Lab 10: Classification Regularization, and Clustering

Math Tools for Neuroscience, 2020

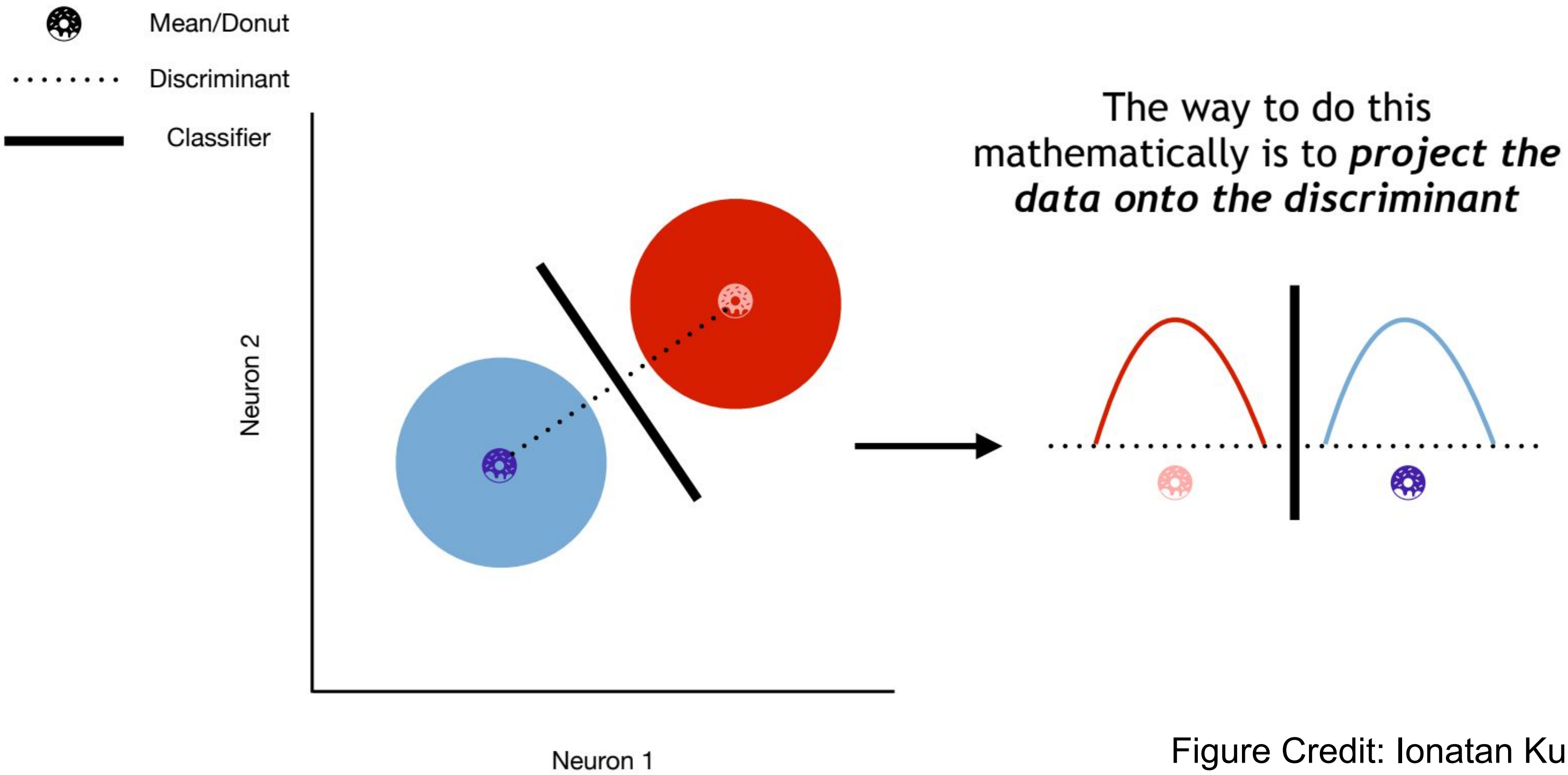
Road Map

- Classification
    - Prototype, FLD, and QDA as solutions to nested special cases of expectation maximization
    - Coding Exercises: Using the contour drawing method of computing the boundaries for data generated in the nested special case fashion to demonstrate the above
- Regularization
    - L1 and L2 Regularization as enforcing different priors
    - Coding Exercise: L1 and L2 regularized polynomial regression for various values of lambda (and sigma)
- Clustering
    - The k-means algorithm
    - Expectation maximizations
    - Running the Hard K-means

Classification 1: The Prototype Classifier

- Assume that samples from both classes come from normal distribution come from normal distributions with covariance matrix = lambda * Identity, and that the only the difference between them comes from the difference between the class means.
- Under this assumption, the likelihood that a sample at a given point came from either class is proportional to the euclidean distance of said point is from each mean.
    - Thus the “decision boundary,” or the set of points for which membership to each class is equally likely is simply the set of points that is equidistant from the two centers.
        - HW 6 Q2: Use this intuition to calculate the decision boundary in closed form!

Classification 1: The Prototype Classifier



The way to do this mathematically is to ***project the data onto the discriminant***

Figure Credit: Ionatan Kuperwajs

Classification 2: The Fisher Linear Discriminant

- Assume that samples from both classes come from normal distribution come from normal distributions with the same (arbitrary/elliptical) covariance matrix.
- It turns out this restriction is also sufficient to derive a closed form solution for the decision boundary as well! (Appropriately the decision boundary is still linear)
    - HW 6 Q2b: prove the above + find a closed form solution for the decision boundary
        - HINT: Can you think of some transformed domain in which this problem may be easier to solve? If so you can solve in this “nice,” domain, then transform back into the original data domain after solving.

Classification 2: Fisher’s Linear Discriminant

Now, let's say we have data that may ***not be well approximated by an identity covariance***. What happens if we don't take that into account?

Now taking into account the covariances with LDA...

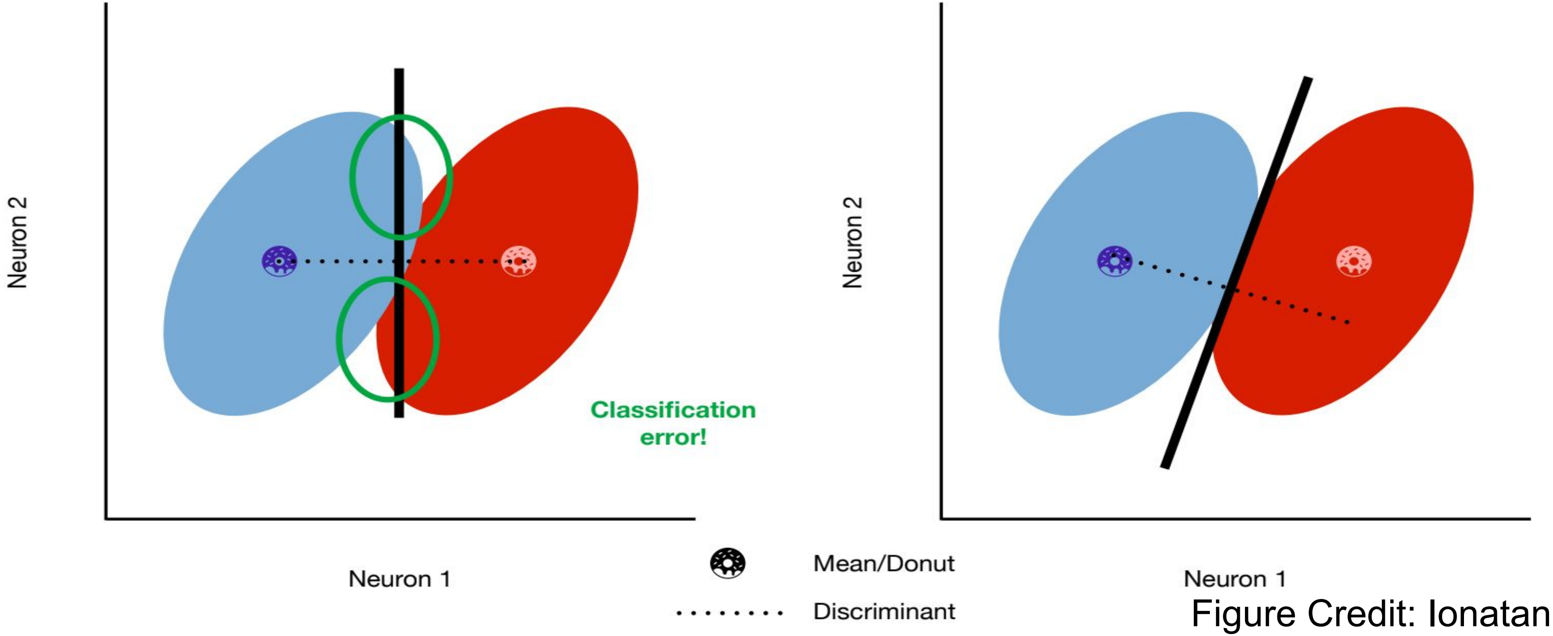


Figure Credit: Ionatan Kuperwajs

Classification 3: Quadratic Discriminant Analysis

- Assume that samples from both classes come from normal distribution come from normal distributions AND NOTHING ELSE.
- This problem does not yield itself to simple closed form analysis, but the same principles still apply: points are assigned to clusters that maximize their likelihood, and the boundary is the set of points where the likelihoods are equal.
    - Appropriately, the decision boundaries in this general case are quadratic forms (as opposed to lines/planes).

Classification 2: Fisher’s Linear Discriminant

Now, we may have data that is not well approximated by assuming equal covariances, even if they are not the identity. Let's see what happens with each classifier in this case:

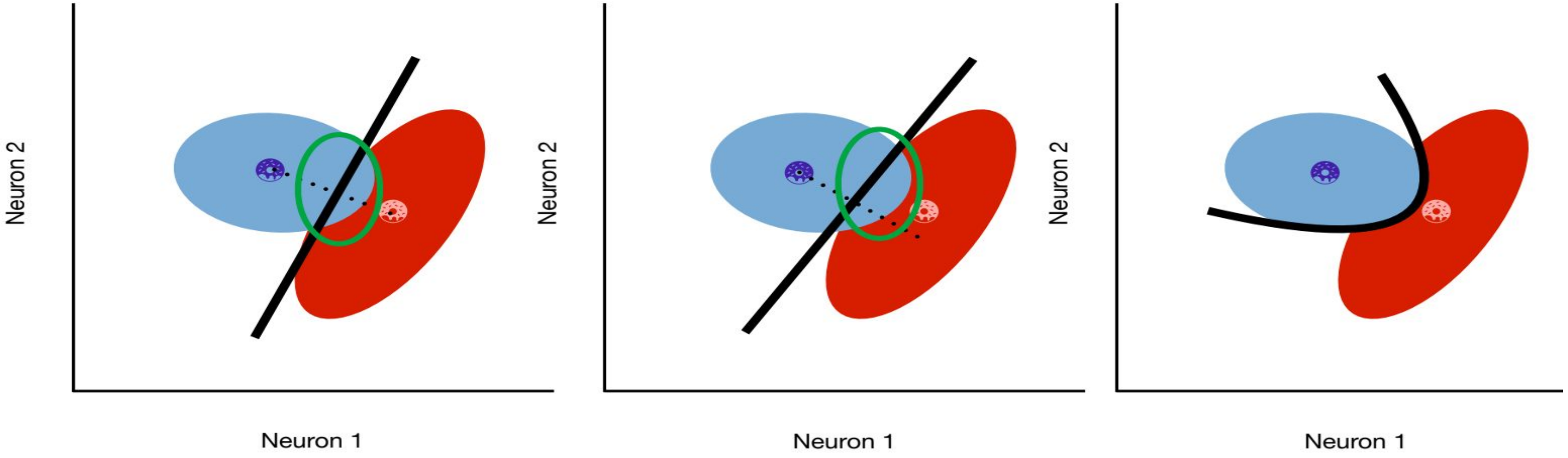


Figure Credit: Ionatan Kuperwajs

Coding Section 1: The prototype and LDA as special cases of QDA

Coding Section 1 Takeaways

- The advantage of using simpler models, that may be less powerful in general, is that they may extrapolate better, and that they require comparatively less data to become accurate .

Regularization: Ridge Regression

Regularization Perspective:

$$\arg\min_{\vec{\beta}} ||\vec{y} - X\vec{\beta}||^2 + \lambda||\vec{\beta}||^2$$

Maximum A Posteriori Perspective:

Minimum of negative log of posterior
<==> Maximum of Posterior

$$\textbf{arg max}_{\vec{\beta}} \; \underset{\substack{\Big\uparrow \\ \text{Gaussian Likelihood}}}{e^{-||\vec{y}-X\vec{\beta}||^2}} \; \underset{\substack{\Big\uparrow \\ \text{Gaussian Prior}}}{e^{-\lambda||\vec{\beta}||_2^2}}$$

Regularization: LASSO

Regularization Perspective:

$$\arg\min_{\vec{\beta}} \; ||\vec{y} - X\vec{\beta}||^2 + \lambda||\vec{\beta}||_1^1$$

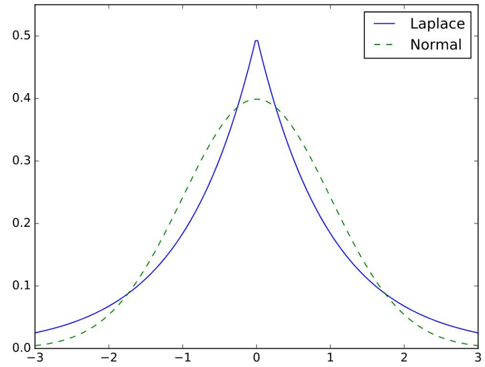
Maximum A Posteriori Perspective:

$$\arg\max_{\vec{\beta}} \; e^{-||\vec{y}-X\vec{\beta}||^2} e^{-\lambda||\vec{\beta}||_1^1}$$

Gaussian Likelihood

Laplacian Prior

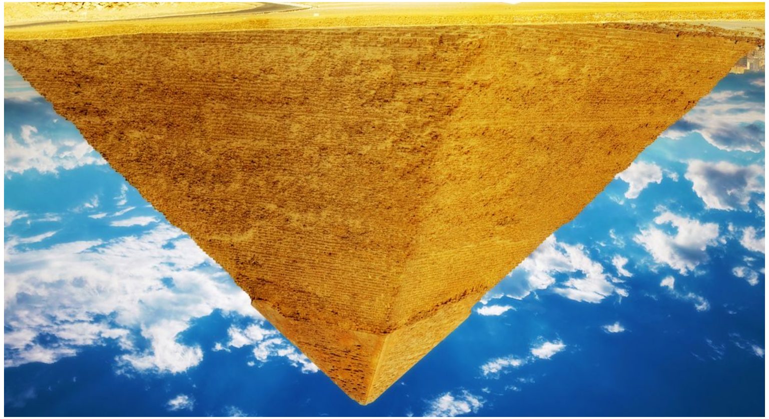
Minimum of negative log of posterior
<==> Maximum of Posterior



Coding Section 2: Lasso and Ridge regression

Coding Section 2: Takeaways

- Each regularization has pro's and con's that make one or the other preferable in various settings (often this depends on the "true," distribution of the parameters).
- Lasso may offer something in the way of "interpretability,"



VS

Clustering: Hard K-means

- Thus far we have been using *known examples* in order to develop a strategy for how to optimally predict future examples (to extrapolate), by making some assumptions about the underlying processes generating the data.
- What happens if, say in the classification context from the first part of this lab, we are not told which class each data point belongs to (or even how many classes are present)?

Clustering: Hard K-means

```
clusters_stable = False;
centers = random(d, k); % initialize k cluster centroids to random points in d-dimensional space

% initialize each data point’s cluster as the nearest centroid
for data_point in data_set:
    cluster_ids(data_point) = argmin_k {||data_point - center_k||^2}

while{! Clusters_stable}
    old_ids = cluser_id;
    % new cluster centroids are mean of each cluster
    for each cluster indexed by k:
        centers(k) =  mean(data_set(cluster_id==k))

    % re-assign each data point to possibly new nearest centroid
    for data_point in data_set:
        cluster_ids(data_point) = argmin_k {||data_point - center_k||^2}

    % Check for convergence
    Clusters_stable = (old_ids - cluster_ids) == 0
```

Clustering: Extensions

- Thinking alllll the way back to the prototype classifier, recall that if we assume each cluster is comprised of samples from normal distributions with (some multiple of) identity covariance then the distance from each point to each center is directly proportional the likelihood that point belongs to that cluster!
    - Modifying the algorithm presented previously can yield a more generalized “Expectation Maximization,” clustering algorithm, in analogy to extending from the prototype to QDA classifiers.
- One immediate issue: how to set the hyperparameter k? As with lambda in the classification case best practice is to use cross validation to decide between models.