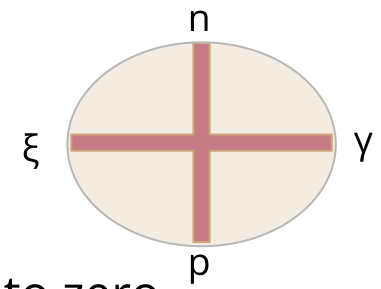
Sparse vs Dense Classification

Assignment 3

Overview - Datasets

Radar-Plot depiction of dataset properties:



Simulate ***β*** ~ $N(0, I)$, and set a certain fraction of components to zero According to sparsity $\xi \in [0,1)$

Simulate data ***X*** with dimensions (n = #samples, p = #features)

Add noise ***ε*** to the responses based on a chosen signal-to-noise ratio γ

In the following: Varying n, p, γ. Always look at $\xi \in \{0.0, 0.9\}$

Binary classification of $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}$

Overview - Classification

Compare dense and sparse classifiers.

Classifiers used:

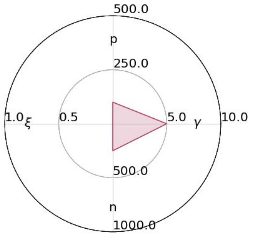
- Logistic Regression with L2 & L1 penalty
  - (default regularization strength $\lambda=1$)
- Nearest Centroids & Shrunken Centroids (shrink_threshold=0.5) (cf. sklearn)

Used 5-fold Cross Validation to estimate performance

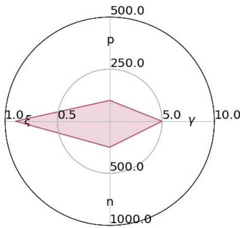
Results averaged over 100 repetitions

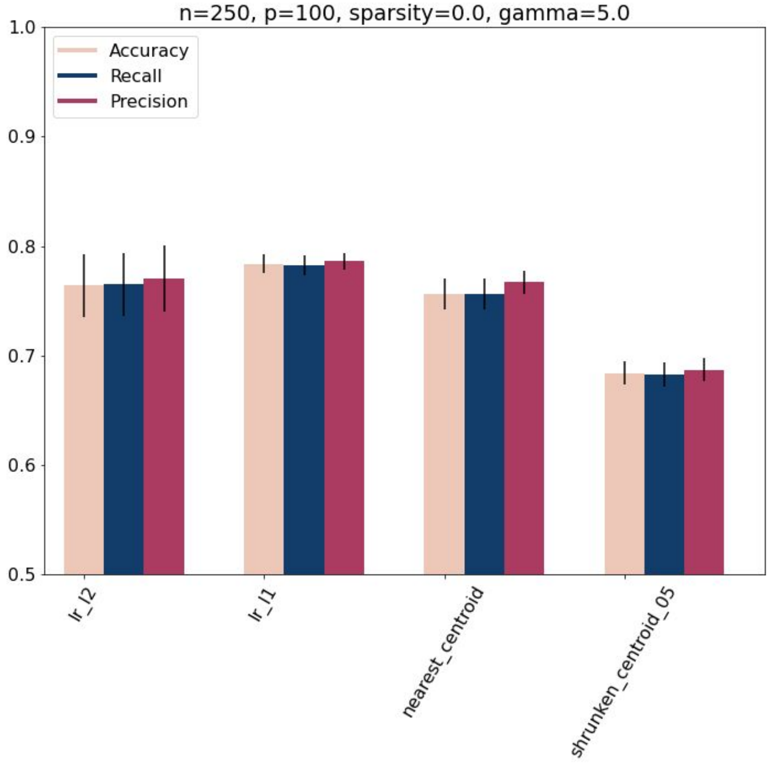
- Standard error represented in bar plots by black, vertical lines

Focus on the impact of sparsity ξ in different parameter settings



Dataset Dimension - Small *n*

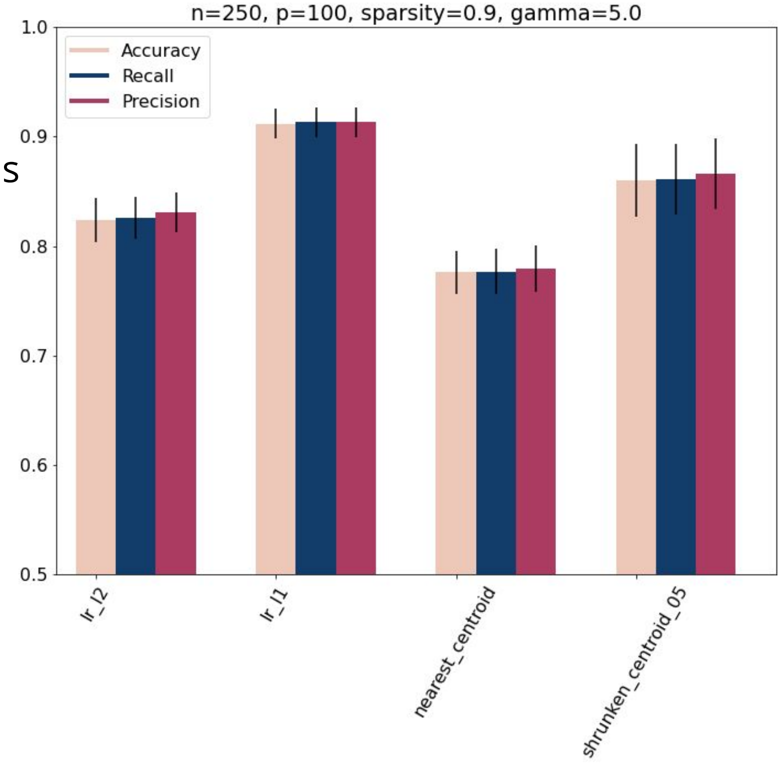


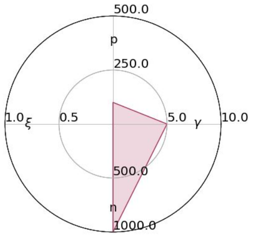


Left:
Only shrunken centroids performance stands out

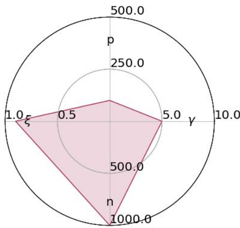
Right:
With high sparsity, Sparse algorithms improve

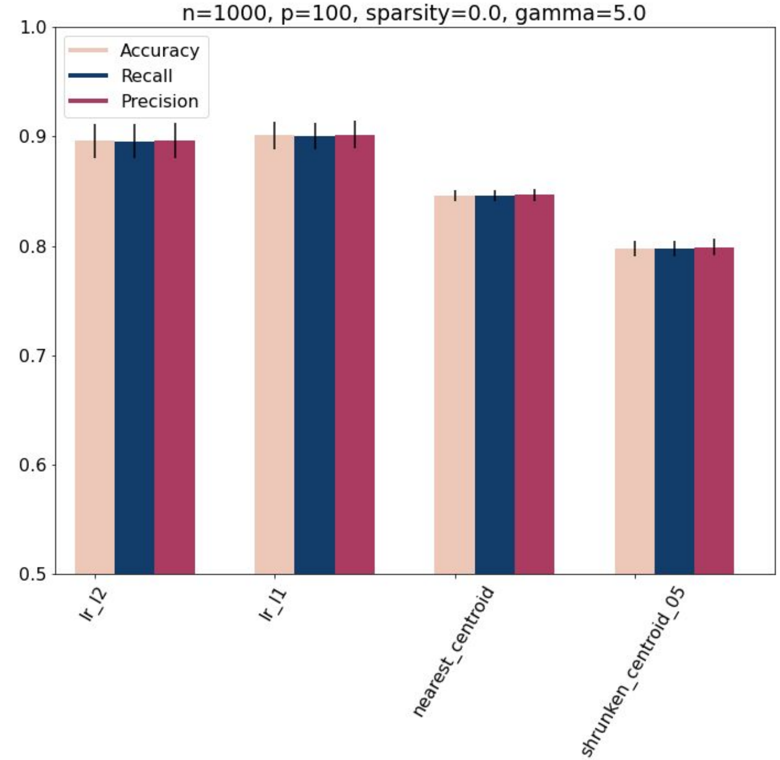
Dense algorithms stay put (mostly)





Dataset Dimension - Large *n*

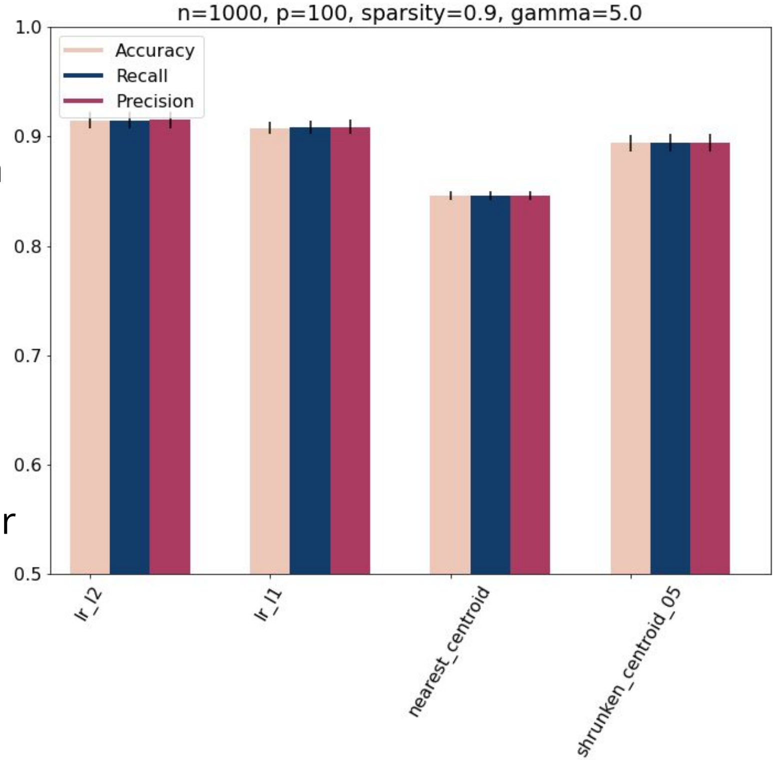


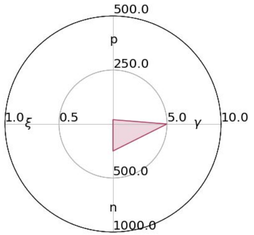


For larger sample sizes:

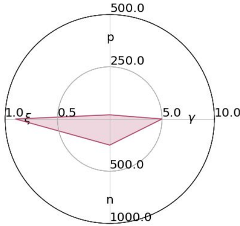
Difference between sparse and dense algorithms not as large for Logistic Regression

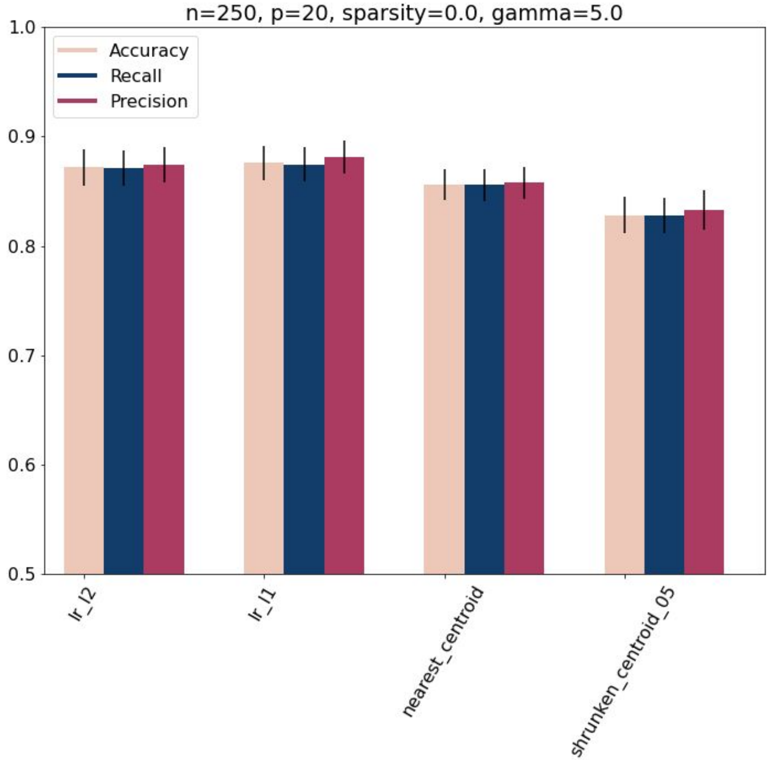
Large sample size seems to rectify effects of sparsity or noise





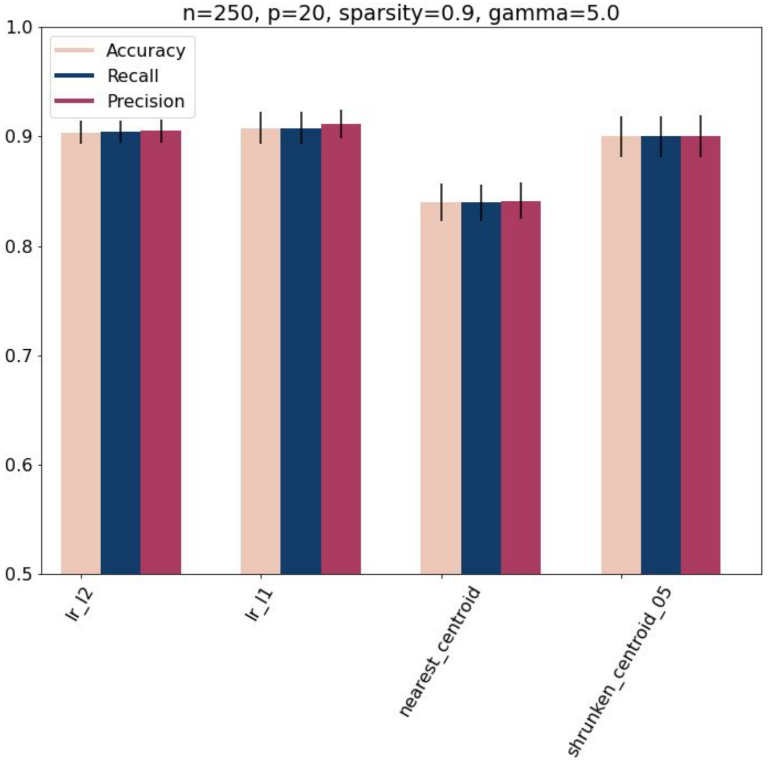
Dataset Dimension - Small *p*

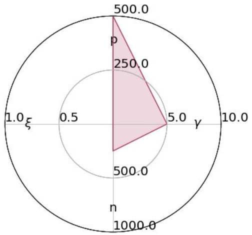




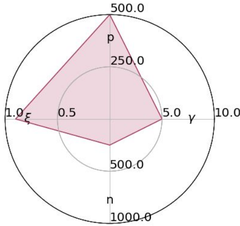
Plots look very similar to the slide before

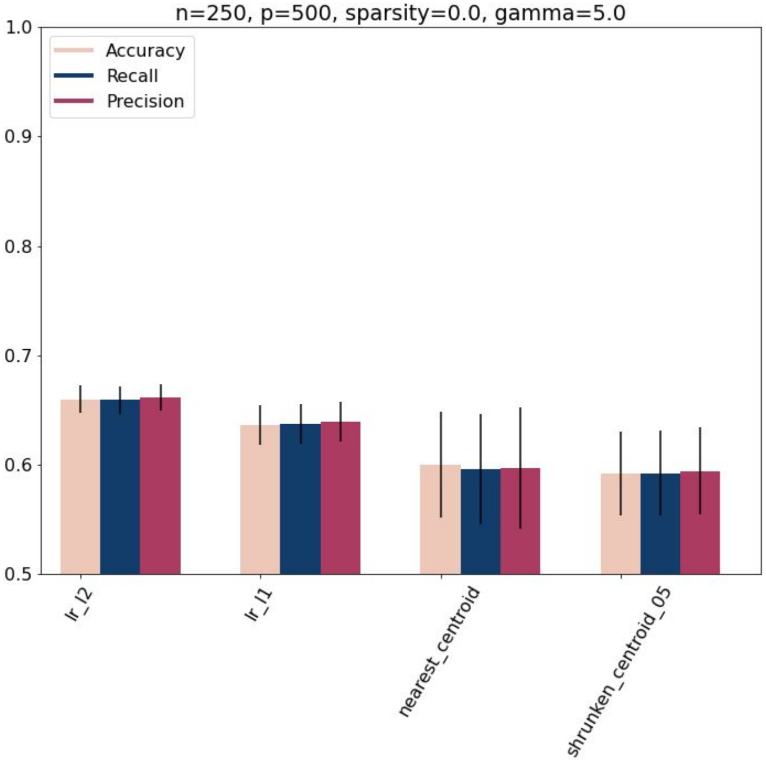
Performance seemingly governed by ratio *n/p*



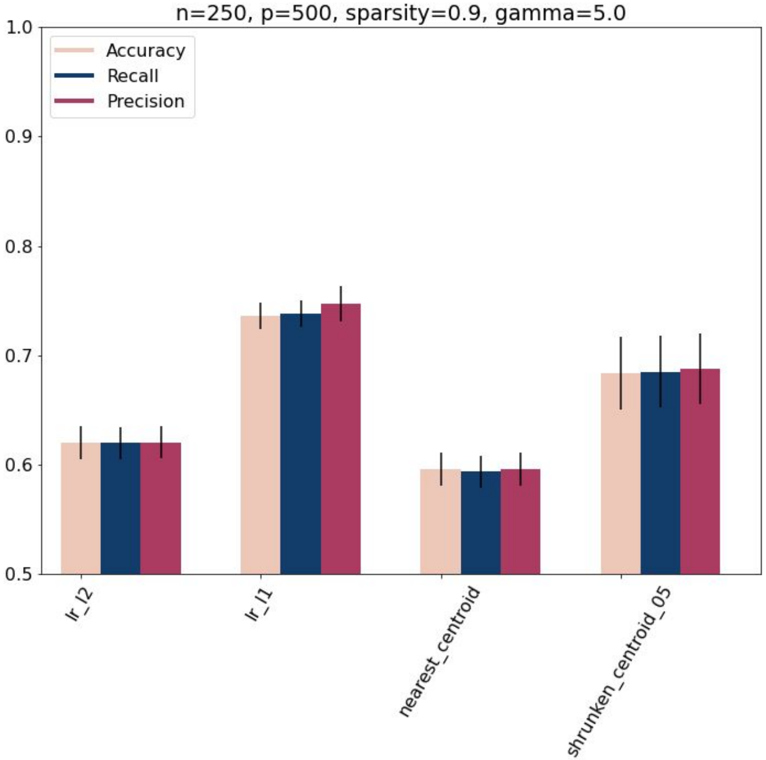


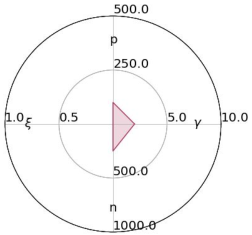
Dataset Dimension - Large *p*



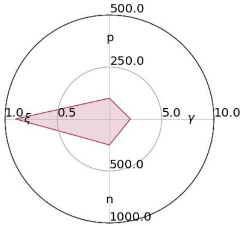


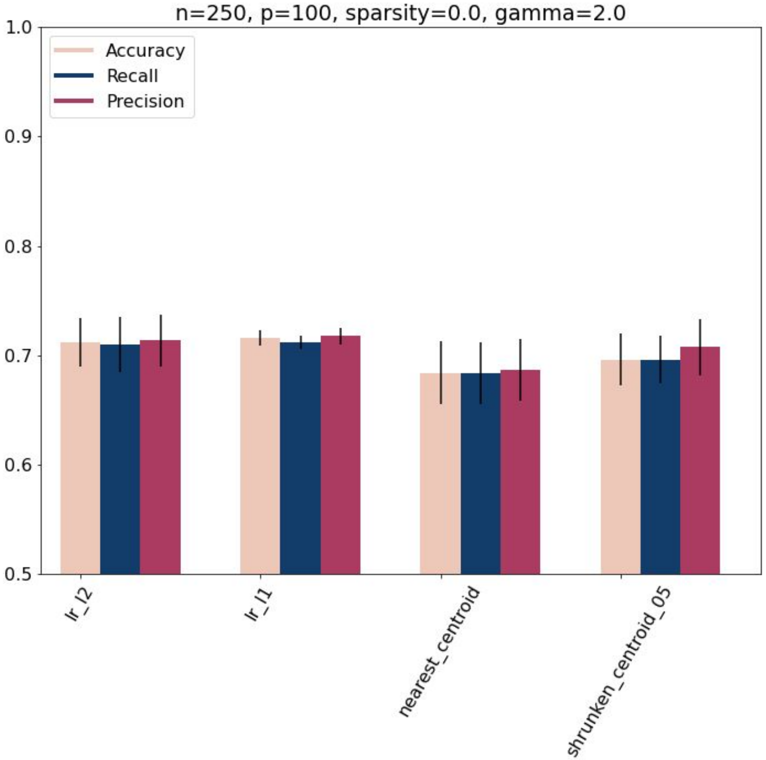
Significant jump in performance for sparse algorithms in a sparse *p*>*n* setting





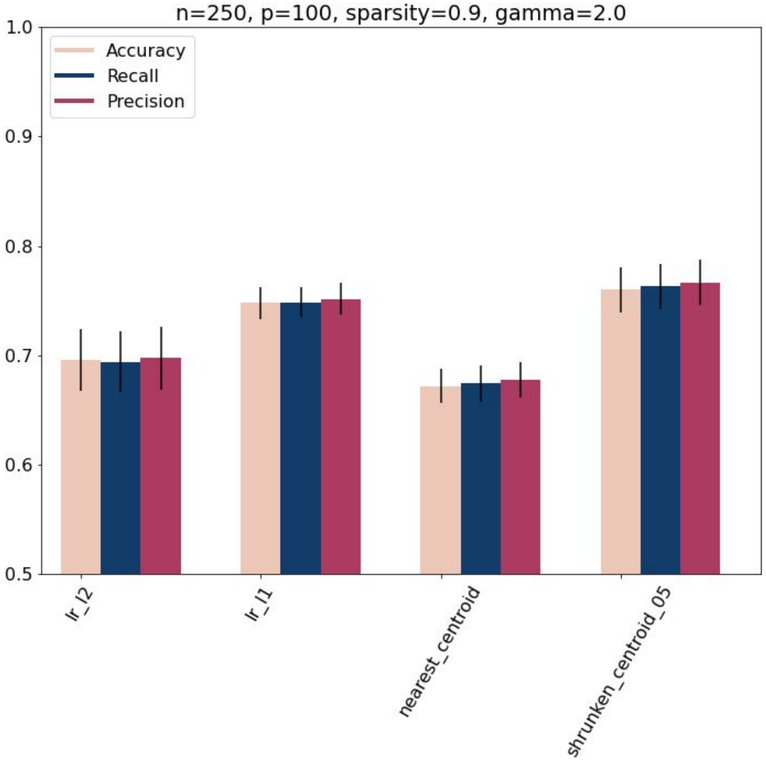
Noise - Small γ

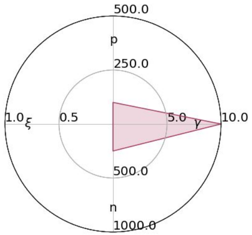




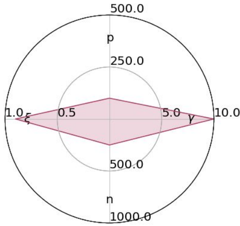
Lower γ means more noise

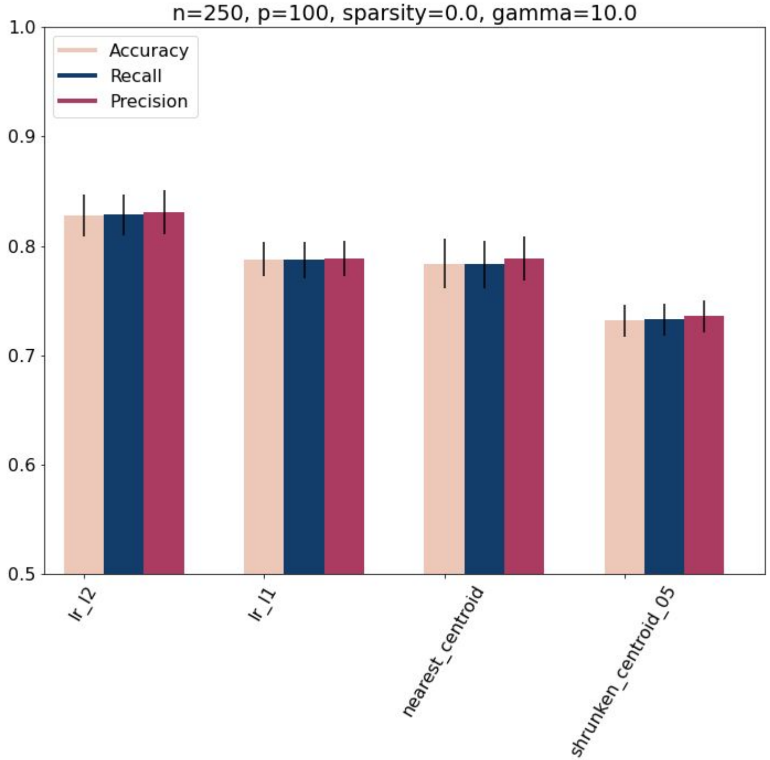
Right:
In sparse & noisy setting,
sparse algorithms better than their dense counterparts





Noise - Large γ



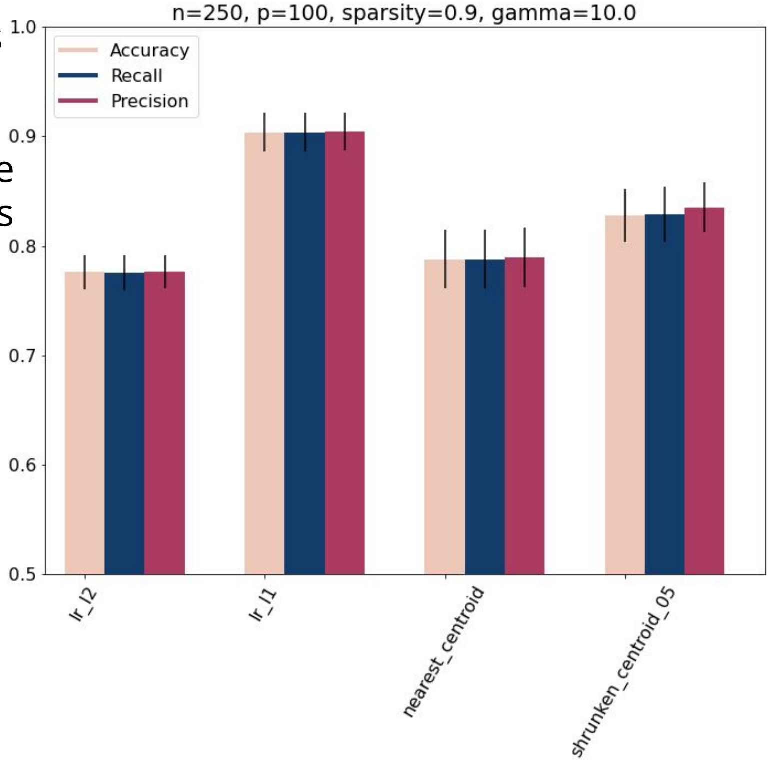


Higher *γ* means less noise

Here, optimal choice of classifier depends on sparsity of data:

<u>Left:</u> Dense data, dense classifiers better

<u>Right:</u> Sparse data, sparse classifiers better



Conclusion

With many samples: Dense Classifiers can deal with both dense and sparse data (cf. slide 5)

- Dataset provides enough information for classifier
- Significance of sparse features can be learned even by dense algorithm
- Shrunken centroids performs worse on dense data, since centroids are closer to the origin than the *real* class centroids

Conclusion

In *p*>*n* setting (or generally small *n*): Sparse Classifiers are desirable as they perform equally as good (dense data) or better (sparse data) and are easier to interpret

- Sparse dataset provides too uncertain statistics for dense algorithms to properly learn the significance of features

For low-noise data the choice of sparse vs. dense classifier can play a significant role based on data sparsity

- Statistics good enough for dense algorithm to learn significance of each dense feature, but fails to penalize irrelevant features on sparse data.
- Sparse algorithms might try to filter actually relevant features from dense data