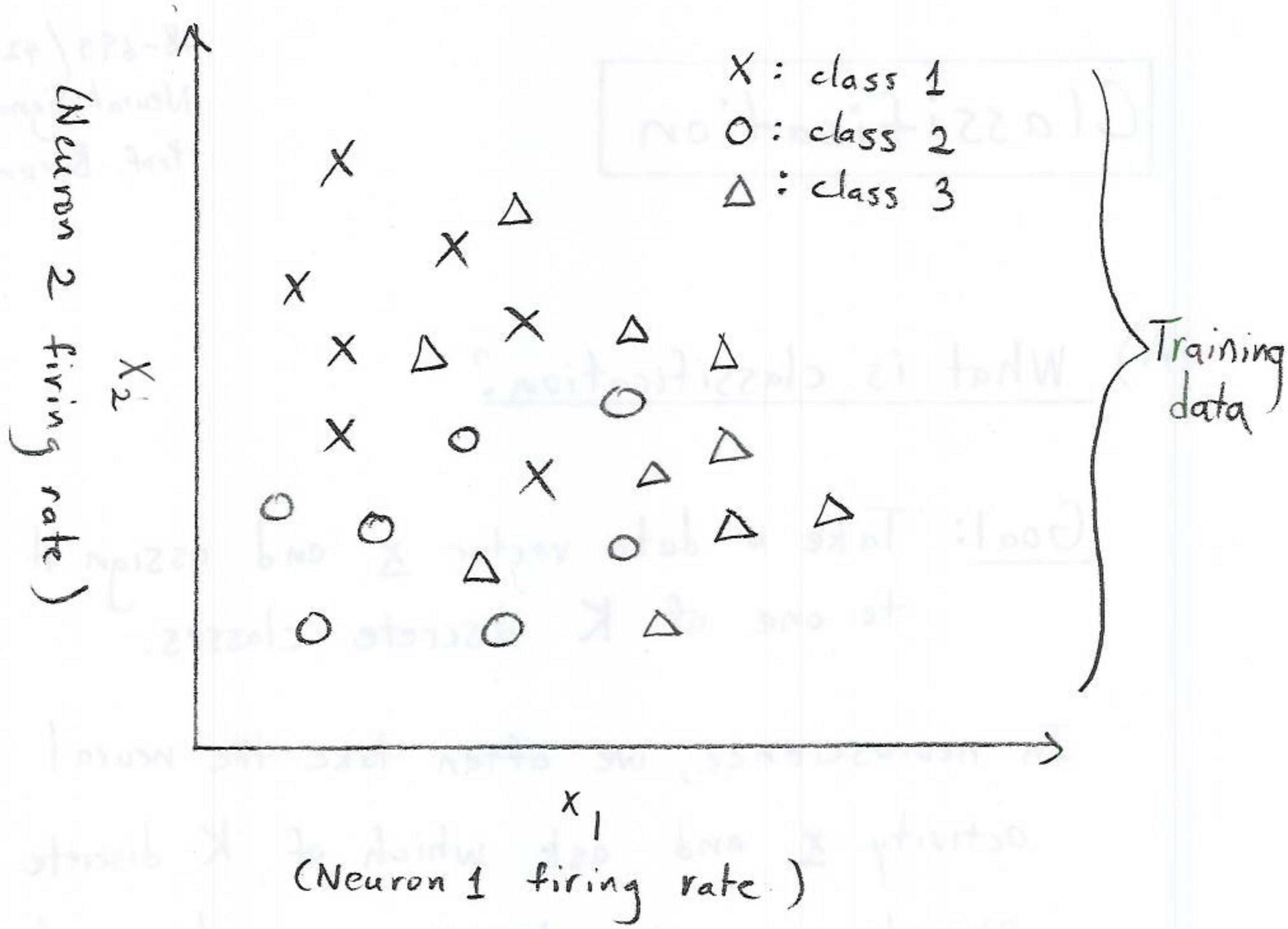
Classification

A) What is classification?

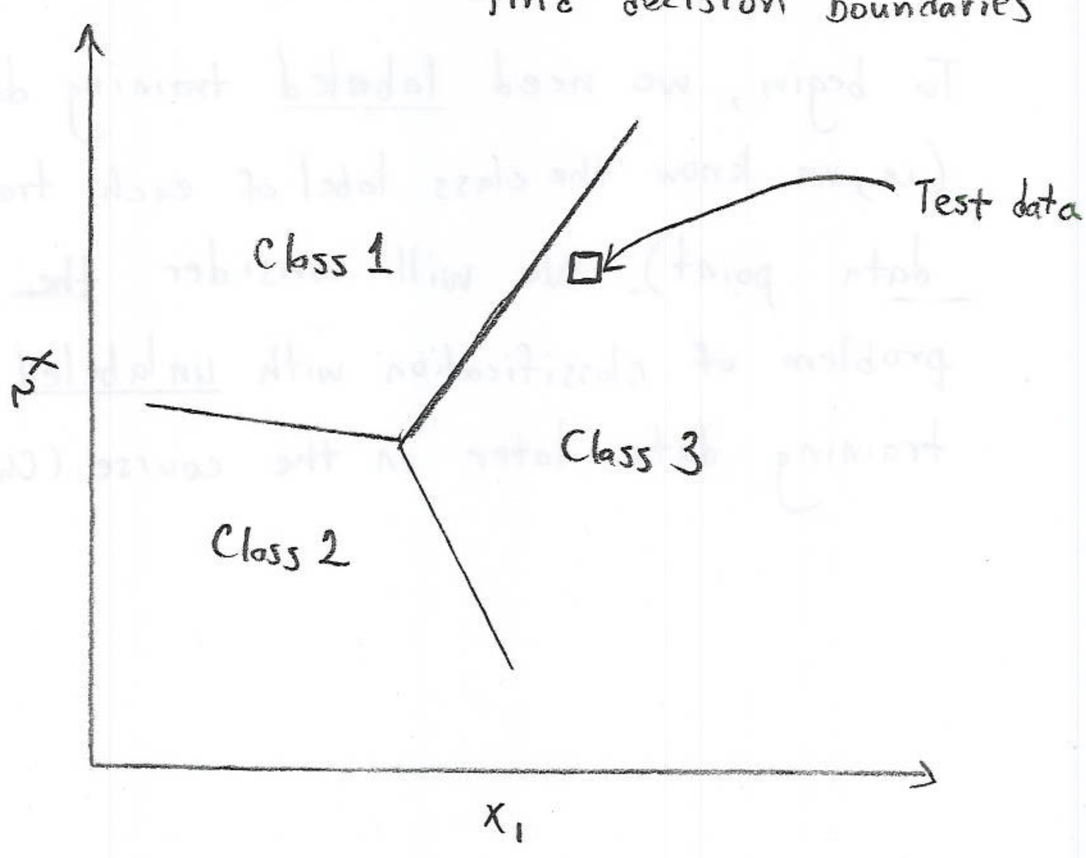
Goal: Take a data vector $\underline{x}$ and assign it to one of K discrete classes.

In neuroscience, we often take the neural activity $\underline{x}$ and ask which of K discrete stimuli gave rise to the observed neural activity.

To begin, we need labeled training data (i.e., we know the class label of each training data point). We will consider the problem of classification with unlabeled training data later in the course (Ch. 9).



⇓ using methods we will discuss, find decision boundaries



B) Classifying Using Generative Models

Training phase:

- Fit class-conditional densities $P(\underline{x} \mid C_k)$ and class priors $P(C_k)$ to training data. $(k = 1, \dots, K)$

Test phase:

- Compute $P(C_k \mid \underset{\text{test data}}{\underline{x}})$ using Bayes' rule

$$P(C_k \mid \underline{x}) = \frac{P(\underline{x} \mid C_k)\, P(C_k)}{P(\underline{x})}$$

$$= \frac{P(\underline{x} \mid C_k)\, P(C_k)}{\sum_{j=1}^{K} P(\underline{x} \mid C_j)\, P(C_j)}$$

- Assign class $\hat{k} = \underset{k}{\operatorname{argmax}}\, P(C_k \mid \underline{x})$ to test data $\underline{x}$.

8.1) Generative models

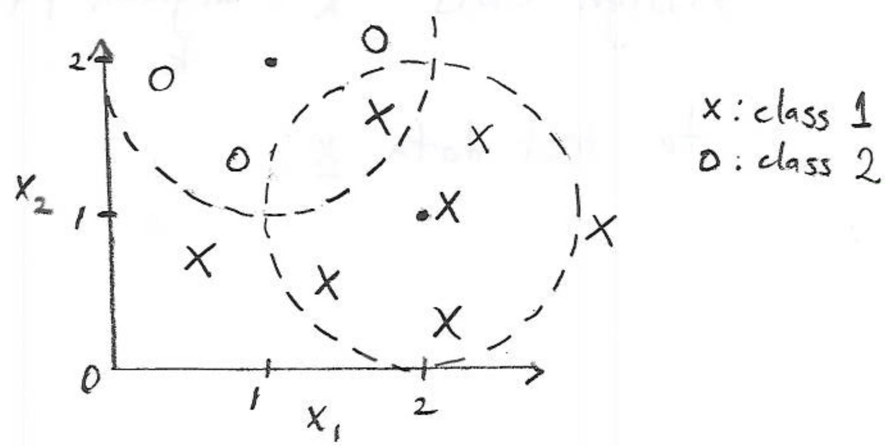
$P(\underline{x} \mid C_k)$ and $P(C_k)$ define a "probabilistic generative model". This means that we can generate synthetic data from the model.

For example, say there are two classes and $\underline{x} \in \mathbb{R}^2$

$$P(C_1) = 0.7$$
$$P(C_2) = 0.3$$
$$P(\underline{x} \mid C_1) = N\left(\begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$$
$$P(\underline{x} \mid C_2) = N\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

To generate one synthetic data vector $\underline{x}$, first flip a biased coin with probability 0.7 of coming up heads.

- If heads, draw from the Gaussian $P(\underline{x} \mid C_1)$.
- If tails, draw from the Gaussian $P(\underline{x} \mid C_2)$.



Philosophy of generative models:

If we generate synthetic data from the model and it looks a lot like the real data we're trying to model, then we have a good model for our real data.

We can then use the generative model to make optimal inferences, decisions, etc.

B.2) Training phase: Maximum likelihood parameter estimation

Maximize the likelihood of the observed data w.r.t. model parameters.

<u>Example</u>: Two classes with Gaussian class-conditional density with shared covariance

Training data: $\{\underline{x}_n, t_n\}$ $n = 1, \ldots, N$

$t_n = 1$ denotes class C_1

$t_n = 0$ denotes class C_2

Let $P(t_n=1) = P(C_1) = \pi$

$P(t_n=0) = P(C_2) = 1-\pi$

For a data point $\underline{x}_n \in \mathbb{R}^D$,

$$P(\underline{x}_n, C_1) = P(\underline{x}_n | C_1)P(C_1) = N(\underline{x}_n | \underline{\mu}_1, \Sigma) \cdot \pi$$

$$P(\underline{x}_n, C_2) = P(\underline{x}_n | C_2)P(C_2) = N(\underline{x}_n | \underline{\mu}_2, \Sigma) \cdot (1-\pi)$$

Data likelihood for N data points together:

$$\mathcal{L} = P(\{\underline{x}_n, t_n\} | \pi, \underline{\mu}_1, \underline{\mu}_2, \Sigma)$$

$$= \prod_{n=1}^{N} \left(N(\underline{x}_n | \underline{\mu}_1, \Sigma) \cdot \pi\right)^{t_n} \left(N(\underline{x}_n | \underline{\mu}_2, \Sigma) \cdot (1-\pi)\right)^{1-t_n}$$

$$\log \mathcal{L} = \sum_{n=1}^{N} \Big[t_n \log N(\underline{x}_n | \underline{\mu}_1, \Sigma) + t_n \log \pi$$

$$+ (1-t_n) \log N(\underline{x}_n | \underline{\mu}_2, \Sigma) + (1-t_n) \log (1-\pi) \Big]$$

where

$$\log N(\underline{x}_n | \underline{\mu}_k, \Sigma) = -\frac{1}{2}(\underline{x}_n - \underline{\mu}_k)^T \Sigma^{-1} (\underline{x}_n - \underline{\mu}_k)$$

$$-\frac{1}{2} \log |\Sigma| - \frac{D}{2} \log (2\pi)$$

i) Find π

$$\frac{\partial \log \mathcal{L}}{\partial \pi} = \sum_{n=1}^{N} \left[t_n \cdot \frac{1}{\pi} - (1-t_n) \frac{1}{1-\pi} \right] = 0$$

$$(1-\pi) \sum_{n=1}^{N} t_n - \pi \sum_{n=1}^{N} (1-t_n) = 0$$

let N_1 = number of data points from C_1

$$(1-\pi) N_1 - \pi (N - N_1) = 0$$

$$\boxed{\pi = \frac{N_1}{N}}$$

$$N_1 = \sum_{n=1}^{N} t_n$$

$$N_2 = \sum_{n=1}^{N} (1-t_n)$$

ii) Find $\underline{\mu}_1$

$$\frac{\partial \log \mathcal{L}}{\partial \underline{\mu}_1} = \sum_{n=1}^{N} \left(t_n \cdot \cancel{\tfrac{1}{2}} \cdot \cancel{2} \, \Sigma^{-1} (\underline{x}_n - \underline{\mu}_1) \right) = 0$$

$$\Sigma^{-1} \left(\sum_{n=1}^{N} t_n \underline{x}_n \right) = \Sigma^{-1} \left(\underline{\mu}_1 \sum_{n=1}^{N} t_n \right)$$

$$\boxed{\underline{\mu}_1 = \frac{1}{N_1} \sum_{n=1}^{N} t_n \underline{x}_n}$$

Analogously,

$$\boxed{\underline{\mu}_2 = \frac{1}{N_2} \sum_{n=1}^{N} (1-t_n) \underline{x}_n}$$

iii) Find Σ

Focusing only on terms that involve Σ,

$$\log \mathcal{L} = \sum_{n=1}^{N} \Big[t_n \Big(-\tfrac{1}{2} \mathrm{Tr}\big(\Sigma^{-1} (\underline{x}_n - \underline{\mu}_1)(\underline{x}_n - \underline{\mu}_1)^T\big) - \tfrac{1}{2} \log |\Sigma| \Big)$$

$$+ (1 - t_n) \Big(-\tfrac{1}{2} \mathrm{Tr}\big(\Sigma^{-1} (\underline{x}_n - \underline{\mu}_2)(\underline{x}_n - \underline{\mu}_2)^T\big) - \tfrac{1}{2} \log |\Sigma| \Big) \Big]$$

$$\frac{\partial \log \mathcal{L}}{\partial \Sigma} = \sum_{n=1}^{N} \Big[t_n \Big(-\tfrac{1}{2} \cdot -\Sigma^{-1} (\underline{x}_n - \underline{\mu}_1)(\underline{x}_n - \underline{\mu}_1)^T \Sigma^{-1} - \tfrac{1}{2} \Sigma^{-1} \Big)$$

$$+ (1 - t_n) \Big(-\tfrac{1}{2} \cdot -\Sigma^{-1} (\underline{x}_n - \underline{\mu}_2)(\underline{x}_n - \underline{\mu}_2)^T \Sigma^{-1} - \tfrac{1}{2} \Sigma^{-1} \Big) \Big]$$

$$= [0]$$

Rearranging yields

$$\tfrac{1}{2} \sum_{n \in C_1} (\underline{x}_n - \underline{\mu}_1)(\underline{x}_n - \underline{\mu}_1)^T - \tfrac{1}{2} N_1 \Sigma$$

$$+ \tfrac{1}{2} \sum_{n \in C_2} (\underline{x}_n - \underline{\mu}_2)(\underline{x}_n - \underline{\mu}_2)^T - \tfrac{1}{2} N_2 \Sigma = [0]$$

$$\boxed{\begin{aligned} &\Sigma = \frac{N_1}{N} S_1 + \frac{N_2}{N} S_2, \text{ where} \\ &S_1 = \frac{1}{N_1} \sum_{n \in C_1} (\underline{x}_n - \underline{\mu}_1)(\underline{x}_n - \underline{\mu}_1)^T \\ &S_2 = \frac{1}{N_2} \sum_{n \in C_2} (\underline{x}_n - \underline{\mu}_2)(\underline{x}_n - \underline{\mu}_2)^T \end{aligned}}$$

B.3) Test phase: Assigning a new data point to a class

$$\hat{k} = \underset{k}{\operatorname{argmax}} \; P(C_k \mid \underline{x})$$

$$= \underset{k}{\operatorname{argmax}} \; \frac{P(\underline{x} \mid C_k) P(C_k)}{P(\underline{x})}$$

$$= \underset{k}{\operatorname{argmax}} \; P(\underline{x} \mid C_k) P(C_k)$$

$$= \underset{k}{\operatorname{argmax}} \left(\log P(\underline{x} \mid C_k) + \log P(C_k) \right)$$

$$= \underset{k}{\operatorname{argmax}} \underbrace{\left(\underline{\mu}_k^T \Sigma^{-1} \underline{x} - \frac{1}{2} \underline{\mu}_k^T \Sigma^{-1} \underline{\mu}_k + \log P(C_k) \right)}_{\text{Call this } a_k(\underline{x})}$$

What do the decision boundaries look like in $\underline{x}$ space?

C) Hyperplanes

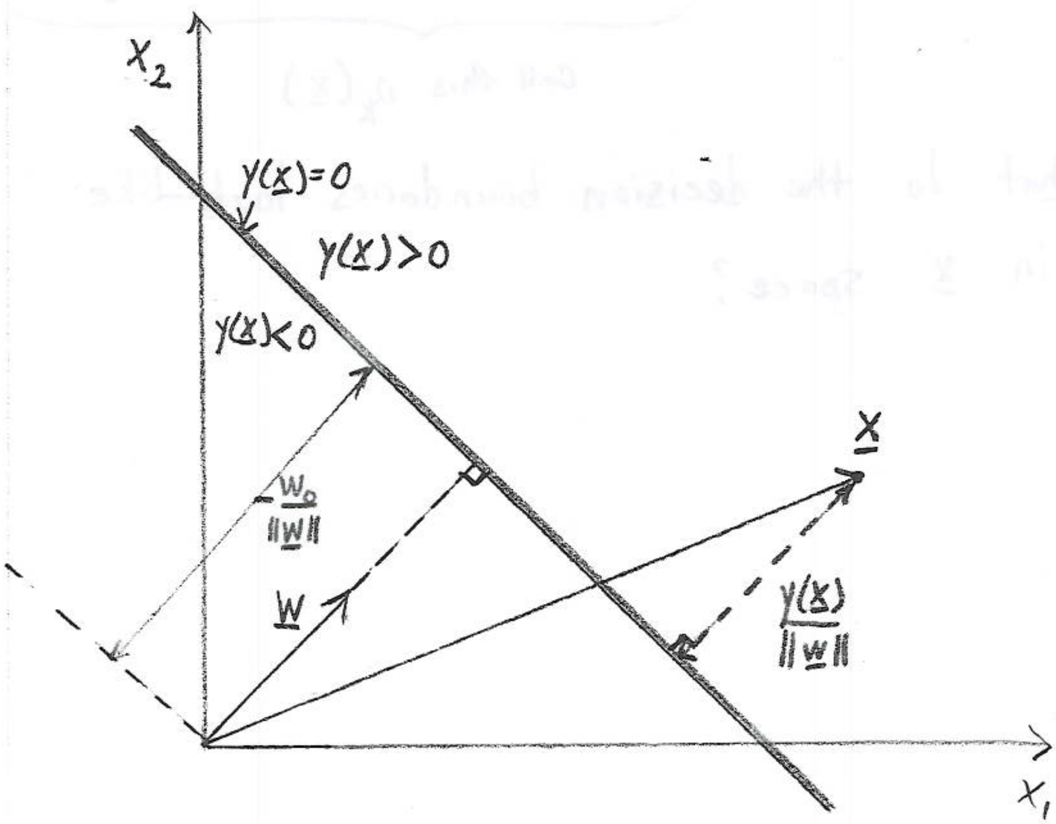
A hyperplane is the D-dimensional generalization of a line in 2-dim space and a plane in 3-dim space.

A hyperplane is defined as the set of all $\underline{x}$ such that

$$y(\underline{x}) = \underline{w}^T \underline{x} + w_0 = 0 \qquad (1)$$

$\underline{w}$ determines the direction of the hyperplane

w_0 determines its offset from the origin.



Facts:

i) $\underline{w}$ is orthogonal to hyperplane

Consider two points $\underline{x}_A$ and $\underline{x}_B$ which lie on hyperplane.

$$y(\underline{x}_A) = y(\underline{x}_B) = 0$$

$$\underline{w}^T \underline{x}_A + w_0 = \underline{w}^T \underline{x}_B + w_0$$

$$\underline{w}^T \underbrace{(\underline{x}_A - \underline{x}_B)}_{\text{vector lying in hyperplane}} = 0$$

$\Rightarrow$ $\underline{w}$ is orthogonal to any vector lying in hyperplane.

ii) Normal distance from origin to hyperplane is $-\dfrac{w_0}{\|\underline{w}\|}$

Let $\underline{x}$ be a point on hyperplane $\Rightarrow$ $\underline{w}^T \underline{x} + w_0 = 0$

Normal distance is projection of $\underline{x}$ onto $\underline{w}$

$$\left(\frac{\underline{w}}{\|\underline{w}\|}\right)^T \underline{x} = -\frac{w_0}{\|\underline{w}\|} \quad //$$

iii) Normal distance from any point $\underline{x}$ to hyperplane is $\frac{y(\underline{x})}{\|\underline{w}\|}$.

Project $\underline{x}$ onto $\underline{w}$, then subtract $-\frac{w_0}{\|\underline{w}\|}$

$$\left(\frac{\underline{w}}{\|\underline{w}\|}\right)^T \underline{x} + \frac{w_0}{\|\underline{w}\|} = \frac{y(\underline{x})}{\|\underline{w}\|}. \quad //$$

D) Linear Decision Boundaries

From p.9, a point $\underline{x}$ is assigned to class C_k if $a_k(\underline{x}) > a_j(\underline{x})$ for all $j \neq k$.

Thus, the decision boundary between class C_k and class C_j is given by $a_k(\underline{x}) = a_j(\underline{x})$.

Let $a_k(\underline{x}) = \underline{w}_k^T \underline{x} + w_{k0}$, where

$$\underline{w}_k = \Sigma^{-1} \underline{\mu}_k$$

$$w_{k0} = -\frac{1}{2} \underline{\mu}_k^T \Sigma^{-1} \underline{\mu}_k + \log P(C_k)$$

The decision boundary is thus

$$(\underline{w}_k - \underline{w}_j)^T x + (w_{k0} - w_{j0}) = 0$$

Note that this takes the same form as (1), so the decision boundary is a (D-1) dimensional hyperplane in $\mathbb{R}^D$.

Appendix: Useful matrix properties

A symmetric

$$\frac{d}{d\underline{x}} \underline{x}^T A \underline{x} = (A + A^T)\underline{x} \overset{\downarrow}{=} 2A\underline{x}$$

vector

$$\frac{d}{dX} \operatorname{Tr}(X^{-1}A) = -X^{-T}A^T X^{-T}$$

matrix

$$\frac{d}{dX} \log|X| = X^{-T}$$

matrix

$$\operatorname{Tr}(ABC\ldots) = \operatorname{Tr}(BCD\ldots A) = \operatorname{Tr}(CD\ldots AB)$$

A good reference is:

http:// www.ee.ic.ac.uk/hp/staff/dmb/matrix/intro.html

or

simply google "matrix reference manual".