Statistical Learning for Data Science

Lecture 14 Appendix

唐晓颖
电子与电气工程系
南方科技大学

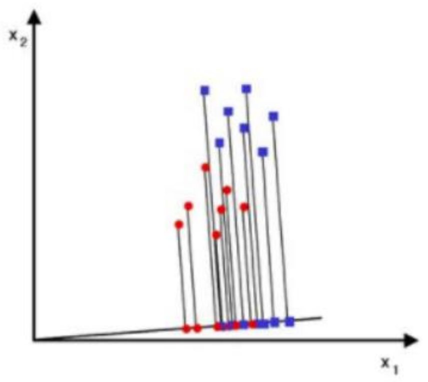
April 17, 2023

Linear Discriminant Analysis (LDA)

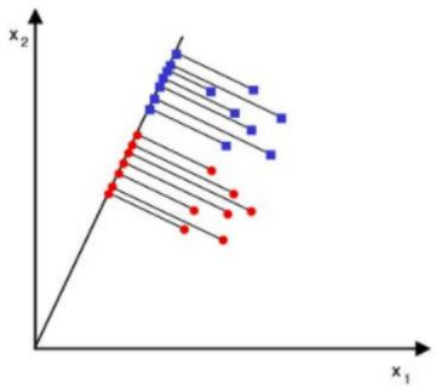
- **linear discriminant analysis** (LDA) is a classical method by Fisher, also called by Fisher discriminant analysis

- basic idea: **project** the data onto a line such that examples within a class are close to each other, while examples from different classes are depart from each other

- reduce dimensionality, preserve as much class discriminatory information as possible



A projection with non-ideal separation



A projection with ideal separation

Linear Discriminant Analysis (LDA) definition

- the within-class distance

$$\sum_{\mathbf{x}\in X_0} \mathbf{w}^\top(\mathbf{x}-\mu_0)(\mathbf{x}-\mu_0)^\top\mathbf{w}+\sum_{\mathbf{x}\in X_1} \mathbf{w}^\top(\mathbf{x}-\mu_1)(\mathbf{x}-\mu_1)^\top\mathbf{w} = \mathbf{w}^\top S_w\mathbf{w}$$

where the within-class scatter matrix:

$$S_w = \sum_{\mathbf{x}\in X_0}(\mathbf{x}-\mu_0)(\mathbf{x}-\mu_0)^\top + \sum_{\mathbf{x}\in X_1}(\mathbf{x}-\mu_1)(\mathbf{x}-\mu_1)^\top$$

- the between-class distance

$$(\mathbf{w}^\top\mu_0 - \mathbf{w}^\top\mu_1)^2 = \mathbf{w}^\top S_b\mathbf{w}$$

Where S_b is the between-class scatter matrix defined by:

$$S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^\top$$

Linear Discriminant Analysis (LDA) optimization

- Objective function

$$\max \ J = \frac{\mathbf{w}^\top S_b \mathbf{w}}{\mathbf{w}^\top S_w \mathbf{w}} \qquad \min_{\mathbf{w}} \ -\mathbf{w}^\top S_b \mathbf{w} \qquad \text{s.t. } \mathbf{w}^\top S_w \mathbf{w} = 1$$

- Lagrangian formulation

$$L = -\mathbf{w}^\top S_b \mathbf{w} + \lambda(\mathbf{w}^\top S_w w - 1)$$

- setting the gradient of Lagrangian to zero

$$S_b \mathbf{w}^* = \lambda S_w \mathbf{w}^*$$

$$S_b \mathbf{w}^* = \underbrace{(\mu_0 - \mu_1)^\top \mathbf{w}^*}_{:=\tilde{\lambda} \in \mathbb{R}} (\mu_0 - \mu_1) = \tilde{\lambda}(\mu_0 - \mu_1) \qquad \mathbf{w}^* = S_w^{-1}(\mu_0 - \mu_1)$$

Quadratic Discriminant Analysis (QDA)

Quadratic Discriminant Analysis (QDA)

- A generalization to linear discriminant analysis is quadratic discriminant analysis (QDA).

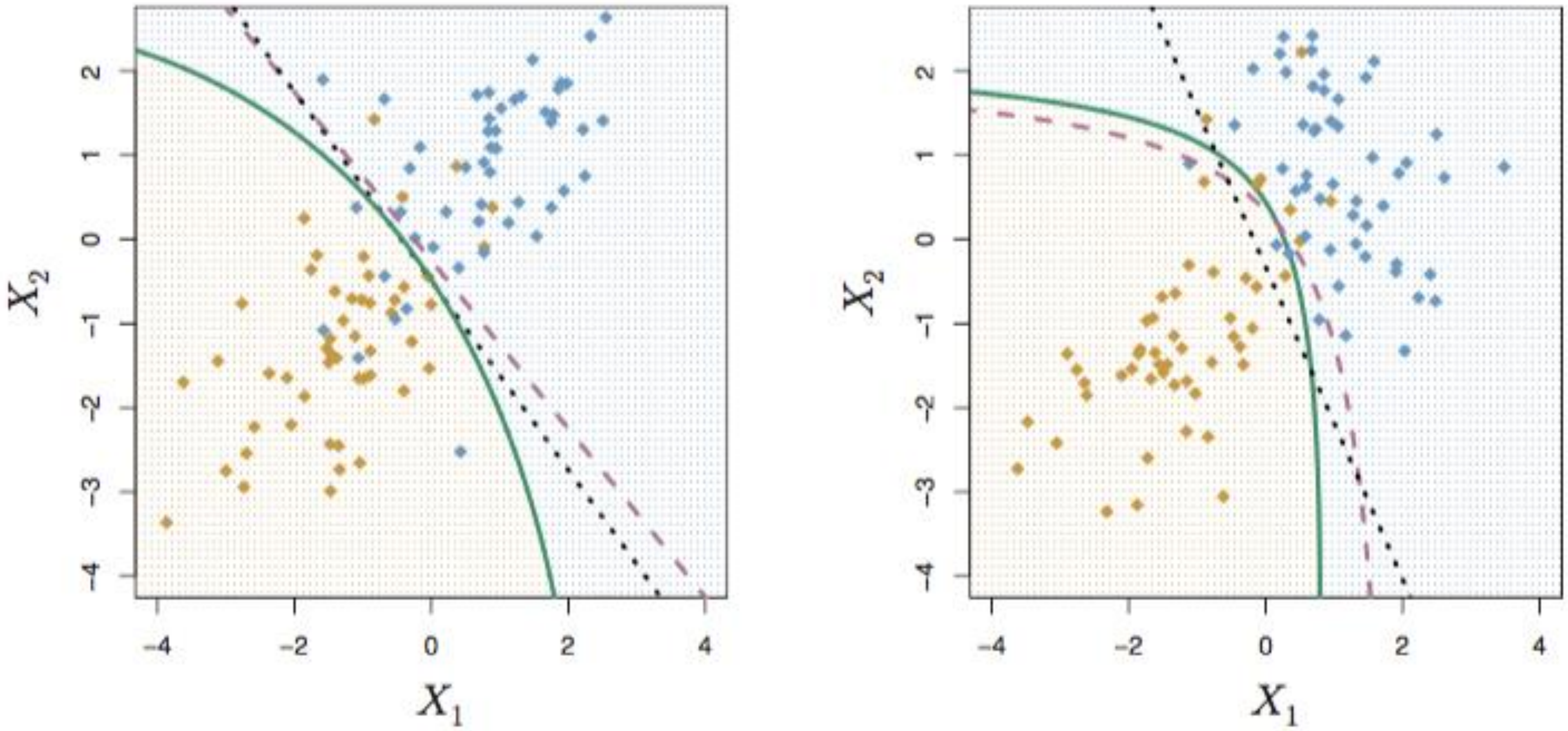
- Why do you suppose the choice in name?

- The implementation is just a slight variation on LDA. Instead of assuming the covariances of the MVN distributions within classes are equal, we instead allow them to be different.

- This relaxation of an assumption completely changes the picture...

QDA in a picture

- A picture can be very illustrative:



QDA (cont.)

- When performing QDA, performing classification for an observation based on its predictors $\vec{x}$ is equivalent to maximizing the following over the *K* classes:

$$\delta_k(\vec{x}) = -\frac{1}{2}\vec{x}^T\Sigma_k^{-1}\vec{x} + \vec{x}^T\Sigma_k^{-1}\vec{\mu}_k - \frac{1}{2}\vec{\mu}_k^T\Sigma_k^{-1}\vec{\mu}_k - \frac{1}{2}\log|\Sigma_k| + \log\pi_k$$

- Notice the `quadratic form' of this expression. Hence the name QDA.

- Now how many parameters are there to be estimated?

- There are *pK* means, *pK* variances, *K* prior proportions, and $\binom{p}{2}K = \left(\frac{p(p-1)}{2}\right)K$ covariances to estimate. This could slow us down very much if *K* is large...

Discriminant Analysis in Python

- LDA is already implemented in Python via the `sklearn.discriminant_analysis` package through the `LinearDiscriminantAnalysis` function.

- QDA is in the same package and is the `QuadraticDiscriminantAnalysis` function.

- It's very easy to use. Let's see how this works

QDA vs. LDA

- So both QDA and LDA take a similar approach to solving this classification problem: they use Bayes' rule to flip the conditional probability statement and assume observations within each class are multivariate Normal (MVN) distributed.

- QDA differs in that it does not assume a common covariance across classes for these MVNs. What advantage does this have? What disadvantage does this have?

QDA vs. LDA (cont.)

- So generally speaking, when should QDA be used over LDA? LDA over QDA?

- The extra covariance parameters that need to be estimated in QDA not only slow us down, but also allow for another opportunity for overfitting. Thus if your training set is small, LDA should perform better for ***out-of-sample prediction***, aka, predicting future observations (how do we mimic this process?)

Comparison of Classification Methods (so far)

Quadratic Discriminant Analysis (QDA)

- We have seen 3 major methods for doing classification:
- Logistic Regression
- *k*-NN
- Discriminant Analysis (LDA and QDA)
- For a specific problem, which approach should be used?

- Well of course, it depends on the nature of the data. So how should we decide?

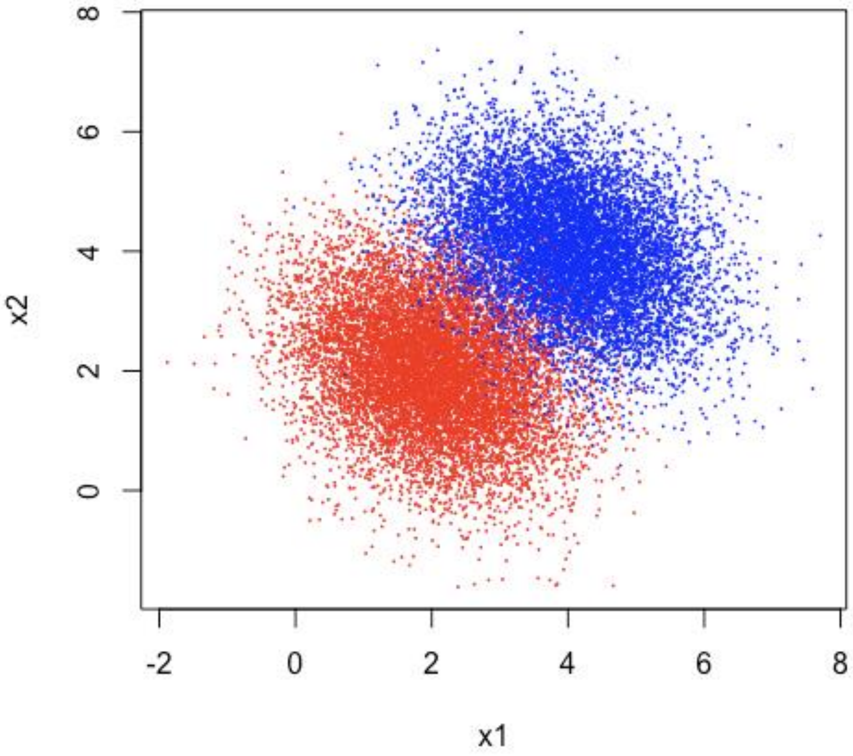
- Visualize the data!

Six Classification Models We'll Compare

- Let's investigate which method will work the best (as measured by lowest overall classification error rate), by considering 6 different models for 4 different data sets (each data set as a pair of predictors...you can think of them as the first 2 PCA components…to come later in the lecture). The 6 models to consider are:
- A logistic regression with only 'linear' main effects}
- A logistic regression with only 'linear' and 'quadratic' effects}
- LDA
- QDA
- *k*-NN where *k* = 3
- *k*-NN where *k* = 25
- What else will also be important to measure (besides error rate)?

Which method should perform better? #1

- n = 20,000, p = 2, K = 2, $\pi_1 = \pi_2 = 0.5$



method	misclass rate	run time (ms)
logit1	0.04410	417.95
logit2	0.04405	229.71
lda	0.04425	50.63
qda	0.04410	49.08
knn3	0.05225	1856.11
knn25	0.04500	2166.57

Notice anything fishy about our answers? What did Kevin do? What should he have done?

Easy to implement in Python

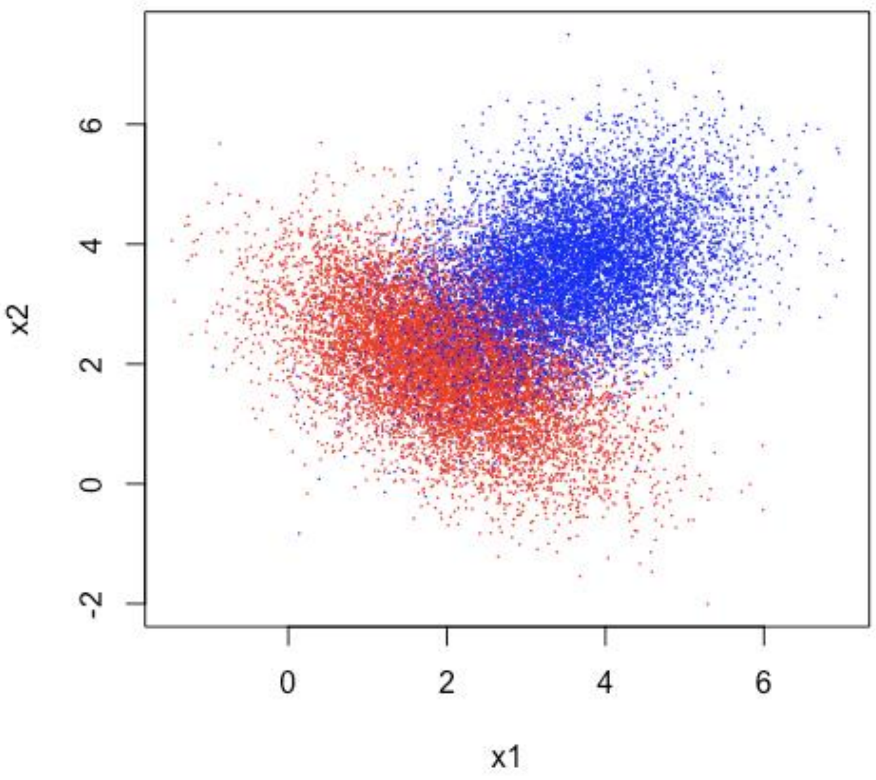
```
lda = da.LinearDiscriminantAnalysis()
qda = da.QuadraticDiscriminantAnalysis()
lda.fit(X,y)
qda.fit(X,y)
logit2 = sk.linear_model.LogisticRegression(C = 1000000)
logit1 = sk.linear_model.LogisticRegression(C = 1000000)
logit1.fit(X,y)
logit2.fit(X2,y)

print("Overall misclassification rate of Logit1 is",(1-logit1.score(X,y)))
print("Overall misclassification rate of Logit2 is",(1-logit2.score(X2,y)))
print("Overall misclassification rate of LDA is",(1-lda.score(X,y)))
print("Overall misclassification rate of QDA is",(1-qda.score(X,y)))
```

```
Overall misclassification rate of Logit1 is 0.0441
Overall misclassification rate of Logit2 is 0.0441
Overall misclassification rate of LDA is 0.04425
Overall misclassification rate of QDA is 0.0441
```

Which method should perform better? #2

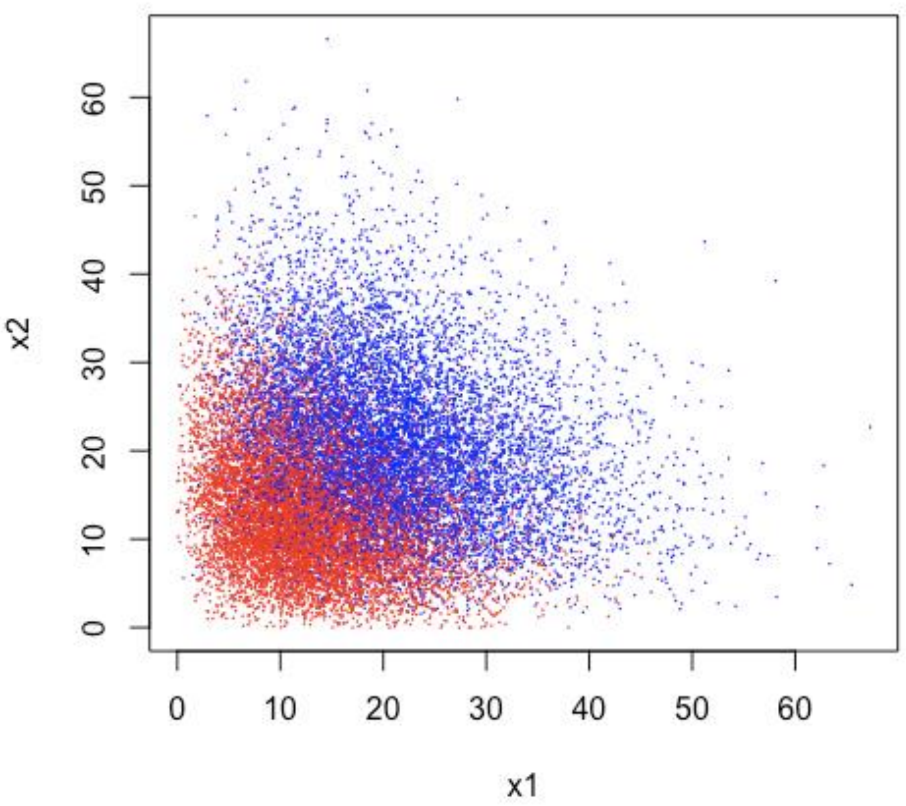
- n = 20,000, p = 2, K = 2, $\pi_1 = \pi_2 = 0.5$



method	misclass rate	run time (ms)
logit1	0.12230	169.53
logit2	0.11860	196.42
lda	0.12215	47.93
qda	0.11445	47.03
knn3	0.14380	1861.90
knn25	0.12015	2223.13

Which method should perform better? #3

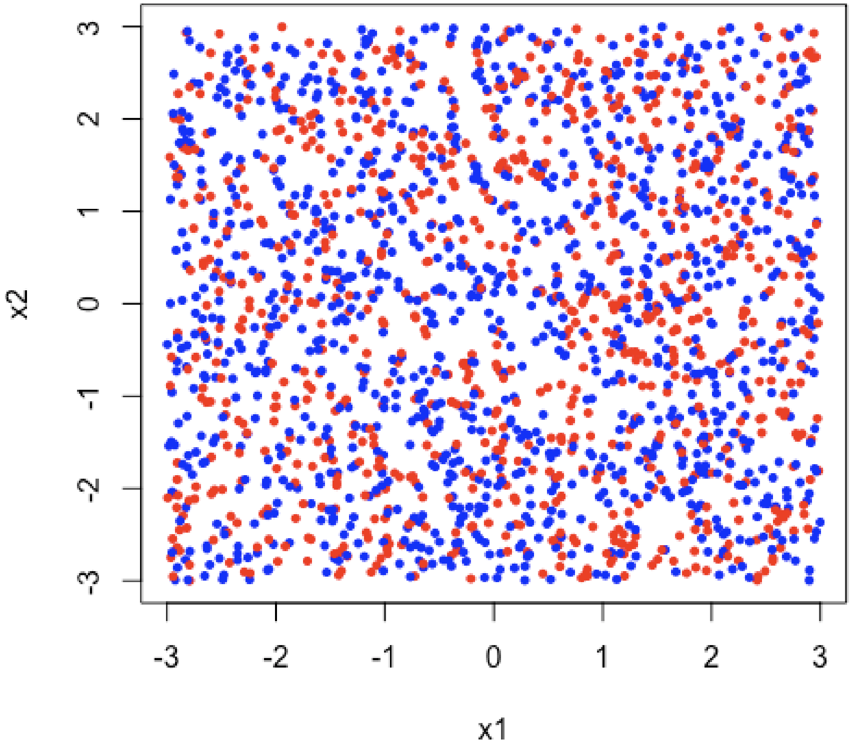
- n = 20,000, p = 2, K = 2, $\pi_1 = \pi_2 = 0.5$



method	misclass rate	run time (ms)
logit1	0.20260	1234.35
logit2	0.19535	192.99
lda	0.21450	49.08
qda	0.20320	60.61
knn3	0.23300	1869.44
knn25	0.20270	2166.77

Which method should perform better? #4

- n = 20,000, p = 2, K = 2, $\pi_1 = \pi_2 = 0.5$

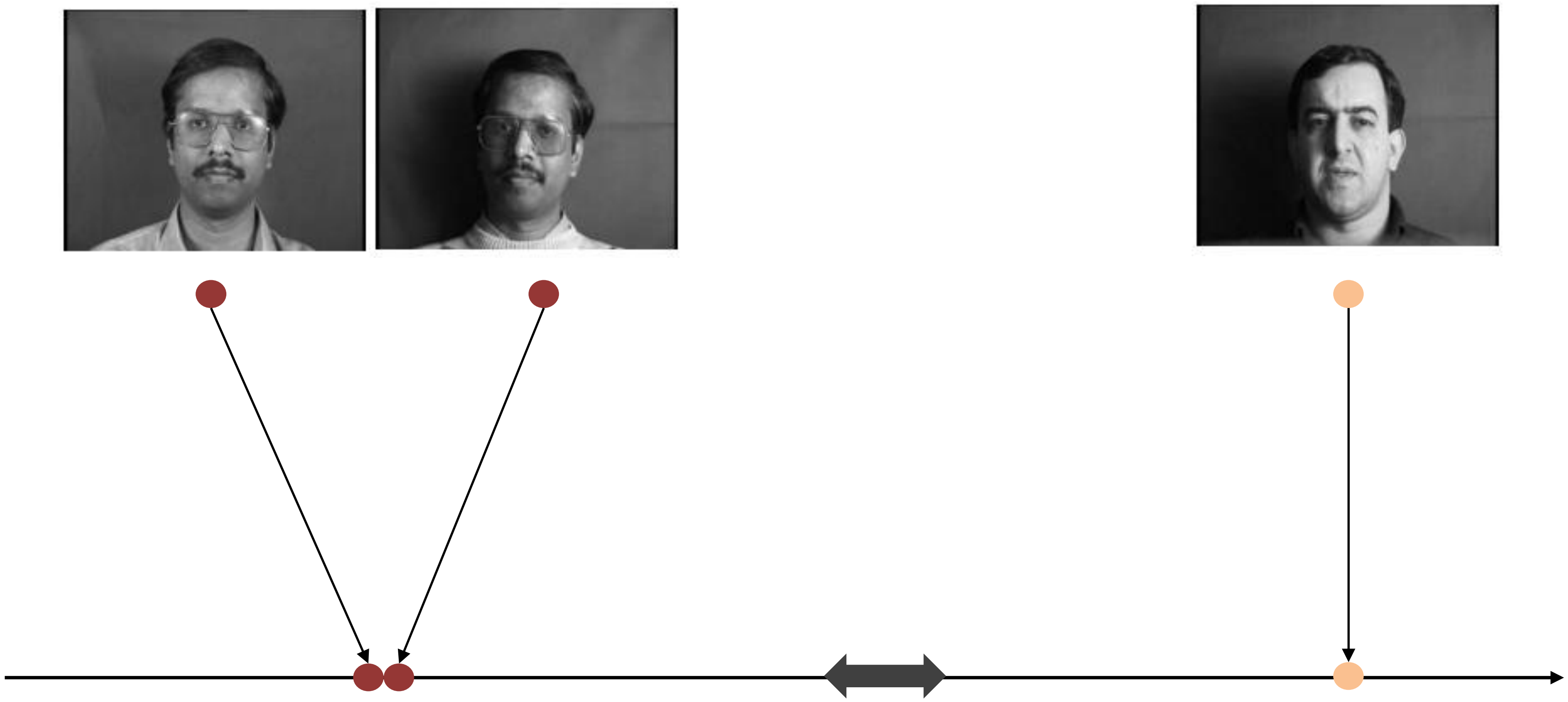


method	misclass rate	run time (ms)
logit1	0.45690	1181.44
logit2	0.37880	147.95
lda	0.45770	51.06
qda	0.40705	44.04
knn3	0.34820	1835.42
knn25	0.30655	2126.38

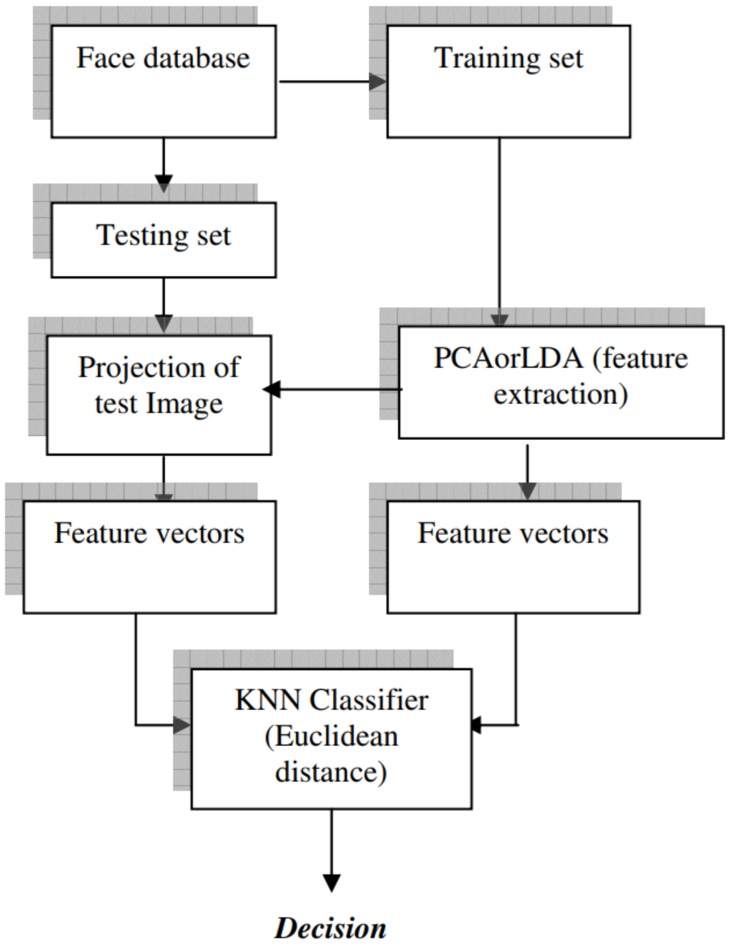
Summary of Results

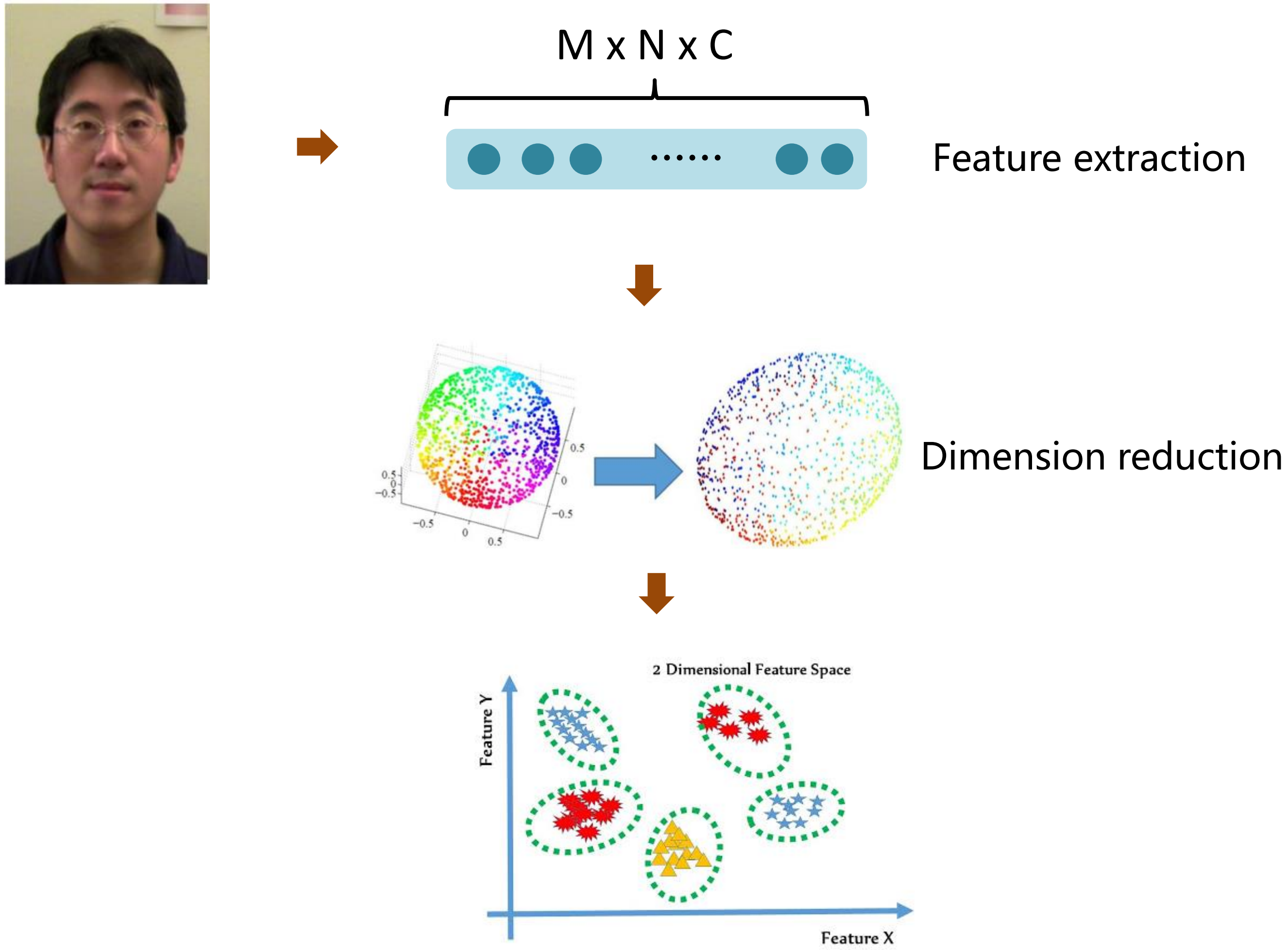
- Generally speaking:
- LDA outperforms Logistic Regression if the distribution of predictors is reasonably MVN (with constant covariance).
- QDA outperforms LDA if the covariances are not the same in the groups.
- k-NN outperforms the others if the decision boundary is extremely non-linear.
- Of course, we can always adapt our models (logistic and LDA/QDA) to include polynomial terms, interaction terms, etc... to improve classification (watch out for overfitting!)
- In order of computational speed (generally speaking, it depends on K, p, and n of course):
  - LDA > QDA > Logistic > k-NN

Application: Face recognition



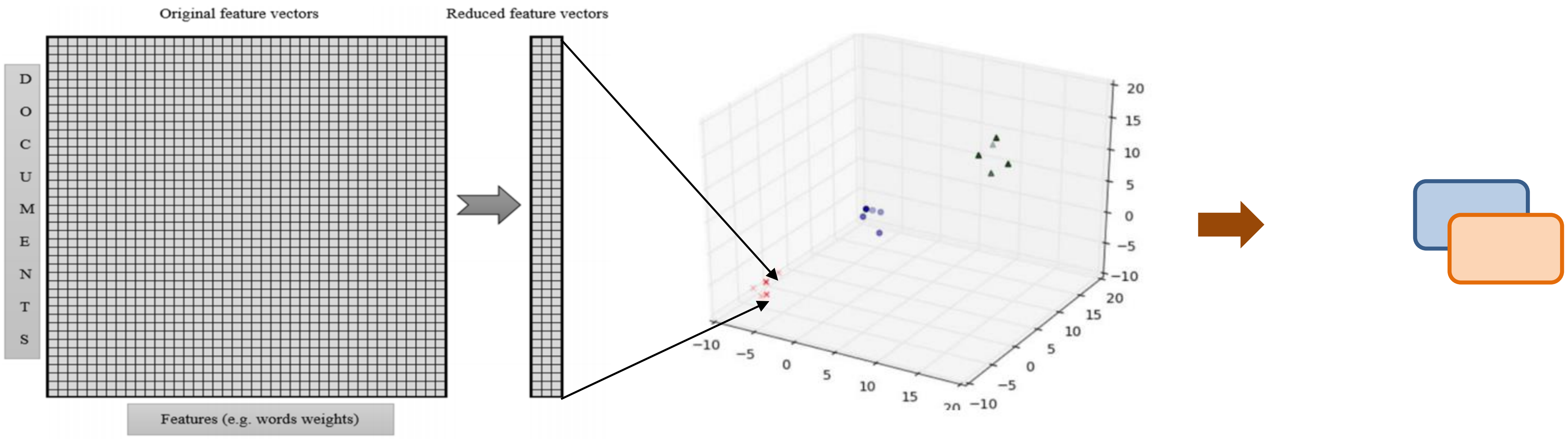
Application: Face recognition





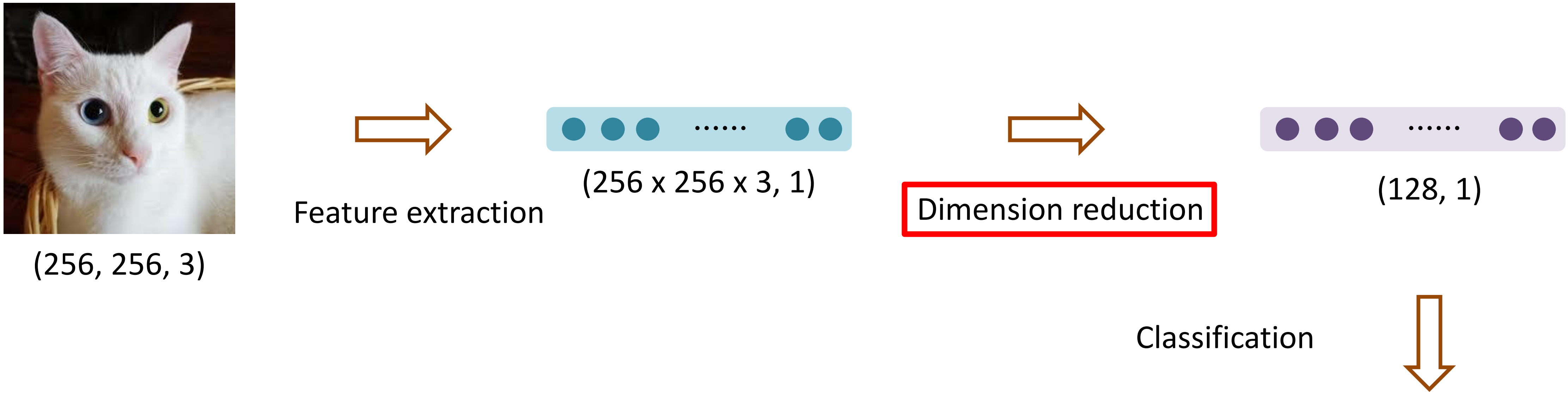
Chelali, F. Z., Djeradi, A., & Djeradi, R. (2009, April). Linear discriminant analysis for face recognition. In *2009 International Conference on Multimedia Computing and Systems* (pp. 1-10). IEEE.

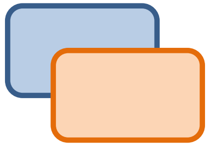
Application: Text classification



Al-Anzi, F. S., & AbuZeina, D. (2017, May). Arabic text classification using linear discriminant analysis. In *2017 International Conference on Engineering & MIS (ICEMIS)* (pp. 1-6). IEEE.

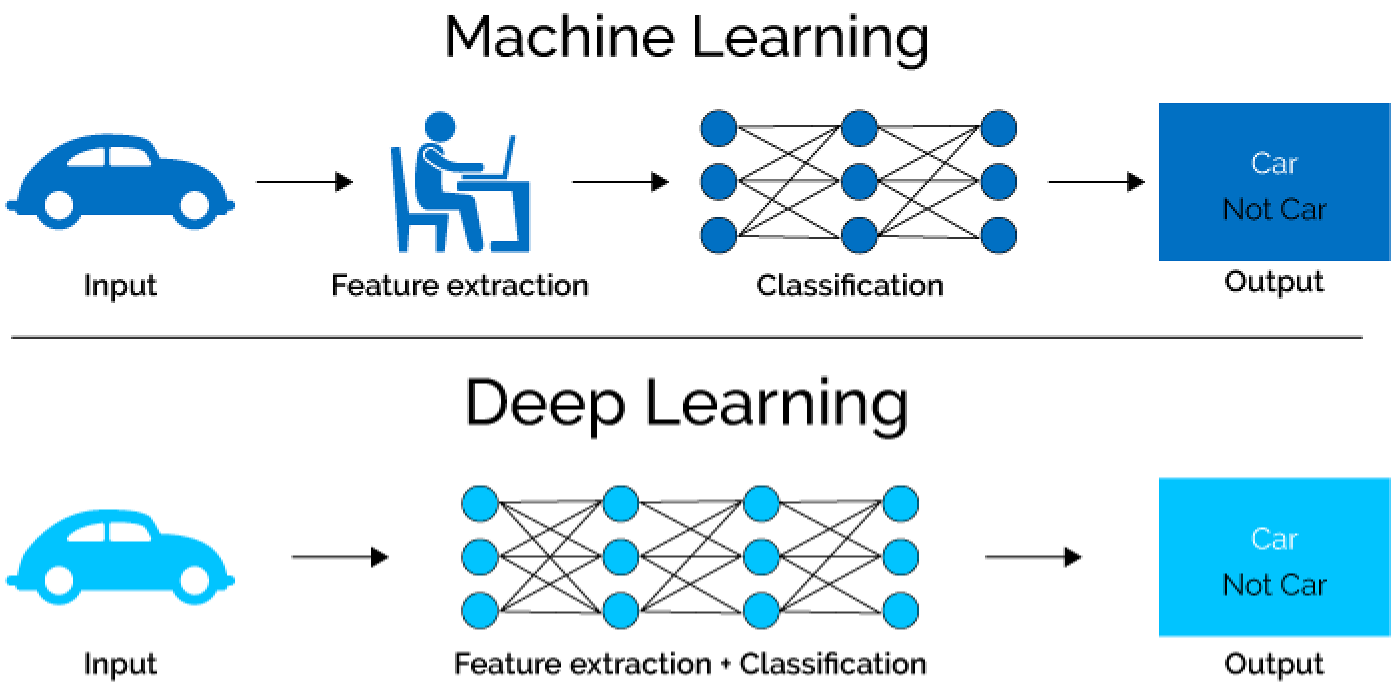
Application: What does LDA do?





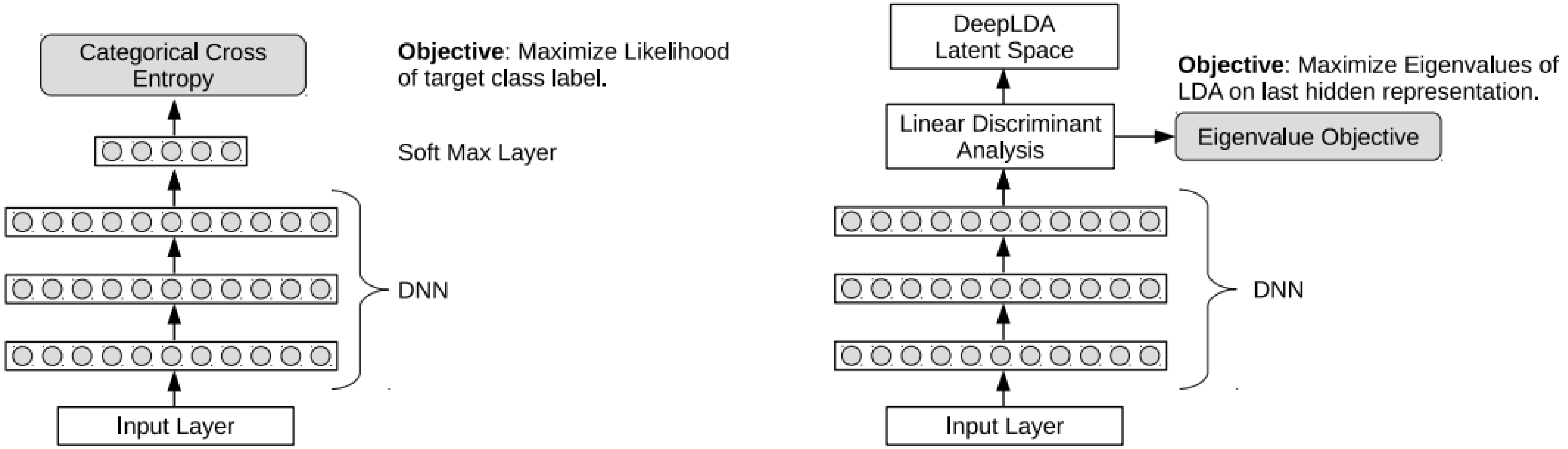
LDA mainly focuses on the **dimensional reduction** processing.

Application & Development: Data-driven feature extraction



Learning a strategy to extract feature, reduce dimension and classify from large-scale dataset!

Application: Deep LDA



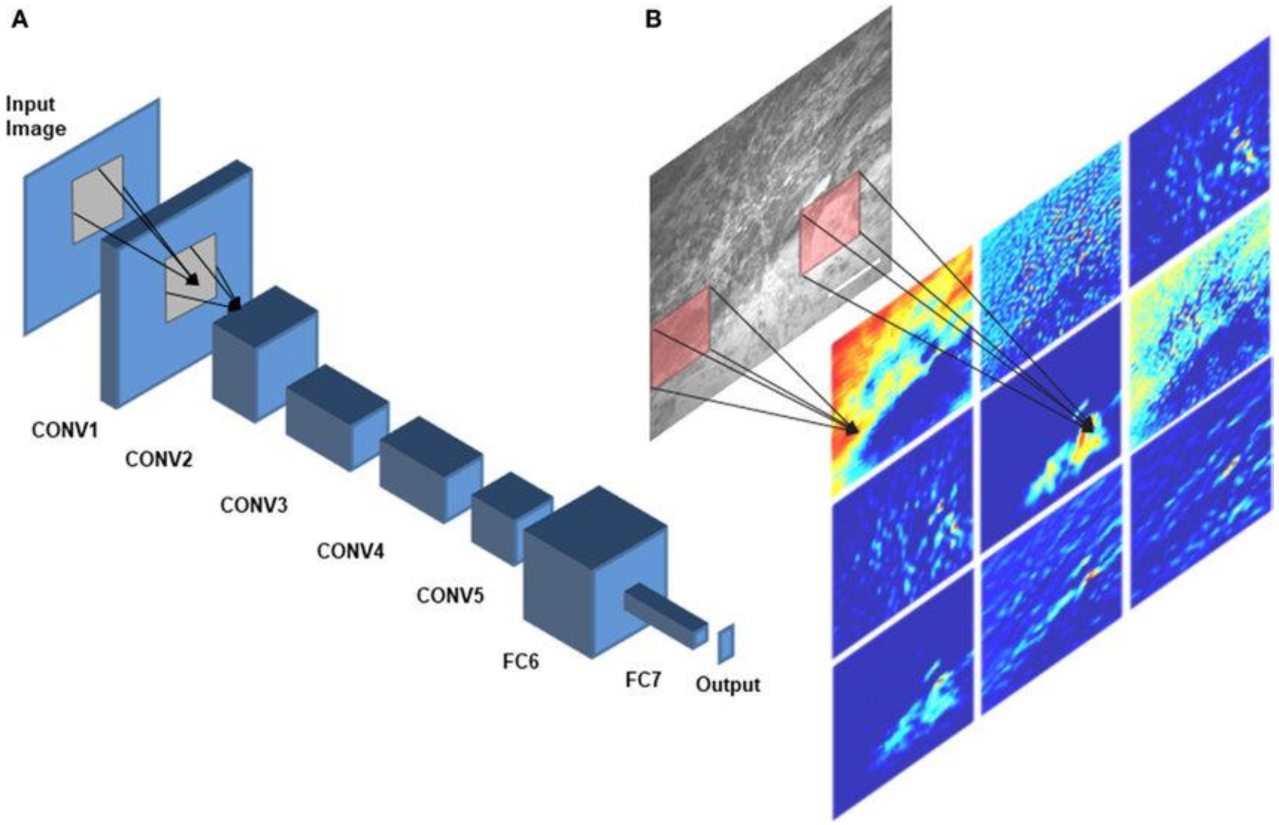
(a) The output of the network gets normalized by a soft max layer to form valid probabilities. The CCE objective maximizes the likelihood of the target class under the model.

(b) On the topmost hidden layer we compute an LDA which produces corresponding eigenvalues. The optimization target is to maximize those eigenvalues.

Figure 1: Schematic sketch of a DNN and DeepLDA. For both architectures the input data is first propagated through the layers of the DNN. However, the final layer and the optimization target are different.

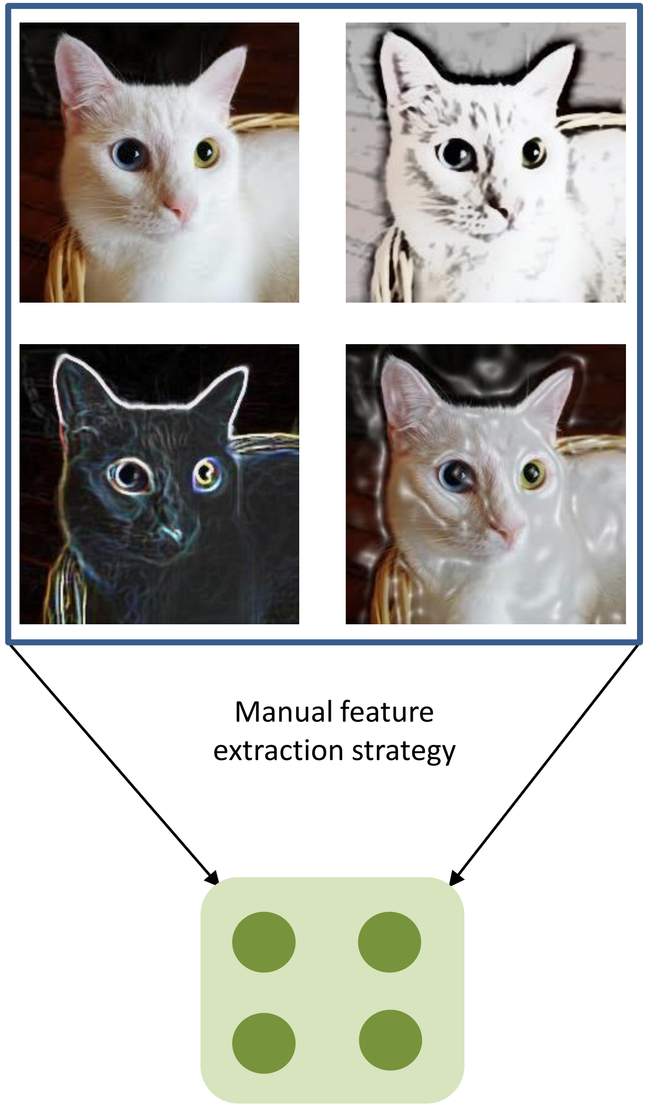
Dorfer, M., Kelz, R., & Widmer, G. (2015). Deep linear discriminant analysis. *arXiv preprint arXiv:1511.04707*.

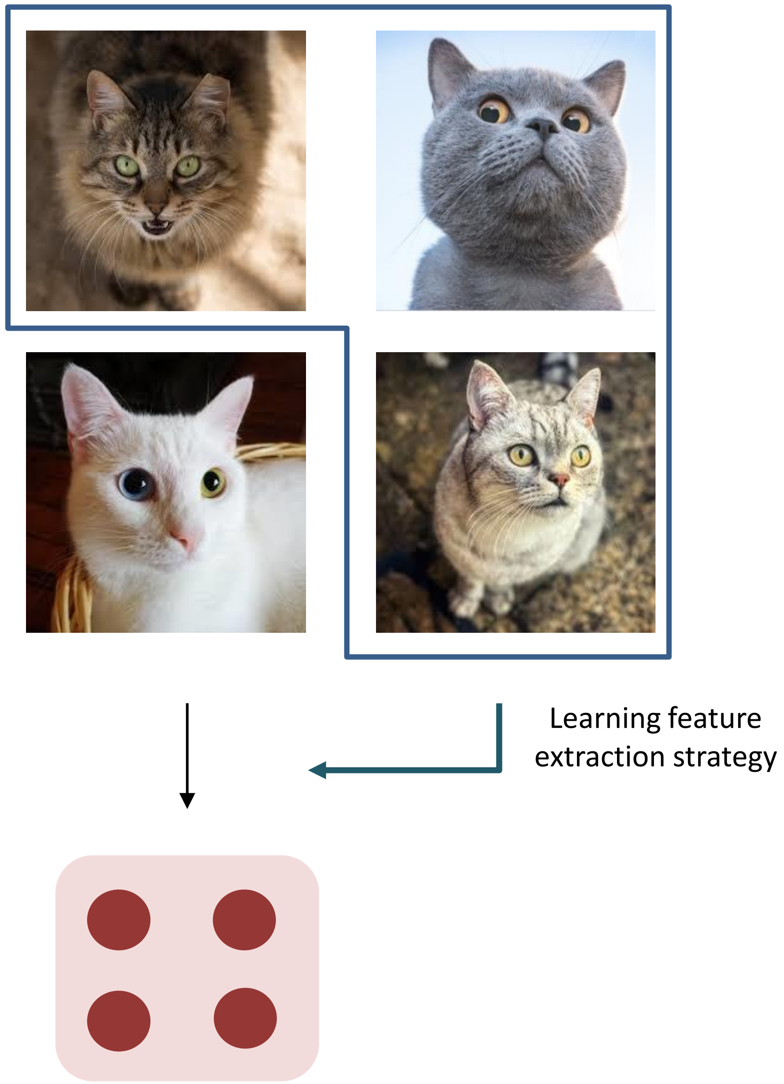
Application: feature reduction in deep learning



786432	(512, 512, 3)	Convolutions & Poolings
4194304	(256, 256, 64)	
2097152	(128, 128, 128)	
1048576	(64, 64, 256)	
* 524288	(32, 32, 512)	
262144	(16, 16, 1024)	
1024	(1, 1024)	Pooling & Flatten

Manual VS Data-driven





Disadvantages & Advantages of LDA framework

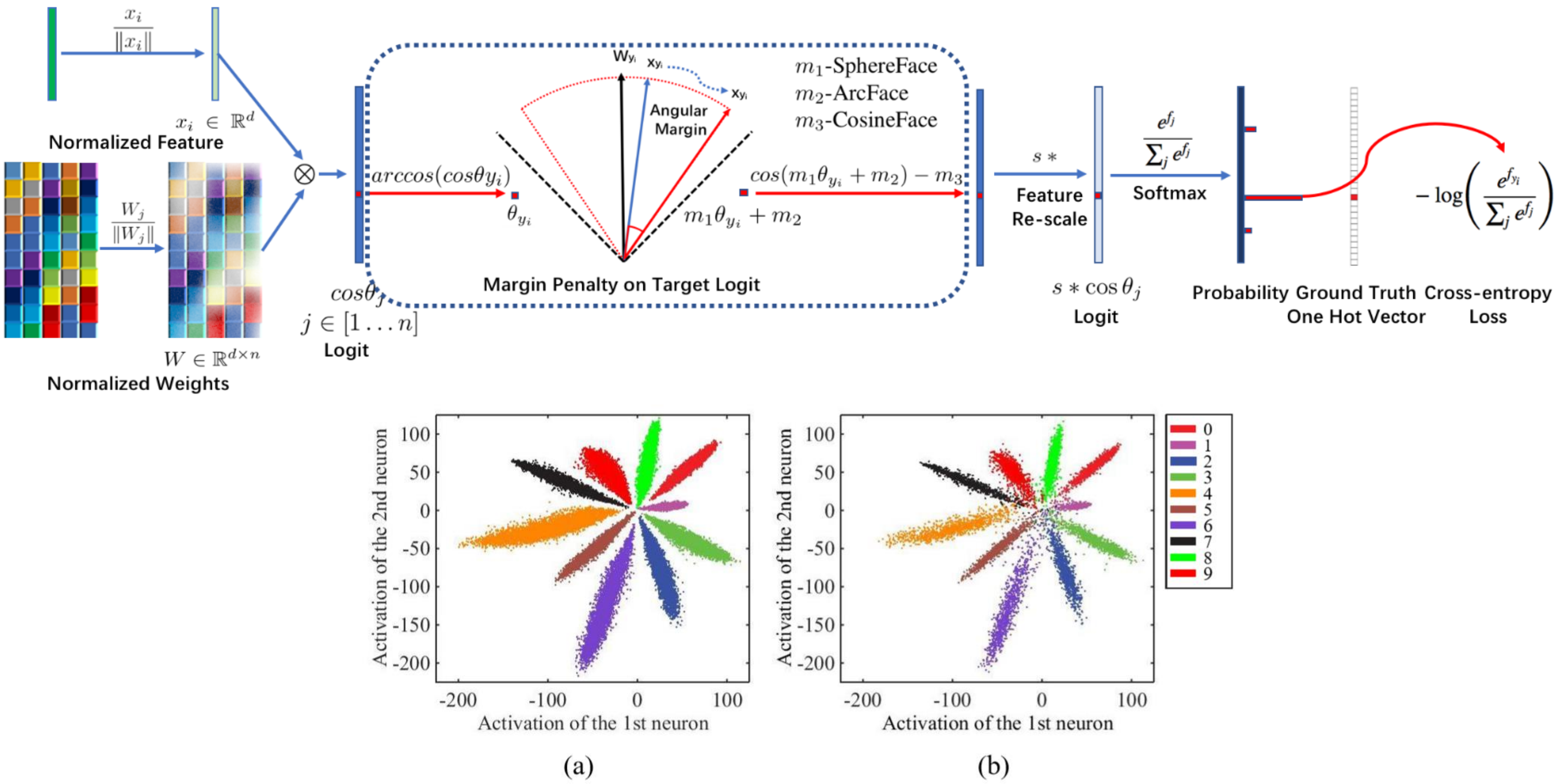
Advantages

- Be independent on the feature extraction.
- Focus on the inter-class and between-class distance
- Be dependent on the prior knowledge

Disadvantages

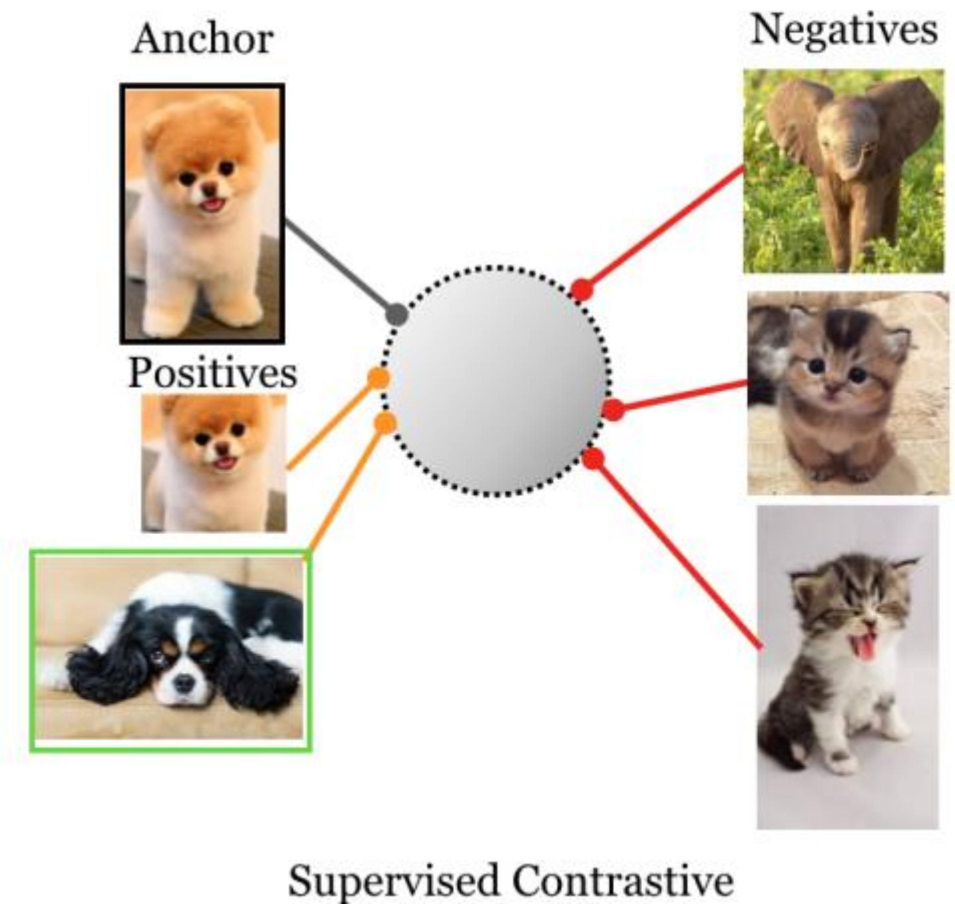
- Lack of semantic information and data-driven features
- Strict constraint of pdf and variance consistency assumption

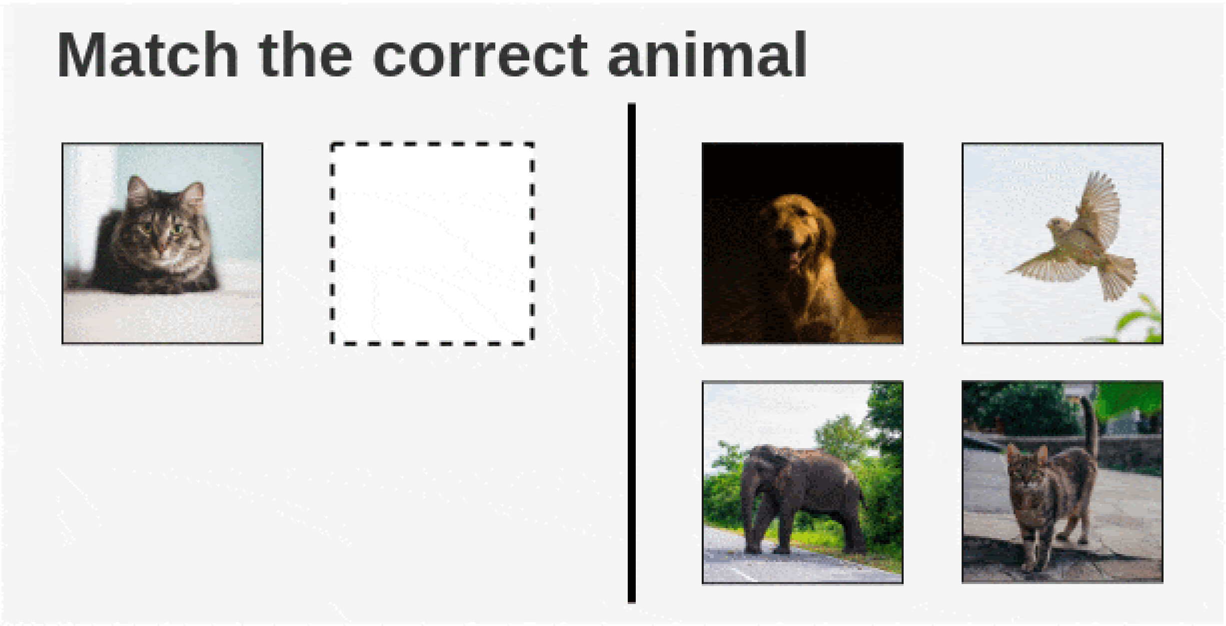
Replacement of LDA: ArcFace



Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4690-4699).

Replacement of LDA: Contrastive loss





Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., ... & Krishnan, D. (2020). Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*.