T-61.3050 Machine Learning: Basic Principles

Dimensionality Reduction

Kai Puolamäki

Laboratory of Computer and Information Science (CIS)
Department of Computer Science and Engineering
Helsinki University of Technology (TKK)

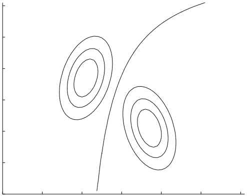
Autumn 2007

Outline

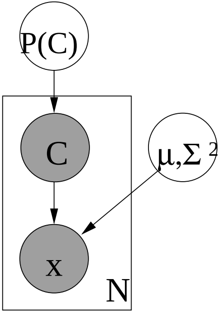
1. Multivariate Methods
   - Bayes Classifier
   - Discrete Variables
   - Multivariate Regression
2. Dimensionality Reduction
   - Subset Selection
   - Principal Component Analysis (PCA)
   - Linear Discriminant Analysis (LDA)

Bayes Classifier

- Data are real vectors.
- Idea: vectors are from class-specific multivariate normal distributions.
- Full model: covariance matrix has $O(Kd^2)$ parameters.

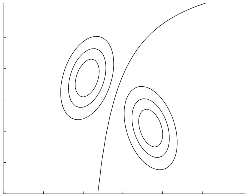


From Figure 5.3 of Alpaydin (2004).

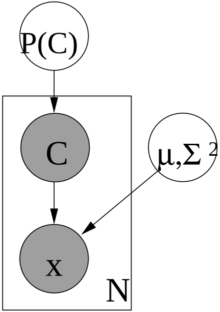


Bayes Classifier

- Data are real vectors.
- Idea: vectors are from class-specific multivariate normal distributions.
- Full model: $O(Kd^2)$ parameters in the covariance matrix.



From Figure 5.3 of Alpaydin (2004).



Bayes Classifier
Common covariance matrix

- Idea: the means are class-specific, covariance matrix Σ is common.
- $O(d^2)$ parameters in the covariance matrix.

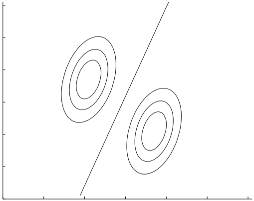


Figure 5.4: Covariances may be arbitary but shared by both classes. *From: E. Alpaydın. 2004.* Introduction to Machine Learning. ©*The MIT Press.*

Bayes Classifier
Common diagonal covariance matrix

- Idea: the means are class-specific, covariance matrix Σ is common and diagonal (Naive Bayes).
- d parameters in the covariance matrix.
- Discriminant: $g_i(\mathbf{x}) = -\frac{1}{2}\sum_{j=1}^{d}\left(x_j^t - m_{ij}\right)^2/s_j^2 + \log \hat{P}(C_i)$.

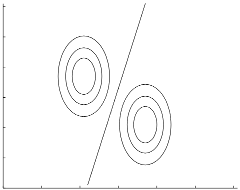
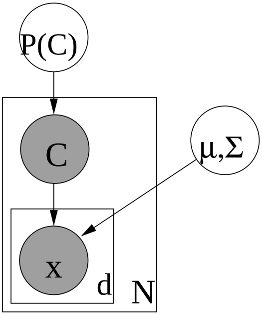


Figure 5.5: All classes have equal, diagonal covariance matrices but variances are not equal. *From: E. Alpaydın. 2004.* Introduction to Machine Learning. *©The MIT Press.*



Bayes Classifier

Nearest mean classifier

- Idea: the means are class-specific, covariance matrix Σ is common and proportional to unit matrix $\Sigma = \sigma^2 \mathbf{1}$.
- 1 parameter in the covariance matrix.
- Discriminant: $g_i(\mathbf{x}) = -\|\mathbf{x} - \mathbf{m}_i\|^2$.
- Nearest mean classifier. Each mean is a prototype.

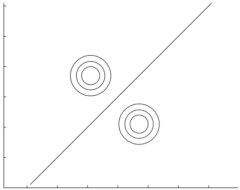
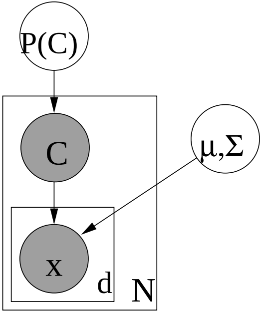


Figure 5.6: All classes have equal, diagonal covariance matrices of equal variances on both dimensions. *From: E. Alpaydın. 2004.* Introduction to Machine Learning. © *The MIT Press.*



Outline

Discrete Features

Most straightforward using Naive Bayes (replace Gaussian with Bernoulli):

- Binary features: $p_{ij} \equiv p(x_j = 1 \mid C_i)$

  if x_j are independent (Naive Bayes')

  $$p(x \mid C_i) = \prod_{j=1}^{d} p_{ij}^{x_j} (1 - p_{ij})^{(1-x_j)}$$

  the discriminant is linear

$$g_i(\mathbf{x}) = \log p(\mathbf{x} \mid C_i) + \log P(C_i)$$
$$= \sum_j \left[x_j \log p_{ij} + (1 - x_j) \log (1 - p_{ij}) \right] + \log P(C_i)$$

Estimated parameters $\hat{p}_{ij} = \dfrac{\sum_t x_j^t r_i^t}{\sum_t r_i^t}$

Lecture Notes for E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)

Outline

Multivariate Regression

$$r^t = g\left(x^t \mid w_0, w_1, \ldots, w_d\right) + \varepsilon$$

- Multivariate linear model

$$w_0 + w_1 x_1^t + w_2 x_2^t + \cdots + w_d x_d^t$$

$$E\left(w_0, w_1, \ldots, w_d \mid \mathcal{X}\right) = \frac{1}{2}\sum_t \left[r^t - w_0 - w_1 x_1^t - \cdots - w_d x_d^t\right]^2$$

- Multivariate polynomial model:

  Define new higher-order variables

  $z_1 = x_1$, $z_2 = x_2$, $z_3 = x_1^2$, $z_4 = x_2^2$, $z_5 = x_1 x_2$

  and use the linear model in this new $\boldsymbol{z}$ space

  (basis functions, kernel trick, SVM: Chapter 10)

Outline

Why Reduce Dimensionality?

1. Reduces time complexity: Less computation
2. Reduces space complexity: Less parameters
3. Saves the cost of observing the feature
4. Simpler models are more robust on small datasets
5. More interpretable; simpler explanation
6. Data visualization (structure, groups, outliers, etc) if plotted in 2 or 3 dimensions

Feature Selection vs. Extraction

- Feature selection: Choosing $k<d$ important features, ignoring the remaining $d - k$

  Subset selection algorithms

- Feature extraction: Project the

  original x_i , $i = 1,\ldots,d$ dimensions to

  new $k<d$ dimensions, z_j , $j = 1,\ldots,k$

  Principal components analysis (PCA), linear discriminant analysis (LDA), factor analysis (FA)

Subset Selection

- There are 2^d subsets of d features
- Forward search: Add the best feature at each step
  - Set of features F initially Ø.
  - At each iteration, find the best new feature $j = \text{argmin}_i \, E(F \cup x_i)$
  - Add x_j to F if $E(F \cup x_j) < E(F)$
- Hill-climbing $O(d^2)$ algorithm
- Backward search: Start with all features and remove one at a time, if possible.
- Floating search (Add k, remove l)

Subset Selection

Example

- Toy data set consists of 100 10-dimensional vectors from two classes (1 and 0).
- First two dimensions x_1^t and x_2^t: drawn from Gaussian with unit variance and mean of 1 or -1 for the classes 1 and 0, respectively.
- Remaining eight dimensions: drawn from Gaussian with zero mean and unit variance, that is, they contain no information of the class.
- Optimal classifier: If $x_1 + x_2$ is positive the class is 1, otherwise the class is 0.
- Use nearest mean classifier.
- Split data in random into training set of 30+30 items and validation set of 20+20 items.

Subset Selection

Example

Forward selection:

Features	E_{VALID}
$\emptyset$	0.500
1	0.175
1, **2**	**0.100**
1, 2, **4**	0.100
1, 2, 4, **5**	0.100
1, 2, 4, 5, **3**	0.075
1, 2, 4, 5, 3, **8**	**0.050**
1, 2, 4, 5, 4, 8, **6**	0.075
1, 2, 4, 5, 4, 8, 6, **7**	0.075
1, 2, 4, 5, 4, 8, 6, 7, **10**	0.100
1, 2, 4, 5, 4, 8, 6, 7, 10, **9**	0.150

Backward selection:

Features	E_{VALID}
9, 10, 4, 6, 7, 8, 3, 5, 2, 1	0.150
10, 4, 6, 7, 8, 3, 5, 2, 1	0.100
4, 6, 7, 8, 3, 5, 2, 1	0.075
6, 7, 8, 3, 5, 2, 1	0.075
7, 8, 3, 5, 2, 1	0.075
8, 3, 5, 2, 1	**0.050**
3, 5, 2, 1	0.075
5, 2, 1	0.100
2, 1	**0.100**
1	0.175
$\emptyset$	0.500

Optimal solution would be features 1, 2!

Outline

Principal Component Analysis (PCA)

- PCA finds low-dimensional linear subspace such that when $\mathbf{x}$ is projected there information loss (here defined as variance) is minimized.
- Finds directions of maximal variance.
- Projection pursuit: find direction $\sqsupseteq$ such that some measure (here variance $\mathrm{Var}(\mathbf{w}^T\mathbf{x})$) is maximized.
- Equivalent to finding eigenvalues and -vectors of covariance or correlation matrix.
- Can also be derived probabilistically (see Tipping ME, Bishop CM (1999) Mixtures of Probabilistic Principal Component Analyzers. Neural Computation 11: 443–482); probabilistic interpretation is important in deriving discrete variants.

Principal Component Analysis (PCA)

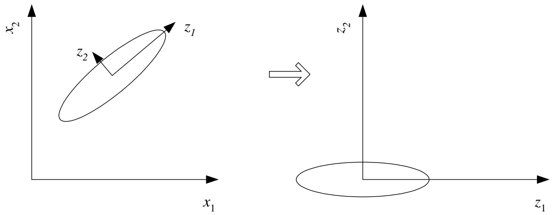


Figure 6.1: Principal components analysis centers the sample and then rotates the axes to line up with the directions of highest variance. If the variance on z_2 is too small, it can be ignored and we have dimensionality reduction from two to one. *From: E. Alpaydın. 2004.* Introduction to Machine Learning. ©*The MIT Press.*

Principal Component Analysis (PCA)

Example

```
> TOY1.pca <- princomp(TOY1[,1:10])
> summary(TOY1.pca)
Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4     Comp.5     Comp.6     Comp.7
Standard deviation     1.7310919 1.1681823 1.1206590 1.0989518 1.01793023 0.96416923 0.86400744
Proportion of Variance 0.2637625 0.1201141 0.1105401 0.1062992 0.09120295 0.08182375 0.06570642
Cumulative Proportion  0.2637625 0.3838767 0.4944168 0.6007160 0.69191899 0.77374274 0.83944917
                           Comp.8     Comp.9    Comp.10
Standard deviation     0.83517208 0.78446118 0.71496201
Proportion of Variance 0.06139384 0.05416463 0.04499236
Cumulative Proportion  0.90084301 0.95500764 1.00000000
```

Principal Component Analysis (PCA)

Example

Previous 10-dimensional toy example:

```
> TOY1.pca <- princomp(TOY1[,1:10])
> summary(TOY1.pca)
Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4     Comp.5     Comp.6     Comp.7
Standard deviation     1.7310919 1.1681823 1.1206590 1.0989518 1.01793023 0.96416923 0.86400744
Proportion of Variance 0.2637625 0.1201141 0.1105401 0.1062992 0.09120295 0.08182375 0.06570642
Cumulative Proportion  0.2637625 0.3838767 0.4944168 0.6007160 0.69191899 0.77374274 0.83944917
                           Comp.8     Comp.9    Comp.10
Standard deviation     0.83517208 0.78446118 0.71496201
Proportion of Variance 0.06139384 0.05416463 0.04499236
Cumulative Proportion  0.90084301 0.95500764 1.00000000
```

Principal Component Analysis (PCA)

Example

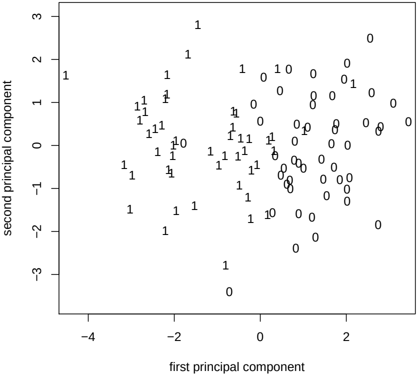
```
> TOY1.pca$loadings
```

Loadings:

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8	Comp.9	Comp.10
X1	-0.581		0.221	-0.431	0.188	0.187	0.488		0.174	-0.298
X2	-0.775	-0.166	-0.218	0.194	-0.200	-0.161	-0.353	-0.187		0.233
X3		-0.190	0.660	0.258	0.121	-0.352		-0.172	-0.495	-0.219
X4		0.162		-0.305	-0.384	-0.186	-0.428	-0.277	0.211	-0.623
X5	0.135	-0.655			-0.554	-0.239	0.346	0.101	0.230	
X6	0.109	0.252			-0.140		0.361	-0.813		0.323
X7	-0.131	0.596	0.235		-0.310	-0.461	0.145	0.411		0.255
X8		0.175	-0.272	0.682		-0.200	0.333		0.308	-0.411
X9		-0.160	-0.302	-0.338	0.532	-0.685				
X10			0.496	0.155	0.220		-0.267	-0.125	0.718	0.273

Principal Component Analysis (PCA)

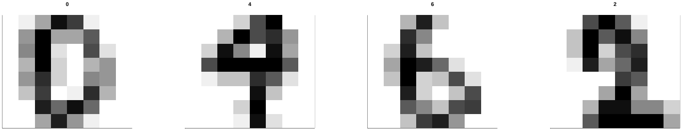
Example



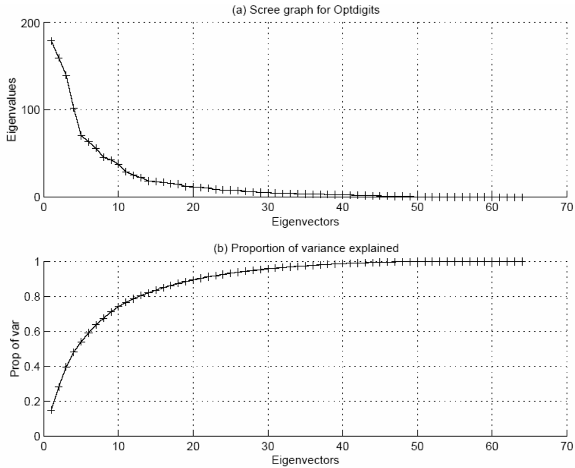

```
> TOY1.pc  <- predict(TOY1.pca)
> eqscplot(TOY1.pc[,1:2],type="n",
+         xlab="first principal component",
+         ylab="second principal component")
> text(TOY1.pc[,1:2],labels=as.character(TOY1[,"Class"]))
```

Example: Optdigits

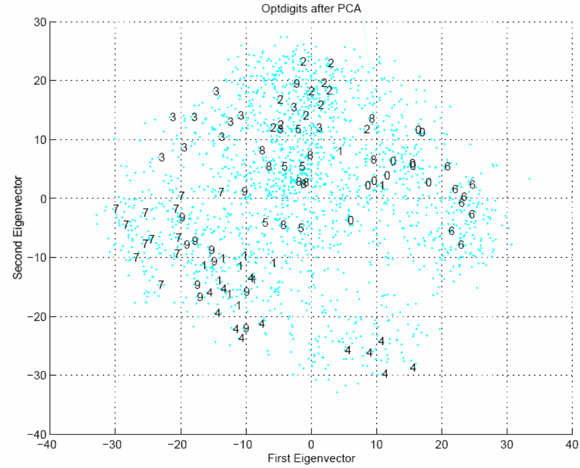
- OPTDIGITS data set contains 5620 instances of digitized handwritten digits in range 0–9.
- Each digit is a $\mathbb{R}^{64}$ vector: $8 \times 8 = 64$ pixels, 16 grayscales.



Principal Component Analysis (PCA)



Lecture Notes for E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)



Lecture Notes for E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)

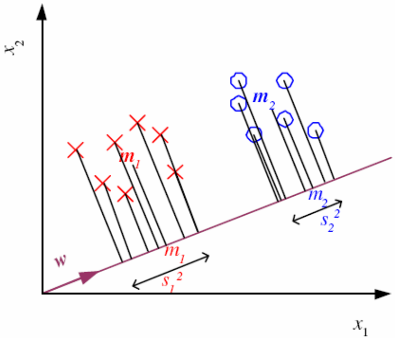
Outline

1. Multivariate Methods
   - Bayes Classifier
   - Discrete Variables
   - Multivariate Regression
2. Dimensionality Reduction
   - Subset Selection
   - Principal Component Analysis (PCA)
   - Linear Discriminant Analysis (LDA)

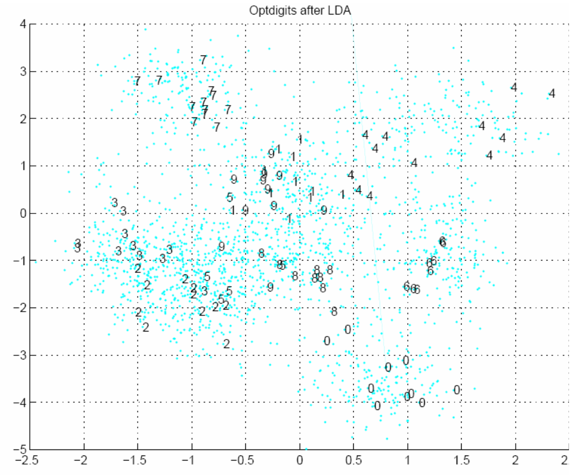
Linear Discriminant Analysis (LDAA)

- Find a low-dimensional space such that when $\boldsymbol{x}$ is projected, classes are well-separated.
- Find $\boldsymbol{w}$ that maximizes

$$J(\boldsymbol{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$$



$$m_1 = \frac{\sum_t \boldsymbol{w}^T \boldsymbol{x}^t r^t}{\sum_t r^t} \qquad s_1^2 = \sum_t \left(\boldsymbol{w}^T \boldsymbol{x}^t - m_1\right)^2 r^t$$



Lecture Notes for E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)