SAND2023-06443C

Modeling Correlated Features for Machine Learning Classification

Rich Field, Mike Smith, Joe Ingram, Eva Domschot

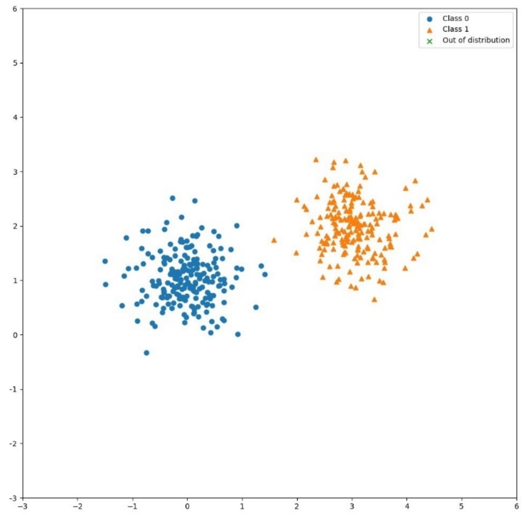
Applied Information Sciences

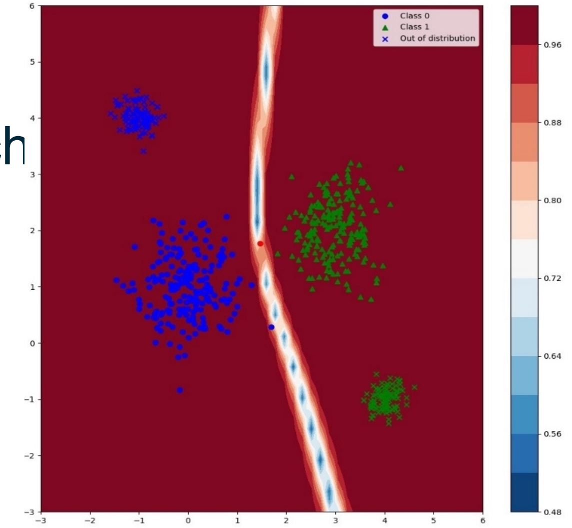
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Motivation

- Machine learning (ML) models can classify with high accuracy
  - Frequently assume any new test data is similar to the training data (small statistical distance)
  - This is often not true in practice
- Hence ML models can sometimes be over-confident in their predictions
  - Want the model to report low confidence on test points that are far from training data (large statistical distance)
- There are methods to measure statistical distance
  - Most are defined assuming two distributions; we want to look at each test point independently
  - Many assume Gaussian and independent features
  - Many out of distribution methods depend on ML model
  - Mahalanobis distance (assumes multivariate Gaussian features)
- Our objective: build statistical model of features and use for indicating when test point is far from training set





Confidence is the normalized softmax value (ML output)

Feature Modeling

- Objective: Build probabilistic model for each feature
  - Can be used for ML classification
  - Can synthesize new data that is consistent with the training data
  - Can be used to measure distance of new data from training data
- Let X_i denote a random variable that models feature i; train the model to match properties of the original data
  - Marginal distributions

$$F_i(x_i) = \Pr(X_i \leq x_i),\ \ i = 1, \ldots, d$$

  - Correlations (2nd-order property)

$$\mathbb{E}[X_i\, X_j],\ \ i, j = 1, \ldots, d$$

  - Other (higher-order) properties

$$\Pr(X_i \leq x_i, X_j \leq x_j, X_k \leq x_k),\ \ \mathbb{E}[X_i\, X_j^2]$$

  - Full joint distribution function

$$\Pr(X_1 \leq x_1, X_2 \leq x_2, \ldots, X_d \leq x_d)$$

Practical problems limit us to marginal distributions and correlations

Accuracy increases

More data is needed for training

Yet Another Discriminant Analysis (YADA)

- YADA is a probabilistic model for the feature data
  - Training involves matching the marginal distributions and pairwise correlations
- The YADA model for the i^{th} feature is

$$X_i = \mu_i + \sigma_i \, h_i(G_i)$$

  - μ_i and σ_i are the sample mean and standard deviation of X_i
  - h_i is a nonlinear function of the marginal distribution of X_i
  - G_i is a Gaussian random variable with zero mean and unit variance
  - G_i and G_j for two features $i \neq j$ are correlated based on the sample correlation matrix
  - The joint marginal distribution is available in closed-form
- Conditioned on the class labels; one YADA model per class
- Based on the translation random variable model* developed for engineering mechanics

Pairwise correlations are	Features are: Gaussian	Features are: non-Gaussian
Ignored	Linear discriminant analysis (LDA)	
Included	Quadratic discriminant analysis (QDA)	Yet another discriminant analysis (YADA)

Some related methods

*M. Grigoriu. Crossings of non-Gaussian translation processes. *Journal of Engineering Mechanics*, 110(4):610–620, 1984.

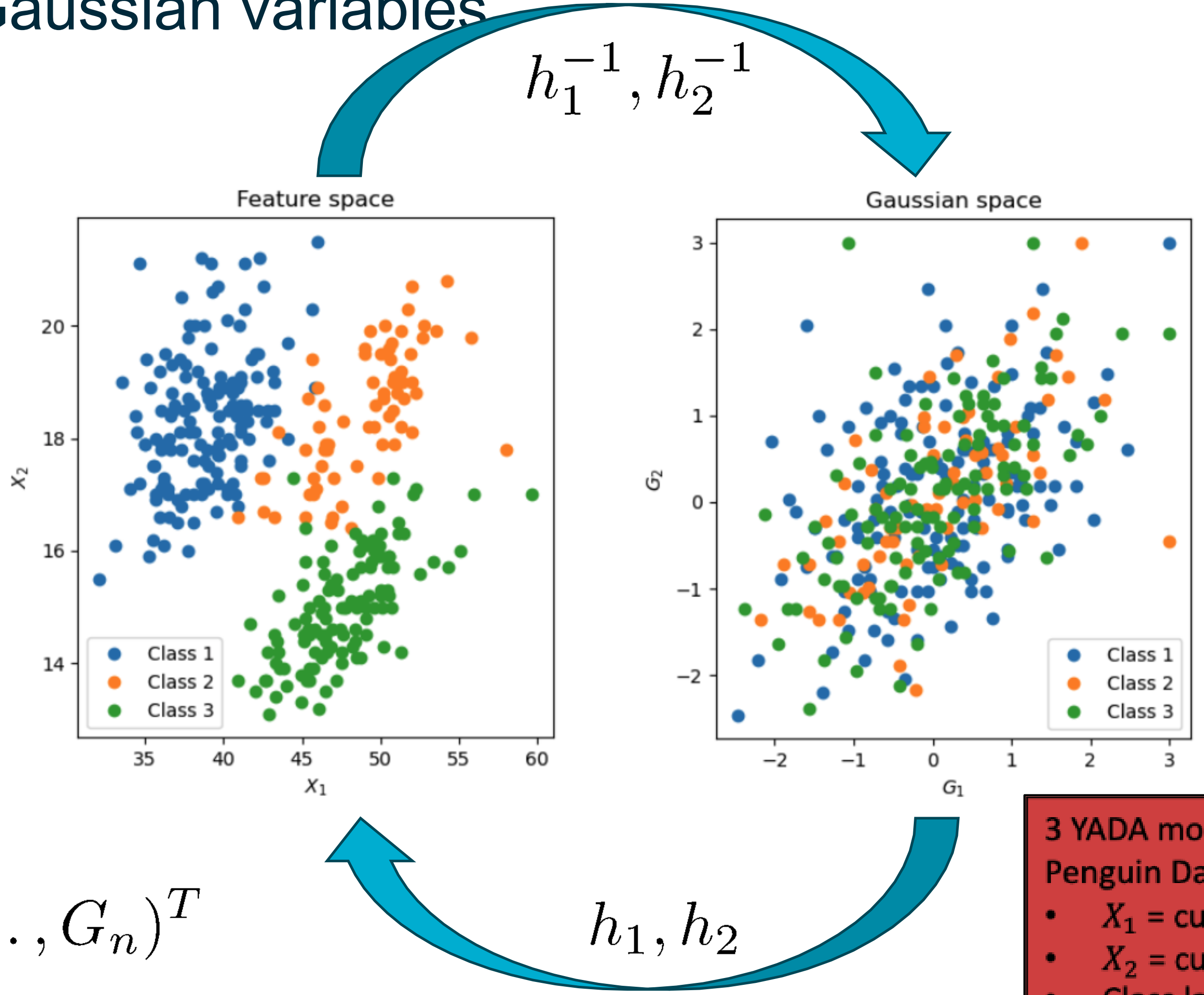
YADA Maps Features to their Gaussian Image

- The YADA model provides an invertible mapping to the space of multivariate (correlated) Gaussian variables

$$G_i = h_i^{-1}\left(\frac{X_i - \mu_i}{\sigma_i}\right)$$

- Mahalanobis distance
  - Statistical distance of point to a distribution
  - Can be applied in the multivariate Gaussian space to assess the statistical distance of new test point $\boldsymbol{X} = (X_1, \ldots, X_d)^T$ from a trained YADA model

$$\sqrt{\mathbf{G}^T \mathbf{c}^{-1} \mathbf{G}},\ \mathbf{G} = (G_1, \ldots, G_n)^T$$



3 YADA models trained on the Penguin Dataset*
- X_1 = culmen length (mm)
- X_2 = culmen width (mm)
- Class labels are {Adelie, Chinstrap, Gentoo}

*https://www.kaggle.com/code/parulpandey/penguin-dataset-the-new-iris/notebook

Training a YADA Model

- Given training data $\{(\boldsymbol{x}, y)_j, j = 1, \dots, n\}, \boldsymbol{x} \in \mathbb{R}^d, y \in \{1, \dots, \kappa\}$, partition it according to the class label
- For each class:

1. Compute the sample mean μ_i and standard deviation σ_i for each feature
2. Normalize the training set $z_{ij} = \frac{x_{ij} - \mu_i}{\sigma_i}, i = 1, \dots, d, j = 1, \dots, n$
3. Compute the sample cumulative distribution function F_i of $\boldsymbol{z}_i$ for each feature
   - Empirical methods
   - Kernel methods (e.g., kernel density estimation)
4. Determine the Gaussian image of the training set $g_{ij} = \Phi^{-1} \circ F_i(z_{ij})$
5. Compute the sample Pearson correlation matrix $\boldsymbol{c}$ from $\{g_{ij}\}$
6. Compute the inverse and (log) determinant of $\boldsymbol{c}$

Note: it might also be useful to train a single YADA model to all data regardless of the class label

Model Uncertainty

- Given
  - Training data $\{(\boldsymbol{x}, y)_j, j = 1, \ldots, n\}, \boldsymbol{x} \in \mathbb{R}^d, y \in \{1, \ldots, \kappa\}$
  - κ trained YADA models
  - Test point $\boldsymbol{x}'$ with predicted label from an ML model
- Assess uncertainty in the predicted label for $\boldsymbol{x}'$ that is due to possible inconsistency between test and training data
  - Use Mahalanobis distance from each YADA model to quantify how "far" test point is from the training data

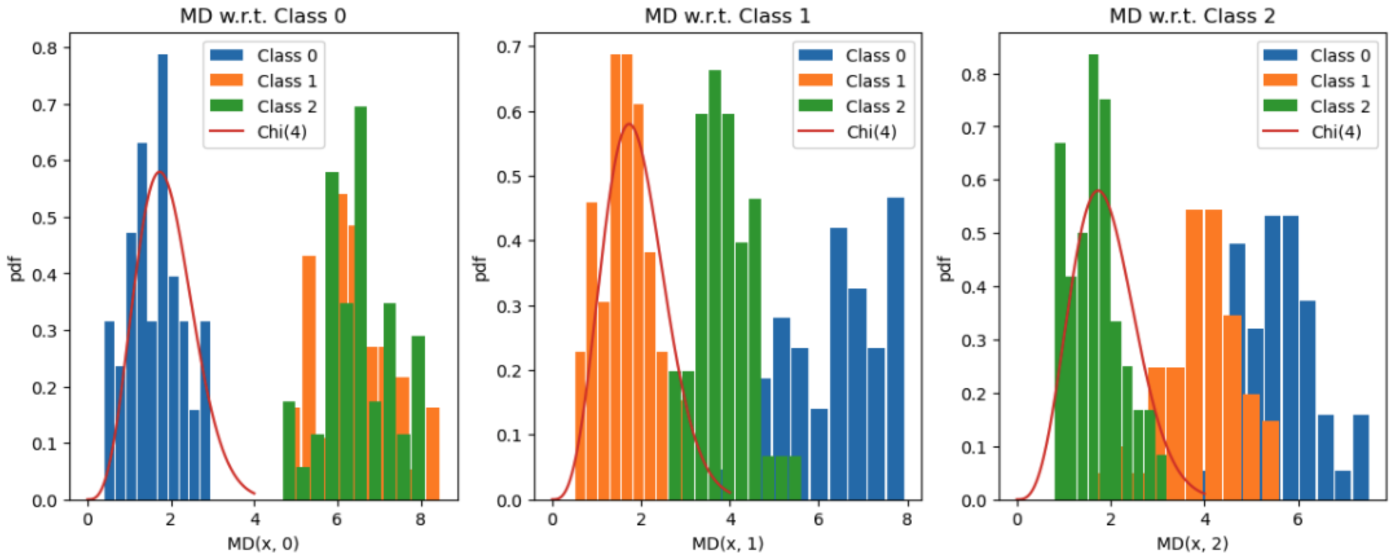
$$\mathrm{MD}(\mathbf{x}', j) = \sqrt{\left(\mathbf{g}^{(j)}\right)^T \left(\mathbf{c}^{(j)}\right)^{-1} \mathbf{g}^{(j)}}, \ \mathbf{g}^{(j)} \text{ is the Gaussian image of } \mathbf{x}' \text{ with respect to model } j$$

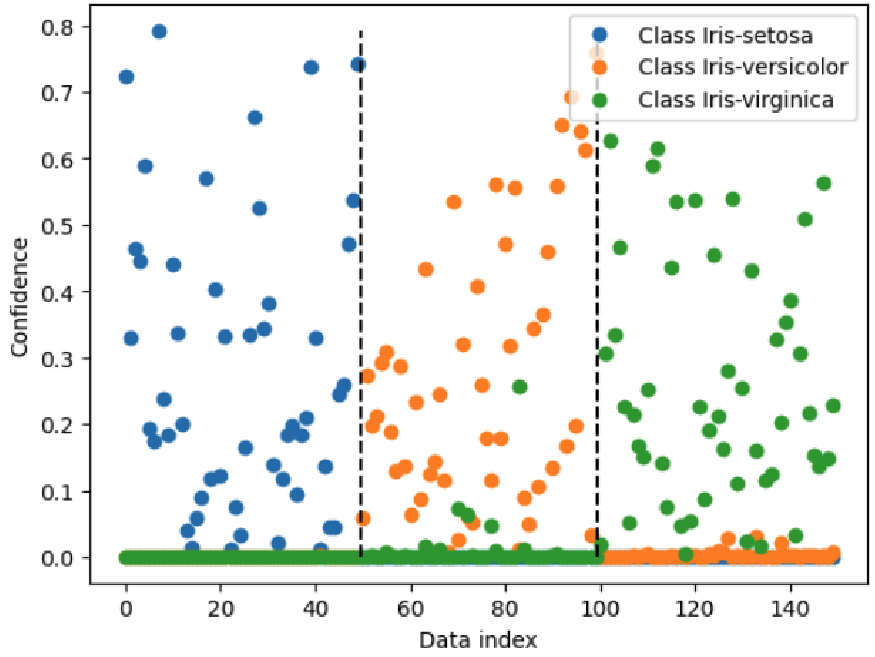
  - Compare $\mathrm{MD}(\boldsymbol{x}', j)$ to the $\{\mathrm{MD}(\boldsymbol{x}, j)\}$ calculated from the training data
  - Given a random test point, the probability distribution of its Mahalanobis distance from a model is known in closed-form (the chi distribution with d degrees of freedom)
  - This means we can evaluate the likelihood of any particular $\mathrm{MD}(\boldsymbol{x}', j)$
  - We define confidence as this likelihood, scaled to take values in $(0, 1)$

$$\mathrm{conf}(\mathbf{x}', j) = \frac{1}{(d-1)^{(d-1)/2}} \exp\left(-\frac{1}{2}\left(\mathrm{MD}(\mathbf{x}', j)^2 - d + 1\right)\right)$$

Results for Iris Dataset

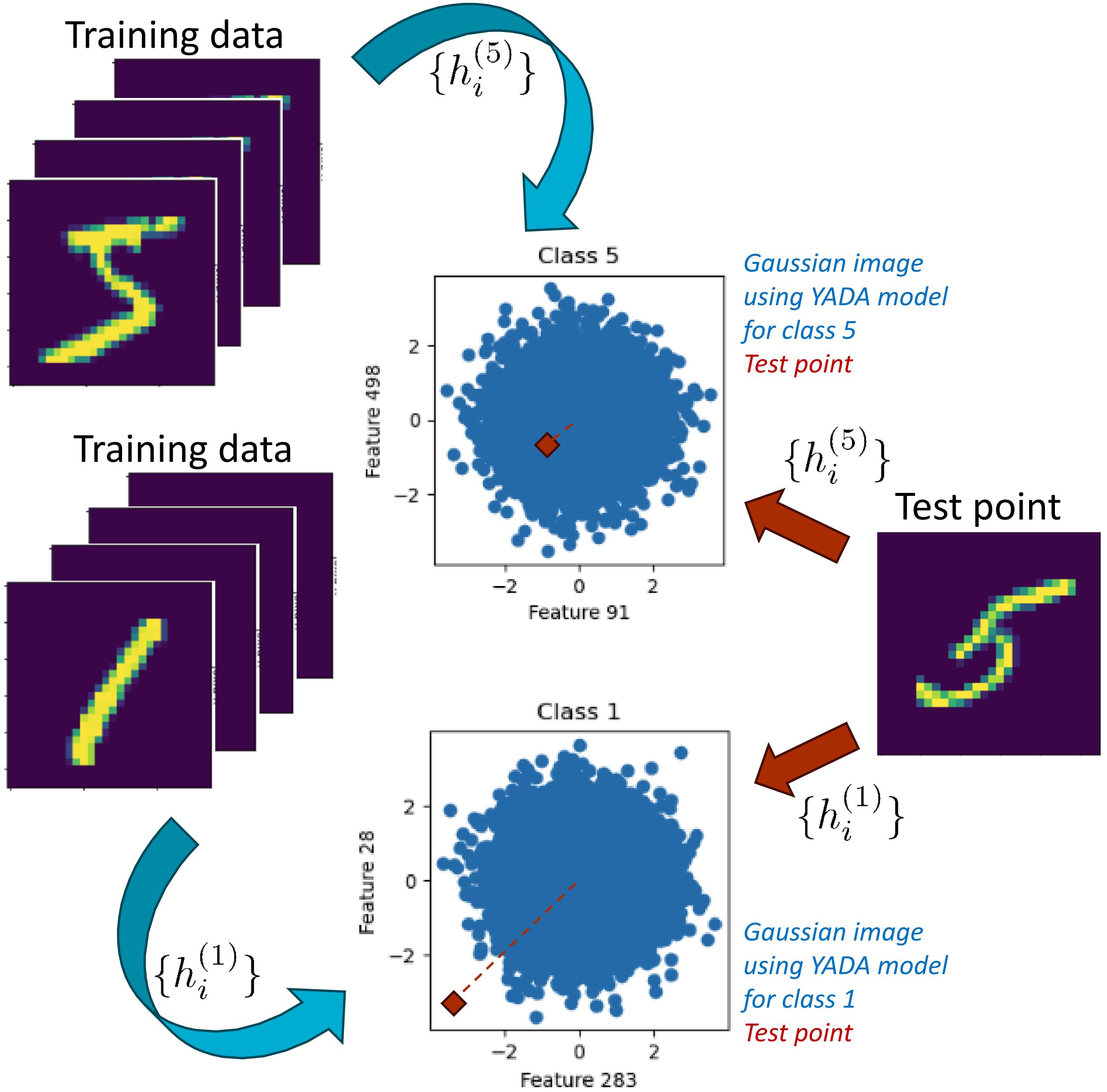
- Left: Normalized histograms of the Mahalanobis distances of each training point from each YADA model
  - MD(x, correct class) follows the chi distribution
  - MD(x, incorrect class) > MD(x, correct class) in most cases
- Right: Confidence that a training point comes from each YADA model



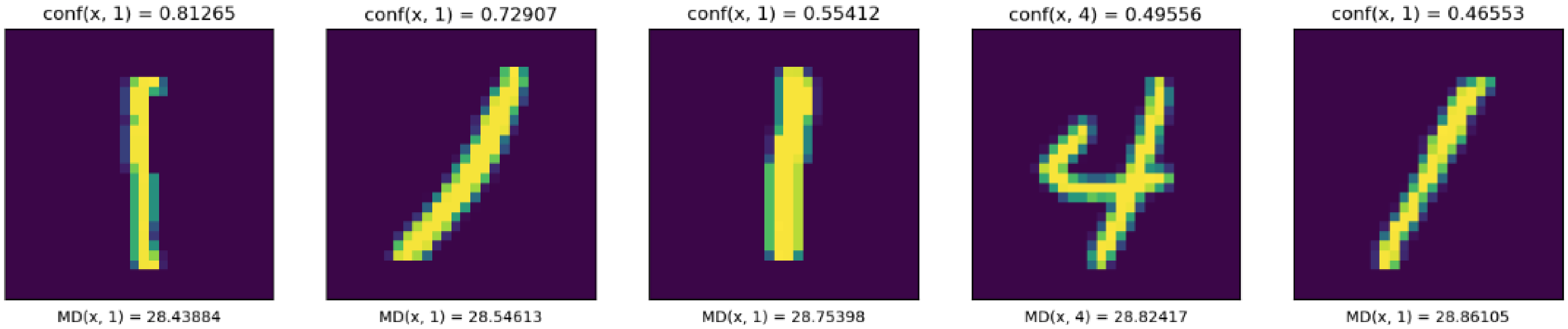


YADA Applied to MNIST Dataset

- Training data: 60,000 images of handwritten digits
  - Approximately equal number of each of the 10 classes
- Each image is 28x28 pixels
- Treat each pixel value as a feature
  - Integer in {0, …, 255}; map to [0, 1]
  - 28 x 28 = 784 features
- Train 10 YADA models
- Testing data: 10,000 additional images
- Compute MD of test point to each YADA model
  - Confidence measure

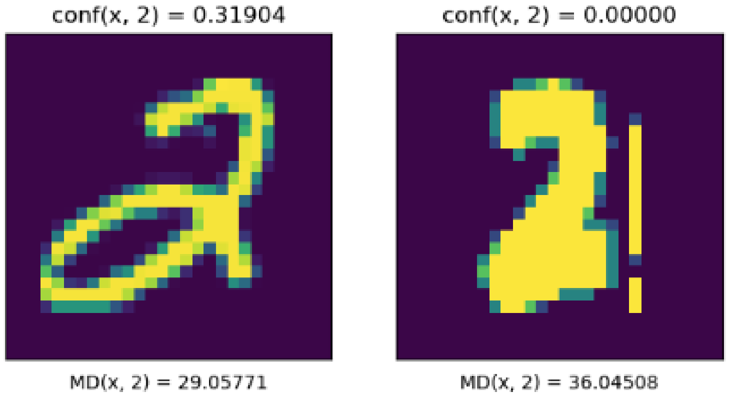


Results for MNIST Dataset

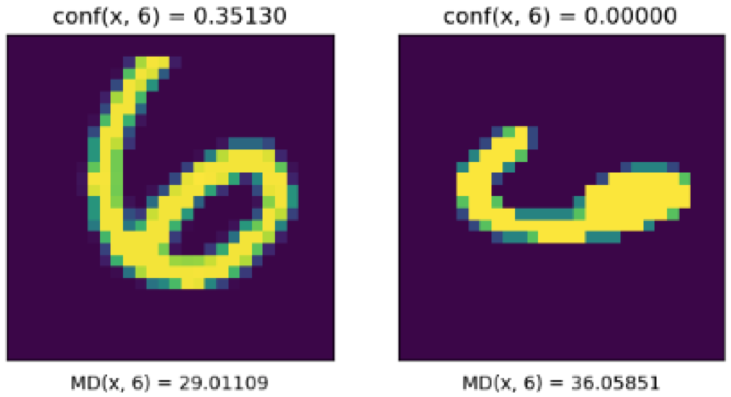


5 test images with the least uncertainty

Images of ‘2’ with the least and most uncertainty

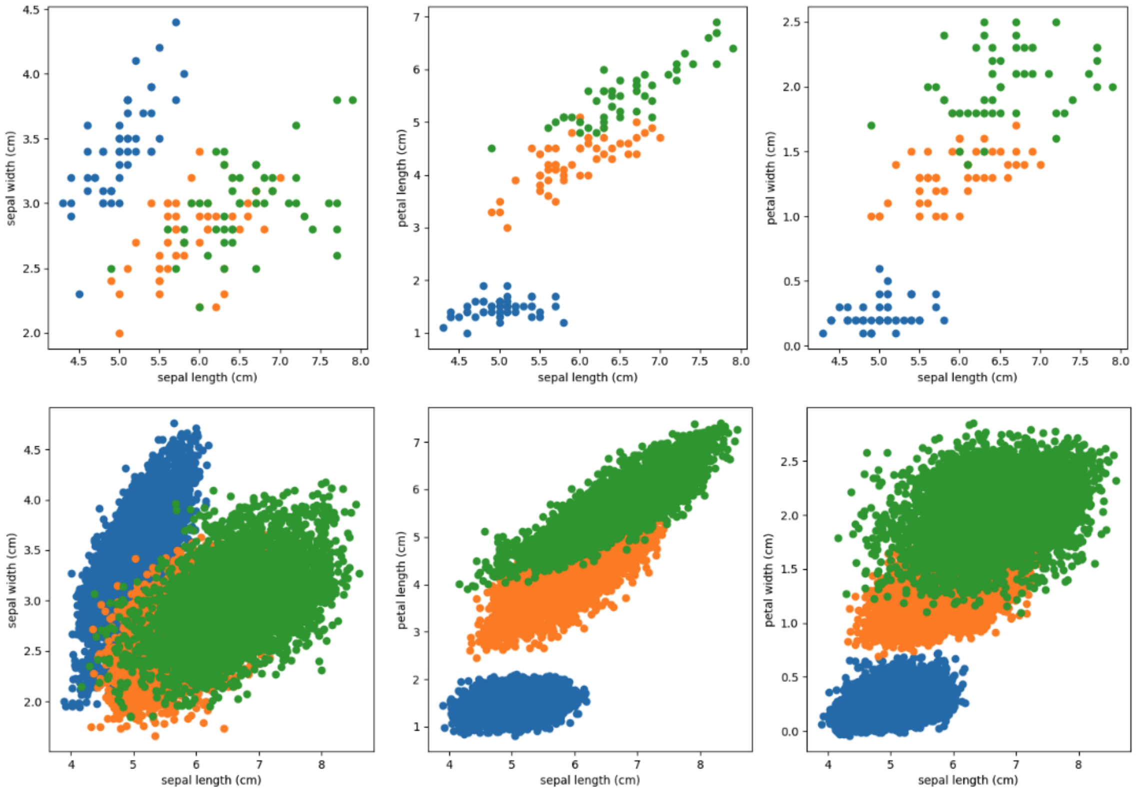


Images of ‘6’ with the least and most uncertainty



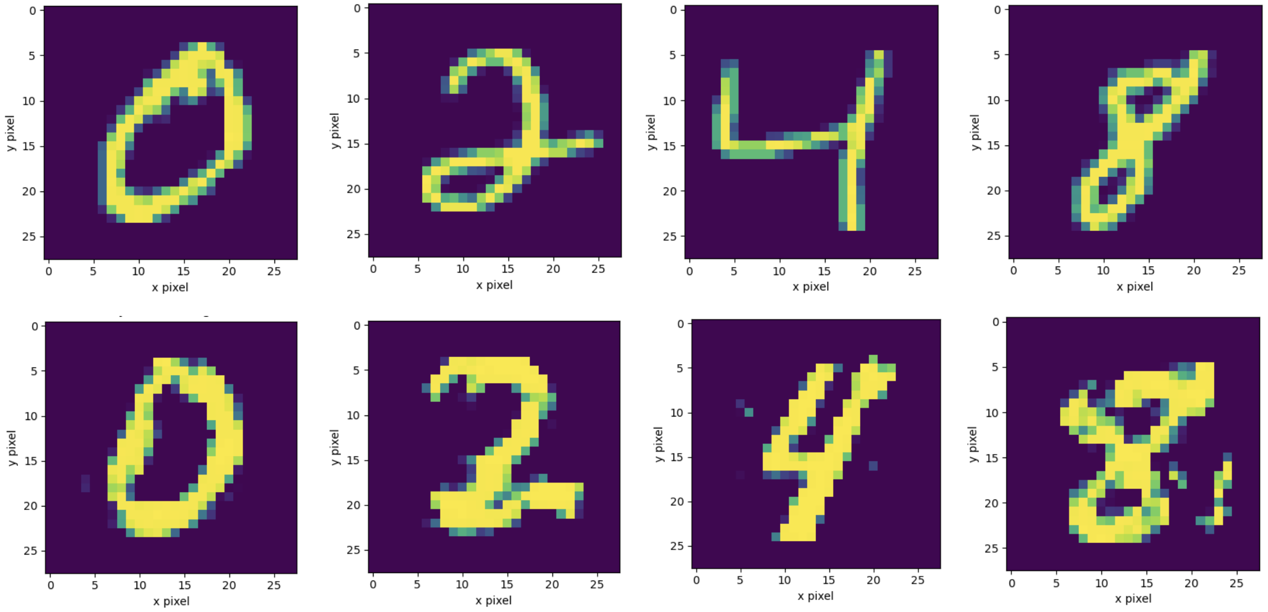
YADA: Create Synthetic Data

- YADA is a probabilistic model, so we can draw random samples from it to produce synthetic data



<u>Algorithm</u>

1. Create samples of correlated Gaussian variables
2. Map each sample to the feature space



YADA model trained on the Fisher iris Dataset*
- Top row are real data
- Bottom row are synthetic data produced by YADA

*https://archive.ics.uci.edu/dataset/53/iris

YADA model trained on the MNIST Dataset* (images of handwritten digits 0-9)
- Top row are real images
- Bottom row are synthetic images produced from YADA models

*http://yann.lecun.com/exdb/mnist/

YADA: Classification Based on Maximum Joint Likelihood

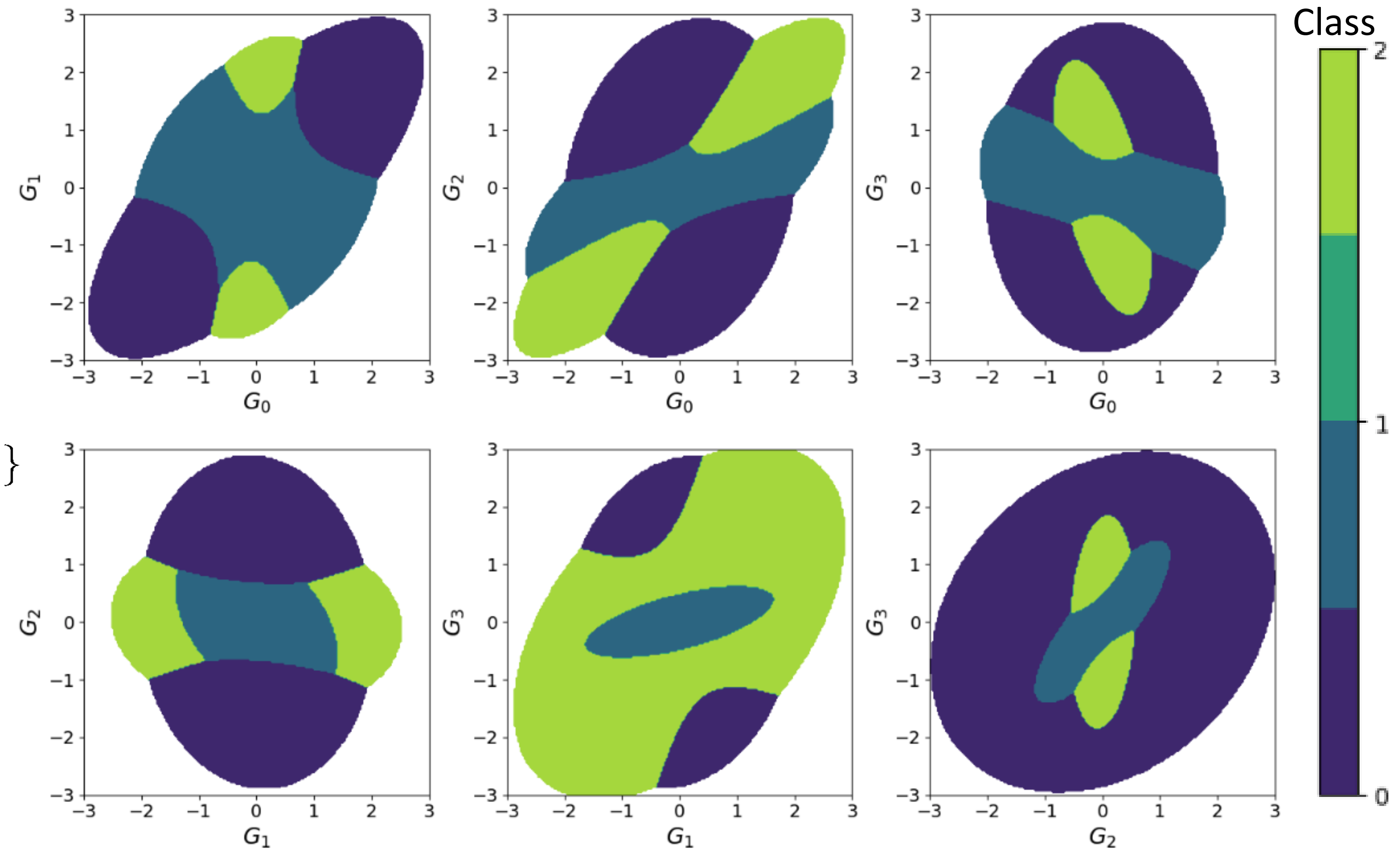
- Classification is defined using the Gaussian image of a test point $\boldsymbol{X}$
- The set of points belonging to class i:

$$\mathcal{C}_i = \{\mathbf{G}\colon \phi_n(\mathbf{G};\mathbf{0},\mathbf{c}^{(i)}) > \phi_n(\mathbf{G};\mathbf{0},\mathbf{c}^{(j)}), \forall j \neq i\}$$

$$\phi_n = \text{multivariate normal PDF}$$

$$\mathbf{c}^{(i)} = \text{covariance matrix for class } i$$

- YADA predicts that $\boldsymbol{X}$ is from class i if its Gaussian image $\mathbf{G} \in \mathcal{C}_i$
  - White regions = the likelihood of all YADA models is very small



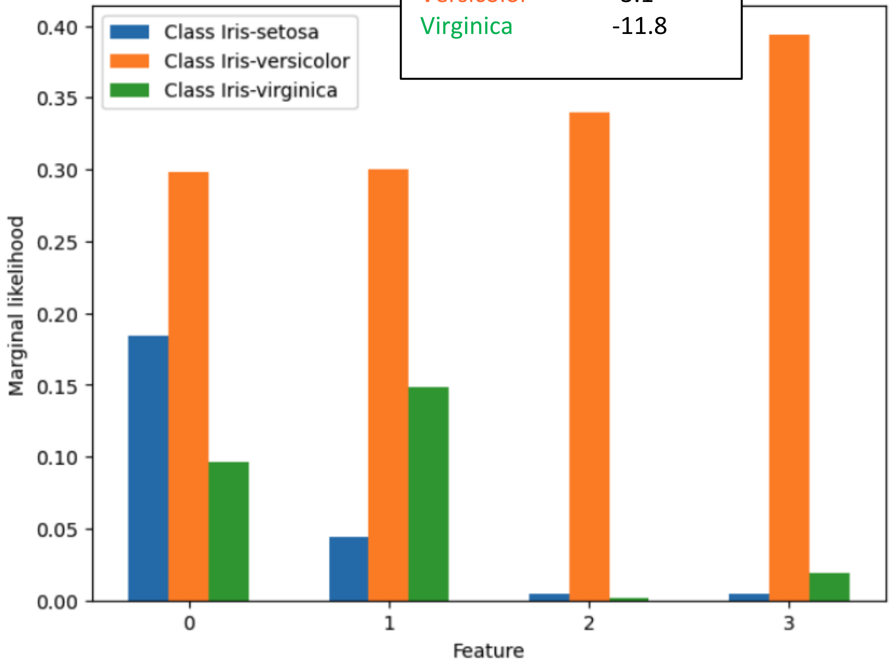
Decision boundaries for the Fisher iris dataset
- Each plot illustrates a different pairing of features with the other two set to zero
- Drawn in the Gaussian space

YADA: Marginal Likelihoods Can Provide Explanations

- Classification is based on the joint likelihood function
- The marginal likelihood functions can be used for explanations
  - Compute the Gaussian image of a test point w.r.t. each class *j*

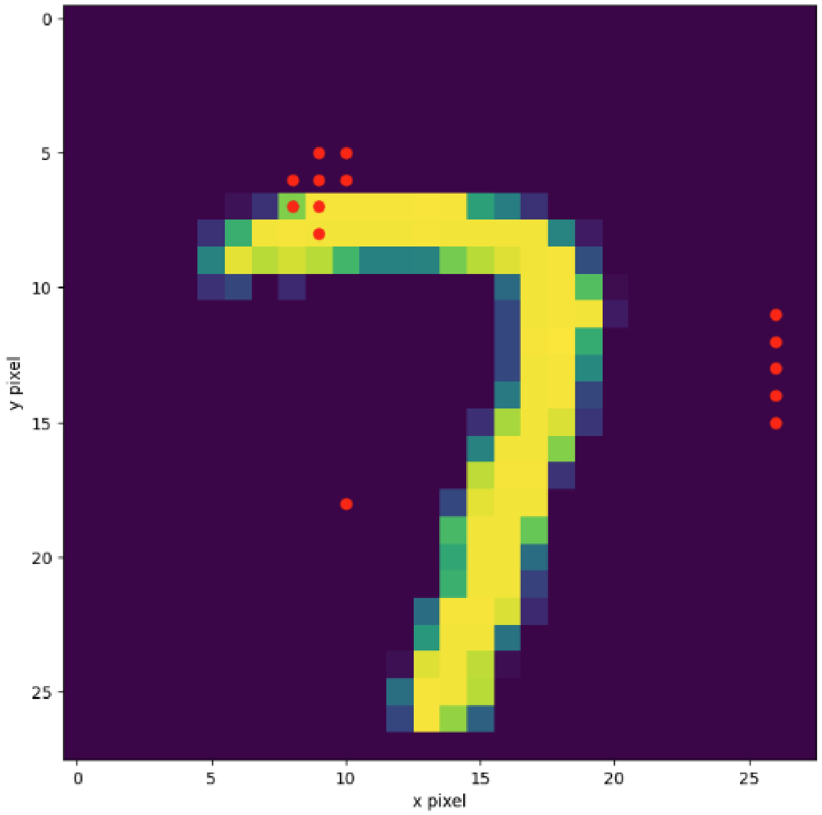
$$\phi(G_i^{(j)}), i = 1, \ldots, n$$

$$\phi = \text{ Univariate normal PDF}$$



Log of joint likelihood	
Setosa	-21.4
Versicolor	-3.1
Virginica	-11.8

Marginal likelihoods for one test point from the Fisher iris dataset
- YADA predicts the label to be Versicolor
- Marginal likelihoods shown for each feature



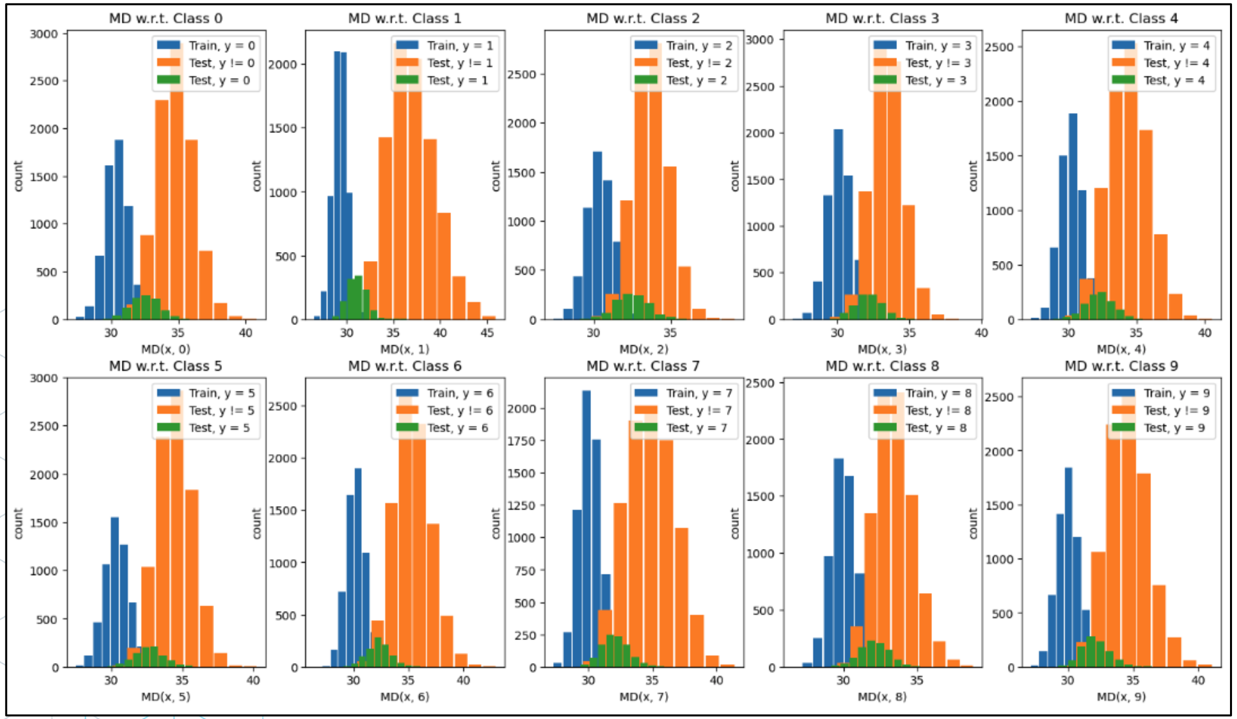
One test image from the MNIST dataset
- Each pixel is a feature
- YADA predicts the label to be ‘7’
- Highlighted pixels are features where the marginal likelihood for ‘7’ was large while small for all other classes

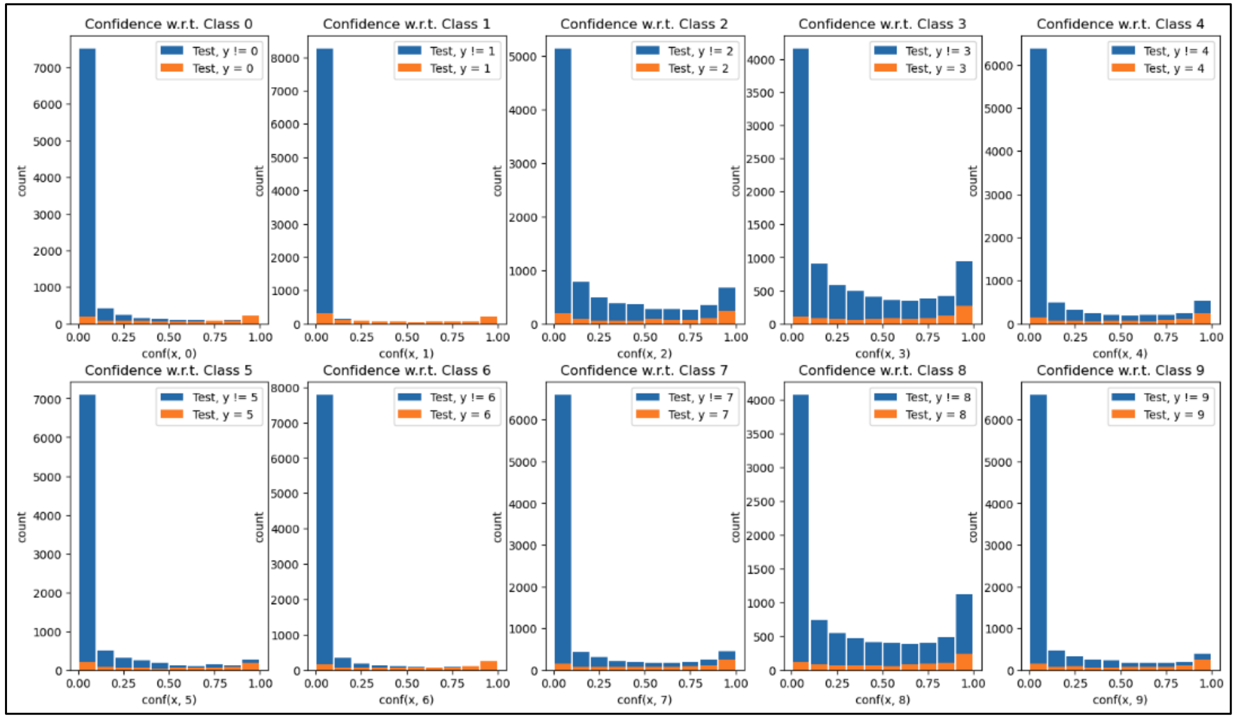
Summary

- ML models can sometimes be over-confident in their predictions
  - Want the model to report low confidence on test points that are far from training data (large statistical distance)
  - The YADA model can achieve this
- YADA – a statistical model of features for indicating when a test point is far from training set
  - Mahalanobis distance of test point from the YADA model for each class
  - An uncertainty or confidence measure can be obtained using the MD
  - Showed results for MNIST image dataset
- YADA can also be used: (1) for creating synthetic data; and (2) as an alternative ML classifier that can provide explanations
- One possible extension: Include feature importance values as weights during YADA training

Results for MNIST Dataset

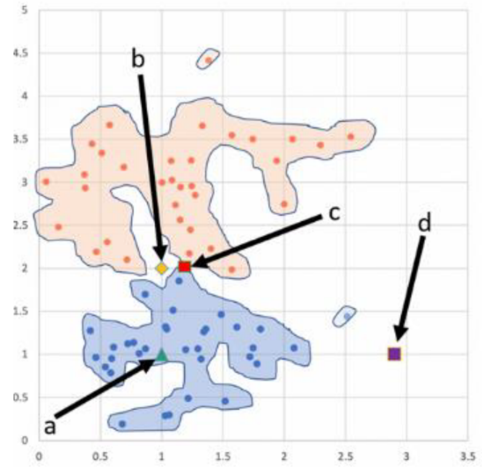
- Left: Histograms of the M distances of each a point from each YADA model
  - Blue = training data with correct label
  - Green = test data with correct label
  - Orange = test data with incorrect label
- Right: Histograms of the confidence for each test point for each YADA model



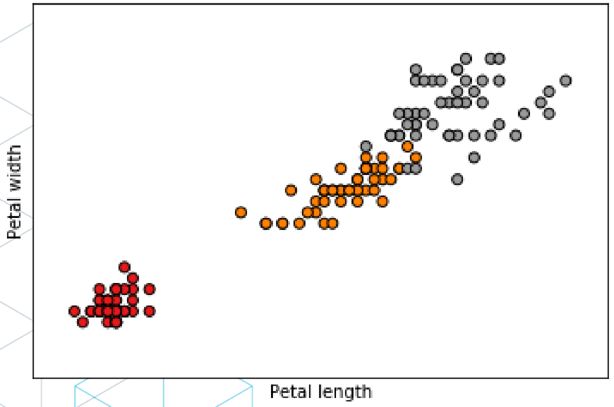


Density-based Trustworthiness

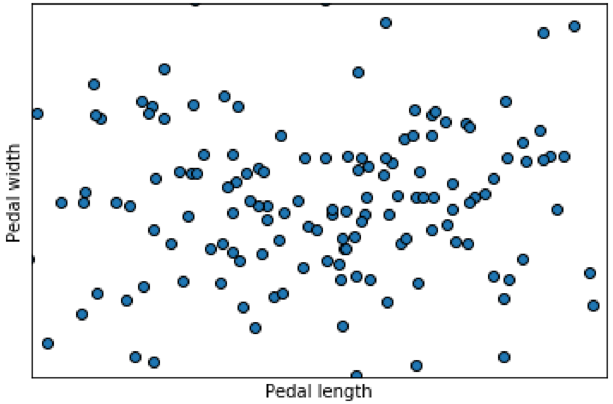
- **YADA: Yet Another Discriminant Analysis**
  - Probabilistic model
  - Based on a Translational Random Variables model (converts features to a Gaussian space)
  - Accounts for correlations (second order/pair-wise)
  - Non-Gaussian features



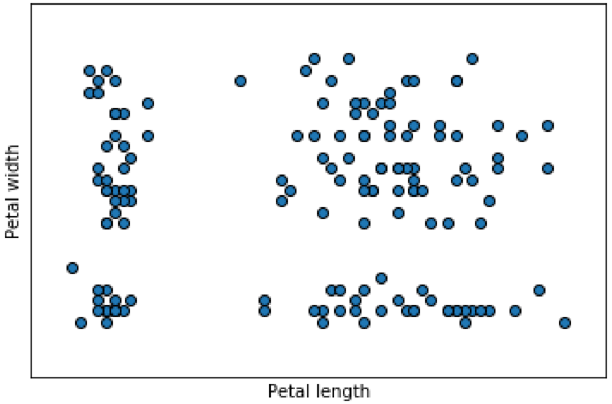
Density: Are test points represented by training data? Do classes overlap?



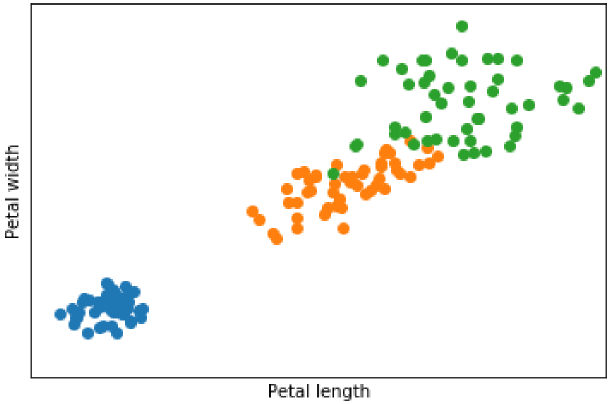
Orig



LIME



Bootstrapping



YADA