# Advanced Machine Learning, Lecture 4

András Kornai

BME 2020/1

# Linear stuff cont'd

## Another significant quote

One may rightly ask just why he should conisider a linear decision function. Is there any guarantee that it will work? In general, this question can only be answered by designing the categorizer, and then deciding whether the resulting system is good enough (Highleyman 1962)

- Even when you just take an algorithm off the shelf, working with linear methods is a good first step
- They are easy to visualize/understand
- Already give a good idea of where the problems are
- Help in selecting the "true" model

# Data normalization

- Data matrix $D$ has rows (observations, 1520 in PB data) and columns (measurements, 4 in PB)
- We begin with *means centering*, subtracting the mean of each column from each entry in the column
- Optionally, we may divide by the variance as well. (Note that constant column causing DIVZERO could be omitted to begin with!)
- Covariance matrix $C$ is formed by arranging scalar products of col $i$ and col $j$ in (symmetric) square marix $C = D^T D$
- $C$ is positive semidefinite (definite if data rows were linearly independent), variance in an arbitrary direction $\vec{x}$ is given by $\vec{x}^T C \vec{x}$
- To maximize this we need to solve

$$\frac{d}{d\vec{x}} \vec{x}^T C \vec{x} - \lambda \vec{x}^T \vec{x}$$

# SVD to the rescue

- The Lagrange multiplier $-\lambda \vec{x}^T \vec{x}$ makes clear the critical points are solutions to $C\vec{x} = \lambda\vec{x}$
- So the solutions $\lambda_i$ are, by definition, the eigenvalues and the $x_i$ are the corresponding eigenvectors
- Let the SVD of $D$ be $UGV^T$. The columns of $V$ are the eigenvectors of $C$, and the positive singular values found in the diagonal matrix $G$ (conventionally arranged to run from larger to smaller) are the square roots of the eigenvalues $\lambda_i$ of $C$
- We use these to measure the "goodness" of the principal components. If $\Lambda = \sum_{i=1}^{c} \lambda_i$, we say that each $\lambda_i$ *accounts for* a fraction $\lambda_i/\Lambda$ of the total variance
- **Theorem** (Eckart–Young 1936) the first $r$ columns of $U, G, V$ can be used to form $U_r G_r V_r^T$ which is the best rank $r$ approximation of $D$ (in Frobenius norm)
- HW4.1: do this for PB data, plot goodness as function of $r$
- What is your expectation *before* doing the plot? Write it down

# PCA and LDA

- Normalizing your data by PCA generally helps: nonlinearities are dampened
- On occasion, very significant dimension reduction (e.g. from 300 to 20) is achievable
- HW4.2 On PB data do any of the methods you used improve by running PCA first? No need to add methods you haven't tried yet, but if you are inspired by the leaderboard, you can
- Another classical method is Linear Discriminant Analysis (LDA)
- Not just for data reduction, can be used as a standalone classifier
- Invented by Fisher (1936), another classic dataset 'iris' was used (but we stay with PB)

# LDA (BINARY CASE)

- We assume two classes $y = 0, 1$ and feature vectors $\vec{x}$. Also, we assume these are sampled from two normal distributions $p(\vec{x}|0)$ and $p(\vec{x}|1)$ which have means $\mu_0, \mu_1$ and covariances $\Sigma_0, \Sigma_1$ (for $n$ dim we have a total of $n(n+1)$ parameters)
- Especially for biological distributions, normal assumption often makes very good sense
- $(\vec{x}-\vec{\mu}_0)^T\Sigma_0^{-1}(\vec{x}-\vec{\mu}_0)+\ln|\Sigma_0|-(\vec{x}-\vec{\mu}_1)^T\Sigma_1^{-1}(\vec{x}-\vec{\mu}_1)-\ln|\Sigma_1| > T$ (where $T$ is the discriminant threshold)
- When we assume $\Sigma_0 = \Sigma_1$ (homoscedasticity) this simplifies to $\vec{x}\vec{w} > c$ where $\vec{w} = \Sigma^{-1}(\vec{\mu}_1 - \vec{\mu}_0)$ and $c = \vec{w} \cdot \frac{1}{2}(\vec{\mu}_1 + \vec{\mu}_0)$
- The hyperplane bisecting the vector connecting the means offers the best separation of the normal distributions (as long as they have the same variance)
- In high dimension, we may go for a *max margin* hyperplane instead, leading to SVMs
- Multiclass (m-way) is generally treated as a $\frac{1}{2}m(m-1)$ binaries

# More HW, project discussion

- HW4.3 study the PB data: how homoscedactic is it? How about after PCA?
- At this point, you are not asked to actually build an LDA model of the PB data, but later you might be, so you may want to look at https://scikit-learn.org/0.16/modules/generated/sklearn.lda.LDA.html
- Project topics claimed so far: sentiment analysis; QA; NER; summarization; medical (w/ heart dataset); medical (w/ lung dataset); adversarial attack on image classifiers; time series prediction (forex);
- Nobody needs to beat SOTA to get As!
- Everybody needs to develop a simple (but better than random) baseline first (possibly instead of a midterm)
- Non-programming (academic) projects are still feasible (possibly in Hungarian)