both ‘one vs. rest’ and pairwise:
 - ▶ $\hat{y} = i$ if $P(\mathbf{x}|\theta^{(i)}) > P(\mathbf{x}|\theta^{(j)}) \forall j$.
- ▶ However, this seems like a hack?

Can we resolve these ambiguities?

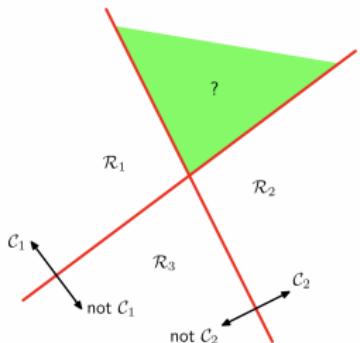


Menti-Quiz (Code: 1373-7721)

Are the posterior probabilities even comparable?

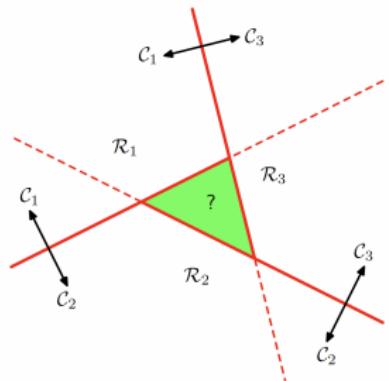


Can we resolve these ambiguities?

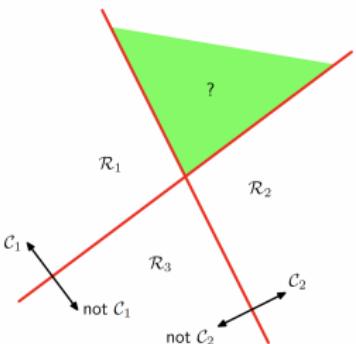


Menti-Quiz (Code: 1373-7721)

Are the posterior probabilities even comparable?



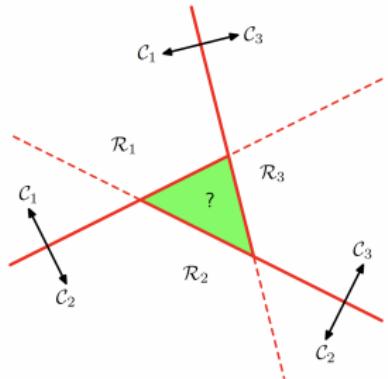
Can we resolve these ambiguities?



Menti-Quiz (Code: 1373-7721)

Are the posterior probabilities even comparable?

What we need:

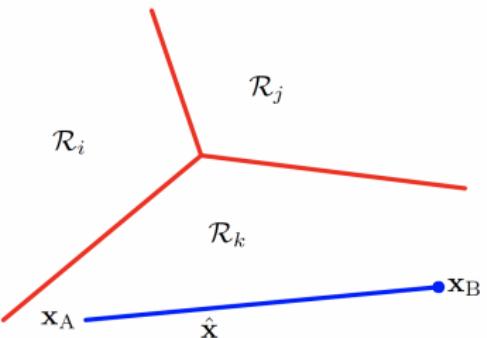
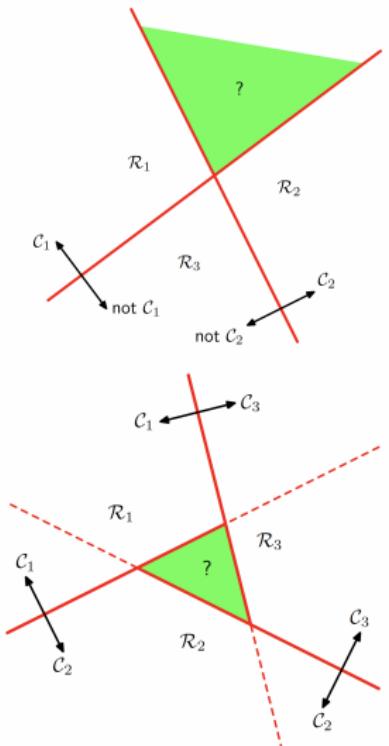


- ▶ A single model that is trained **jointly**.
- ▶ A decision for one class needs to be done in **awareness of the other possibilities**.

Softmax Regression

- ▶ What then needs to be changed?
 - ▶ Short answer: The notion of decision boundary - because it only applies for two classes.
- ▶ Long answer:
 - ▶ Linear \mapsto Logistic: Objective function was made to represent a decision boundary.
 - ▶ Logistic \mapsto Multi-class: Objective function needs to be **jointly modeling the posteriors** of each class.
 - ▶ k -class discriminant: composed of k linear functions **trained jointly**.
- ▶ Joint Modeling means that:
 - ▶ **Not independent decision boundaries** anymore.
 - ▶ Our hacky idea of taking the maximum over the posteriors now works!
 - ▶ Because the **probabilities are now comparable!**

Softmax Regression



- ▶ Joint training ensures:
 - ▶ Piecewise decision boundaries.
 - ▶ **Convex** decision regions.
 - ▶ No ambiguity.

Softmax Regression

Logistic Regression

- ▶ Distribution: Bernoulli
- ▶ $h_\theta(\mathbf{x})^y(1 - h_\theta(\mathbf{x}))^{1-y}$
- ▶ Model:
- ▶
$$h_\theta(\mathbf{x}) = g(\theta\mathbf{x}) = \frac{1}{1 + \exp(-\theta\mathbf{x})}$$

Softmax Regression

- ▶ Distribution: Multinomial

$$h_{\theta^{(1)}}(\mathbf{x})^{y=1} h_{\theta^{(2)}}(\mathbf{x})^{y=2} \dots h_{\theta^{(k)}}(\mathbf{x})^{y=k}$$
- ▶ Model:
- ▶
$$h_{\theta^{(i)}}(\mathbf{x}) = \phi_i / \Phi$$
 - ▶ $\phi_i = \exp(\theta^{(i)}\mathbf{x})$
 - ▶ $\Phi = \sum_{j=1}^k \phi_j$
 - ▶ Belief in class i
relative to the other class beliefs.

Menti-Quiz (Code: 1373-7721)

What is the key insight which tells us that the decision region is convex?

Softmax Regression

Logistic Regression

- ▶ Distribution: Bernoulli
- ▶ $h_\theta(\mathbf{x})^y(1 - h_\theta(\mathbf{x}))^{1-y}$
- ▶ Model:
- ▶
$$h_\theta(\mathbf{x}) = g(\theta\mathbf{x}) = \frac{1}{1 + \exp(-\theta\mathbf{x})}$$

Softmax Regression

- ▶ Distribution: Multinomial

$$h_{\theta^{(1)}}(\mathbf{x})^{y=1} h_{\theta^{(2)}}(\mathbf{x})^{y=2} \dots h_{\theta^{(k)}}(\mathbf{x})^{y=k}$$
- ▶ Model:
- ▶
$$h_{\theta^{(i)}}(\mathbf{x}) = \phi_i / \Phi$$
 - ▶ $\phi_i = \exp(\theta^{(i)}\mathbf{x})$
 - ▶ $\Phi = \sum_{j=1}^k \phi_j$
 - ▶ Belief in class i
relative to the other class beliefs.

Menti-Quiz (Code: 1373-7721)

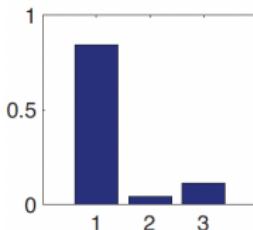
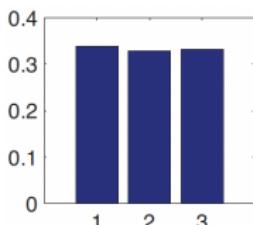
What is the key insight which tells us that the decision region is convex?

Ground-truth Labels and Objective Function

- ▶ Need to employ the cross-entropy objective, meaning:
 - ▶ Predicted probability for a class (close to 1) and a target value of 1 → good prediction.
 - ▶ **Cross-entropy can't work directly on y values** because probabilities are bounded in $\{0, 1\}$, whereas the target values are bounded in $\{0, \dots, k - 1\}$.
- ▶ Intuitively, the target variables need to also be comprised of '1's and '0's as was the case for logistic regression.
- ▶ Encode each target variable (a class id) $y \in \mathbb{Z}_k$ as a k -dimensional 1-hot vector \mathbf{y} .
 - ▶ $\mathbf{y}_i = 1$ if $y(\mathbf{x}) = i$ (i.e., $\mathbf{x} \in C_i$)
 - ▶ Example: Consider a 3-class classification problem.
 - ▶ For $y = 1$, $\mathbf{y} = (1, 0, 0)$
 - ▶ For $y = 2$, $\mathbf{y} = (0, 1, 0)$
 - ▶ For $y = 3$, $\mathbf{y} = (0, 0, 1)$

Softmax Regression

- ▶ Each ground-truth value (a 1-hot vector) is now the **ideal probability distribution**
 - ▶ That is, when you **know that** $\mathbf{x} \in C_i$, the probability distribution **collapses** to $(y_1 = 0, \dots, y_i = 1, \dots, y_k = 0)$.

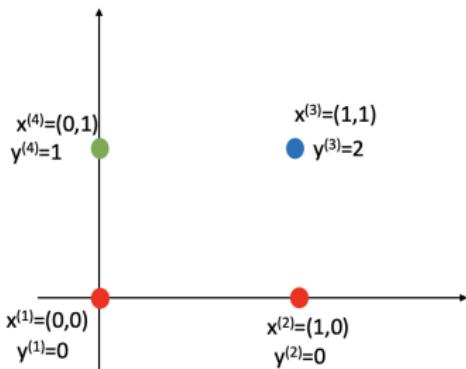


Generalized Cross-Entropy Loss

- ▶ $h_{\Theta}(\mathbf{x}) = (\frac{\phi_1}{\Phi}, \dots, \frac{\phi_k}{\Phi})$
- ▶ Good prediction: When the correct class gets a high probability and every other probability value is low.
- ▶ Bad predictions:
 - ▶ When the incorrect class gets a high predicted probability.
 - ▶ When the distribution is close to uniform.

Multi-class Numerical Working Example

- ▶ Recall: In Week 6, we worked out the gradients of a sample problem with logistic regression.
- ▶ We modify the problem slightly to introduce one more class (everything else remaining the same).

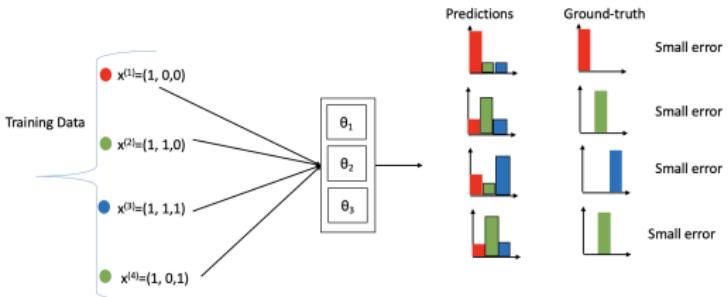
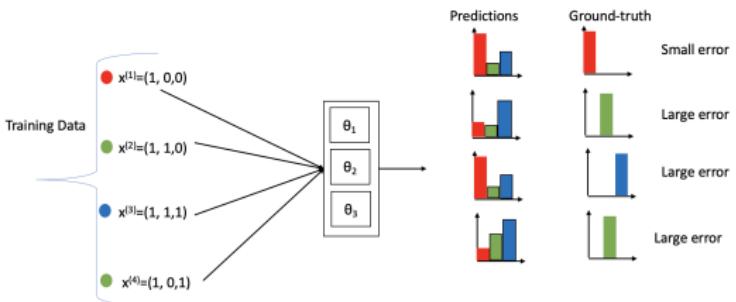


- ▶ We now need 3 Θ vectors, call them Θ_1 , Θ_2 and Θ_3 .

Multi-class
ClassificationMulti-Label
Classification

K-NN classifier

Mixture of Experts



Gradient Descent for Softmax Regression

- ▶ For each training set instance $\mathbf{x} \in \mathbb{R}^{d+1}$ (1 additional for bias)
- ▶ Compute: $\Phi = \sum_{k=1}^K \exp(\theta_k \cdot \mathbf{x})$ ($\theta_k \in \mathbb{R}^{d+1}$)
- ▶ For each class $k = 1, \dots, K$
- ▶ Compute the probability
 $P(y = k) = \hat{\mathbf{y}}_k = \exp(\theta_k \cdot \mathbf{x}) / \Phi.$
- ▶ For each parameter vector component $\theta_{k,j}$
($j = 1, \dots, d + 1$)
- ▶ $\theta_{k,j} \leftarrow \theta_{k,j} + (y_k - \hat{y}_k) \cdot x_j.$

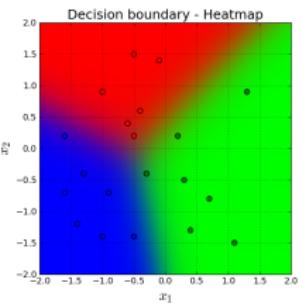
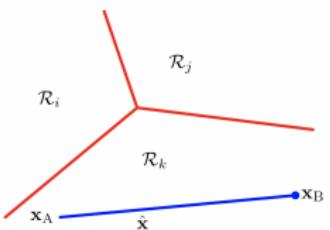
Compact vector form of the SGD update

$$\theta_k \leftarrow \theta_k + (\mathbf{y} - \hat{\mathbf{y}}) \mathbf{x}_j$$

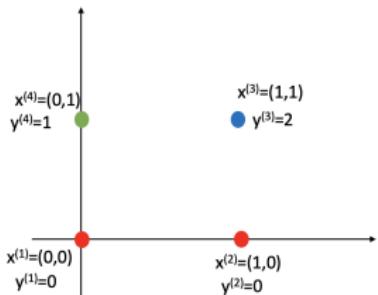
Evolution of the Decision Boundary

Softmax regression

Produces a convex decision boundary comprised of piece-wise linear segments.



Softmax Regression Numerical Working Example



Gradient update for $\mathbf{x}^{(3)} = (1, 1, 1)$.

- ▶ One-hot encoding of $y^{(3)}$:
 $\mathbf{y}^{(3)} = (0, 0, 1)$ (Why?)
- ▶ $\Phi =$

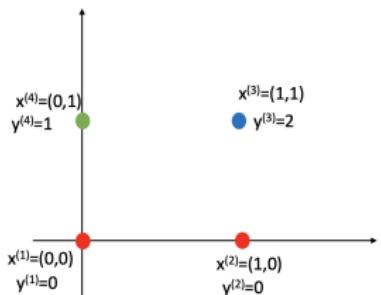
$\underbrace{\exp(0 \times 1 + 0 \times 1 + 0 \times 1)}_{\theta^{(1)} \mathbf{x}} +$ $\underbrace{\exp(0)}_{\theta^{(2)} \mathbf{x}} +$ $\underbrace{\exp(0.1 + 0.1 + 0.2)}_{\theta^{(3)} \mathbf{x}} =$

$1 +$ $1 +$ $1.49 = 3.49$

- ▶ Posterior probability $\hat{\mathbf{y}} =$
 $(1/3.49, 1/3.49, 1.49/3.49) =$
 $(0.28, 0.28, 0.44)$
- ▶ $\mathbf{x}^{(3)}$ is thus predicted to be blue

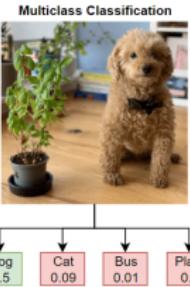
Softmax Regression Numerical Working Example

Updates:



- ▶ $\theta^{(1)} \leftarrow (0, 0, 0) + ((0 - 0.28) \times 1, (0 - 0.28) \times 1, (1 - 0.44) \times 1) = (-0.28, -0.28, 0.66) = \theta^{(2)}$
- ▶ $\theta^{(3)} \leftarrow \underbrace{(0.1, 0.1, 0.2)}_{\text{initial parameter}} + \underbrace{(-0.28, -0.28, 0.66)}_{\text{gradient-based error}} = (-0.18, -0.18, 0.86)$
- ▶ The modified posteriors are better because it assigns a higher probability to the third component — the true class of $x^{(3)}$.

- ▶ **Homework:** Do one more step of gradient updates with $x^{(2)}$.



Multi-class
Classification

Multi-Label
Classification

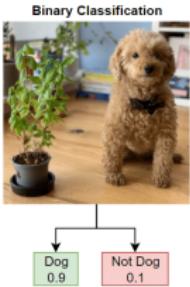
K-NN classifier

Mixture of Experts

- ▶ When an instance is of **composite** type.
- ▶ Examples:
 - ▶ Multi-topical documents, e.g., long Wikipedia articles.
 - ▶ Images containing various objects.
- ▶ We saw the journey from linear \mapsto logistic \mapsto softmax

Menti-Quiz (Code: 1373-7721)

Is MLC more constrained than MCC?



Multi-class
Classification

Multi-Label
Classification

K-NN classifier

Mixture of Experts

- ▶ When an instance is of **composite** type.
- ▶ Examples:
 - ▶ Multi-topical documents, e.g., long Wikipedia articles.
 - ▶ Images containing various objects.
- ▶ We saw the journey from linear \mapsto logistic \mapsto softmax

Menti-Quiz (Code: 1373-7721)

Is MLC more constrained than MCC?

Solution in class.

MLC is algorithmically simpler than MCC!

MCC

- ▶ Output: 1-hot vector of dimension k (#classes)
- ▶ Parameter: k parameter vectors each of dimension $d + 1$.
- ▶ Objective function:

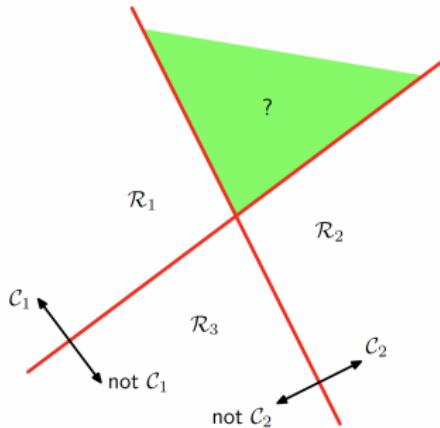
$$\sum_{i=1}^k \log y_i \frac{\exp(\theta^{(i)} \mathbf{x})}{\Phi}$$

- ▶ Cross Entropy

MLC

- ▶ Output: p -hot vector of dimension k (#classes), where $p \leq k$.
 - ▶ Parameter: k parameter vectors each of dimension $d + 1$ (**identical to MCC**).
 - ▶ Objective function:
- $$\sum_{i=1}^k \log y_i \frac{\exp(\theta^{(i)} \mathbf{x})}{1 + \exp(\theta^{(i)} \mathbf{x})}$$
- ▶ **Sum over Binary Cross Entropy values**

We had seen MLC before!



MLC

- ▶ Objective function:

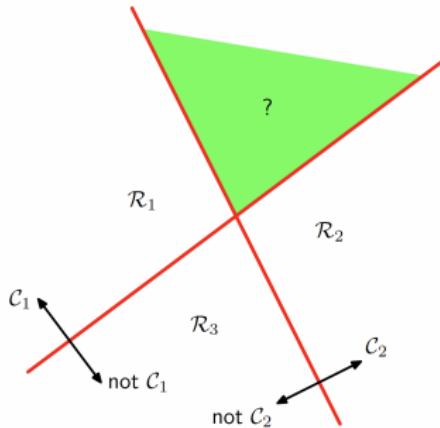
$$\sum_{i=1}^k \log y_i \frac{\exp(\theta^{(i)} \mathbf{x})}{1 + \exp(\theta^{(i)} \mathbf{x})}$$

- ▶ Sum over Binary Cross Entropy values
- ▶ Similar to one vs. rest for MCC.

Menti-Quiz (Code: 1373-7721)

No problems with the green region here! (Why?)

We had seen MLC before!



MLC

- ▶ Objective function:

$$\sum_{i=1}^k \log y_i \frac{\exp(\theta^{(i)} \mathbf{x})}{1 + \exp(\theta^{(i)} \mathbf{x})}$$

- ▶ Sum over Binary Cross Entropy values
- ▶ Similar to one vs. rest for MCC.

Menti-Quiz (Code: 1373-7721)

No problems with the green region here! (Why?)

K-Nearest Neighbors (KNN)

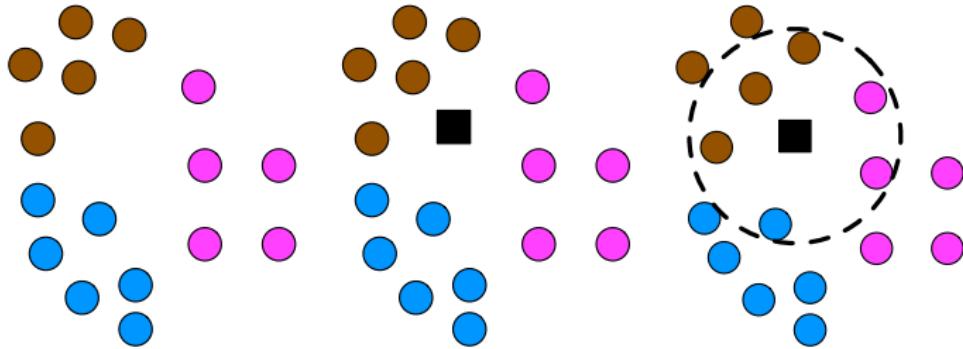
- ▶ Linear/Logistic/Softmax regression belong to the family of **parametric models**.
 - ▶ Learn a set of weights/parameters w/θ from a given set of observations (training set of labeled data).
 - ▶ During inference, **we don't need the training set**.
 - ▶ All we need is the learned parameter vector.
- ▶ In general, **non-parametric models** refer to a class of methods which **doesn't use a fixed set of parameters**, i.e., the **number of parameters can dynamically grow**.
- ▶ What's the closest we have seen so far? –

K-Nearest Neighbors (KNN)

- ▶ Linear/Logistic/Softmax regression belong to the family of **parametric models**.
 - ▶ Learn a set of weights/parameters w/θ from a given set of observations (training set of labeled data).
 - ▶ During inference, **we don't need the training set**.
 - ▶ All we need is the learned parameter vector.
- ▶ In general, **non-parametric models** refer to a class of methods which **doesn't use a fixed set of parameters**, i.e., the **number of parameters can dynamically grow**.
- ▶ What's the closest we have seen so far? –
 - ▶ Higher-order features via feature maps – but it is not non-parametric. **Why?**
- ▶ K-NN is one of the simplest non-parametric models.
- ▶ **It doesn't learn any parameters.**

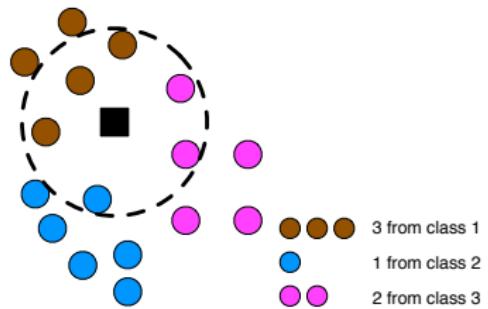
So how does KNN work?

- ▶ Uses the **locality of reference** hypothesis.
- ▶ Similar inputs will have similar labels.
 - ▶ Example: If two houses are quite similar (in terms of the features, e.g., #bedrooms, front_yard_area, back_yard_area, furnished_kitchen, . . .), then they will be in the same price range.
- ▶ Advantages:
 - ▶ Same algorithm works for binary and multi-class (all the trouble of devising different objective functions saved).

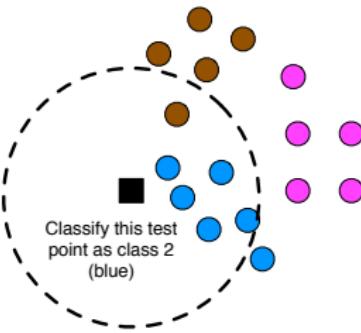


- ▶ Choose a K (a small integer) to define the **neighborhood size**.
- ▶ For a test object x_{new} :
 - ▶ Find the K closest points from the training set.
 - ▶ Find majority class of these K neighbors.
 - ▶ Resolve ties arbitrarily.

KNN

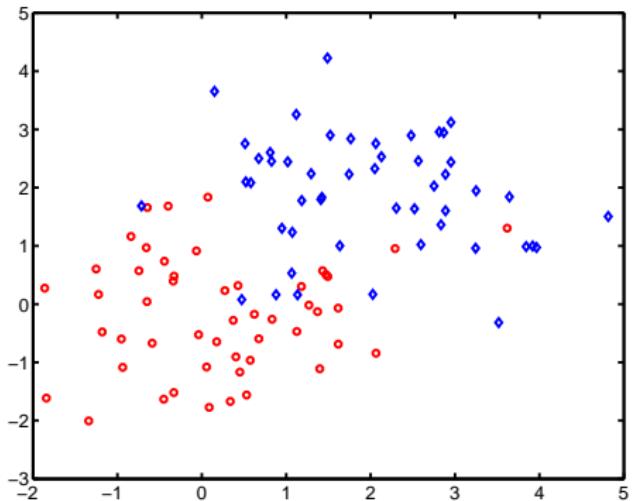


- ▶ Class one (brown) has most votes – classify x_{new} as belonging to class 1.



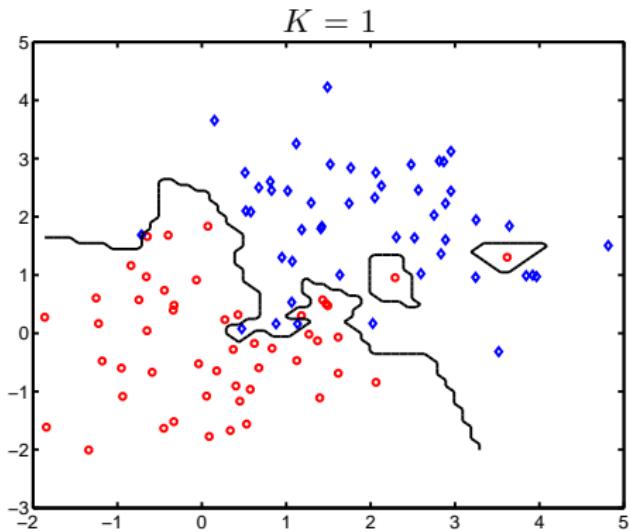
- ▶ Class 2 has most votes.

KNN – real example



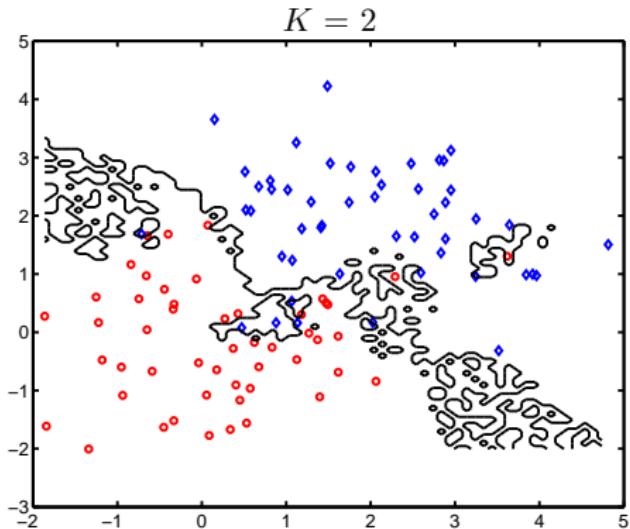
- ▶ Binary data.

KNN – real example



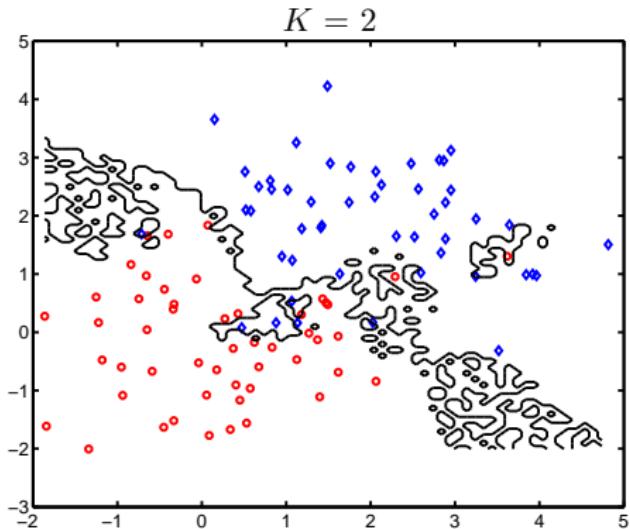
- ▶ 1-Nearest Neighbour.
- ▶ Line shows decision boundary.
- ▶ Too complex – should the islands exist?

KNN – real example



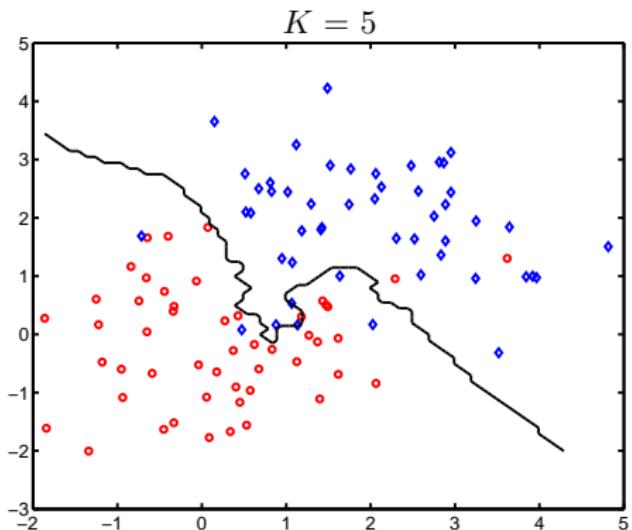
- ▶ 2-Nearest Neighbour.
- ▶ What's going on?

KNN – real example



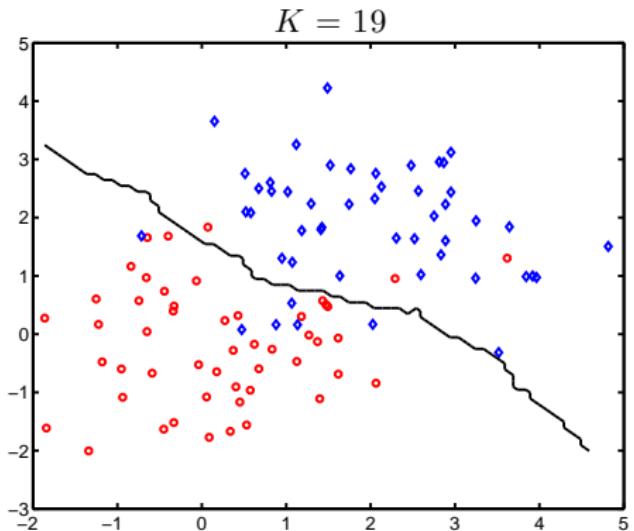
- ▶ 2-Nearest Neighbour.
- ▶ What's going on?
- ▶ Lots of ties – random guessing.

KNN – real example

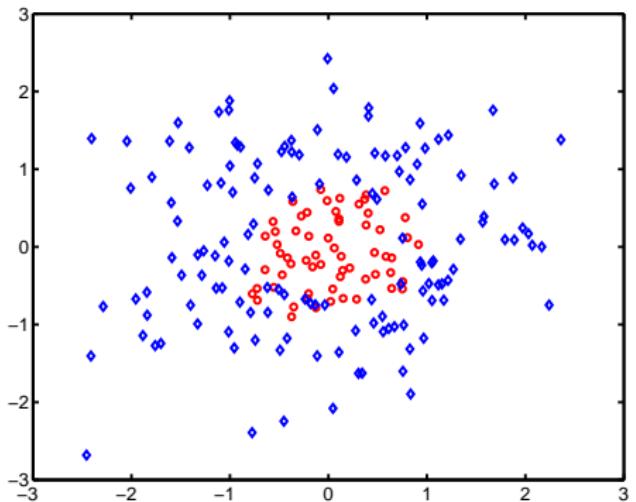


- ▶ 5-Nearest Neighbour.
- ▶ Much smoother.

KNN – real example

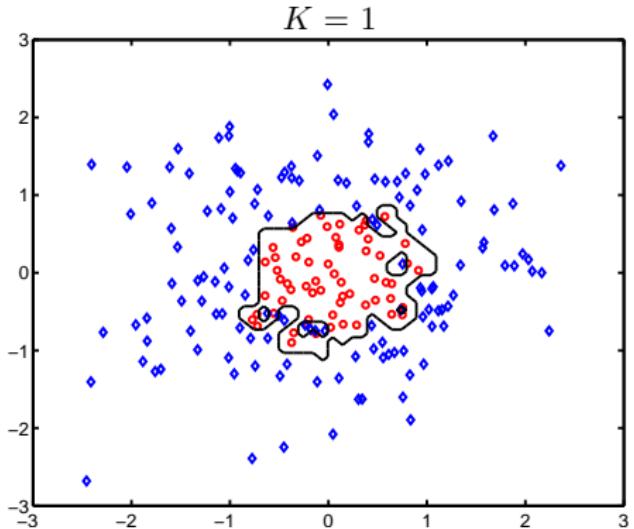


- ▶ 19-Nearest Neighbour.
- ▶ Very smooth.



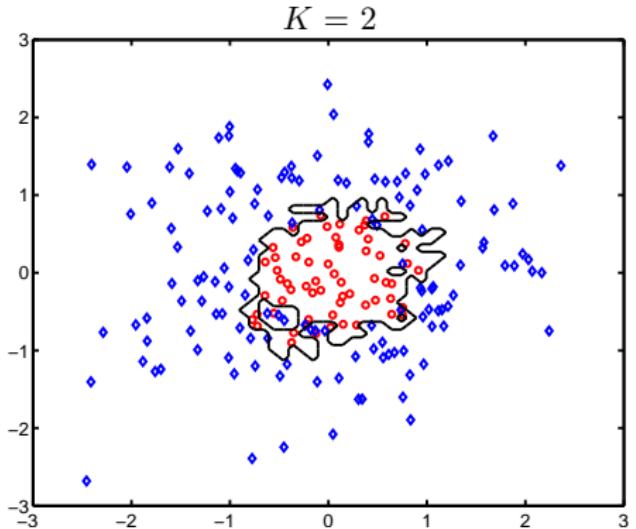
- ▶ Binary data.

KNN – real example 2

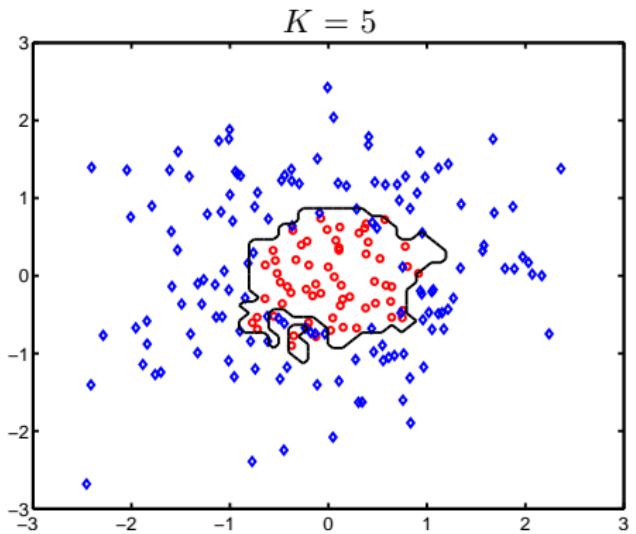


- ▶ Non-smooth – too complex again?

KNN – real example 2

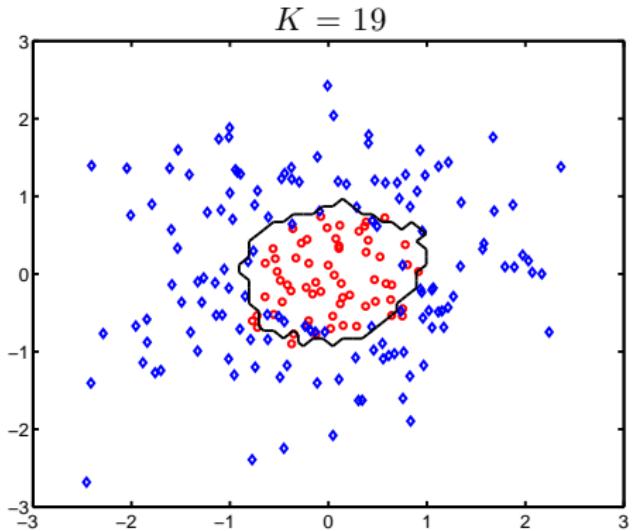


- ▶ Random effects again...



- ▶ Getting smoother.

KNN – real example 2



- ▶ Smoother still.

Problems with KNN

- ▶ Class imbalance
 - ▶ As K increases, small classes will disappear!
 - ▶ Imagine we had only 5 training objects for class 1 and 100 for class 2.
 - ▶ For $K \geq 11$, class 2 will **always** win!

Homework: Neighborhood size for which class with least prior disappears from predictions?

Work out a general formula for the minimum values of $K(i)$ that make the classes (C_i s) disappear with respective priors $\alpha_1, \dots, \alpha_M$ (assume wlog $\alpha_1 < \alpha_2 \dots < \alpha_M$).

Problems with KNN

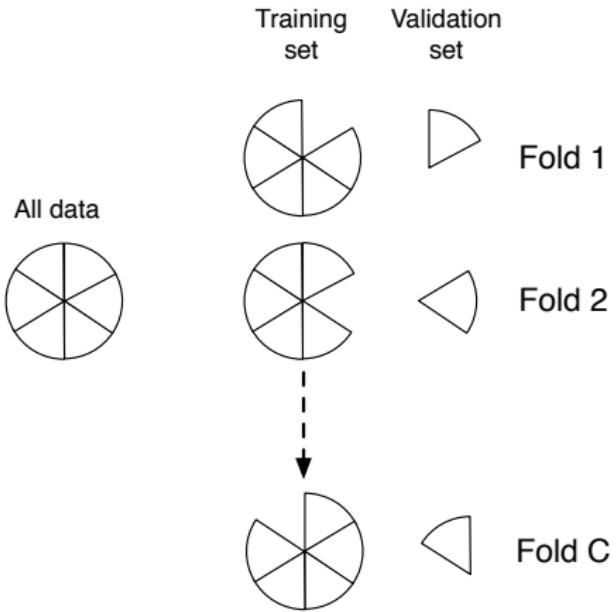
- ▶ Class imbalance
 - ▶ As K increases, small classes will disappear!
 - ▶ Imagine we had only 5 training objects for class 1 and 100 for class 2.
 - ▶ For $K \geq 11$, class 2 will **always** win!

Homework: Neighborhood size for which class with least prior disappears from predictions?

Work out a general formula for the minimum values of $K(i)$ that make the classes (C_i s) disappear with respective priors $\alpha_1, \dots, \alpha_M$ (assume wlog $\alpha_1 < \alpha_2 \dots < \alpha_M$).

- ▶ What's a pragmatic choice for K ?
 - ▶ Right value of K will depend on data.
 - ▶ Cross-validation!

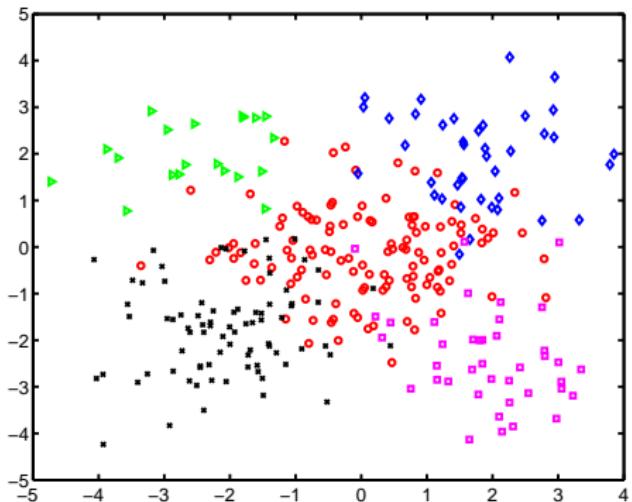
Cross-validation to select optimal K



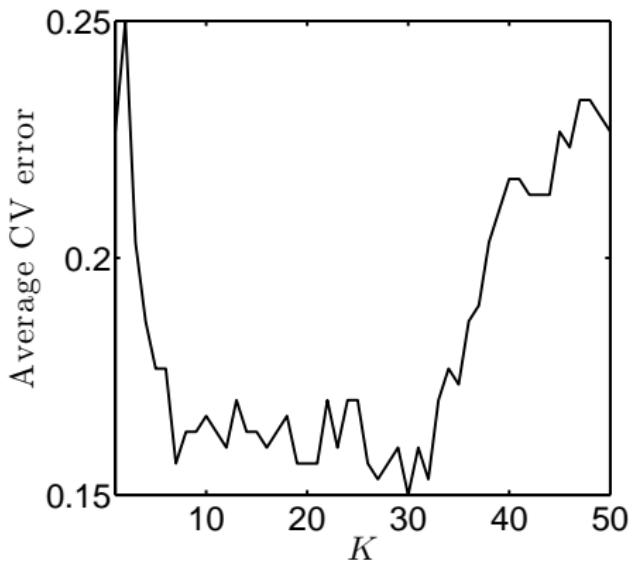
- ▶ Maximize the average performance measure across folds.

- ▶ Vary K from 1 to some max size.
- ▶ Select the one that leads to highest average performance measure, e.g.,
 - ▶ accuracy
 - ▶ precision, or
 - ▶ recall depending on the task.

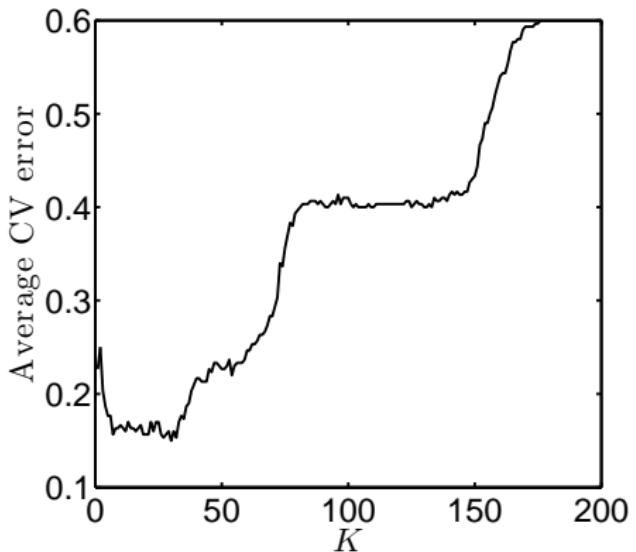
Example – 5 classes



- ▶ 5 classes.
- ▶ Smallest has 20 instances, biggest 120.



- ▶ Curve shows average misclassification error for 10-fold CV.
- ▶ Minimum at approximately $K = 30$.



- ▶ As K increases, classes ‘disappear’
- ▶ Causes the ‘steps’ in error.

Any other problems of KNN?

Menti-Quiz (Code: 1373-7721)

Can you think of any other problems of K-NN that I didn't enlist in the last slide?

Any other problems of KNN?

Menti-Quiz (Code: 1373-7721)

Can you think of any other problems of K-NN that I didn't enlist in the last slide?

- ▶ KNN is non-probabilistic (assigns equal weights to each training instance in the neighbourhood).

Menti-Quiz (Code: 1373-7721)

Can there be a probabilistic version. How might it work?

- ▶ KNN is non-probabilistic (assigns equal weights to each training instance in the neighbourhood).

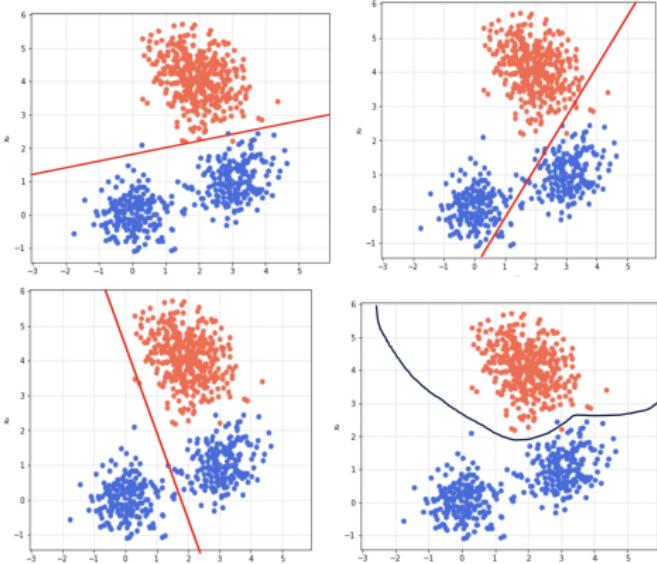
Menti-Quiz (Code: 1373-7721)

Can there be a probabilistic version. How might it work?

Multi-class
ClassificationMulti-Label
Classification

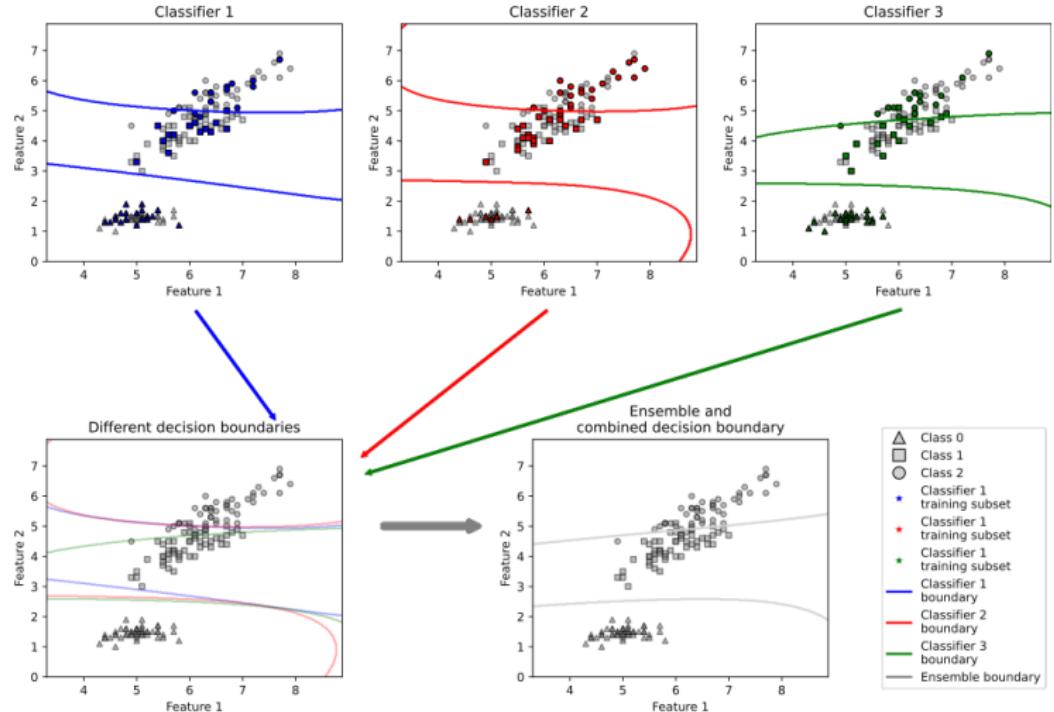
K-NN classifier

Mixture of Experts



- ▶ Different regions of the input space often requires a different local model.
- ▶ Learn local models on each cluster of the data.
- ▶ Combine the models to get a better decision boundary.

Combination of Models



- ▶ Is particularly useful when data is available in streams.
- ▶ Is also able to handle outliers in data similar to Bayesian (next week).

Mixture of Experts

- ▶ By this time you may have realised that this appears in the practical coursework.

Training individual models on parts of data

- ▶ Cluster the data X into X_1, \dots, X_n .
- ▶ For each subset X_i , train a model θ_i .
 - ▶ θ_i 's don't need to be the same model, e.g., one could be logistic regression (week 6), and another could be K-NN (this week).

A weighted combination of the models

- ▶ $P(y|\mathbf{x}; \theta; \phi) = \sum_{i=1}^n \phi_i P(y|\mathbf{x}; \theta_i)$
- ▶ ϕ 's are the additional set of n parameters — the weights (or prior beliefs) of each classifier.
- ▶ We need to learn the ϕ 's.

Mixture of Experts

Introduction

D. Ganguly

Multi-class
Classification

Multi-Label
Classification

K-NN classifier

Mixture of Experts

Menti-Quiz (Code: 1373-7721)

How will you learn the weights of the experts?

Mixture of Experts

Introduction

D. Ganguly

Multi-class
Classification

Multi-Label
Classification

K-NN classifier

Mixture of Experts

Menti-Quiz (Code: 1373-7721)

How will you learn the weights of the experts?

Solution in class!

Inference with MoE

Introduction

D. Ganguly

Multi-class
Classification

Multi-Label
Classification

K-NN classifier

Mixture of Experts

Menti-Quiz (Code: 1373-7721)

How will you infer the class of a test instance?

Inference with MoE

Introduction

D. Ganguly

Multi-class
Classification

Multi-Label
Classification

K-NN classifier

Mixture of Experts

Menti-Quiz (Code: 1373-7721)

How will you infer the class of a test instance?

Solution in class!

- ▶ Parametric Classification
 - ▶ Generalized Logistic Regression to multiple classes.
 - ▶ Also discussed multi-label classification
- ▶ Discussed a non-parametric approach - KNN
- ▶ MoE which can combine different classifiers.
 - ▶ MoE is the fundamental concept behind decision tree based classifiers, which we will not cover.