Fast classification of Big Data:
proximal methods for sparse discriminant analysis

Summer Atkins*
Brendan Ames*
Line Clemmensen**
Gudmundur Einarsson**

*Department of Mathematics, The University of Alabama
**Department of Applied Mathematics and Computer Science Technical University of Denmark

SIAM Southeastern Atlantic Section Conference
March 13, 2016

Outline

Proposes three new heuristics for Sparse Discriminant Analysis (SDA) by using the following techniques: proximal gradient method, accelerated proximal gradient method and alternating direction method of multipliers.

Outline:

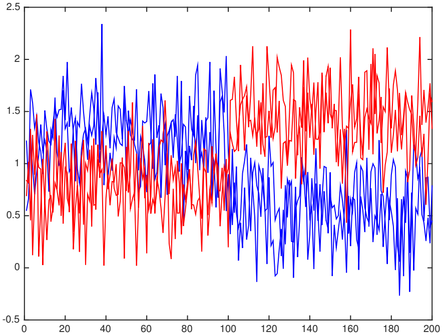
- Review of linear discriminant analysis.
- Our proximal heuristics for SDA.
- Numerical results.

The Classification Problem

Given n observations $\boldsymbol{x}_i \in \mathbf{R}^p$. Stored as matrix $\boldsymbol{X} \in \mathbf{R}^{n \times p}$.

Each $\boldsymbol{x}_i$ belongs to exactly one of K classes $C_1, C_2, \ldots, C_K$.

Problem: Design a decision rule to assign new observations to exactly one of the K classes.



Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a technique for supervised classification where data is linearly projected to a subspace where class discrimination is maximized using a set of discriminant vectors $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_q \in \mathbf{R}^p$, $q < K$.

There are three equivalent approaches that result in the LDA classifier:

1. The multivariate Gaussian model
2. Fisher's discriminant problem
3. The optimal scoring problem

Optimal Scoring

Idea: Turns categorical variables (class labels) into quantitative ones (class score).

Suppose we know $\{(\boldsymbol{\theta}_\ell, \boldsymbol{\beta}_\ell)\}_{i=1}^{k-1}$. Find $(\boldsymbol{\theta}_k, \boldsymbol{\beta}_k)$ by solving

$$\begin{aligned} \min_{\boldsymbol{\beta}_k, \boldsymbol{\theta}_k} \quad & \|\boldsymbol{Y}\boldsymbol{\theta}_k - \boldsymbol{X}\beta_k\|^2 \\ \text{s.t.} \quad & \frac{1}{n}\boldsymbol{\theta}_k^T \boldsymbol{Y}^T \boldsymbol{Y}\boldsymbol{\theta}_k = 1 \\ & \boldsymbol{\theta}_k^T \boldsymbol{Y}^T \boldsymbol{Y}\boldsymbol{\theta}_\ell = 0 \quad \forall \ell < k \end{aligned}$$

- $\boldsymbol{Y}$ is an $n \times K$ indicator matrix for class membership,
- $\boldsymbol{X}$ is an $n \times p$ data matrix,
- β_k is the k^{th} discriminant vector in $\mathbf{R}^p$, and
- $\boldsymbol{\theta}_k$ is the k^{th} scoring vector in $\mathbf{R}^k$.

Challenges in LDA

LDA is known to fail in the following cases:

1. When data is in a high-dimensional setting, meaning when the number of predictor variables p is larger than the number of observations n.
2. When linear boundaries are unable to separate the classes.

Sparse Discriminant Analysis

Clemmensen et al. 2011: Take the Optimal Scoring formulation of LDA and applies an elastic net penalty to the coefficient vectors.

$$\begin{aligned}
&\min_{\boldsymbol{\beta}_k, \boldsymbol{\theta}_k} \|\boldsymbol{Y}\boldsymbol{\theta}_k - \boldsymbol{X}\boldsymbol{\beta}_k\|^2 + \gamma \boldsymbol{\beta}_k^T \boldsymbol{\Omega} \boldsymbol{\beta}_k + \boldsymbol{\lambda}\|\boldsymbol{\beta}_k\|_1 \\
&\text{s.t.} \quad \frac{1}{n}\boldsymbol{\theta}_k^T \boldsymbol{Y}^T \boldsymbol{Y} \boldsymbol{\theta}_k = 1 \\
&\qquad\quad \boldsymbol{\theta}_k^T \boldsymbol{Y}^T \boldsymbol{Y} \boldsymbol{\theta}_\ell = 0 \quad \forall \ell < k
\end{aligned}$$

- γ, $\boldsymbol{\lambda}$ are non-negative tuning parameters and $\boldsymbol{\Omega}$ is a positive definite matrix.
- The ℓ_1 term encourages sparsity and the $\boldsymbol{\Omega}$ term encourages smoothness.

SDA is **not convex** in $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ jointly.

Block Coordinate Descent for SDA

For β_k fixed, SDA is a least squares problem:

$$\min_{\theta} \left\{ \|\boldsymbol{Y}\theta - \boldsymbol{X}\beta_k\|^2 : \theta^T \boldsymbol{D}\theta = 1, \theta^T \boldsymbol{D}\theta_\ell = 0 \quad \forall \ell < k \right\}$$

where $\boldsymbol{D} := \frac{1}{n}\boldsymbol{Y}^T\boldsymbol{Y}$.

Closed form solution is given by

$$\theta_k = r(\boldsymbol{I} - \boldsymbol{Q}_k\boldsymbol{Q}_k^T\boldsymbol{D})\boldsymbol{D}^{-1}\boldsymbol{Y}^T\boldsymbol{X}\beta_k$$

where $\boldsymbol{Q}_k = [\theta_1|\theta_2|\ldots|\theta_{k-1}|\boldsymbol{e}]$, $\boldsymbol{e}$ is the all-ones vector, $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{r}$ is a proportionality constant.

Block Coordinate Descent for SDA (2)

For θ_k fixed, SDA becomes

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{Y}\boldsymbol{\theta}_k - \boldsymbol{X}\beta\|^2 + \gamma\beta^T\boldsymbol{\Omega}\beta + \lambda\|\beta\|_1.$$

This is a convex problem in β! We propose several proximal methods:

1. **Proximal Gradient Method (PGM)**
2. **Accelerated Proximal Gradient Method (APGM)**
3. **Alternating Direction Method of Multiplication (ADMM)**

Proximal Gradient Method

Decompose objective as $f(\beta) + g(\beta)$ where

- $f(\beta) = \frac{1}{2}\beta^T \boldsymbol{A}\beta - \beta^T \boldsymbol{d}$ with $\boldsymbol{A} = 2(\boldsymbol{X}^T\boldsymbol{X} + \gamma\boldsymbol{\Omega})$, $\boldsymbol{d} = 2\theta_k^T \boldsymbol{Y}^T\boldsymbol{X}$, and
- $g(\beta)) = \lambda\|\beta\|_1$.

Taking a gradient step with respect to f, giving us

$$\boldsymbol{p}^t = \beta^t - \alpha\nabla \boldsymbol{f}(\beta^t) = \beta^t - \alpha(\boldsymbol{A}\beta^t + \boldsymbol{d})$$

Then, we take proximal step with respect to g, giving us

$$\beta^{t+1} = \mathrm{sign}(\boldsymbol{p}^t)\max\{|\boldsymbol{p}^t| - \lambda\alpha\boldsymbol{e}, 0\} = \boldsymbol{S}_{\lambda\alpha}(\boldsymbol{p}^t)$$

where $\alpha > \mathbf{0}$ is a fixed step size.

Accelerated Proximal Gradient Method

Accelerated Proximal Gradient Method (APGM): A version of PGM that adds a momentum term in order to accelerate convergence.

Results are generated by the following iterates:

1. $\boldsymbol{y}^{t+1} = \beta^t - \omega_t(\beta^t - \beta^{t-1})$
2. $\boldsymbol{p}^t = y^{t+1} - \alpha \nabla \boldsymbol{f}(\beta^t) = \beta^t - \alpha(\boldsymbol{A}\beta^t + \boldsymbol{d})$
3. $\beta^{t+1} = \boldsymbol{S}_{\lambda\alpha}(\boldsymbol{p}^t)$

where $\omega_t \in [0, 1)$ is an extrapolation parameter, a standard choice for ω_t is $\frac{t}{t+3}$.

ADMM

Alternating Direction Method of Multipliers (ADMM):
Algorithm that blends dual decomposition and augmented Lagrangian for solving constrained optimization problems.

By splitting β as $\beta = \boldsymbol{x} = \boldsymbol{y}$, we can form the **Augmented Lagrangian**:

$$\boldsymbol{L}_\delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{d} + \lambda\|\boldsymbol{y}\|_1 + \boldsymbol{z}^T(\boldsymbol{x} - \boldsymbol{y}) + \frac{\delta}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2$$

where δ is a nonnegative penalty parameter.

Using **ADMM** will generate a sequence of iterates $\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\}$ by

1. $\boldsymbol{x}^{t+1} = (\delta\boldsymbol{I} + \boldsymbol{A})^{-1}(\boldsymbol{d} + \delta\boldsymbol{y}^t - \boldsymbol{z}^t)$
2. $\boldsymbol{y}^{t+1} = \boldsymbol{S}_\lambda(\boldsymbol{x}^{t+1} + \frac{\boldsymbol{z}^t}{\delta})$
3. $\boldsymbol{z}^{t+1} = \boldsymbol{z}^t + \delta(\boldsymbol{x}^{t+1} - \boldsymbol{y}^{t+1})$.

Numerical Experiments

We performed a series of numerical experiments using Matlab2014a through UA's cluster RC2 to compare classification performance of the following heuristics:

1. Proximal Gradient Method for SDA (SDAP)
2. Accelerated Proximal Gradient Method for SDA (SDAAP)
3. Alternating Direction Method of Multipliers for SDA (SDAD)
4. Sparse Zero Variance Discriminant Analysis (SZVD)
5. Sparse Discriminant Analysis (SDA).

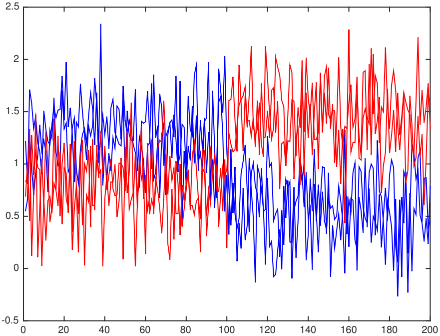
We used cross-validation to train parameters.

Time Series Data

- Penicillium (Pen) data set of multi-spectral imaging of 3 Penicillium that are almost visually indistinguishable ($p = 3542$, $k = 3$, $n = 36$, training size $= 24$, testing size$= 12$) (Clemmensen, 2007).
- Electrocardiogram measurements (ECG) data set of 884 heartbeat signals that were either classified as healthy or unhealthy ($p = 136$, $k = 2$, $n = 884$, training size$= 23$, testing size$= 861$) .
- Coffee data set consists of 56 food spectrogram observations of either Arabica or Robusta variants of instant coffee ($p = 286$, $k = 2$, $n = 56$, training\testing size $= 28$).
- Olive Oil data set of 60 food spectrogram observations of extra virgin olive oil that originated in 1 of 4 countries ($p = 570$, $k = 4$, $n = 60$, training\ testing size $= 30$) (E. Keogh, X.Ci, L.Wei, and C.A. Ratanamahatana, 2006) .

Synthetic Data Experiments

- Sample observations from multivariate Normal distributions.
- Different classes had different means, but same covariance matrix.
- We also varied each set of data based upon its constant covariance ($\boldsymbol{r} \in \{0, 0.1, 0.5, 0.9\}$).



Time Series Results

Data		SDAP	SDAAP	SDAD	SZVD	SDA
Pen	Err	0	0	0	0	0
p = 3542	Feat	62	89	26	1587	111
k = 3	Time	2517.2	80.86	326.25	1920.32	1494.3
ECG	Err	38	44	58	19	71
p = 136	Feat	17	10	10	30	16
k = 2	Time	35.36	7.01	13.38	1.87	2.29
Coffee	Err	0	0	0	0	0
p = 286	Feat	12	12	8	98	4
k = 2	Time	217.24	15.32	25.53	13.01	7.24
Olive Oil	Err	2	3	2	1	1
p = 570	Feat	18	30	44	317	60
k = 4	Time	1760.32	388.58	210.46	1066.71	261.01

Synthetic Data Results for k=2

Sim.		SDAP	SDAAP	SDAD	SZVD	SDA
p = 500	Err	12.9(9.48)	19.3(12.96)	63.05(23.47)	17.7(41.22)	96.15(16.99)
r = 0	Err%	2.58(1.90)	3.86(2.59)	12.61(4.69)	3.54(8.24)	19.23(3.40)
k = 2	Feat	62.15(15.60)	52.75(17.12)	21.3(8.29)	435.2(82.72)	11(0)
feat = 0.3	Feat%	12.43(3.12)	10.55(3.42)	4.26(1.66)	87.04(16.55)	2.2(0)
Trials= 20	Time	31.07(1.23)	11.467(1.18)	82.57(4.66)	235.14(2.35)	33.77(1.75)
p = 500	Err	7.85(7.82)	13.85(9.82)	54.2(34.29)	26.6(64.89)	88.3(9.43)
r = 0.1	Err%	1.57(1.56)	2.77(1.96)	1.08(6.86)	5.32(12.98)	17.66(1.89)
k = 2	Feat	67.6(17.19)	55.1(12.41)	22.4(10.32)	424.45(79.13)	11(0)
feat = 0.3	Feat%	13.52(3.44)	11.02(2.48)	4.48(2.06)	8.49(15.83)	2.2(0)
Trials= 20	Time	46.23(4.1)	13.35(0.9)	86.12(3.83)	232.05(2.35)	32.84(1.27)
p = 500	Err	5.55(7.02)	8(8.93)	27.3(18.33)	14.8(55.81)	60.35(18.95)
r = 0.5	Err%	1.11(1.40)	1.6(1.79)	5.46(3.67)	2.96(11.16)	12.07(3.79)
k = 2	Feat	54.3(23.28)	49.25(18.24)	22.1(8.34)	397.45(113.87)	11.05(0.22)
feat = 0.3	Feat%	10.86(4.66)	9.85(3.65)	4.42(1.67)	79.49(2.23)	2.21(0.04)
Trials= 20	Time	94.81(11.29)	16.31(1.48)	82.18(3.4)	228.06(11.128)	32.96(1.64)
p = 500	Err	1.85(2.76)	2.7(3.8)	1.45(2.89)	37.5(91.59)	3.6(5.96)
r = 0.9	Err%	0.37(0.55)	0.54(0.76)	0.29(0.58)	7.5(18.32)	0.72(1.19)
k = 2	Feat	17.1(9.67)	16.4(9.03)	17.3(6.54)	358.65(122.02)	11(0)
feat = 0.3	Feat%	3.42(1.93)	3.28(1.81)	3.46(1.31)	71.73(24.41)	2.2(0)
Trials= 20	Time	210.93(41.26)	18.12(2.22)	74.75(3.534)	179.66(73.42)	33.52(2.22)

Synthetic Data Results for k=4

Sim.		SDAP	SDAAP	SDAD	SZVD	SDA
p = 500	Err	41.25(18.85)	43.3(12.17)	172.8(75.35)	295.6(290.3)	152.9(24.14)
r = 0	Err%	4.13(1.88)	4.33(1.22)	17.28(7.53)	29.56(29.03)	15.29(2.41)
k = 4	Feat	277.05(53.5)	279.55(35.36)	112.25(34.79)	1137.3(237.56)	99.15(0.37)
feat = 0.3	Feat%	18.47(3.57)	18.64(2.36)	7.48(2.32)	75.82(15.84)	6.61(0.02)
Trials = 20	Time	98.36(2.12)	129.42(13.36)	331.24(2.77)	1030.9(74.0)	529.42(19.15)
p = 500	Err	32.15(14.14)	40.65(13.13)	132.45(49.74)	128.7(133.79)	174.45(21.29)
r = 0.1	Err%	3.22(1.41)	4.07(1.31)	13.25(4.97)	12.87(13.38)	17.44(2.13)
k = 4	Feat	286.45(44.42)	273.5(37.31)	132.5(28.04)	1080(179.18)	99.1(0.31)
feat = 0.3	Feat%	19.10(2.96)	18.23(2.49)	8.83(1.87)	72.0(11.95)	6.61(0.02)
Trials= 20	Time	194.35(24.13)	140.09(16.51)	368.26(38.08)	1028.4(81.3)	514.7(31.52)
p = 500	Err	4.8(6.37)	9.7(6.86)	74.3(48.97)	11.9(32.22)	81.9(17.87)
r = 0.5	Err%	0.48(0.64)	0.97(0.69)	7.43(4.90)	1.19(3.22)	8.19(1.79)
k =4	Feat	281.8(56.39)	247.3(42.72)	116.8(32.92)	718.6(271.16)	99.1(0.31)
feat = 0.3	Feat%	18.79(3.76)	16.49(2.85)	7.79(2.19)	47.91(18.08)	6.61(0.02)
Trials = 20	Time	392.74(51.77)	161.79(19.06)	322.23(26.51)	461.81(306.09)	499.68(48.56)
p = 500	Err	2.25(3.02)	1.55(3.28)	8.5(20.69)	61.65(178.19)	0.15(0.67)
r = 0.9	Err%	0.225(0.30)	0.16(0.32)	0.85(0.02)	6.17(17.82)	0.02(0.07)
k= 4	Feat	89.95(34.22)	107.3(36.67)	86.75(30.13)	736.9(398.63)	99(0)
feat = 0.3	Feat%	6.00(2.28)	7.15(2.44)	5.78(2.01)	49.13(26.58)	6.6(0)
Trials= 20	Time	671.91(56.72)	147.11(15.06)	241.6(4.49)	287.7(200.79)	454.17(13.45)

Conclusion

We have developed three new heuristics which apply the following techniques for solving the SDA problem: **ADMM**, **PGM**, and **APGM**.

Work in progress

- Developing R and Matlab packages of heuristics
- Deriving bounds on classification error
- Analyzing convergence

Thank You!