

bio - finder

PHD DEFENSE



Abstract:

↳ BIO - FINDER stands for Bioinformatics Feature Interference on Noisy Datasets

using Eigenspace Residuals in machine learning

↳ Karhunen-Loeve Expansion (KLE) of a stochastic process = eigenfunction expansion used in time-series analysis, stochastic PDE's, & signal processing
 ↳ also proved to exist in certain L^2 spaces

⇒ Not a lot of literature on KLE's for stochastic elements whose values lie in an arbitrary (not necessarily separable) Hilbert Space (specific type of feature space)

↳ to develop theory for generalized KLE, draw on Bochner spaces & Hilbert-Schmidt spaces & then construct a natural isomorphism between them & then apply this KLE theory to probabilistic binary classification problems

↳ Data sets where the latent feature dimension F greatly exceeds the available number of samples N arise frequently in medical contexts (like genetic + patients' data to detect predispositions to diseases like Alzheimer's) & also prove difficult to classify on traditional classifiers, such as support vector machines (SVMs) & K-Nearest neighbors (KNN).

★ Introduce FINDER, a suite of methods for producing stochastic feature maps which exploits the generalized KLE function. Unlike kernel methods, FINDER provides probabilistic guarantees of class separation that are independent of data distribution.

↳ Can confirm with experiments on 8 two-class biomedical data sets that an SVM, with Gaussian Radial Basis Function (RBF) kernel if suitable, exhibits higher empirical performance on these mapped features than both raw & PCA data features. Furthermore, FINDER is competitive with several Boosting/Bagging classifiers as well.

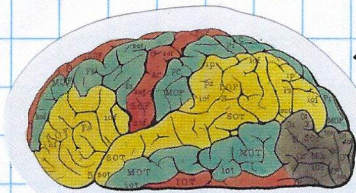
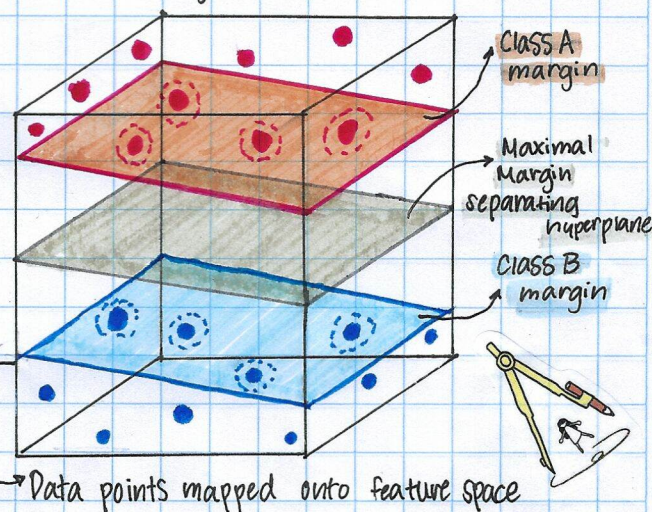


Figure 1



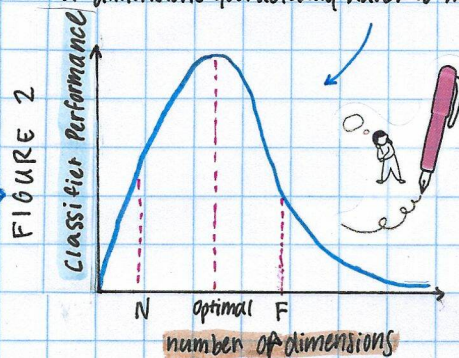
Problem + Initial Solution:

↳ Separate patient data into 2 classes, has Alzheimer's/predisposition to it OR doesn't

↳ Treat patient data as vectors, because data is transformed into a structured format where geometric properties (direction & magnitude) correspond to meaning & relationships + allows for operations like dot products, cross products, matrix multiplication, etc to unlock parallel processing on hardware = really important for machine learning

Problem + Initial Solution: Continued

→ Do it in an F -dimensional space (the ideal amount of dimensions to make data clear) but N (the amount of dimensions you actually have) is much less than F , sample size is way too small to look at the population



→ Solution: elevate SVM by finding suitable subspace & get some kind of geometric separation between data

→ SVM scales very well to a bigger size, neural networks don't in machine learning, hence why using SVM, because of big data set

→ Limit reliance on assumptions of data for more accurate predictions

→ Feature Selection, localizes biomarkers of disease to make classifier easier

→ Principle component analysis to reduce data dimension while minimizing mean squared error $\Rightarrow$ method is similar but more unique & better for class separation

distribution-agnostic methods

→ SVM basic learning: W & B are coordinates of a regression line are found through an optimization problem for the SVM learning, constraint optimization problem which is solved by using Lagrange multiplier and then it becomes a quadrating program problem which is efficient to solve.

→ When data is not linearly separable, a kernel SVM is typically used to separate the data.

→ Potentially infinite dimensions, making it hard to use here

→ For any data set, there can exist a feature mapping function ϕ to make data linearly separable. Using the kernel trick allows us to bypass directly calculating ϕ . So could have computed the kernel function first & then prove that ϕ exists that does so, Gaussian radial basis function show that there does exist said ϕ , but this isn't perfect. If your data is such that is VERY hard to linearly separate or still has noise via multiple dimensions, it becomes very messy

The Karhunen-Loève Expansion: Tool of Choice

→ Stochastic process defined on unit interval, captures all the stochastic data isolated into a sequence that is interpretable, PC is infinite dimensions (F to infinity), easy to estimate eigenfunctions from sampled data $\Rightarrow$ ASSUMPTIONS

→ Missing piece: cannot accommodate all data types & stochastic components are hard to estimate from sampled data $\rightarrow$ not independent data, & this becomes an issue when you don't have much data to work with like now

→ PhD contribution / new research: KLE's in arbitrary Hilbert spaces

→ A lot of literature on expansion, but they focus on specific expansions & you want to make sure they exist across the larger generality but there are still technical limitations

→ Since $N \ll F$, hard to verify independence / distribution assumptions on $\{v_i\}_{i=1}^N$



FORMAL DEFINITION

→ Given a stochastic process $\{v_t\}_{t \in [0,1]}$ on a complete probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with $\int_0^1 \int_{\Omega} v_t(\omega)^2 d\mathbb{P}(\omega) dt < \infty$, v admits the eigenfunction expansion $v_t(\omega) = \mathbb{E}v_t + \sum_{r=1}^{\infty} \lambda_r^{1/2} Y_r(\omega) \phi_r(t)$

The Karhunen Loève Expansion: Continued

- ↳ $K(s, t) := \mathbb{E}[(V_s - \mathbb{E}V_s)(V_t - \mathbb{E}V_t)] \rightarrow$ covariance kernel
- ↳ (λ_r, ϕ_r) solve eigenvalue problem $\lambda \phi(t) = \int_0^1 K(s, t) \phi(s) d\mu(s)$
- ↳ $\{\phi_r\}_{r=1}^\infty$ form orthonormal basis (ONB) of $L^2([0, 1]) \rightarrow$ eigenfunctions
- ↳ $Y_r(\omega)$ are zero-mean, unit variance, mutually uncorrelated RVs $\rightarrow$ stochastic components.

↳ PROOF for generalized KLE

↳ Given a subspace/sample space $L^2(\Omega, \mathcal{H})$ that satisfies first assumptions

$\rightarrow \{v: \Omega \rightarrow \mathcal{H} \mid v \text{ satisfies assumptions}\}$

↳ Can define Hilbert-Schmidt space w Hilbert-Schmidt operator H

$\rightarrow \text{HS}(\mathcal{H}, L^2(\Omega)) = \{H: \mathcal{H} \rightarrow L^2(\Omega) \mid H \text{ is Hilbert-Schmidt}\}$

• an operator H is Hilbert-Schmidt if $\sum_{e \in \mathcal{B}} \|He\|^2 < \infty$ for arbitrary ONB $\mathcal{B}$

↳ $\langle u, v \rangle_{L^2(\Omega, \mathcal{H})} = \int_\Omega \langle u(\omega), v(\omega) \rangle_{\mathcal{H}} d\mathbb{P}$, $\langle H, K \rangle_{\text{HS}(\mathcal{H}, L^2(\Omega))} = \sum_{e \in \mathcal{B}} \langle He, Ke \rangle_{L^2(\Omega)}$

↳ For $v \in L^2(\Omega, \mathcal{H})$, define $H_v: \mathcal{H} \rightarrow L^2(\Omega)$ by $H_v x = \langle v, x \rangle_{\mathcal{H}}$ (isometric isomorphism)

↳ can show H_v is compact (Hilbert-Schmidt even), $\Rightarrow H_v - \mathbb{E}H_v \stackrel{\text{SVD}}{=} \sum_{r=1}^R \lambda_r^{1/2} Y_r \otimes \phi_r^a$

↳ $\lambda_r \in \mathbb{R}$, $Y_r \in L^2(\Omega)$, $\phi_r \in \mathcal{H}$, $Y_r \otimes \phi_r$ is the rank one linear operator $x \mapsto \langle x, \phi_r \rangle Y_r$

↳ KLE properties: let $v_0 = v - \mathbb{E}v$, $v \stackrel{\text{KLE}}{=} \mathbb{E}v + \sum_{r=1}^R \lambda_r^{1/2} Y_r \phi_r$

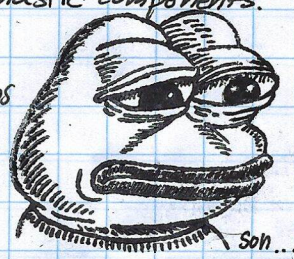
↳ Singular values: $\{\lambda_r^{1/2}\}_{r=1}^R =$ non-increasing, positive sequence in $\mathbb{R}$. $\sum_{r=1}^R \lambda_r = \mathbb{E}(\|v_0\|^2)$

↳ Stochastic components: $\{Y_r\}_{r=1}^R =$ zero mean orthonormal set in $L^2(\Omega)$ - unit variance, mutually correlated random variables (RV)

↳ Right singular vectors/eigenvectors: $\{\phi_r\}_{r=1}^R =$ orthonormal set $\mathcal{H}$

↳ Optimal Truncation Property: Let $S \subseteq \mathcal{H}$ be any M -dimensional subspace. Then

$\mathbb{E}(\| \text{Id} - P_S \|_{\text{Vol}}^2) \geq \sum_{r=M}^R \lambda_r$ with equality if $S = \text{span}\{\phi_r\}_{r=1}^M$.



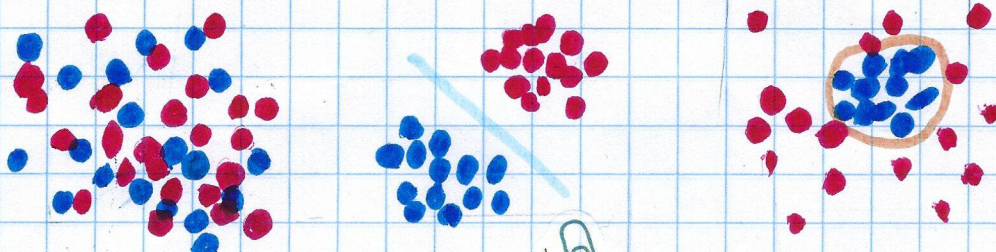
Application to Classification: Residual Component Analysis

↳ Goal: Find a projection operator $P_S: \mathcal{H} \rightarrow \mathcal{H}$ which segregates classes for SVM learning

Raw Data

Linearly Separable

Separable via RBF



↳ Recall v^A, v^B underlying $\mathcal{H}$ -valued RVs.

↳ Let $v^A \stackrel{\text{KLE}}{=} \sum_{r=1}^R (\lambda_r^A)^{1/2} Y_r^A \phi_r^A$

↳ Assume $\mathbb{E}Cv^A = 0$.

FIGURE 3

Application to Classification: Residual Component Analysis - Continued

→ **Class A Concentration Bound**: Let $H_A = \text{Span} \{ \phi_r^A \}_{r=1}^{M_A}$. Suppose $S \perp H_A$ and $\alpha \in (0, 1)$. By Markov's inequality, we have

$$P_r \left(\|P_S V^A\|_{\mathcal{H}}^2 > \alpha^{-1} \sum_{r=1}^{M_A} \lambda_r^A \right) \leq \alpha$$

→ Subspace gives us a distribution and probability that are close to the origin & geometric separation with pc $\Rightarrow$ tradeoff between dimensions of subspace that is not too different from the tradeoff between the res values $\leadsto$ Class A Concentration bound shows that the Class A samples lie close to the origin with high probability

→ Learning Pipeline

→ Data Acquisition - 8 data sets

- Cancer: 16063 gene expression levels $\rightarrow$ GCM
- Alzheimer's Disease: 2053 gene expression levels, 146 blood plasma protein levels (3 stages), 7006 CSF protein levels (3 stages)
 - new AD: Alzheimer's disease from gene expression levels
 - Proteomic CADNI: AD from protein content of blood plasma
 - $\leadsto$ Cognitive Normal (CN)
 - $\leadsto$ Late Mild Cognitive Impairment (LMCI)
 - $\leadsto$ Alzheimer's Disease (AD)
 - CSF: AD from protein content of cerebrospinal fluid (same stages as ADNI)
- N_A (resp. N_B) = number of class A (resp. class B) samples. M_A = truncation parameter



FIGURE 4

	Genetic		Proteomic (CADNI)			CSF		
	GCM	newAD	AD vs. CN	AD vs. LMCI	CN vs. LMCI	AD vs. CN	AD vs. LMCI	CN vs. LMCI
N_A	190	184	97	346	346	167	221	221
N_B	90	145	54	97	54	138	138	167
Nominal	Normal	Normal	AD	LMCI	LMCI	CN	LMCI	LMCI
Anomalous	Tumor	AD	CN	AD	CN	AD	AD	CN
Features	16063	2053		146			7006	
M_A	39	8		5			8	

→ **Reliable Classification**: classification is easiest when the first few eigen vectors of V^A & V^B are close to orthogonal, the eigenvalues of V^A decay quickly (Markov Bound is small), M_A & $\dim(S)$ are "chosen wisely"

→ **Choosing subspaces**: three ways to build an orthonormal basis (CN, B) for S

- Multilevel subspace (MLS)
- Anomalous class adapted - Small method (ACA-S)
- Anomalous class adapted - Large method (ACA-L)

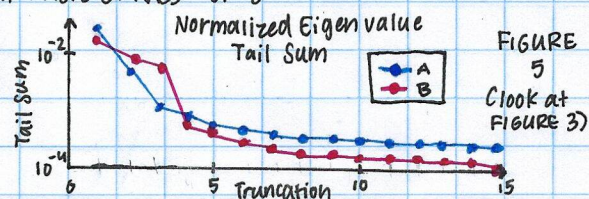


FIGURE 5
Look at FIGURE 3)

MLS & ACA Methods:

→ A multilevel basis

- A full SVD of the data matrix to get an eigenbasis for V^A takes $O(F^3)$
- Reduce to $O(F \log F)$ with a kd-tree division of $\{1, 2, \dots, F\}$ ($K=1$, $d = \lfloor \log_2 \frac{F}{M_A} \rfloor$)
- Idea: Sequence of mutually orthogonal subspaces $\{\mathbb{W}_\ell\}_{\ell=-1}^d$ such that
 - $\bigoplus_{\ell=-1}^d \mathbb{W}_\ell = \mathbb{R}^F$
 - $\mathbb{W}_{-1} = \mathcal{H}_A = \text{Span} \{ \phi_r^A \}_{r=1}^{M_A}$
 - $\bigoplus_{\ell=0}^d \mathbb{W}_\ell = \mathcal{H}_A^\perp$
 - $\mathbb{W}_\ell = \text{Span} \{ \text{sparse vectors} \}$

- Try $S_n = \bigoplus_{\ell=0}^n \mathbb{W}_\ell$ for each $0 \leq n \leq d$
 $\dim(S_n) = (2^{d-n} - 1) M_A$

→ The Anomalous Class Adapted Method

- Previous method doesn't use eigenstructure of V^B (hopes that the classes work together efficiently)
- choose $S \perp \mathcal{H}_A$ which makes $\mathbb{E} \| \Pi_S (V^B - \mathbb{E}(V^B)) \|^2$ small or large compared to $\mathbb{E} \| \Pi_S (V^A) \|^2$ → entire orthogonal complements & present onto that subspace, handled entirely by a linear SVM with RBF kernel (Figure 3)

→ Anomalous Class Adapted Basis

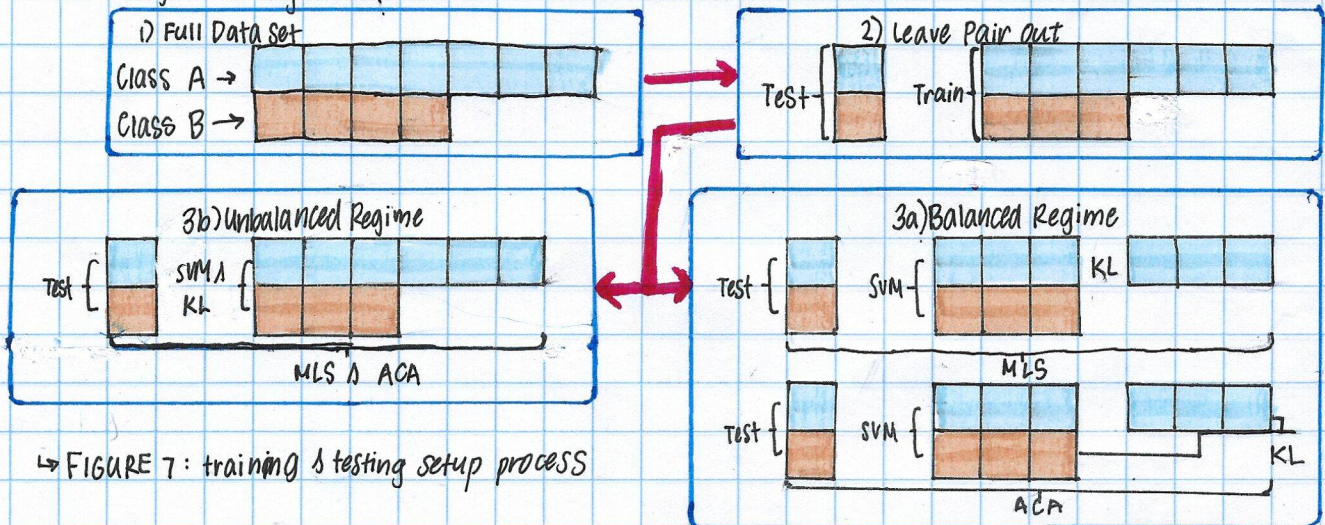
- Let $P_{\mathcal{H}_A^\perp} (V^B - \mathbb{E}(V^B)) \stackrel{KLE}{=} \sum_{r=1}^R \sigma_r^{1/2} Z_r t_r$

- $S = \text{Span} \{ \text{smallest } t_r \} \Rightarrow \mathbb{E}(\cdot)$ is minimized (ACA-S)
- $S = \text{Span} \{ \text{largest } t_r \} \Rightarrow \mathbb{E}(\cdot)$ is maximized (ACA-L)

- Still need a basis for $\mathcal{H}_A^\perp$ & a quick way to estimate all of t_r , can use MLS basis!

- look at Figure 3, with linearly separable → ACA-S & separable via RBF → ACA-L

→ Training & Testing Set-up



→ FIGURE 7: training & testing setup process

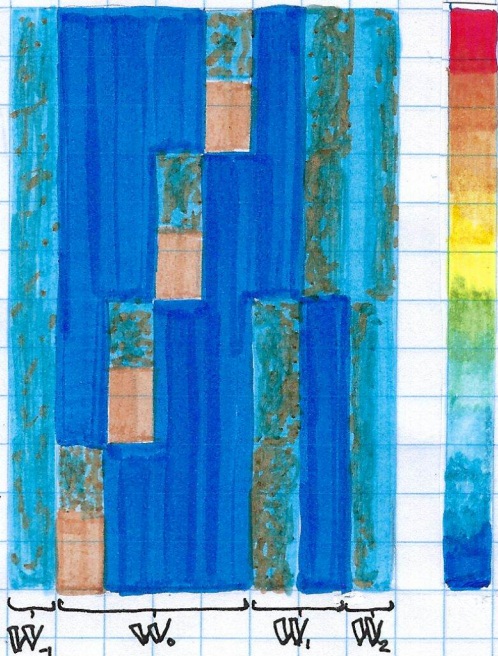
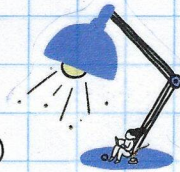


FIGURE 6: Multilevel Basis with $F=1024$, $M_A=128$, $d=2$ (columns = vectors)

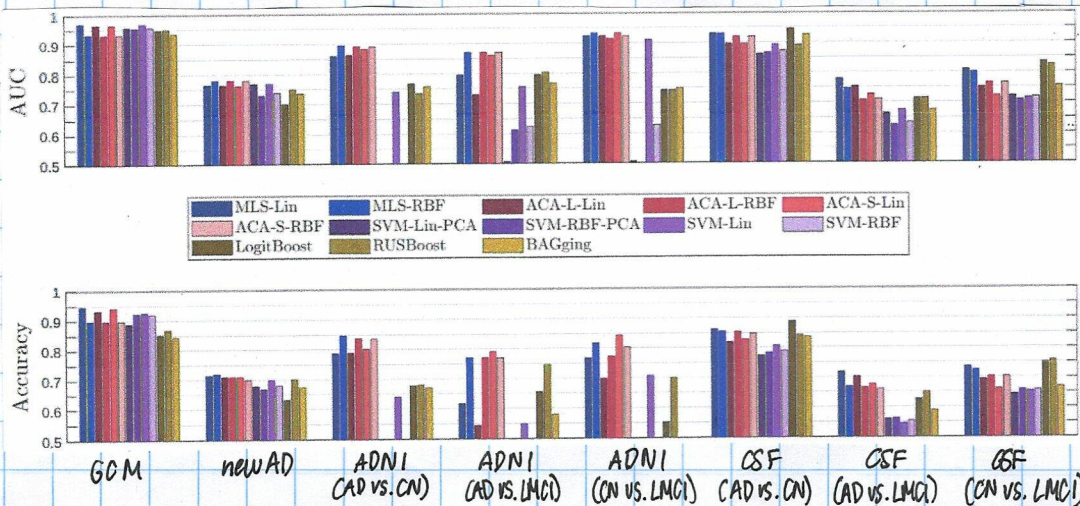


MLS & ACA Methods: Continued

↳ Training & Testing Setup (See Figure 7)

- SVM works well if data is balanced
- ① Hold out one sample from each class for testing, designate γ_A^{test} , γ_B^{test} .
 - ② Remaining data = γ_A^{train} , γ_B^{train} .
 - ③ If Balanced SVM, remove unbalanced portion of γ_A^{train} and use to estimate eigenvectors of V^A . Use remaining to train SVM along with γ_B^{train} .
 - ④ If unbalanced SVM, use all of γ_A^{train} , γ_B^{train} to estimate eigenvectors & train SVM.
 - ⑤ Construct S (residual subspace), project $v_i \leftarrow P_S v_i$ (draw data onto subspace)
 - ⑥ Train SVM.
 - ⑦ Compare against raw SVM (w/ & w/o RBF, on PCA and raw features), LogitBoost, RUSBoost, BAGging.
 - ⑧ Repeat ①-⑦ holding out the i th sample from A & j th sample from B (for half the sample size of each class).

↳ Residual Subspace Methods vs. Benchmarks



for each residual subspace
for each kernel choice
Linear vs. RBF

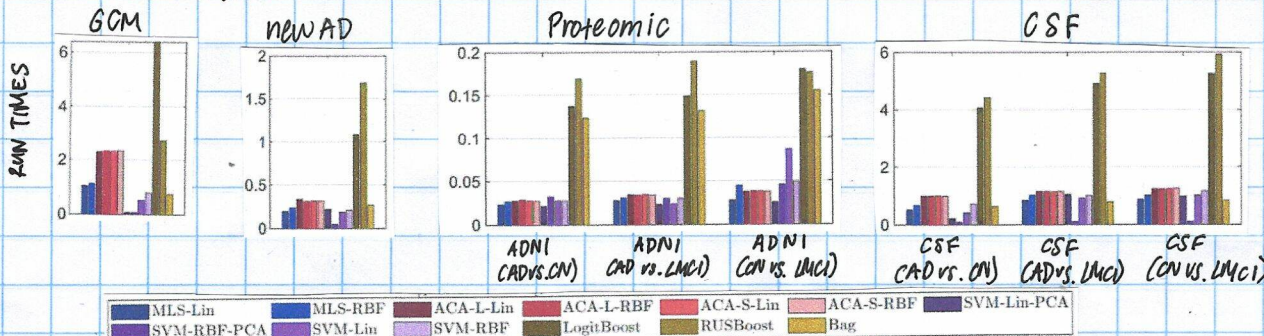
Maximum AUC is
accuracy across all
pre-balancing conditions
& all S dimensions,

→ FIGURE 8

↳ Blue & Red (MLS & ACA methods developed) tend to out do everything else

↳ ACA & RBF tends to do worse than the baseline linear, but the actual subspaces we constructed to do better than the baseline linear (ACA & MLS vs. SVM)

↳ Time Analysis



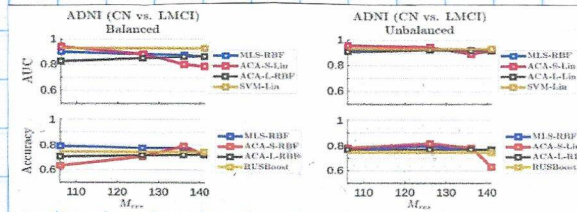
even though
some of the
Boost programs
outperform MLS
& ACA, they take
significantly
much more time.

→ FIGURE 9

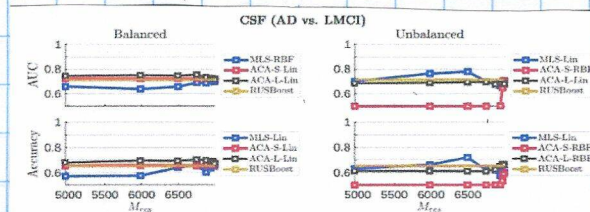
MLS & ACA Methods : Continued

↳ Residual Subspace Dimension Analysis → how it affects AUC & accuracy on data sets

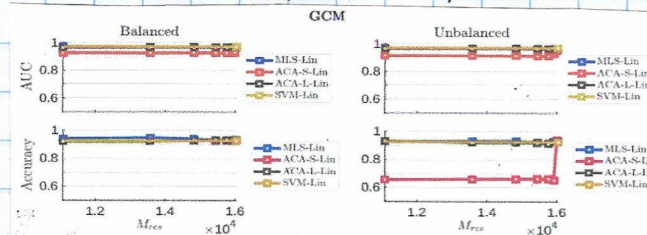
↪ FIGURE 10 : ADNI CCN vs. LMCI. Best AUC : 0.932 (Unbalanced MLS-RBF, $M_{res} = 126$).
Best accuracy : 0.844 (Unbalanced ACA-S-Lin, $M_{res} = 126$).



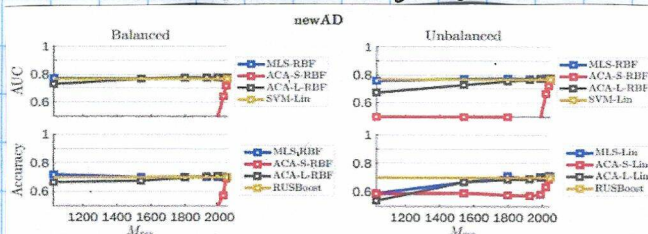
↪ FIGURE 11 : CSF CAD vs. LMCI. Best AUC : 0.779 (Unbalanced MLS-Lin, $M_{res} = 6494$).
Best accuracy : 0.719 (Balanced Unbalanced MLS-Lin, $M_{res} = 6494$)



↪ FIGURE 12 : 6 CM. Best AUC : 0.971 (Balanced MLS-Lin, $M_{res} = 11071$). Best Accuracy : 0.946
(Balanced MLS-Lin, $M_{res} = 13567$)



↪ FIGURE 13 : newAD. Best AUC : 0.780 (Balanced ACA-L-RBF, $M_{res} = 1989$). Best Accuracy :
0.719 (Balanced MLS-RBF, $M_{res} = 1029$).



↳ In figures 12 & 13, in unbalanced data, the MLS RBF & MLS-Lin tends to do better than the ACA methods

↳ In contrast, Figure 10-11 in the balanced data, ACA does better but ~~never~~ underperforming compared to Subspace

MLS & ACA Methods: Continued

↳ Summary of data

- ↳ Residual subspace methods improve SVM over raw & PCA features.
- ↳ Residual subspace methods improve baseline considerably for blood proteomic data.
- ↳ Residual subspace methods underperform on CSF data marginally but save time compared to boosting methods.
- ↳ MLS improves 618 data sets. ACA improves over MLS on 2 data sets
- ↳ FIGURE 14: Wilcoxon signed rank test p-values for each residual subspace method.



	AUC		Accuracy	
	Best SVM	Best Boost/Bag	Best SVM	Best Boost/Bag
MLS	< 0.01	0.03	< 0.01	0.02
ACA-S	< 0.01	0.07	< 0.01	0.04
ACA-L	< 0.01	0.05	< 0.01	0.04

↳ Disadvantages

- ↳ Classification gain diminishes for well-separated data sets (CCM).
- ↳ Concentration bound uses Markov's inequality (not sharp, but can't be tightened).
- ↳ Finitude of $\sum_{r=1}^R \lambda_r Y_r^2$ and non-independence of Y_r , preclude CLT-like analyses.
- ↳ Choice of M_A, M_{res} , balancing isn't obvious $\leadsto$ leads into...

Algorithmic Hyperparameter Selection:

- ↳ Question: how to choose M_A & M_{res} $\leadsto M_A$ too small keeps too much of class A noise & M_A too large eliminates too much class B noise.

- ↳ Idea: create short-list of candidate M_A & M_{res} values and choose the pair that maximize internal validation accuracy.

↳ choose M_A to explain 70%, 75%, ... 95% of class A variance (like PCA).

- ↳ project data $\{V_i\}_{i=1}^N$ onto $\mathbb{R}_A^1$ & create M_{res} short-list

↳ choose M_{res} to explain 5%, 15%, ... 95% of class A variance (in standard basis)

$$\leadsto M_{res} := \min_M \left\{ \frac{E(\sum_{i=1}^M \langle V_i^A, e_i \rangle^2)}{E(\|V^A\|^2)} > 0.05 \right\}$$

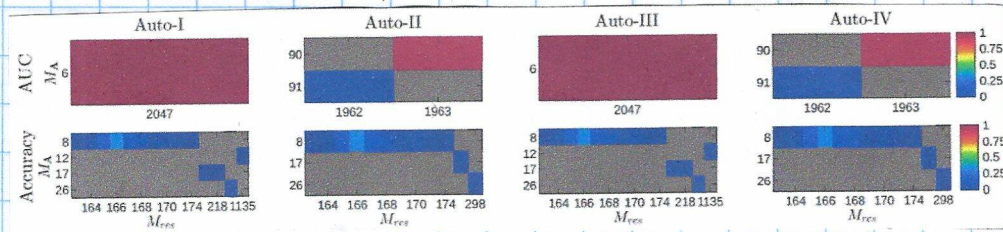
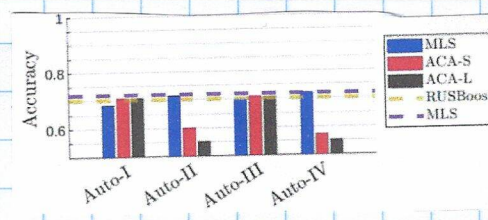
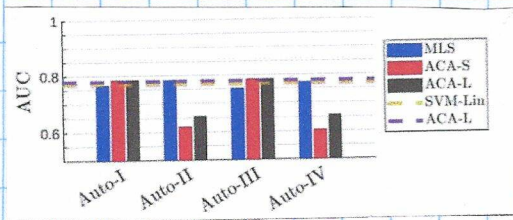
↳ Breaking Ties

- ↳ Auto-I chooses the largest M_A & largest M_{res} from short-list which obtains optimal internal validation accuracy. Auto II chooses the smallest M_A & largest M_{res} from short-list which obtains optimal internal validation accuracy. Auto III chooses the largest M_A & smallest M_{res} from short-list which obtains optimal internal validation accuracy. Auto IV chooses the smallest M_A & smallest M_{res} from short-list which obtains optimal internal validation accuracy.

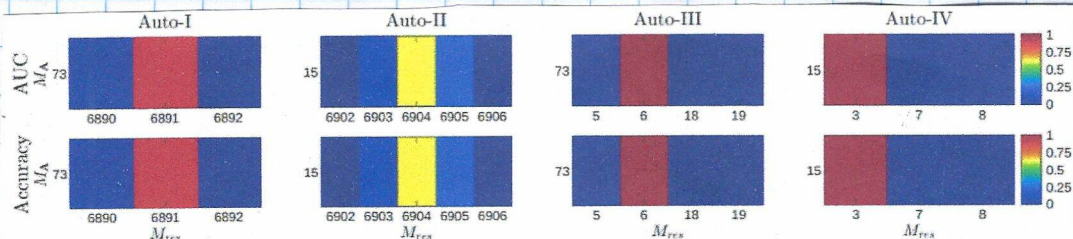
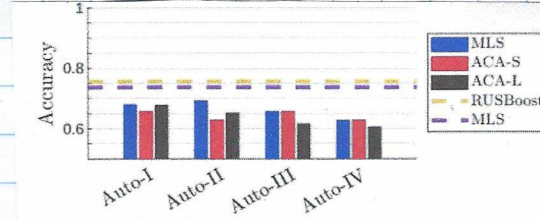
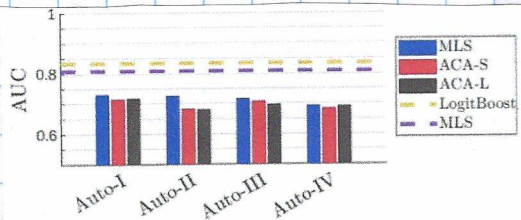
↳ Split into training & testing $\rightarrow$ & then into SVM & KL $\rightarrow$ optimize hyperparameters $\rightarrow$ train & test $\rightarrow$ iterate for LPOCV

Algorithmic Hyperparameter Selection : Continued

↳ The successes : the newAD set



↳ The failures : CSF (CN vs. LMC1)



*SVM training data was prebalanced.
† SVM was kernelized.



↳ Manual vs. Algorithmic : Max AUC & accuracy for each data set across all hyperparameter selection algorithms, balancing conditions, kernels, & residual

	Genetic CECM	Genetic CMCAD	ADNI (AD vs. CN)	ADNI (AD vs. LMC1)	ADNI (CN vs. LMC1)	CSF (AD vs. CN)	CSF (AD vs. LMC1)	CSF (CN vs. LMC1)
Auto	0.966	0.785	0.879	0.869	0.883	0.920	0.727	0.729
Algorithm/Subspace	Auto II / MLS*	Auto I / ACA-S*	Auto II / MLS*	Auto I / ACA-S*	Auto I / MLS*	Auto II / MLS*	Auto I / ACA-S*	Auto I / MLS
Benchmark	0.968	0.769	0.767	0.803	0.910	0.946	0.714	0.834
Classifier	Linear SVM	Linear SVM	Logit Boost	RUS Boost	Linear SVM	Logit Boost	RUS Boost	Logit Boost
Manual Subspace	0.971	0.780	0.896	0.869	0.932	0.931	0.779	0.808
Subspace	MLS*	ACA-L†	MLS†	MLS*	MLS†	MLS	MLS	MLS
Auto	0.933	0.719	0.814	0.771	0.763	0.859	0.685	0.691
Algorithm/Subspace	Auto II / MLS*	Auto II / MLS*	Auto II / MLS†	Auto I / ACA-S*	Auto I / MLS*	Auto II / MLS*	Auto II / MLS*	Auto II / MLS
Benchmark	0.925	0.701	0.681	0.746	0.707	0.888	0.680	0.756
Classifier	Linear SVM	RUS Boost	RUS Boost	RUS Boost	Linear SVM	Logit Boost	RUS Boost	RUS Boost
Manual Subspace	0.946	0.719	0.847	0.791	0.844	0.861	0.719	0.736
Subspace	MLS*	MLS*	MLS†	ACA-S*	ACA-S	MLS*	MLS	MLS*

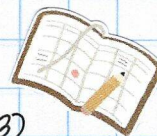
subspace methods along with benchmark & residual AUC method manual hyperparameter selection performance.

Accuracy

Conclusion:

↳ Summary:

- ↳ Residual subspace methods subvert traditional PCA by projecting onto (a subspace of) the residual eigenspace rather than the ~~specific~~ principle eigenspace.
- ↳ KLE provides provable, distribution-independent, geometric separation qualities
 - geometric separation qualities can be exploited by SVM & kernels (Figure 3)
- ↳ Competitive with state of the art classifiers vis-a-vis classification capability or run time
 - Effective for high-feature, low sample data



↳ Hyperparameter selection (Trajan's suggestions)

- ↳ Rule of thumb: M_A smaller & M_{res} larger (Auto-I & II)
- ↳ use short-lists in combination with grid-based search

↳ Future Directions

- ↳ Feature Selection: conserve features where A has low variance and where B has high variance
- ↳ Assume independence on stochastic components $\{y_r\}_{r=1}^R$. Worst case scenario if assumptions are violated?