# Machine Learning I: Supervised Methods

B. Keith Jenkins

## Announcements

- Slido event code: 1872043
- Homework 7 is due Friday
  - For Pr. 2, use dataset Wine_data_v2.csv

## Today's lecture

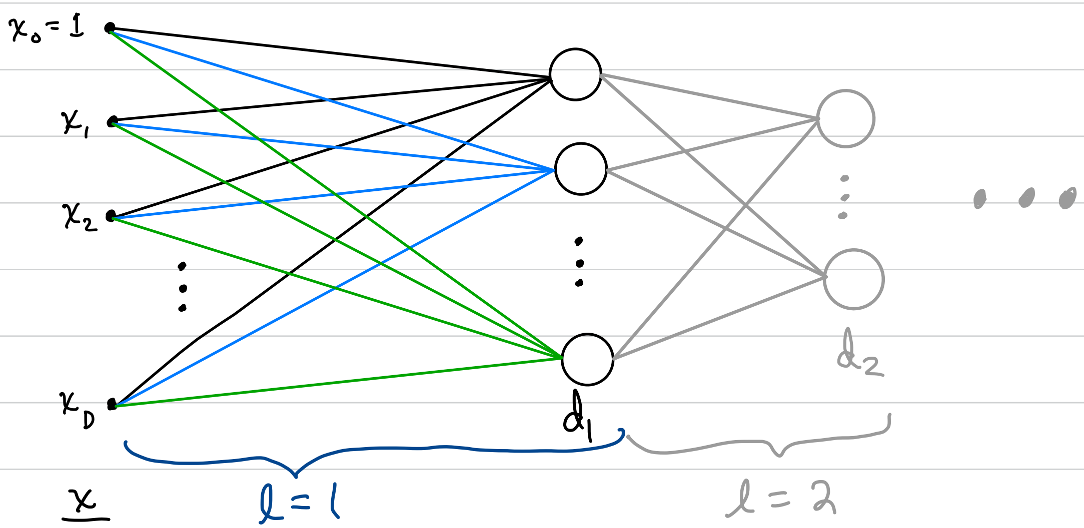
- d.o.f. and constraints in ANNs
  - Ex1: RBF networks
  - Ex2: More general ANNs
- Review of random vectors
  - Definitions
  - First and second order statistics
  - Multivariate Normal
  - Mahalanobis distance
  - Linear transformations
    - Orthonormal transformation
    - Whitening transformation
- Start Statistical Classification
  - Bayes Decision Theory (part 1)

* p. 2: activation fcn. notation changed to h post-lecture.

# Complexity: degrees of freedom and constraints in ANN

Each weight that is varied during learning, is a d.o.f.,
unless it is "tied" or constrained in some way.
Other parameters that are optimized as a function of the data are also d.o.f.

Ex 1: 1-layer (fully connected) ANN:




$d_l$ = # neuron units in layer $l$.

* $l = 1 \Rightarrow$ Assume activation functions h are given (not a fcn. of data)
d.o.f. = # $w_{jd}^{(1)}$ scalars = $(D+1) \cdot d_1$, if each $w_{jd}^{(1)}$ is not constrained or tied to other weights in some way.

- d.o.f. $\Rightarrow N_c = ?$ (1 layer ANN)

  Consider a 1000 x 1000 color image as input. $D = ?$ $3 \cdot 10^6$

  If $d_1 = 200 \times 200$, with $\ell = 1$ fully connected.

  Then how many weights in $\ell = 1$? $4 \cdot 10^4 \times 3 \cdot 10^6 \approx 10^{11}$

- How many images would we need in our training set?

  $\Rightarrow N > (3-10)$ d.o.f. $\approx (3-10) \cdot 10^{11} \sim 10^{12} = 1$ Trillion !

Ex 2: ChatGPT

GPT3: has 175 Billion parameters.
(Wikipedia has 4.2 billion words.)

GPT4: 1.7 Trillion parameters

GPT5: 2T – 5T

Will GPT run out of words to learn from?

Can reduce d.o.f. by:

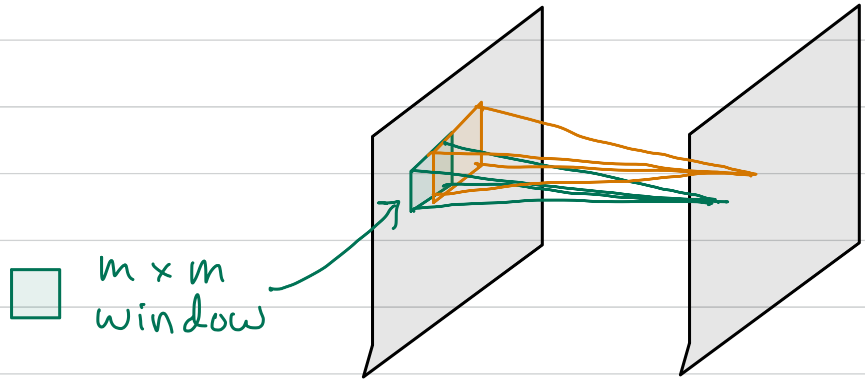
(1) Eliminate some interconnections
e.g., keep interconnections "local".

(2) "Tie" some weights to each other
e.g., same set of weights is repeated.

(1) & (2) are used to reduce d.o.f. in convolutional neural networks (CNNs), commonly used for ML on images and for computer vision.
[Bishop Sec. 5.5.6; Goodfellow Ch. 9]

(3) regularization can be used to constrain the weights in some other way.
[Bishop 5.5 describes some techniques.]

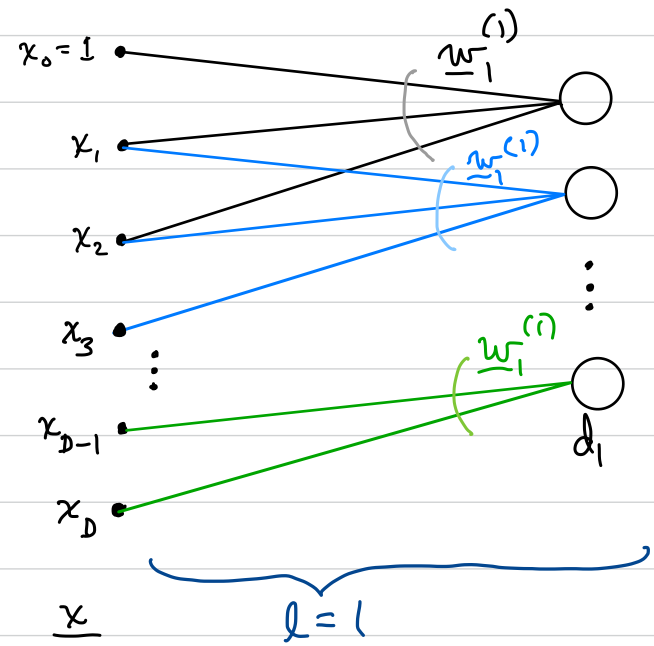
Example for images as input




input
$\sqrt{D} \times \sqrt{D}$ nodes

1st hidden
layer

⟹



# Random Vectors and Their Properties

Ref: Bishop 2.3.0-2.3.3; Duda, Hart, and Stork A.4.8-A.5.2; 2.5

## Definitions and Properties for 1 Random Vector

Let $\underline{x} = \left[ x_1, x_2 \ \cdots, \ x_D \right]^T$ in which the components of $\underline{x}$ are random variables (r.v.).

The joint probability density function $p$ is:

$$p\left(\underline{x}\right) = p\left(x_1, x_2, \cdots x_D\right)$$

in which $p$ is a scalar. Note that:

$$\int p\left(\underline{x}\right) d\underline{x} \triangleq \iint \cdots \int p\left(x_1, x_2, \cdots, x_D\right) dx_1 \, dx_2 \cdots dx_D = 1$$

### Mean or Expected Value

$$\underline{m} = E\left\{\underline{x}\right\} = \int \underline{x} \, p\left(\underline{x}\right) d\underline{x}$$

$$m_i = \iint \cdots \int x_i \, p\left(x_1, x_2, \cdots, x_D\right) dx_1 dx_2 \cdots dx_D = \int x_i \underbrace{\iint \cdots \int}_{\text{all } i \neq k} p(x_1, \cdots, x_D) \, dx_1 \cdots dx_D$$

$$m_i = \int x_i p\left(x_i\right) dx_i$$

in which $p\left(x_i\right) = \iint \cdots \int p\left(x_1, x_2, \cdots, x_D\right) dx_1 \, dx_2 \cdots dx_{i-1} \, dx_{i+1} \cdots dx_D$

p.6

## Autocorrelation and Cross-correlation Matrix

Autocorrelation matrix:

$$E\left\{\underline{x}\,\underline{x}^T\right\} = E\left\{ \begin{bmatrix} x_1 \\ \vdots \\ x_D \end{bmatrix} \begin{bmatrix} x_1 & \cdots & x_D \end{bmatrix} \right\}$$

$$= E\left\{ \begin{bmatrix} x_1x_1 & x_1x_2 & x_1x_3 & \cdots & x_1x_D \\ x_2x_1 & & & & \\ x_3x_1 & & \ddots & & \vdots \\ \vdots & & & & \\ x_Dx_1 & & \cdots & & x_Dx_D \end{bmatrix} \right\}$$

e.g.: living area #rooms

$$= \begin{bmatrix} E\{x_1x_1\} & E\{x_1x_2\} & E\{x_1x_3\} & \cdots & E\{x_1x_D\} \\ E\{x_2x_1\} & & & & \\ E\{x_3x_1\} & & \ddots & & \vdots \\ \vdots & & & & \\ E\{x_Dx_1\} & & \cdots & & E\{x_Dx_D\} \end{bmatrix}$$

in which $E\left\{x_i x_j\right\} = \iint x_i x_j \, p\left(x_i, x_j\right) dx_i \, dx_j$ .

Cross-correlation matrix is $E\left\{\underline{x}\,\underline{y}^T\right\}$.

**Covariance Matrix**

$$\underline{\underline{\Sigma}} = E\left\{\left(\underline{x}-\underline{m}\right)\left(\underline{x}-\underline{m}\right)^T\right\}$$

$$= \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \cdots & \sigma_{1D} \\ \sigma_{21} & & & & \\ \sigma_{31} & & \ddots & & \vdots \\ \vdots & & & & \\ \sigma_{D1} & & \cdots & & \sigma_{DD} \end{bmatrix}$$

in which $\sigma_{ij} = E\left\{\left(x_i - m_i\right)\left(x_j - m_j\right)\right\}$, and $\sigma_{ii} = \sigma_i^2 = E\left\{\left(x_i - m_i\right)^2\right\}$.

covariance of $x_i$ and $x_j$ variance of $x_i$ ($i^{\text{th}}$ feature).

Note that

$$\underline{\underline{\Sigma}} = E\left\{\left(\underline{x}-\underline{m}\right)\left(\underline{x}-\underline{m}\right)^T\right\}$$

$$= E\left\{\underline{x}\,\underline{x}^T\right\} - E\left\{\underline{m}\,\underline{x}^T\right\} - E\left\{\underline{x}\,\underline{m}^T\right\} + E\left\{\underline{m}\,\underline{m}^T\right\}$$

$$= E\left\{\underline{x}\,\underline{x}^T\right\} - \underline{m}\,E\left\{\underline{x}^T\right\} - E\left\{\underline{x}\right\}\underline{m}^T + \underline{m}\,\underline{m}^T$$

$$= E\left\{\underline{x}\,\underline{x}^T\right\} - \underline{m}\,\underline{m}^T - \cancel{\underline{m}\,\underline{m}^T} + \cancel{\underline{m}\,\underline{m}^T}$$

$$\underline{\underline{\Sigma}} = E\left\{\underline{x}\,\underline{x}^T\right\} - \underline{m}\,\underline{m}^T$$

$\underline{\underline{\Sigma}}$ is symmetric and positive semi-definite.

**We will assume $\underline{\underline{\Sigma}}$ is positive definite**, so that $\left|\underline{\underline{\Sigma}}\right| > 0$.

$\left|\underline{\underline{\Sigma}}\right| = 0$ is a degenerate case; for example this will happen when:

$$\sigma_{ii} = 0 \text{ for some } i, \quad \text{or} \quad x_i = \alpha x_j.$$

The covariance matrix is often normalized:

$$\underline{\underline{\Sigma}} = \underline{\underline{\Gamma}}\ \underline{\underline{R}}\ \underline{\underline{\Gamma}}$$

in which

$$\underline{\underline{\Gamma}} = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \mathbf{0} \\ & \mathbf{0} & \ddots & \\ & & & \sigma_D \end{bmatrix}, \quad \underline{\underline{R}} = \begin{bmatrix} 1 & & & \\ & 1 & & r_{ij} \\ & r_{ji} & \ddots & \\ & & & 1 \end{bmatrix}$$

with

$$r_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$$

Note that $0 \le |r_{ij}| \le 1$. $\underline{\underline{R}}$ is often called a "correlation matrix", but this terminology can be confusing. Another term is "normalized covariance matrix".

## Definitions and Properties for 2 Random Vectors

Two random vectors $\underline{x}$ and $\underline{y}$ are defined as:

– Uncorrelated if $E\left\{\underline{x}\,\underline{y}^T\right\} = E\left\{\underline{x}\right\} E\left\{\underline{y}^T\right\}$

– Orthogonal if $E\left\{\underline{x}^T \underline{y}\right\} = 0$

– Independent if $p\left(\underline{x}, \underline{y}\right) = p\left(\underline{x}\right) p\left(\underline{y}\right)$

As a result,

(1) independent => uncorrelated

(2) uncorrelated ≠> independent, in general.

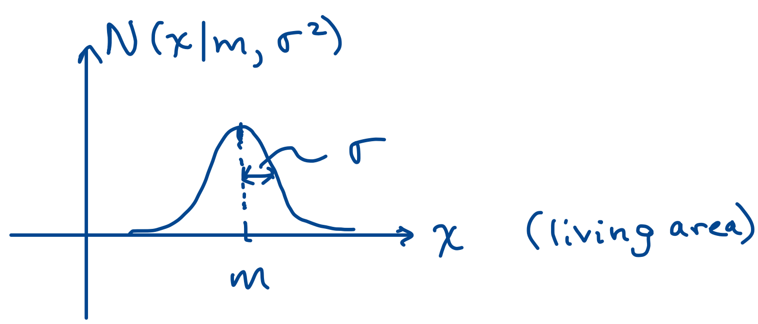
(3) If $E\left\{\underline{x}\right\} = \underline{0}$ or $E\left\{\underline{y}\right\} = \underline{0}$, then uncorrelated => orthogonal.

**Normal or Gaussian density** [DHS 2.5]

**Univariate case**

$$p(x) = N\left(x \middle| m, \sigma^2\right) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left\{-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2\right\}$$

$$E\{x\} = m, \quad E\left\{(x-m)^2\right\} = \sigma^2$$



**Multivariate case**

$$p(\underline{x}) = N(\underline{x} | \underline{m}, \underline{\underline{\Sigma}}) = \frac{1}{(2\pi)^{D/2} |\underline{\underline{\Sigma}}|^{1/2}} \exp\left\{-\frac{1}{2} d_M^2 (\underline{x}, \underline{m}, \underline{\underline{\Sigma}})\right\}$$

in which $|\underline{\underline{\Sigma}}| =$ determinant of $\underline{\underline{\Sigma}}$ ,

$$d_M^2(\underline{x}, \underline{m}, \underline{\underline{\Sigma}}) = (\underline{x} - \underline{m})^T \underline{\underline{\Sigma}}^{-1} (\underline{x} - \underline{m})$$
$$= \mathrm{tr}\left\{\underline{\underline{\Sigma}}^{-1} (\underline{x} - \underline{m})(\underline{x} - \underline{m})^T\right\}$$

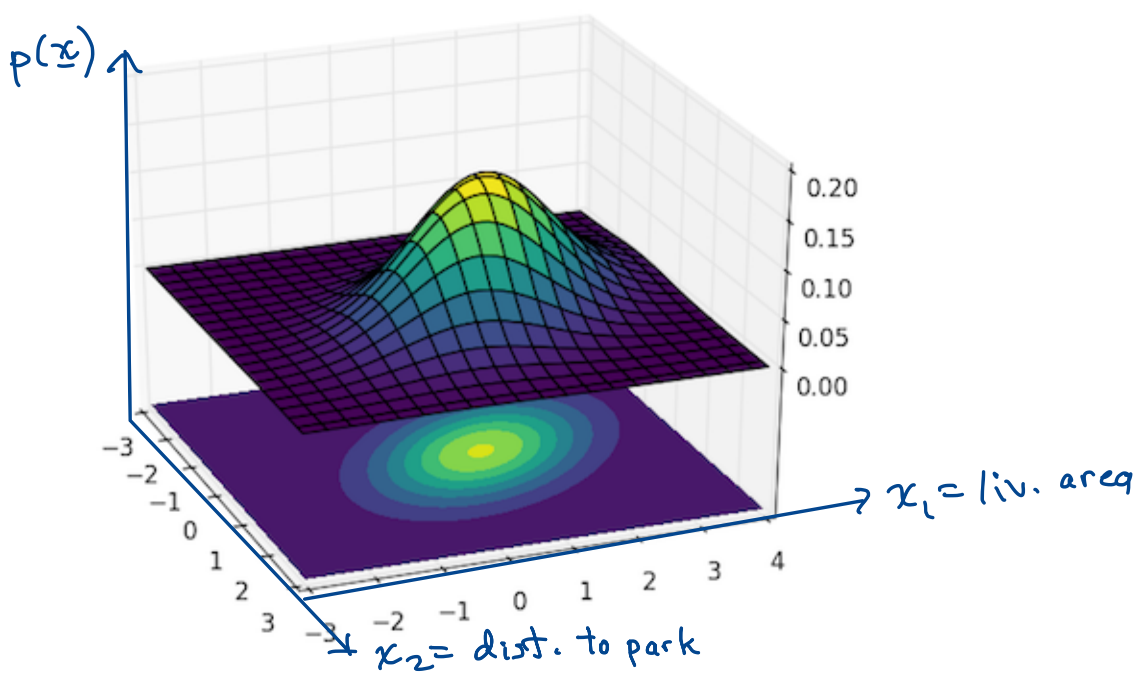
$\mathrm{tr}\{\underline{\underline{A}}\}$ = trace of $\underline{\underline{A}}$, and

$d_M(\underline{x}, \underline{m}, \underline{\underline{\Sigma}})$ = Mahalanobis distance between $\underline{x}$ and $\underline{m}$ .

Thus, the multivariate normal is a function of:

$D$ mean values and

$\frac{D(D+1)}{2}$ variance values.




Plot of bivariate Normal [1]

[1] scipython.com/blog/visualizing-the-bivariate-gaussian-distribution/

**Uncorrelated Features**

If the $x_i$ and $x_j$ are uncorrelated $\forall j \neq i$, then for $i \neq j$ :

$$\begin{aligned}\sigma_{ij} &= E\left\{\left(x_i - m_i\right)\left(x_j - m_j\right)\right\} \\ &= E\left\{x_i x_j\right\} - m_i m_j = E\left\{x_i\right\} E\left\{x_j\right\} - m_i m_j \\ &= 0\end{aligned}$$

Thus

$$\underline{\underline{\Sigma}} = \begin{bmatrix} \sigma_{11} & & & 0 \\ & \sigma_{22} & & \\ & 0 & \ddots & \\ & & & \sigma_{DD} \end{bmatrix}, \quad \underline{\underline{\Sigma}}^{-1} = \begin{bmatrix} \frac{1}{\sigma_{11}} & & & 0 \\ & \frac{1}{\sigma_{22}} & & \\ & & \ddots & \\ 0 & & & \frac{1}{\sigma_{DD}} \end{bmatrix}$$

and

$$d_M^2\left(\underline{x}, \underline{m}, \underline{\underline{\Sigma}}\right) = \left(\underline{x} - \underline{m}\right)^T \underline{\underline{\Sigma}}^{-1} \left(\underline{x} - \underline{m}\right)$$

$$\Rightarrow \; d_M^2\left(\underline{x}, \underline{m}, \underline{\underline{\Sigma}}\right) = \sum_{i=1}^{D} \frac{\left(x_i - m_i\right)^2}{\sigma_{ii}}$$

= Euclidean dist.$^2$ with auto-normalizing factor $\frac{1}{\sigma_{ii}} = \frac{1}{\sigma_i^2}$ for each feature $x_i$.

Also, for the normal case (with uncorrelated features):

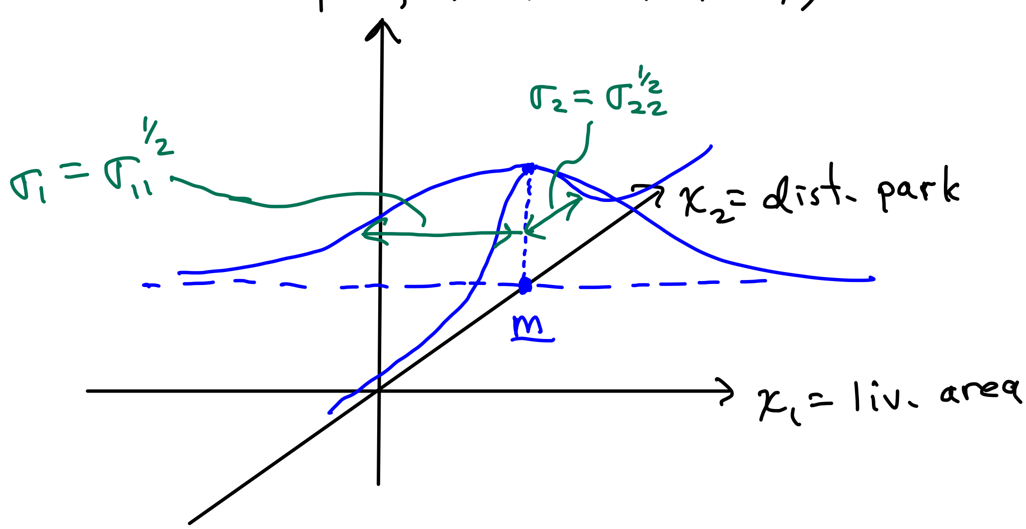
$$p(\underline{x}) = \frac{1}{(2\pi)^{D/2}\left(\prod_{i=1}^{D}\sigma_{ii}\right)^{1/2}} \exp\left\{-\frac{1}{2}\underbrace{\sum_{i=1}^{D}\frac{(x_i - m_i)^2}{\sigma_{ii}}}_{d_m^2(\underline{x},\, \underline{m},\, \underline{\underline{\Sigma}})}\right\}$$

$$= \prod_{i=1}^{D}\left[\frac{1}{(2\pi)^{1/2}\sigma_{ii}^{\ 1/2}}\exp\left\{-\frac{1}{2}\frac{(x_i - m_i)^2}{\sigma_{ii}}\right\}\right]$$

$$= \prod_{i=1}^{D} p(x_i) = \prod_{i=1}^{D} N(x_i | m_i, \sigma_{ii})$$

So uncorrelated => independent for the normal case.

$p(x_1, x_2) = N(\underline{x} | \underline{m}, \underline{\underline{\Sigma}})$, $x_1$ and $x_2$ uncorrelated



## Linear Transformation of Random Vectors

$\boxed{\text{Let } \underline{y} = \underline{\underline{A}}\,\underline{x}\ ,}$ $\quad$ $\underline{x}$ and $\underline{y}$ are random vectors

$\underline{x}$ in orig. feat. space
$\underline{y}$ in transformed feat. space.

Then $\underline{m}_y = \underline{\underline{A}}\,\underline{m}_x$ , $\quad$ $\underline{\underline{A}}$ is deterministic

And

$$\underline{\underline{\Sigma}}_y = \mathrm{E}\left\{\left(\underline{y}-\underline{m}_y\right)\left(\underline{y}-\underline{m}_y\right)^T\right\}$$

$$= \mathrm{E}\left\{\left(\underline{\underline{A}}\,\underline{x}-\underline{\underline{A}}\,\underline{m}_x\right)\left(\underline{\underline{A}}\,\underline{x}-\underline{\underline{A}}\,\underline{m}_x\right)^T\right\}$$

$$= \underline{\underline{A}}\ \mathrm{E}\left\{\left(\underline{x}-\underline{m}_x\right)\left(\underline{x}-\underline{m}_x\right)^T\right\}\underline{\underline{A}}^T$$

$$\Rightarrow \quad \boxed{\underline{\underline{\Sigma}}_y = \underline{\underline{A}}\,\underline{\underline{\Sigma}}_x\,\underline{\underline{A}}^T}$$

Assuming $\underline{\underline{A}}$ is nonsingular,

$$d_M^{\ 2}\left(\underline{y}, \underline{m}_y, \underline{\underline{\Sigma}}_y\right) = \left(\underline{y}-\underline{m}_y\right)^T \underline{\underline{\Sigma}}_y^{-1}\left(\underline{y}-\underline{m}_y\right)$$

$$= \left[\underline{\underline{A}}\left(\underline{x}-\underline{m}_x\right)\right]^T \underbrace{\left[\underline{\underline{A}}\,\underline{\underline{\Sigma}}_x\,\underline{\underline{A}}^T\right]^{-1}}\left[\underline{\underline{A}}\left(\underline{x}-\underline{m}_x\right)\right]$$

$$= \left(\underline{x}-\underline{m}_x\right)^T \cancel{\underline{\underline{A}}^T}\left(\cancel{\underline{\underline{A}}^T}\right)^{-1}\underline{\underline{\Sigma}}_x^{-1}\cancel{\underline{\underline{A}}^{-1}}\,\cancel{\underline{\underline{A}}}\left(\underline{x}-\underline{m}_x\right)$$

$$= \left(\underline{x}-\underline{m}_x\right)^T \underline{\underline{\Sigma}}_x^{-1}\left(\underline{x}-\underline{m}_x\right)$$

$$= d_M^{\ 2}\left(\underline{x}, \underline{m}_x, \underline{\underline{\Sigma}}_x\right)$$

$\Rightarrow$ Mahalanobis distance is preserved under a linear transformation with nonsingular $\underline{\underline{A}}$ .

## Orthonormal Transformation

is a special case of a linear transformation.

Let $\boxed{\underline{y} = \underline{\underline{\mathrm{E}}}^T \underline{x}}$, => $\underline{\underline{\mathrm{A}}} = \underline{\underline{\mathrm{E}}}^T$. Let $\underline{\underline{\mathrm{E}}}^T \underline{\underline{\mathrm{E}}} = \underline{\underline{\mathrm{I}}} = \underline{\underline{\mathrm{E}}}\,\underline{\underline{\mathrm{E}}}^T$ ($\underline{\underline{\mathrm{E}}}$ is orthonormal).

Also let $\underline{\underline{\mathrm{E}}} =$ eigenmatrix of $\underline{\underline{\Sigma}}_x$ : $\underline{\underline{\mathrm{E}}} \triangleq \begin{bmatrix} \underline{e}_1 & \underline{e}_2 & \cdots & \underline{e}_D \end{bmatrix}$

in which $\underline{e}_n$ is the $n^{\text{th}}$ eigenvector of $\underline{\underline{\Sigma}}_x$, and $\{\underline{e}_n, n = 1, 2, \cdots, D\}$ is orthonormal.[1]

$\underline{\underline{\mathrm{E}}}$ satisfies: $\boxed{\underline{\underline{\Sigma}}_x \underline{\underline{\mathrm{E}}} = \underline{\underline{\mathrm{E}}}\,\underline{\underline{\Lambda}},}$ in which $\underline{\underline{\Lambda}} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_D \end{bmatrix}$

and $\lambda_n$ is the $n^{\text{th}}$ eigenvalue of $\underline{\underline{\Sigma}}_x$.[2]

Note that: $\underline{\underline{\Sigma}}_y = \underline{\underline{\mathrm{A}}}\,\underline{\underline{\Sigma}}_x \underline{\underline{\mathrm{A}}}^T = \underline{\underline{\mathrm{E}}}^T \left(\underline{\underline{\Sigma}}_x \underline{\underline{\mathrm{E}}}\right) = \underline{\underline{\mathrm{E}}}^T \underline{\underline{\mathrm{E}}}\,\underline{\underline{\Lambda}} = \underline{\underline{\Lambda}}$. Thus $\underline{\underline{\Sigma}}_y = \underline{\underline{\Lambda}}$

and $\underline{\underline{\mathrm{E}}}$ diagonalizes $\underline{\underline{\Sigma}}_x$ .

$\underline{\underline{\Sigma}}_y =$ diagonal $\Rightarrow$ the components of $\underline{y}$ are uncorrelated.

*Comments:*

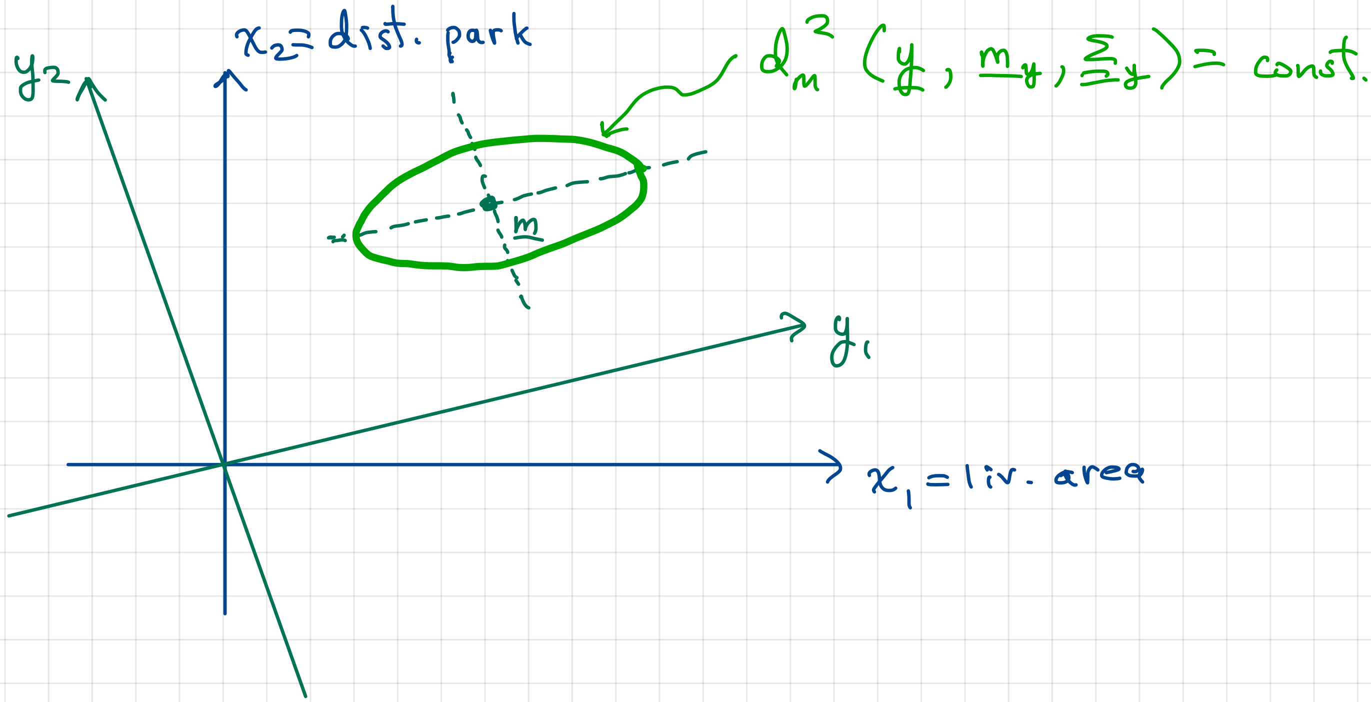
1. Orthonormal transformations are the basis of PCA.
2. It can be shown that Euclidean distance is preserved under an orthonormal transformation.

S

---

[1] Note that $\{\underline{e}_n, n = 1, 2, \cdots, D\}$ is orthogonal, because $\underline{\underline{\Sigma}}_x$ is real and symmetric. The set can be made orthonormal by normalizing each eigenvector to unit length. [2] All $\lambda_n$ are real because $\underline{\underline{\Sigma}}_x$ is symmetric. All $\lambda_n > 0$ because we assumed $\underline{\underline{\Sigma}}_x$ is positive definite.

Ex:


$x_2$ = dist. park
$y_2$
$d_M^2(\underline{y}, \underline{m}_y, \underline{\underline{\Sigma}}_y) = \text{const.}$
$\underline{m}$
$y_1$
$x_1$ = liv. area

## Whitening Transformation

Another special case of a linear transformation

Let $\boxed{\underline{y} = \underline{\underline{\Lambda}}^{-1/2}\, \underline{\underline{E}}^T \underline{x}} \;\Rightarrow\; \underline{\underline{A}} = \underline{\underline{\Lambda}}^{-1/2}\, \underline{\underline{E}}^T$

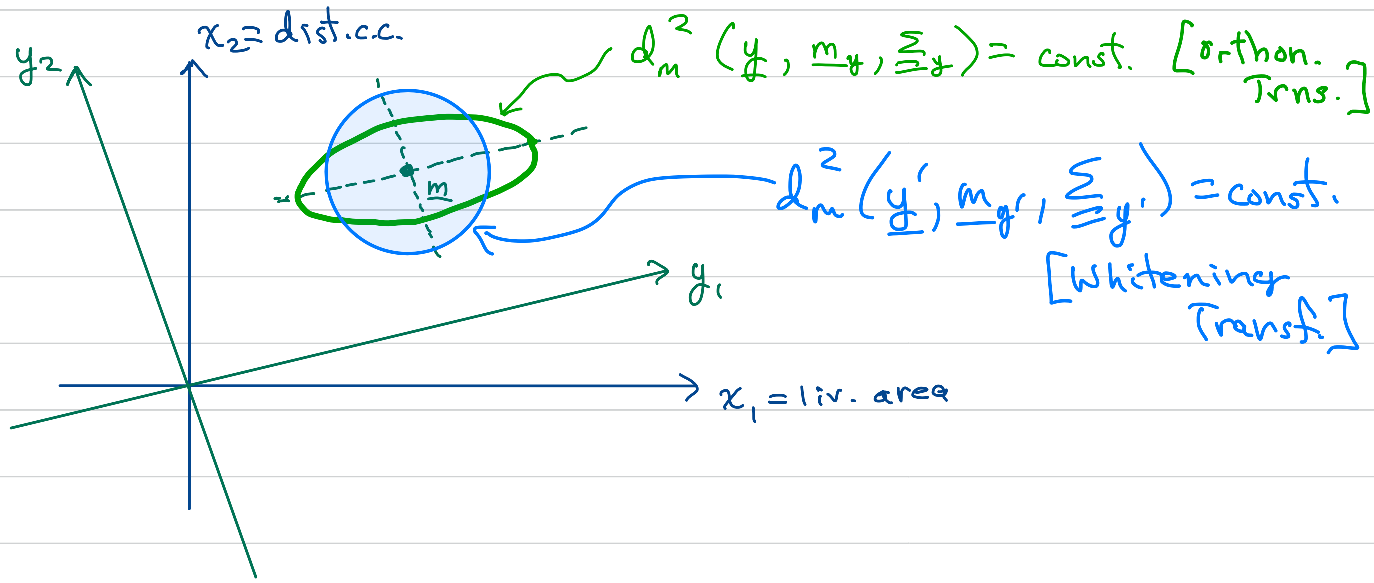
[ in which $\underline{\underline{E}}$ = eigenmatrix of $\underline{\underline{\Sigma}}_x$, and $\underline{\underline{\Lambda}}$ = eigenvalue mtx. of $\underline{\underline{\Sigma}}_x$

$$\underline{\underline{\Sigma}}_y = \underline{\underline{A}}\, \underline{\underline{\Sigma}}_x \underline{\underline{A}}^T = \left(\underline{\underline{\Lambda}}^{-1/2}\, \underbrace{\underline{\underline{E}}^T\right) \underline{\underline{\Sigma}}_x \left(\underline{\underline{E}}}_{\underline{\underline{\Lambda}}}\, \underline{\underline{\Lambda}}^{-1/2}\right)$$

$$\boxed{\underline{\underline{\Sigma}}_y = \underline{\underline{\Lambda}}^{-1/2}\, \underline{\underline{\Lambda}}\, \underline{\underline{\Lambda}}^{-1/2} = \underline{\underline{\Lambda}}^{-1/2}\, \underline{\underline{\Lambda}}^{1/2} = \underline{\underline{I}} = \text{identity matrix}}$$

$\Rightarrow \sigma_{ii}^2 = 1, \quad \sigma_{ij}^2 = 0$ for $i \neq j$.

$\therefore$ The elements of $\underline{y}$ are uncorrelated and have unit variance.




**Comment**: The whitening transformation is not orthonormal, and Euclidean distances are not preserved.

$$\| \underline{y}_1 - \underline{y}_2 \|_2^2 \neq \| \underline{x}_1 - \underline{x}_2 \|_2^2$$

# STATISTICAL CLASSIFICATION

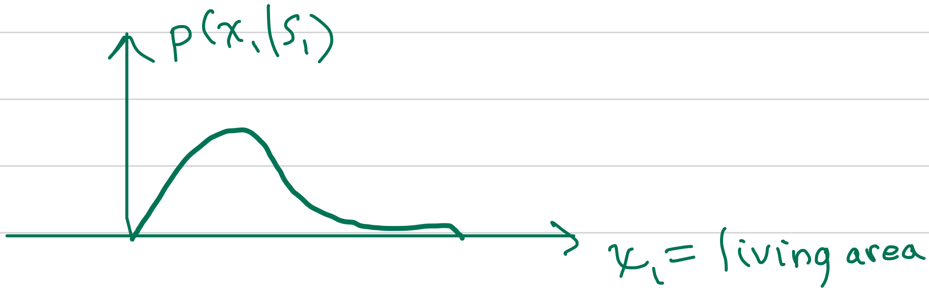
Assume: data pts. $\underline{x}_i$ (and unknowns $\underline{x}$) are drawn i.i.d. from $p(\underline{x})$.

## Bayes Decision Theory [Bishop 1.5]

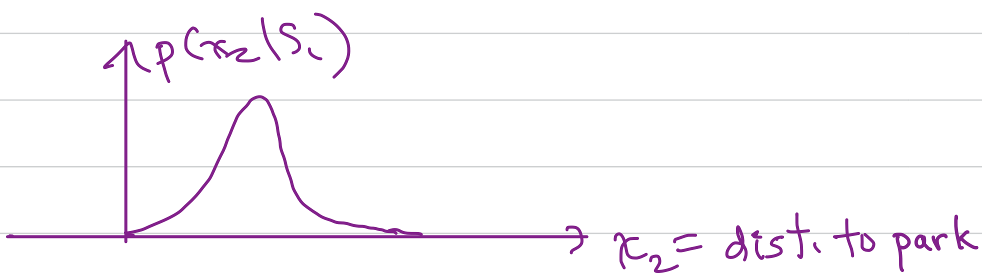
Consider $p(\underline{x} \mid S_i)$ = class-conditional density

e.g.:

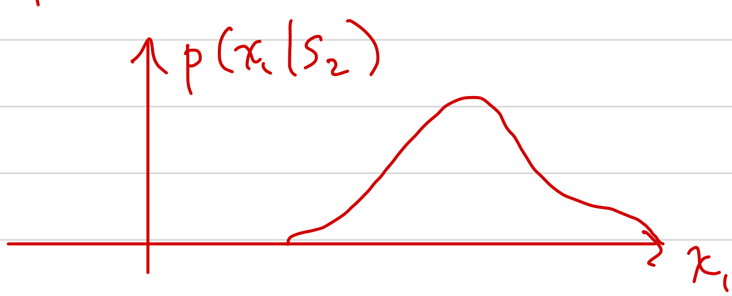
$p(x_1 \mid S_1) = p(\text{living area} \mid \text{price increase})$




$p(x_2 \mid S_1) = p(\text{distance to park} \mid \text{price increase})$




$p(x_1 \mid S_2) = p(\text{living area} \mid \text{price decrease})$



$$\boxed{\text{Assume: } p(\underline{x} \mid S_i) \text{ is known } \forall i}$$

Priors: $P(S_i)$

e.g.: $P(S_1) = P(\text{price increase})$, without knowledge of $\underline{x}$.

$$\boxed{\text{Assume: } P(S_i) \text{ is known } \forall i}$$

If $P(S_i)$ are unknown, can estimate by:

$$\hat{P}(S_i) = \frac{N_i}{N}, \quad \text{in which } N = \sum_{i=1}^{c} N_i$$

## Bayes minimum-error classifier ($c = 2$ classes)

Goal: choose decision boundary and regions that minimize $P(\text{error}) = P_e$

Let "e" denote "error"

$$P_e = P(e, S_1) + P(e, S_2)$$

$$= \underbrace{P(e \mid S_1)}_{\int_{\Gamma_2} p(\underline{x} \mid S_1)\, d\underline{x}} P(S_1) + \underbrace{P(e \mid S_2)}_{\int_{\Gamma_1} p(\underline{x} \mid S_2)\, d\underline{x}} P(S_2)$$

$$\boxed{(1) \quad \therefore P_e = P(S_1) \int_{\Gamma_2} p(\underline{x} \mid S_1)\, d\underline{x} + P(S_2) \int_{\Gamma_1} p(\underline{x} \mid S_2)\, d\underline{x}}$$