

BIC 2007

2nd ISCV Thematic Workshop: Biologically-Inspired Computing
December 3-7, 2007 — Valparaíso Complex Systems Institute, Chile.

Ensemble methods: putting learners to work together



Ricardo Ñanculef (rnancu@inf.utfsm.cl)

*INCA – Grupo de Inteligencia Computacional Aplicada
Departamento de Informática. Universidad Técnica Federico Santa María.*

Rodrigo Salas F. (rodrigo.salas@uv.cl)

*INCA – Grupo de Inteligencia Computacional Aplicada
IRIS – Grupo de Investigadores en Reconocimiento, Inteligencia y del Saber*





- Machine learning concepts
- Ensemble methods in supervised learning
- Ensemble methods in unsupervised learning



- How to build machines that learn?
- Learning: to improve from experience
- Experience: examples, previous situations
- Models of learning: a gateway to the understanding of natural learning?
- The inverse process is common in practice



- Systems trained instead of explicitly programmed to solve a specific task
- Examples of applications:
 - Interfaces and customized software
 - Biometric recognition and security
 - Biomedical engineering and bioinformatics
 - Robotics



Supervised Learning Paradigm

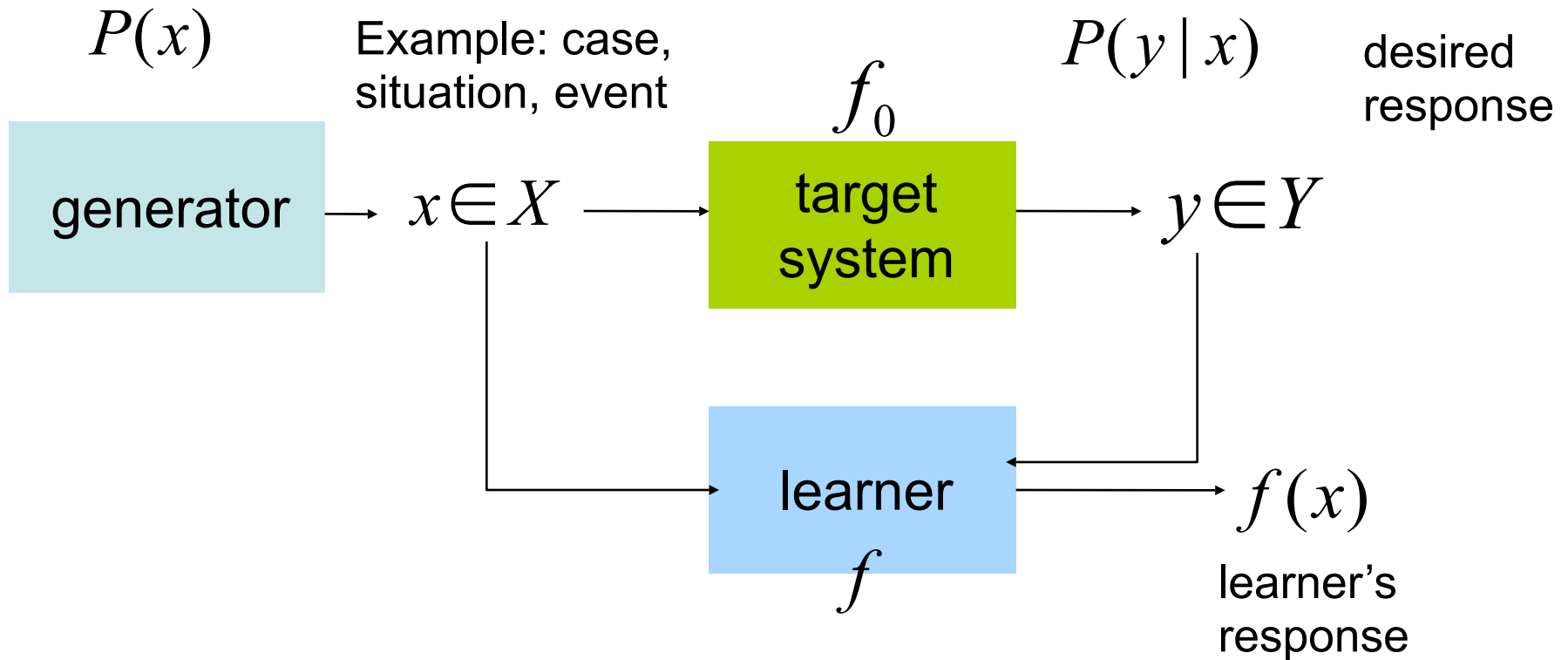
- Examples are input - output pairs
- Desired or ideal response is provided
- **System learns with a “teacher”**
- The learner has to imitate the teacher
- Typical tasks:
 - Classification
 - Regression
 - Ranking



- You have not an explicit “desired answer”
- Typical tasks: to find common configurations, unusual configurations or relations between objects.
- For example: from a set of queries we can conclude “If he searches for a topic A, he also searches for a topic B”



Supervised Learning Model



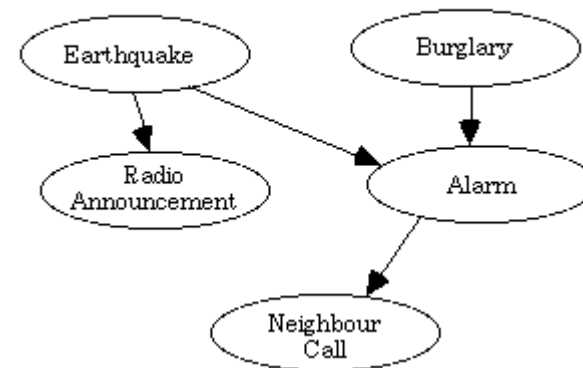
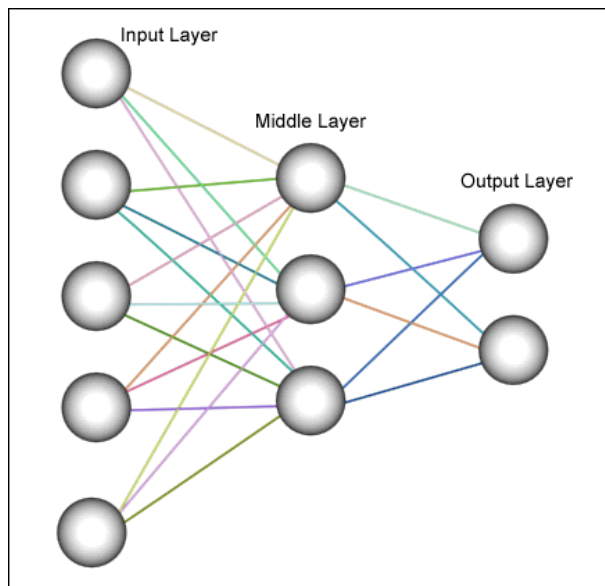


Key elements of the model

- Observations Distributions: $P(x)$, $P(y | x)$
- Training Set: $S = \{z_1, z_2, \dots, z_n\}$ where $z_i = (x_i, y_i)$

Given S , the learner builds a hypothesis f

- Hypothesis Space: H
- Learning map: $f: P(S) \rightarrow H$



$$A_1 \wedge A_2 \wedge \dots \Rightarrow B_1$$

$$A_1 \wedge A_3 \wedge \dots \Rightarrow B_2$$



- **Error-driven Learning**

for a desired answer y and a true answer y' , a **loss function Q** allows the learner to measure the error

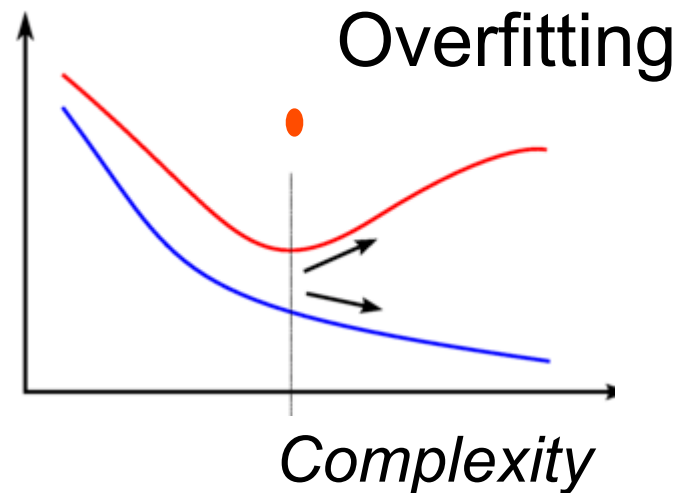
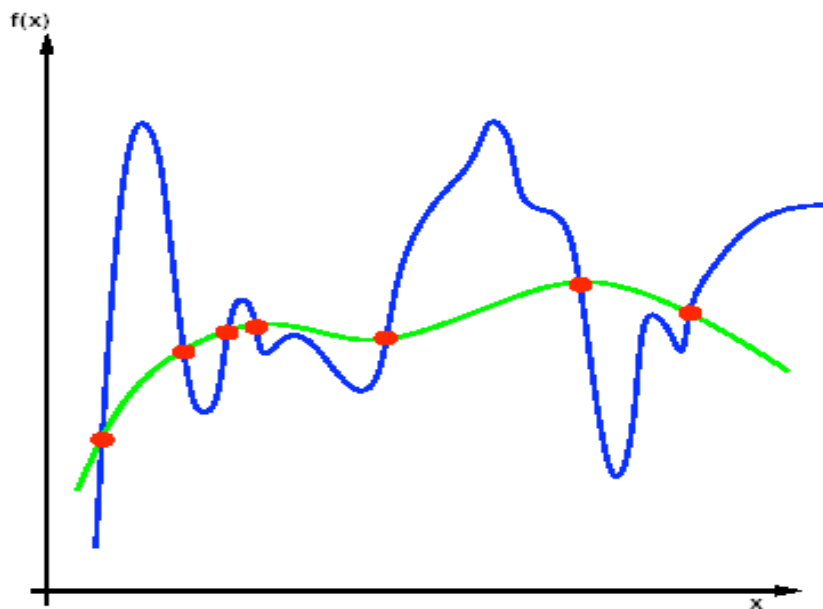
$$Q(y, y') = Q(y, f(x)) = Q(z, f)$$

- **Prediction Error** $R[f] = \int_{\mathcal{Z}} Q(f, z) d\mu(z)$

$\mu(x, y)$ is the **unknown (probability) measure** according to which the examples $z=(x,y)$ appear

- **Empirical or Observed Error** $R_S[f] = \frac{1}{n} \sum_{i=1}^n Q(f, z_i)$

In principle, there are infinite hypothesis consistent with the examples!



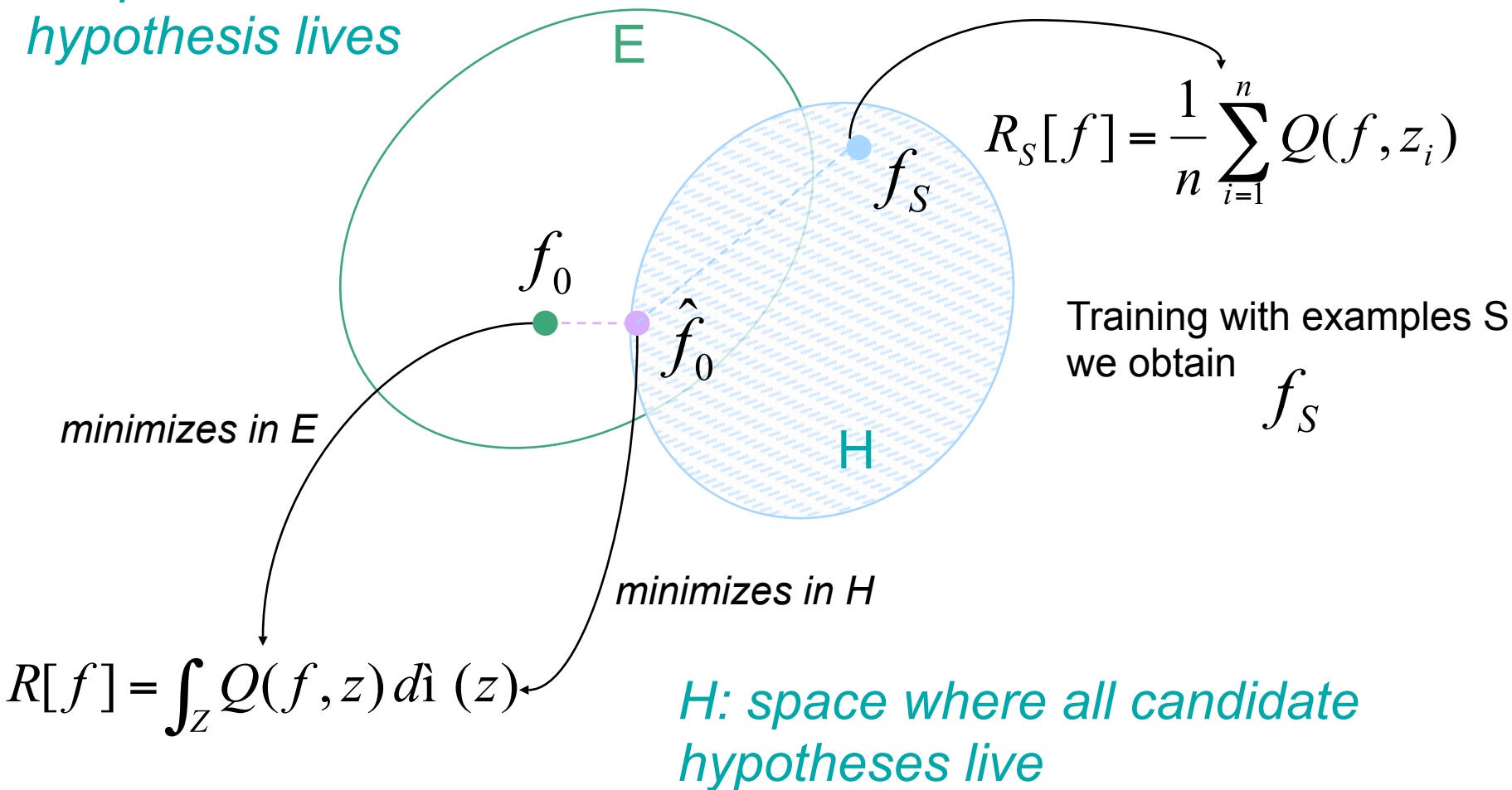
zero training error does not guarantee generalization!



Effect of a finite training set

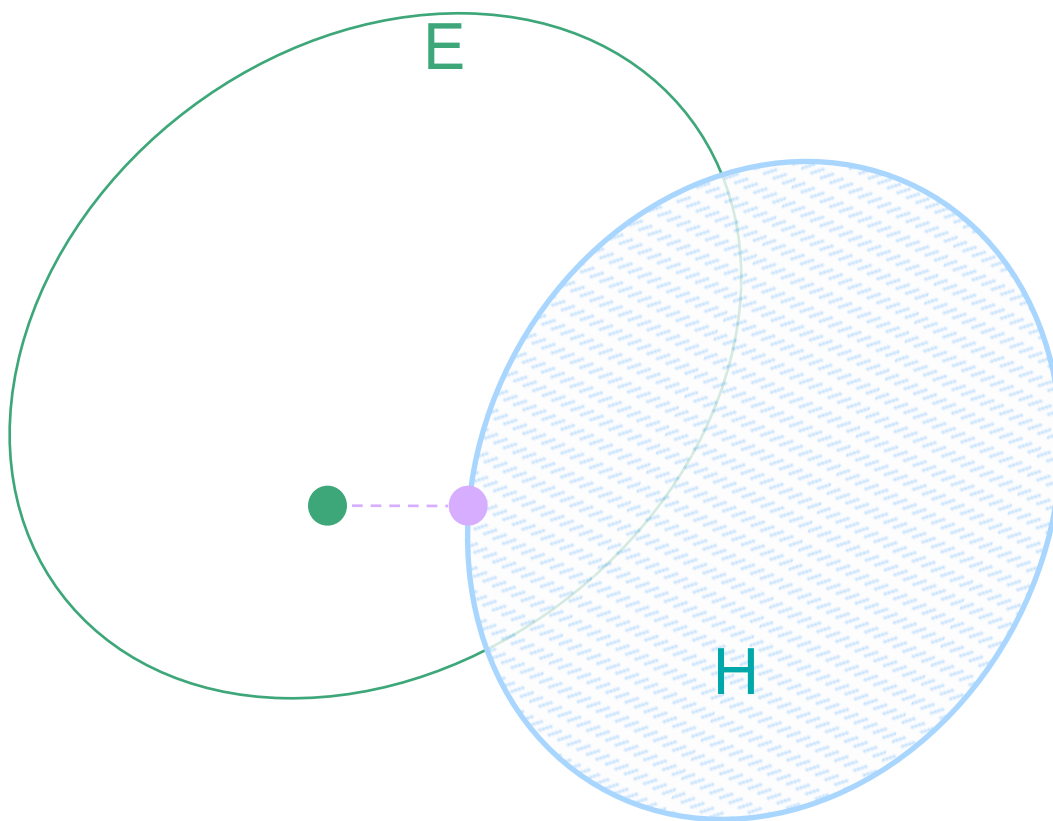
U: space where the real hypothesis lives

minimizes in H





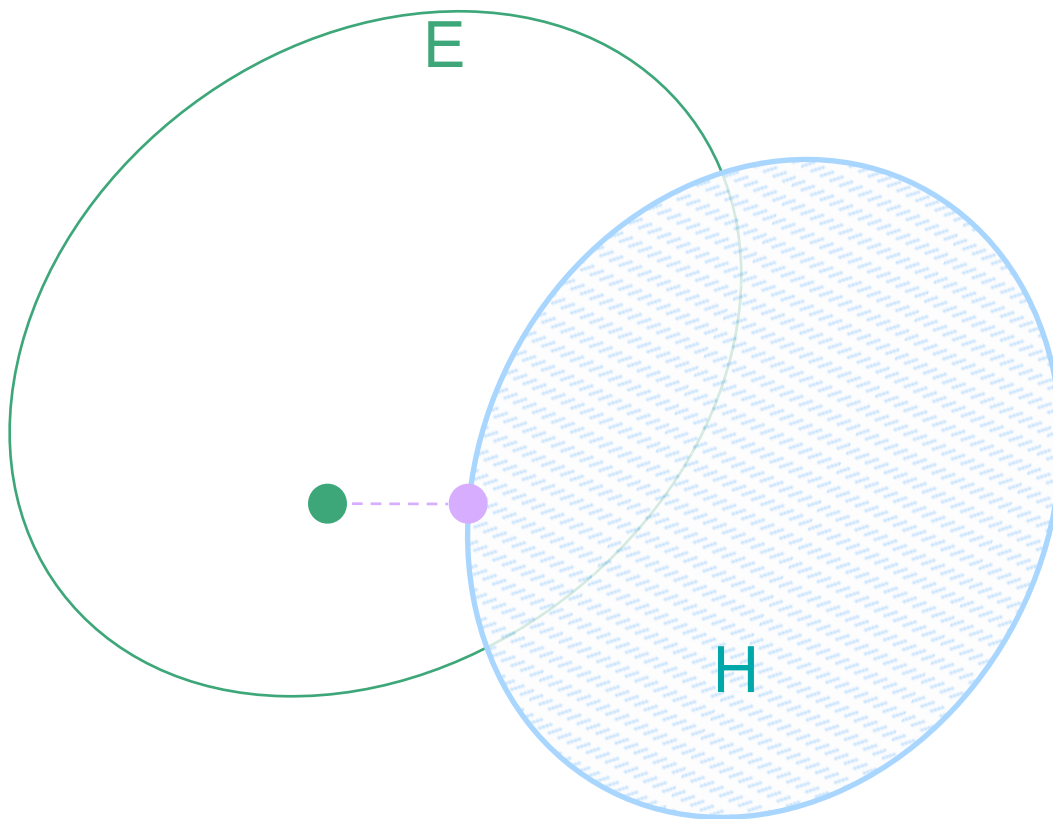
Effect of different training sets



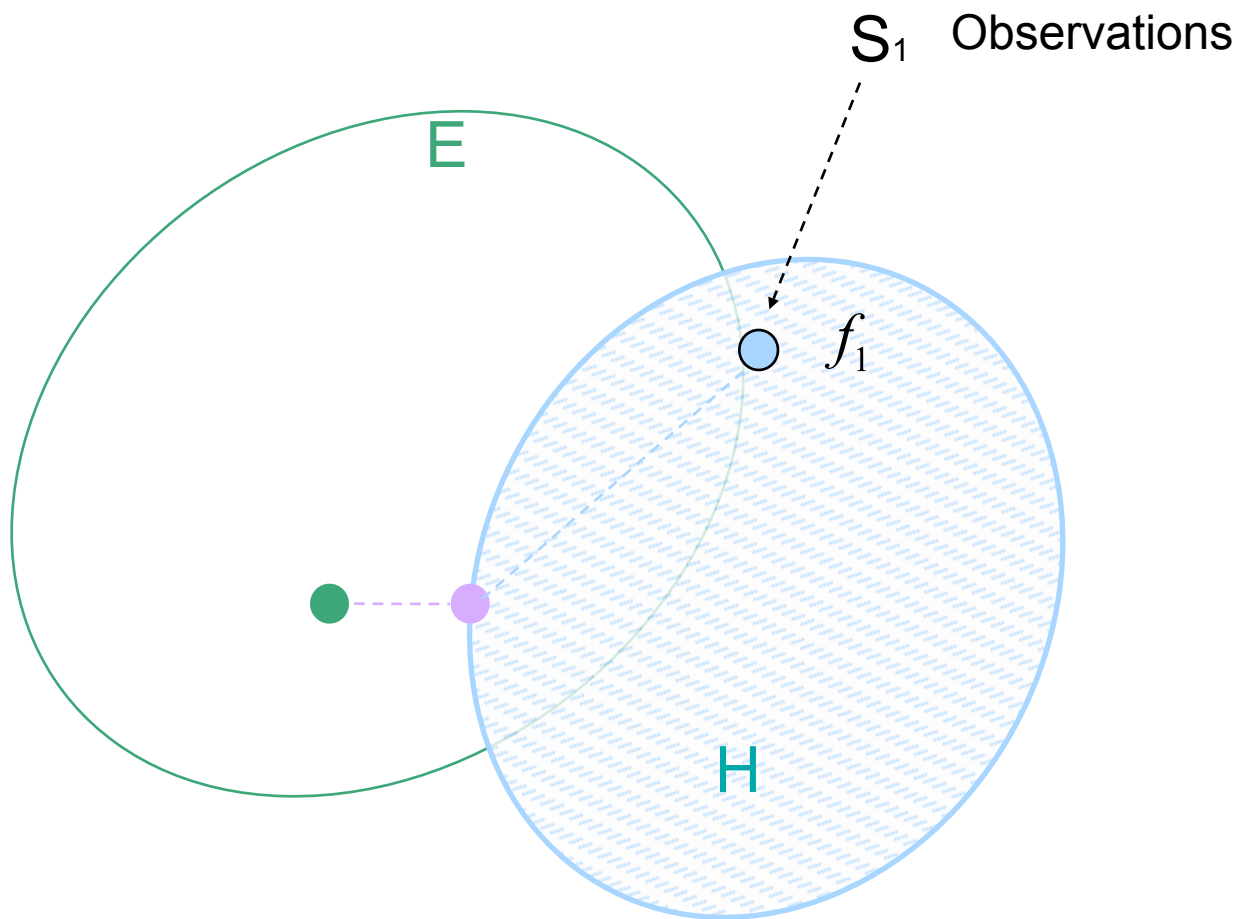


Effect of different training sets

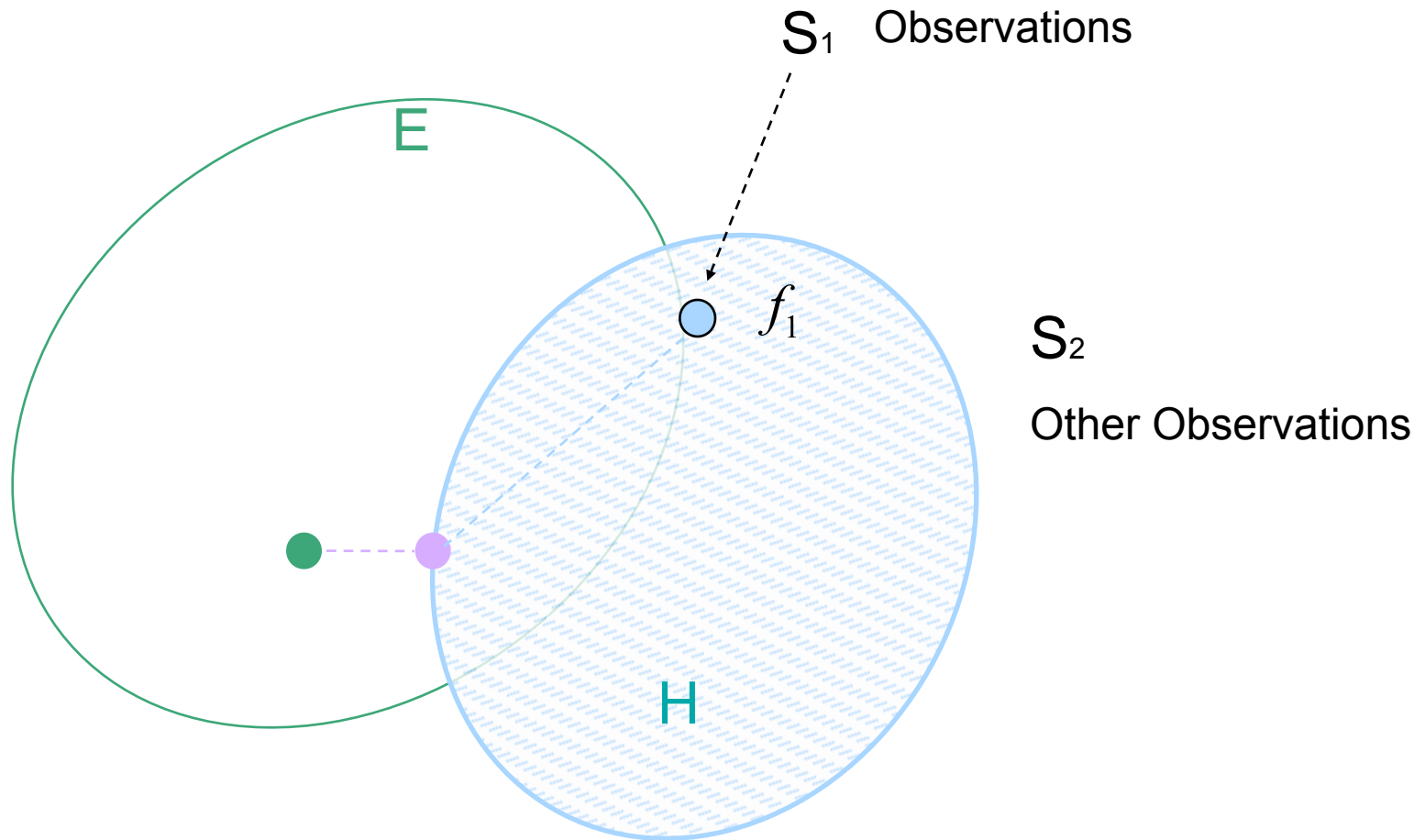
S_1 Observations



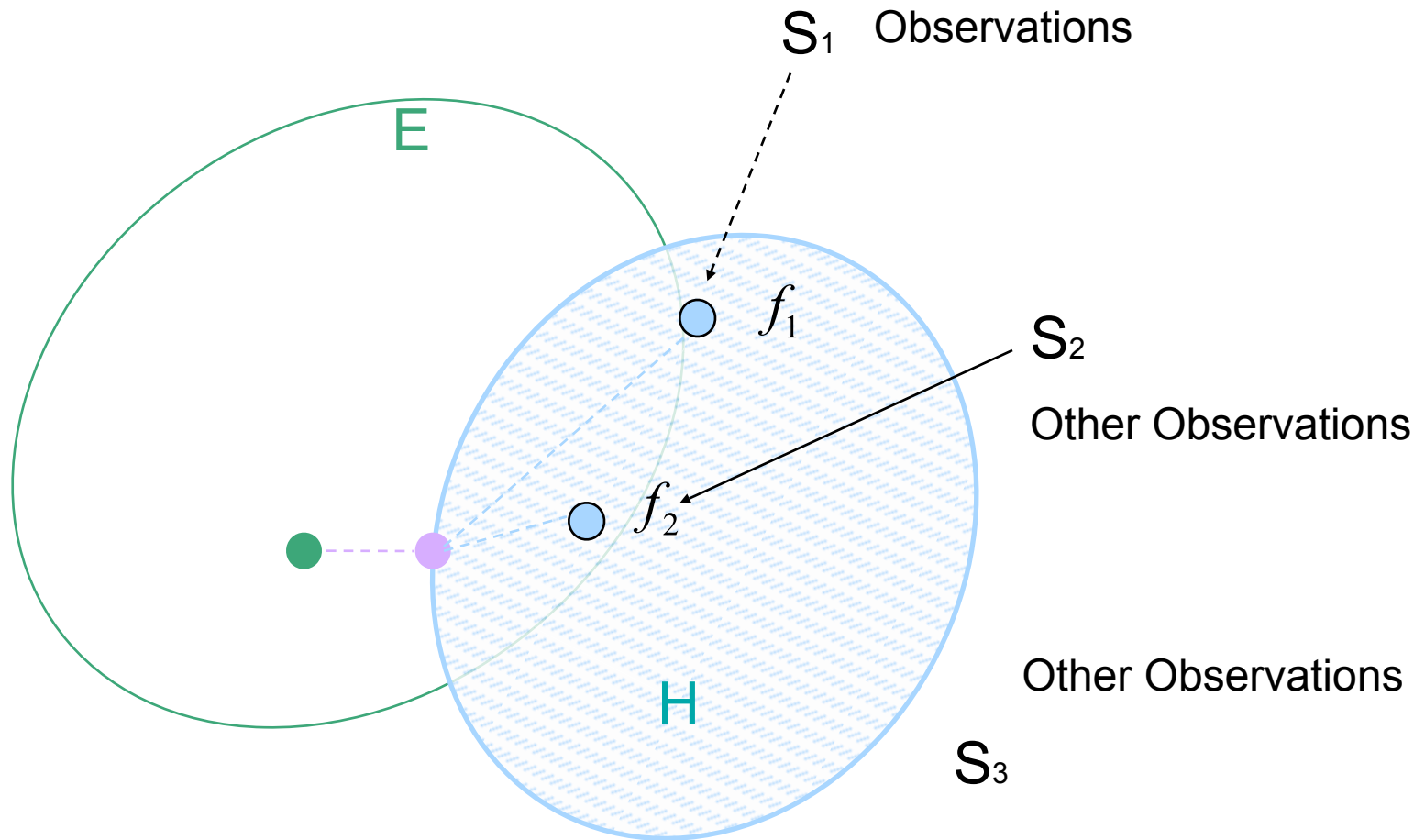
Effect of different training sets



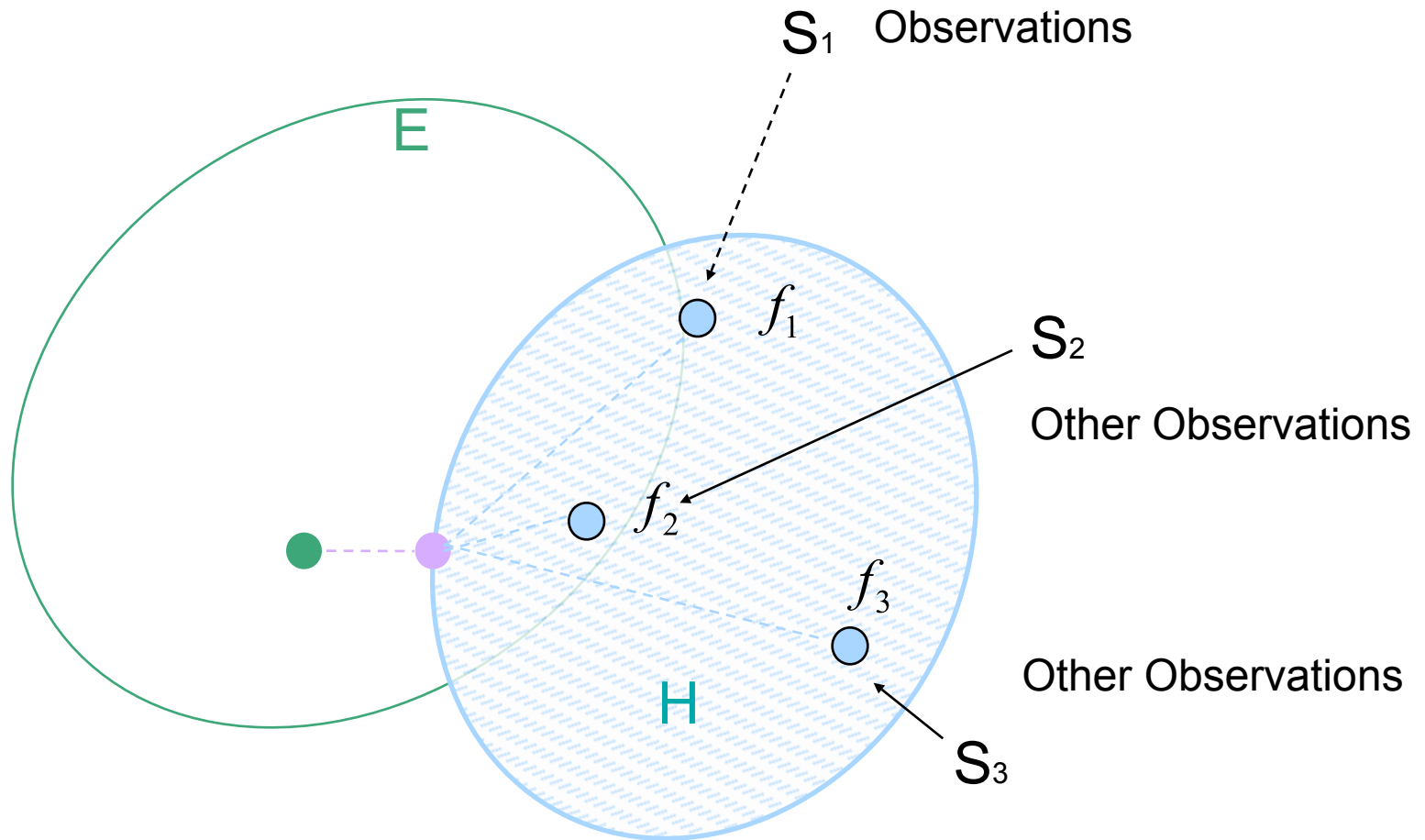
Effect of different training sets



Effect of different training sets

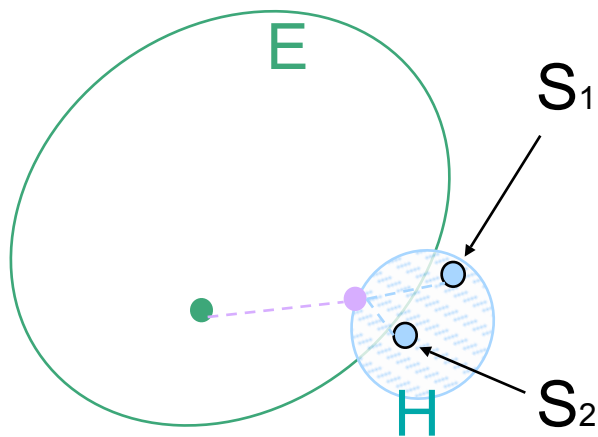


Effect of different training sets

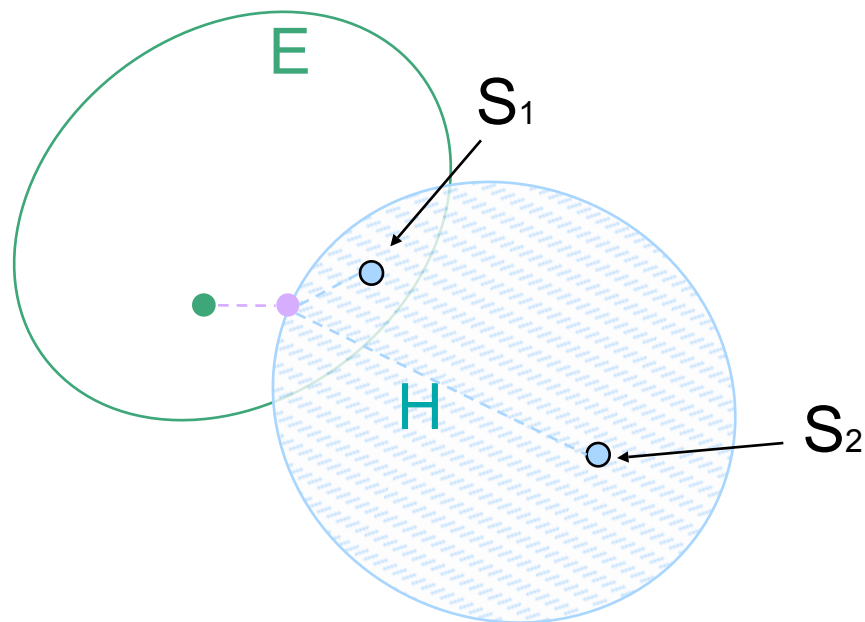


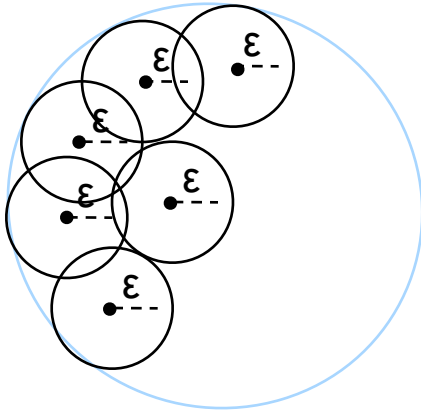
The effect of the hypothesis space “size”

A smaller hypotheses space



A greater hypotheses space





$$N(\varepsilon, S)$$

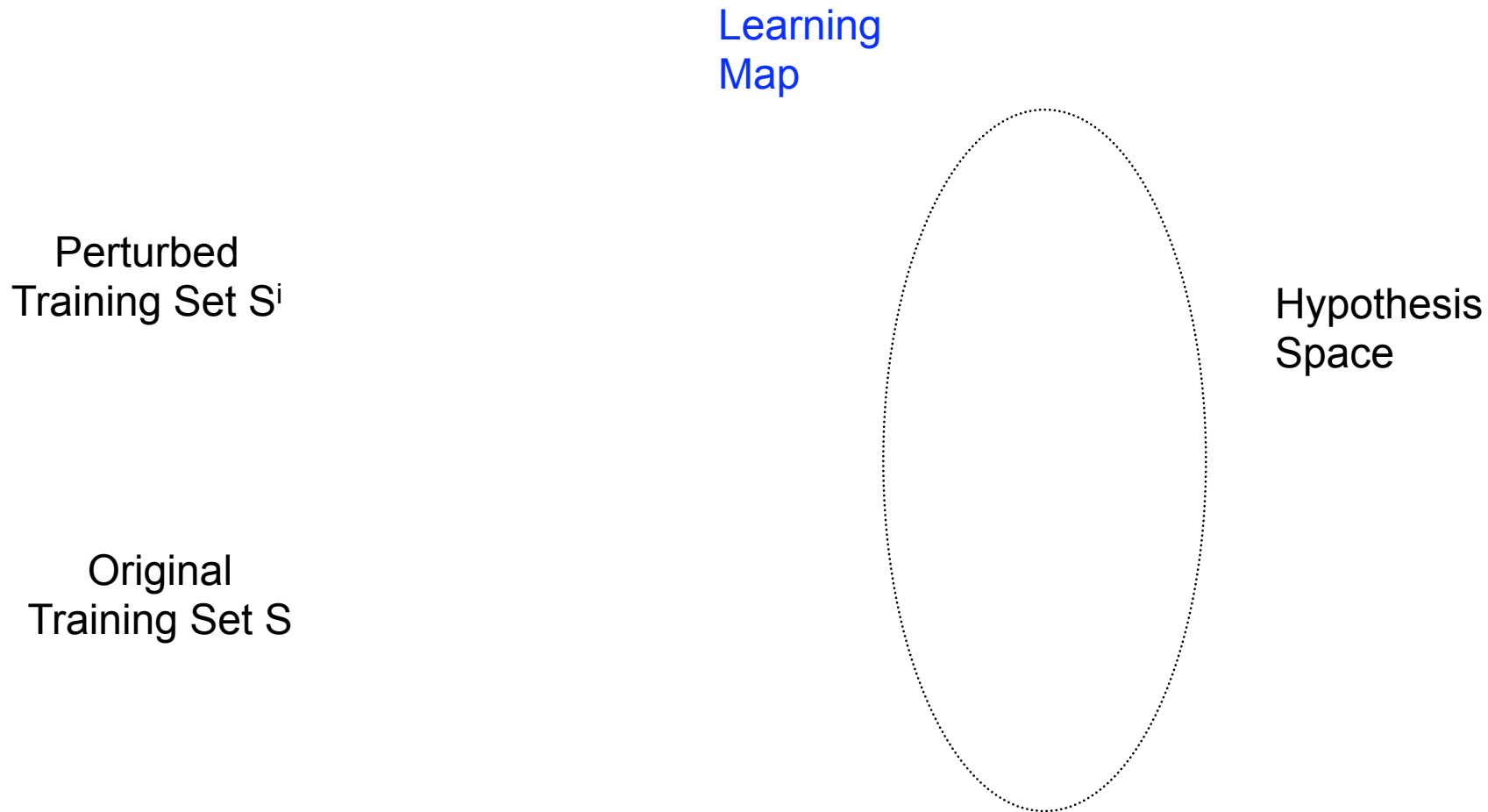
Number of balls needed to cover the hypothesis space

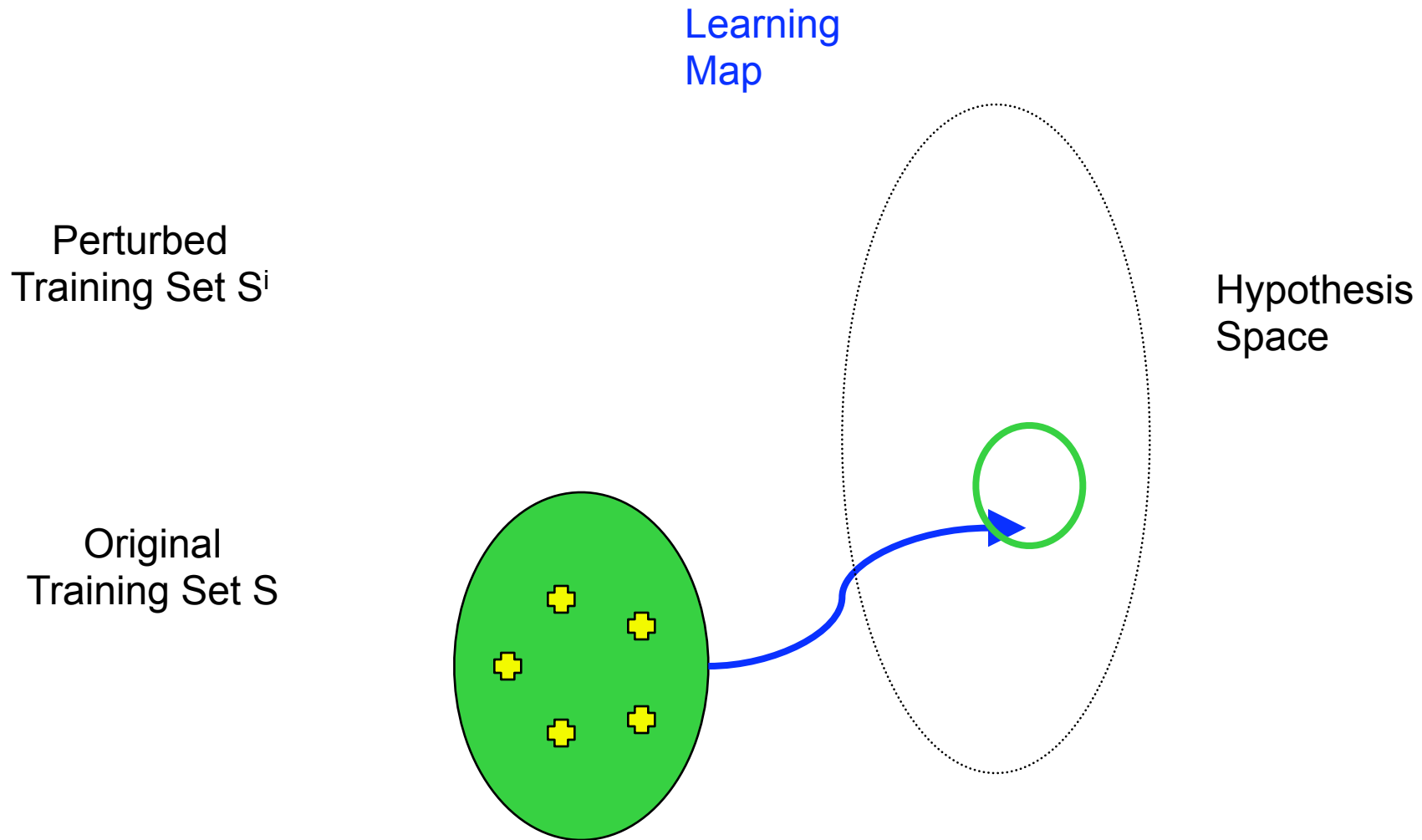
- For an algorithm minimizing the observed error generalization is equivalent to have (for all ε)

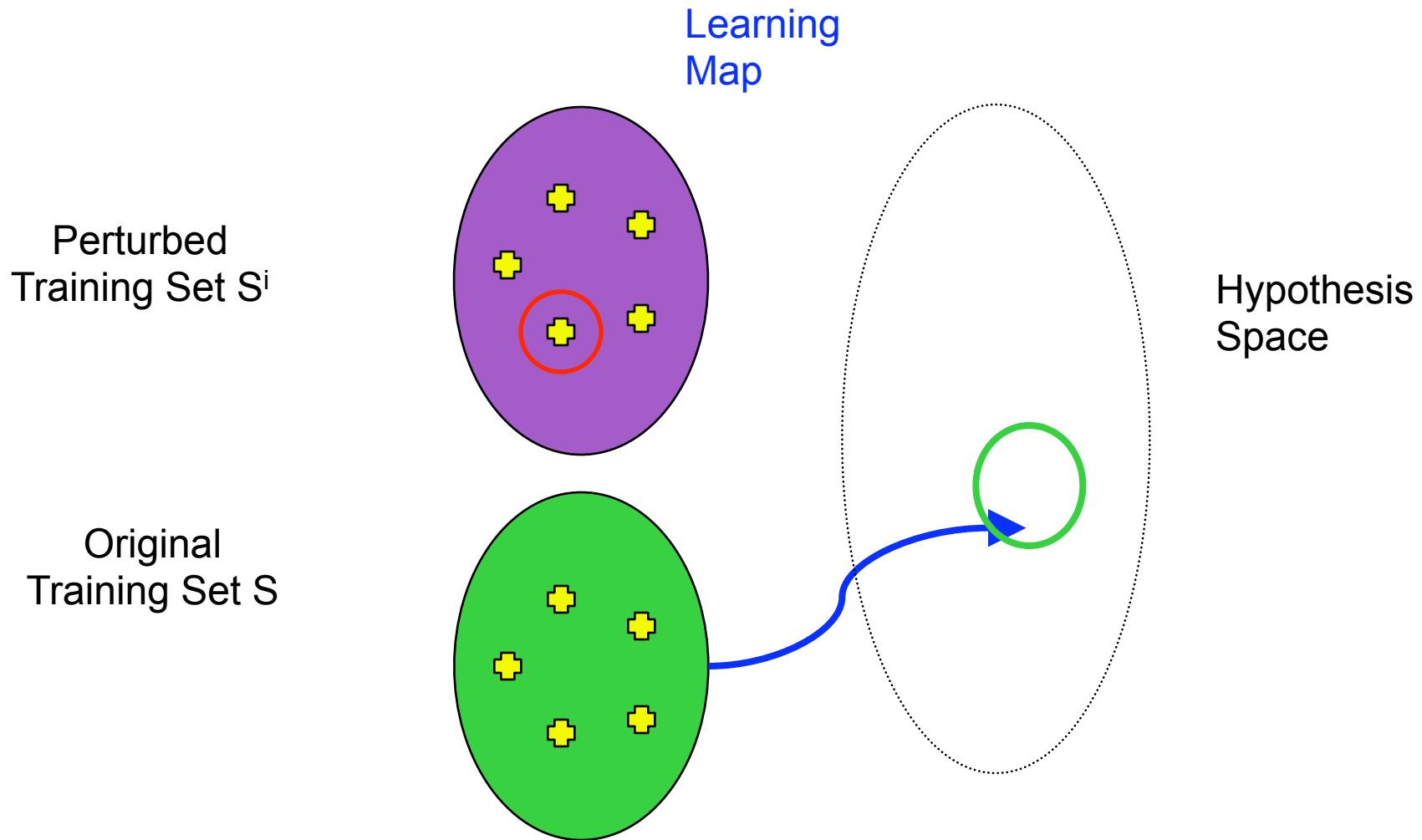
$$\lim_{n \rightarrow \infty} \frac{H(\varepsilon, n)}{n} = 0$$

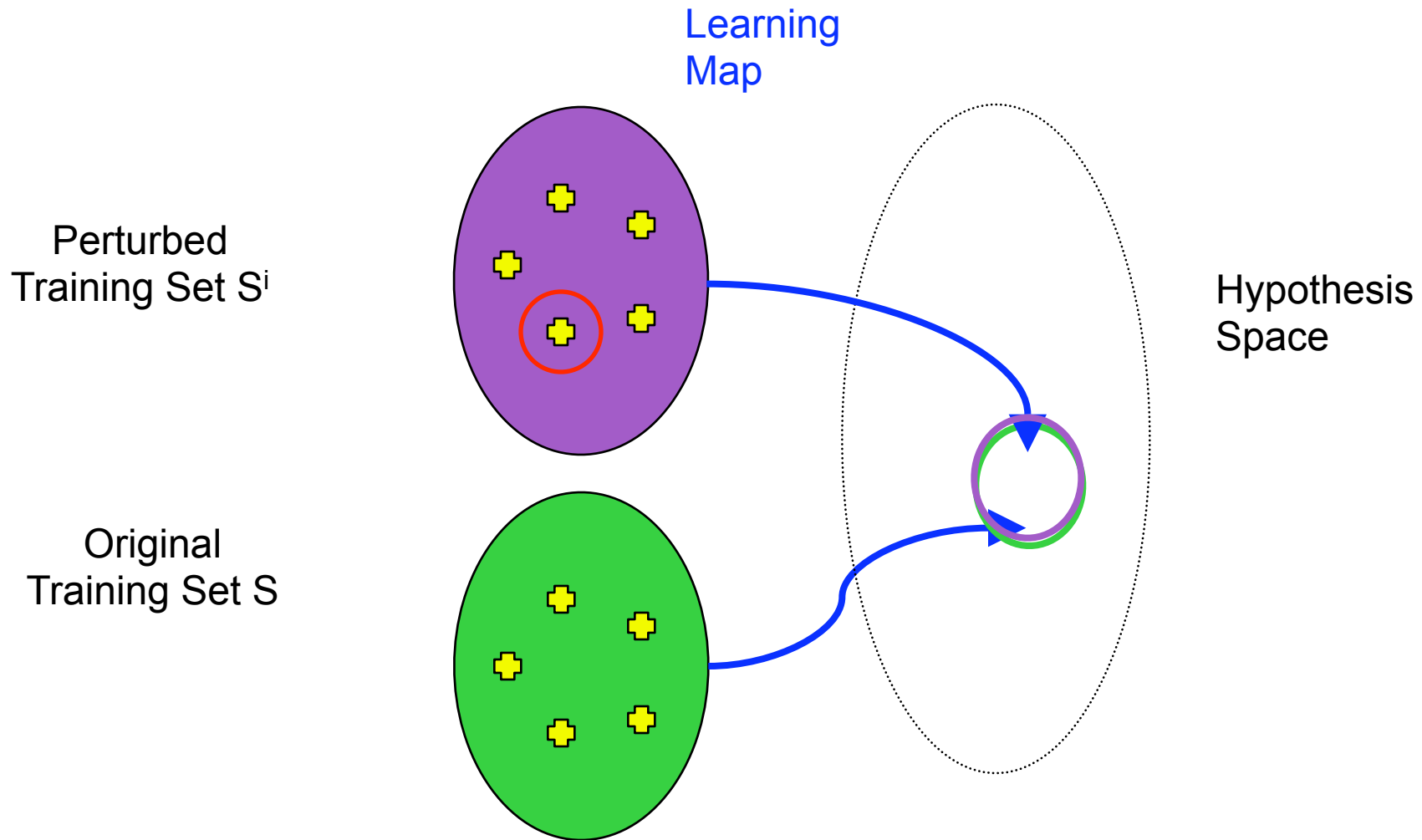


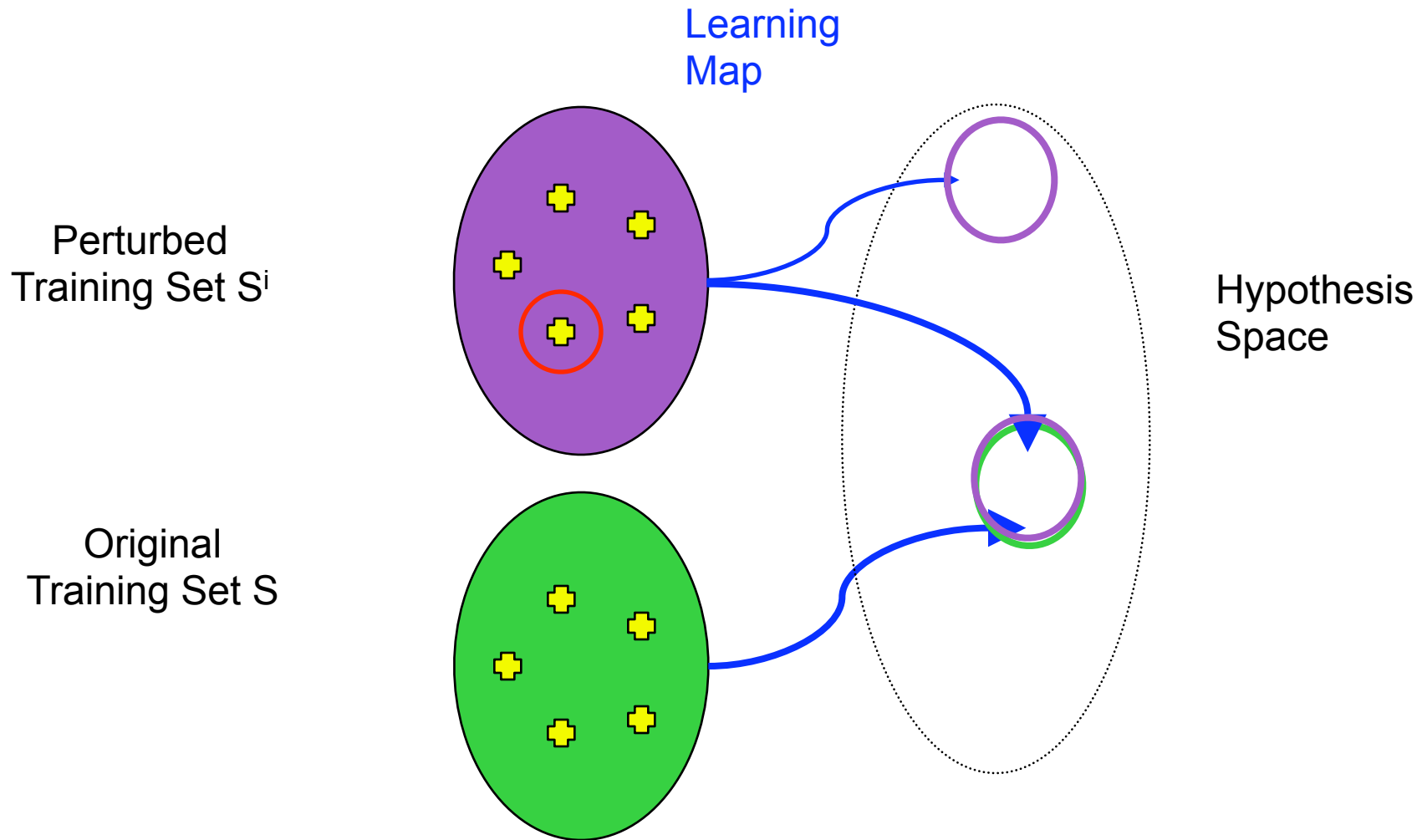
Stability characterization





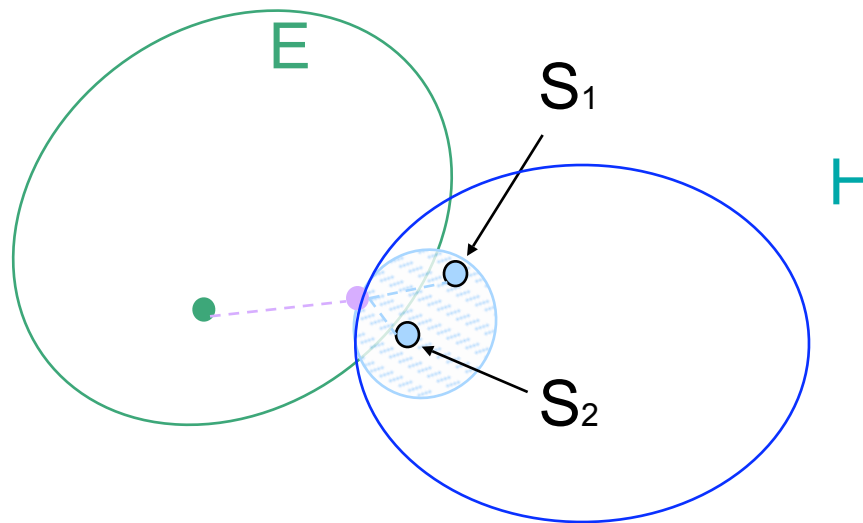




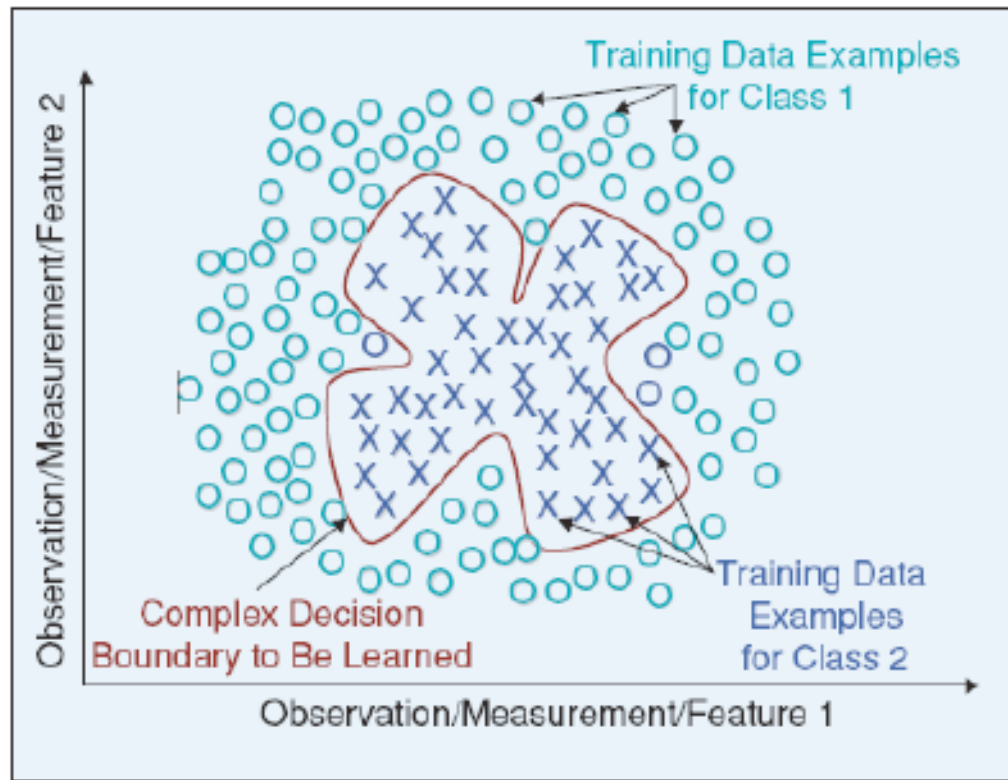


- Recent theoretical findings support the hypothesis that

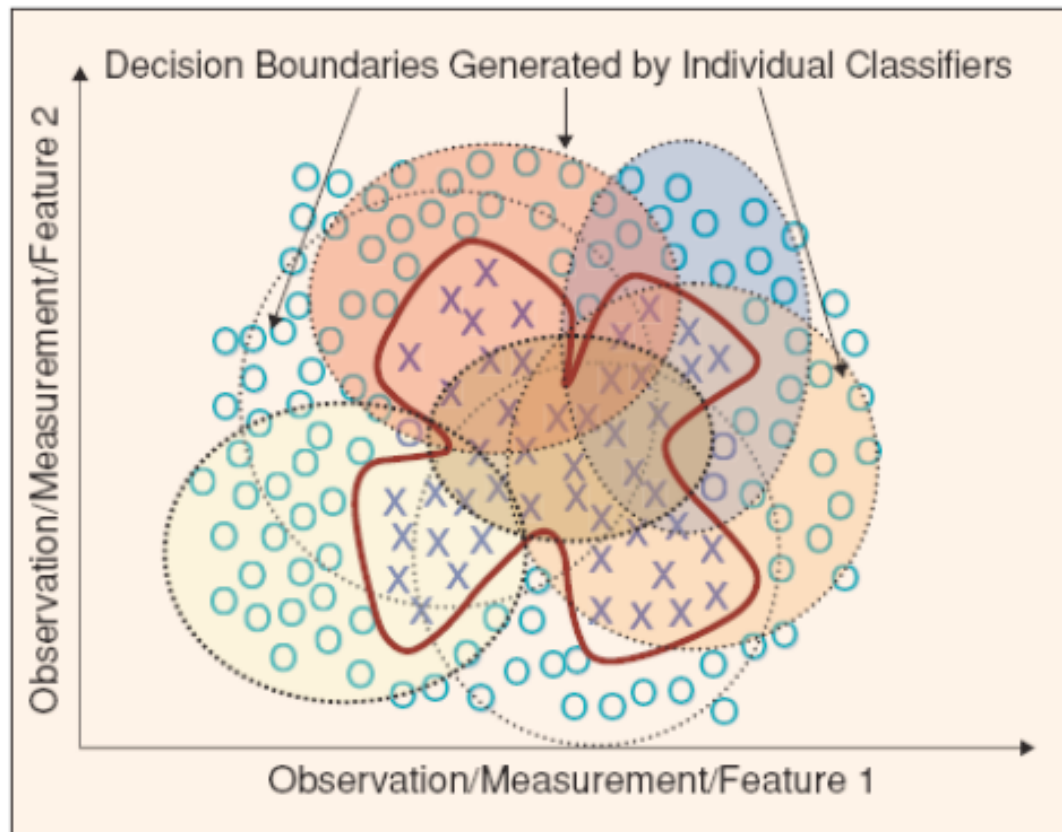
**A learner generalizes if (and only if)
this is stable with respect to training.
(focus on the learning process)**



- Idea: Solving a task by combining a set of simple solutions



- Idea: Solving a task by combining a set of simple solutions





- Complex solutions by reusing simple models
- Natural distribution of the learning task
- Effective handling of overfitting
- Heterogenous data: sources and types
- Easy deal with dynamic environments



- Generate a set of solutions
 - Sequentially? Simultaneously?
 - How to train each learner
- Combine solutions
 - Select only some solutions?
 - How to combine de decisions?
- Some examples: Boosting, Bagging, Mixtures of Experts



- Freund and Schapire, 1996.
- Learners are trained one after the other such that
 - We grow a solution at demand.
 - New solutions are generated to focus on the instances where the previous solutions are bad: **explicit cooperation!**



Let S be the training set and $D_i = 1/m$

1. Get a new training set sampling S with D
2. Train a base learner to get f_t
3. Compute the weighted error $\epsilon_t = \sum_i D_i \times I(y_i \neq f_t(x_i))$
4. Compute the importance of f_t $w_t = (1 - \epsilon_t) / \epsilon_t$
5. Update the sampling distribution

$$D_{t+1}(x_i) = D_t(x_i) \times \exp(-y_i f_t(x_i))$$

- Relation between individual behaviors and the group behavior:

$$P_S(yF(x) \leq 0) \leq \prod_t 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$$

- The error decreases exponentially with the number of learners but apparently also grows the complexity of the solution: does boosting generalize?



- In practice, prediction error continues decreasing after training error reaches zero.
- Schapire, Freund, Bartlett and Lee, 1997:
 - effective complexity of the ensemble hypothesis is not dependent of the number of learners.
- Kutin and Niyogi, 2001: Boosting is a stability preserving algorithm



- Breiman, 1996.
- Sample m sets of p examples from S (with replacement): S_1, S_2, \dots, S_m
- Train a learner on each S_i and obtain a sequence of m solutions f_1, f_2, \dots, f_m



Bagging: why does it work?

- Several works report that neither boosting nor other more sophisticated algorithms are better than bagging with statistical significance
- Why does bagging work?



Bagging: stabilization properties

- Grandvalet 2000, 2001, 2004, 2005: Bagging equalizes the influence of the examples in the learning process.
- Influence of highly influential points is reduced
- This could explain the success and failure of bagging in very general scenarios.



New algorithms, new questions

- After nineties: a lot of new algorithms and very creative ideas to build ensembles.
- Many of the new ideas are difficult to analyze in a mathematical framework using the machine learning principles we have discussed in this talk.
- A key design principle: diversity



.. If the ensemble members are imperfect, they should be different so that at least some of them are correct where the others are wrong.

We call this loosely specified property diversity, and set off to explore why and how it works ...

L. Kuncheva, "Diversity in multiple classifier systems" (Editorial), *Information Fusion* 6(1), 2005



Diversity: Continuous versus Discrete Cases

- In classification advances have been limited by the discrete nature of the problem
- Pairwise versus non-pairwise measures have been studied
- Kuncheva 2003: analyzed 10 measures of diversity and none of them are clearly related with ensemble performance.



Diversity: Continuous versus Discrete Cases

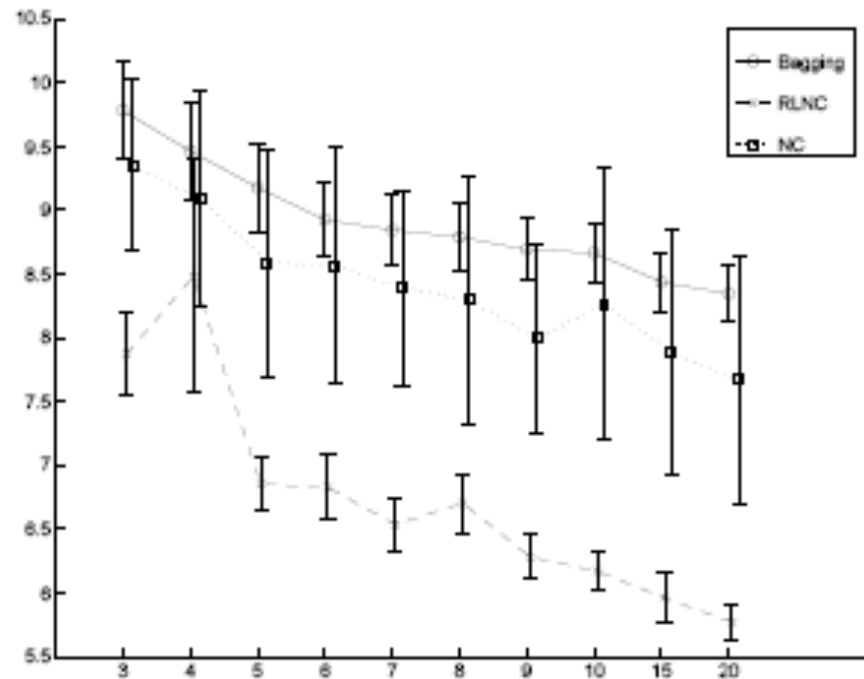
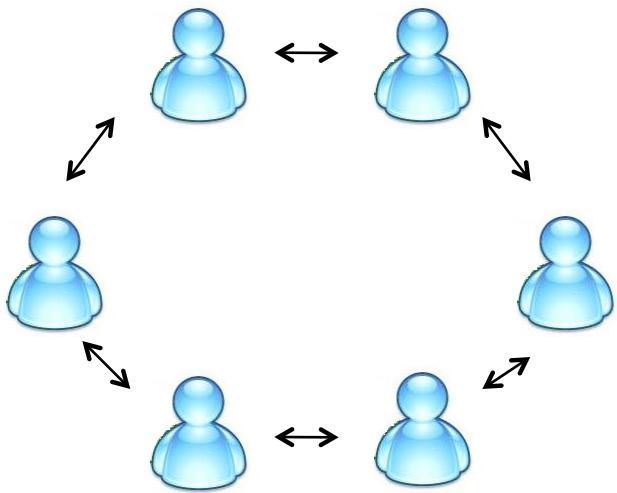
- When the target responses the algorithm attempts to learn are continuous, diversity can be mathematically defined

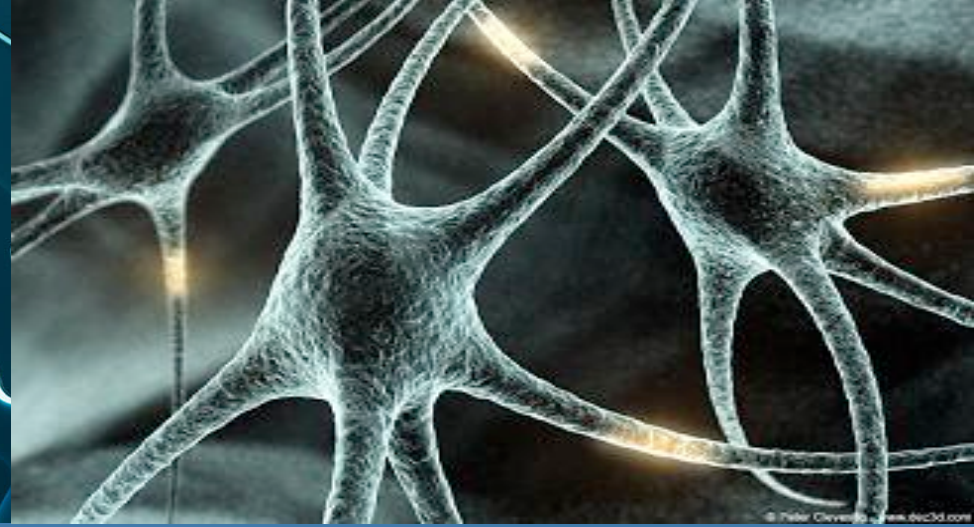
$$(y - F_{ens})^2 = \sum_i w_i (y - f_i)^2 - \sum_i w_i (f_i - F_{ens})^2$$

Average of individual errors

Diversity component

Bagging with Local Diversity





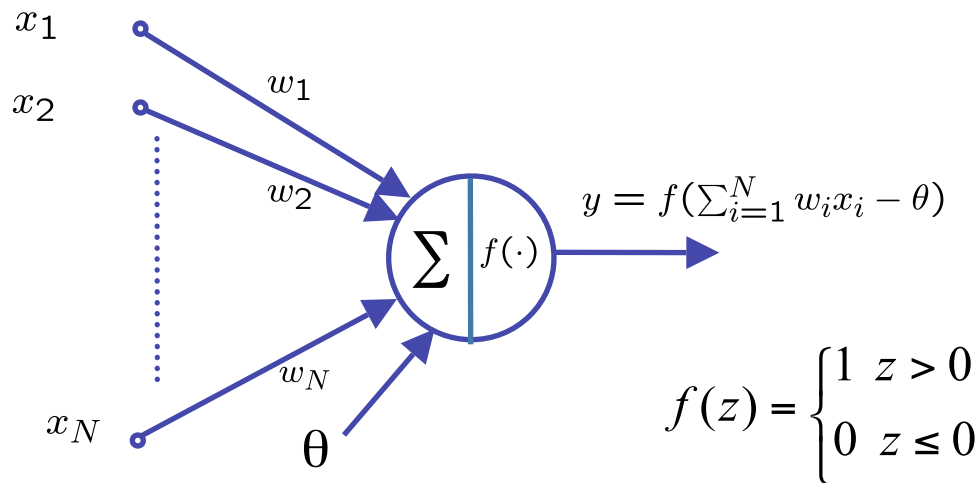
From Artificial Neurons to Ensemble of Nets



Warren
McCulloch
(1898 - 1972)



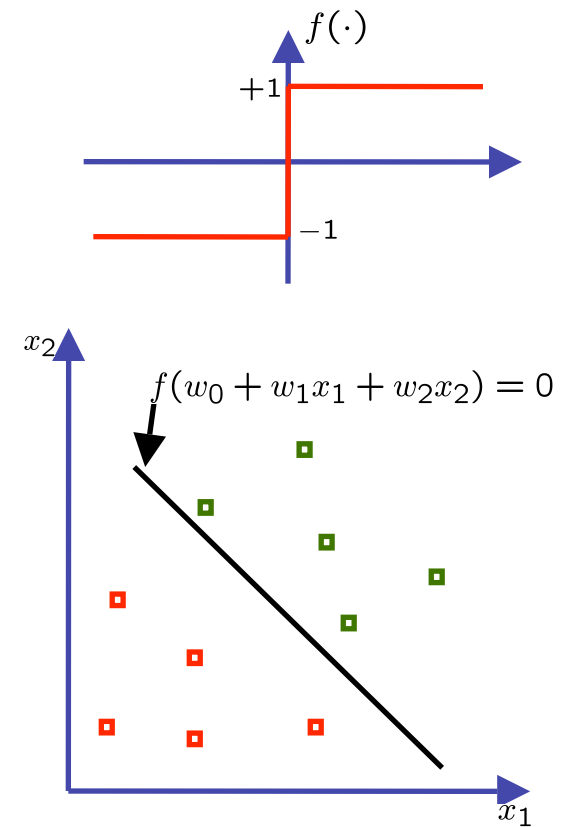
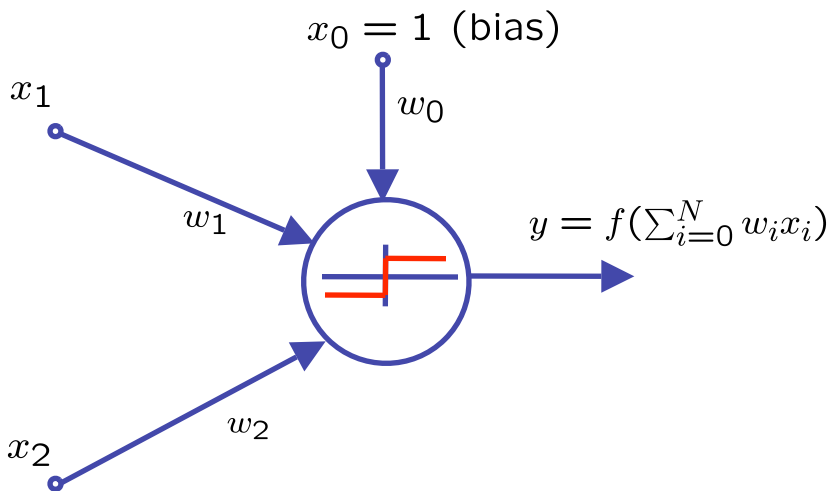
Walter Pitts
(1924 - 1969)



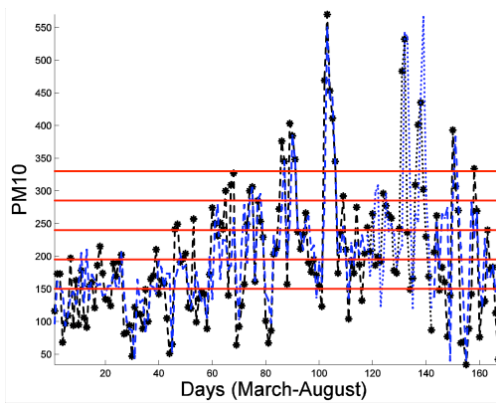
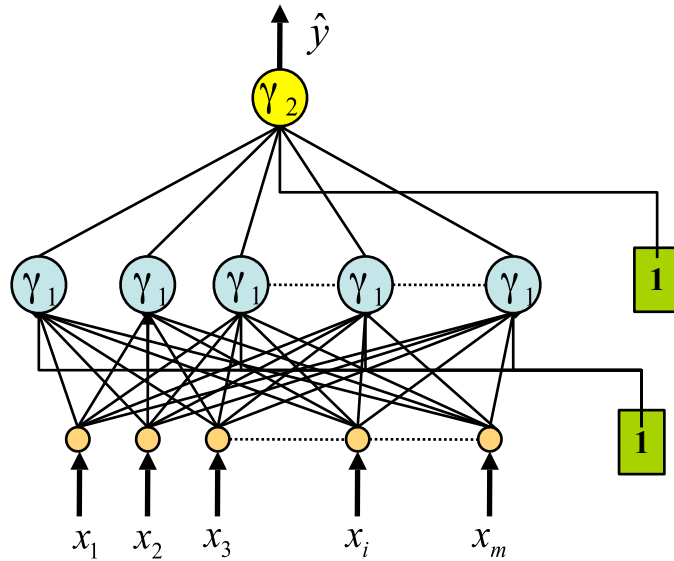
Warren McCulloch and Walter Pitts, A Logical Calculus of Ideas Immanent in Nervous Activity, 1943, Bulletin of Mathematical Biophysics 5:115-133.

- McCulloch and Pitts (1943) showed that networks made from these neurons can implement logic functions, such as AND, OR, XOR. Therefore these networks are universal computation devices.

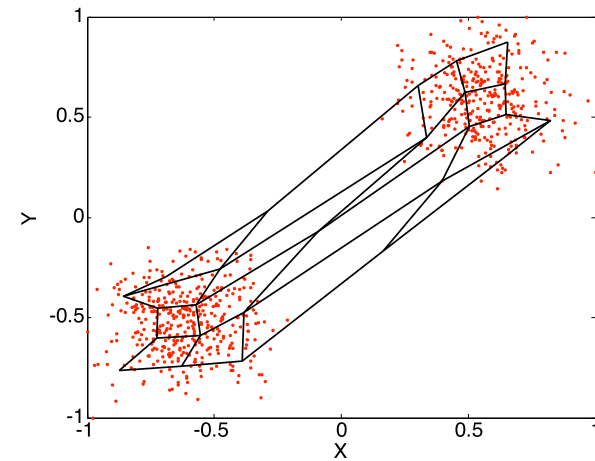
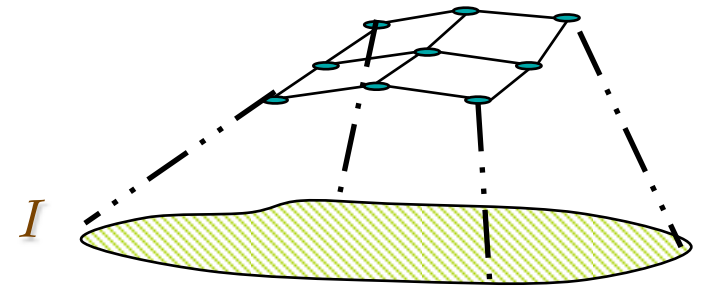
The artificial neuron's activation y depends on the (linear) sum of inputs (including a bias) converging on the neuron through weighted pathways.



Feedforward Artificial Neural Network

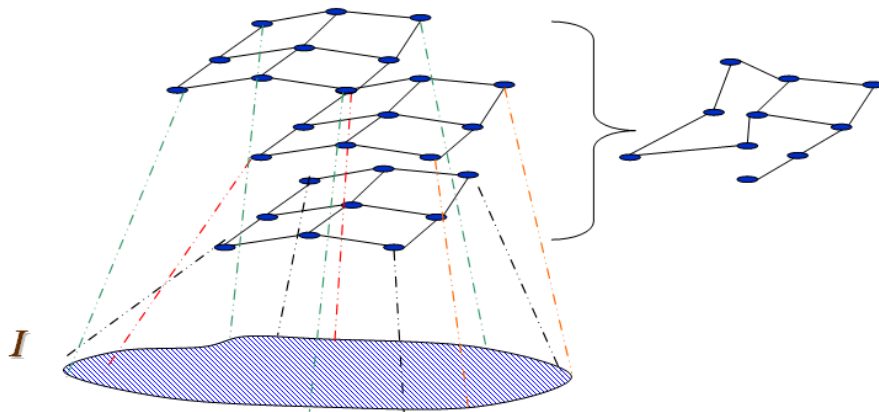


Self Organizing Maps

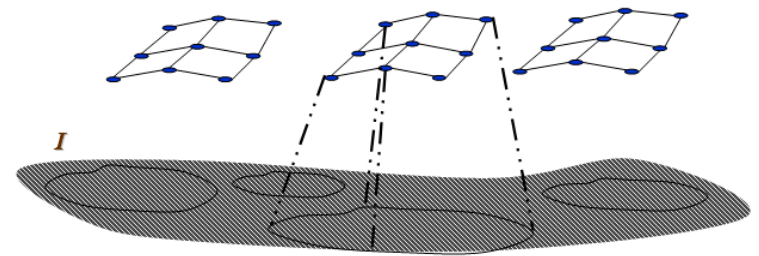


- Create and train a diverse set of base learners.
 - » Different training sets
 - » Different training parameters
 - » Different machines
- Combine the decisions of the individual learners.

Machine fusion

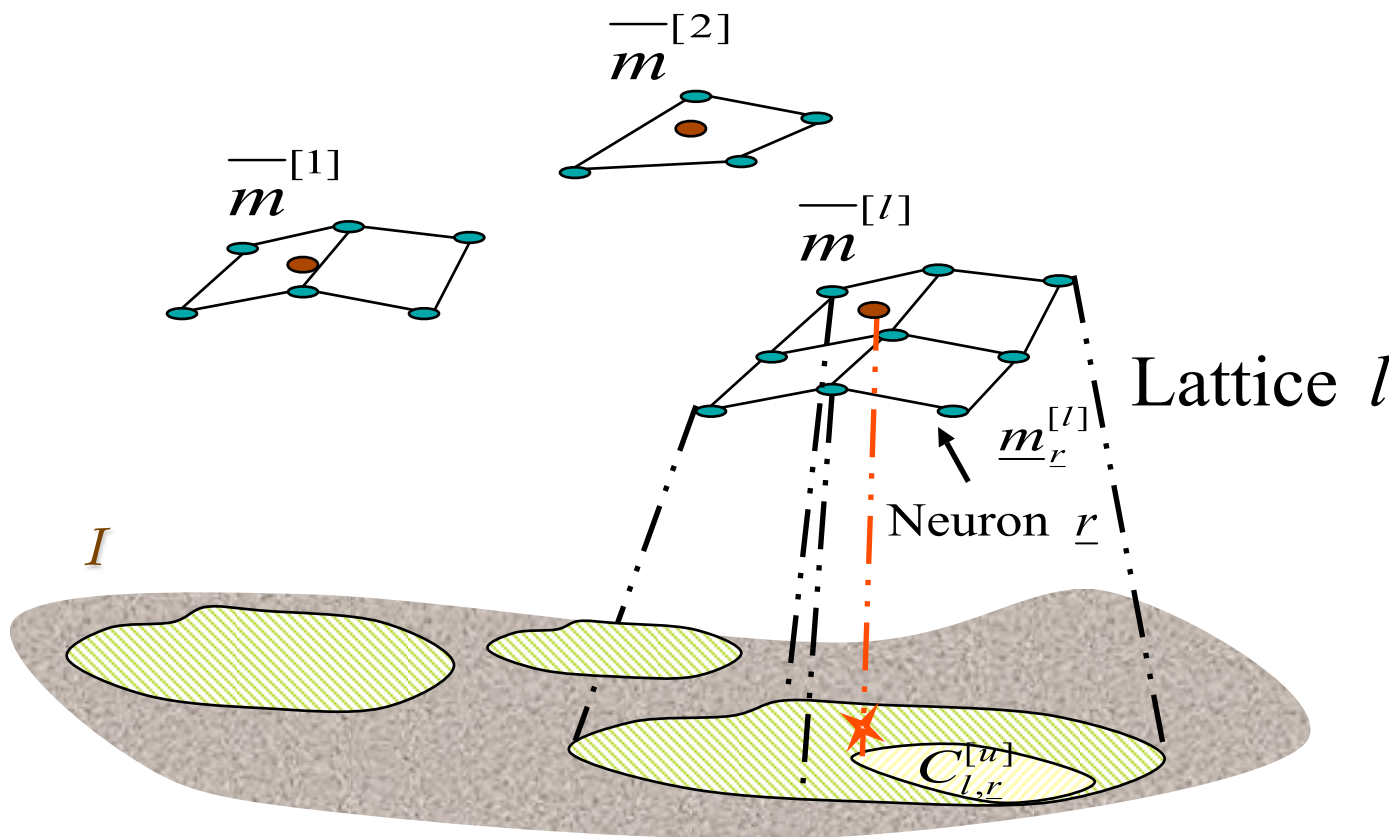


Machine selection



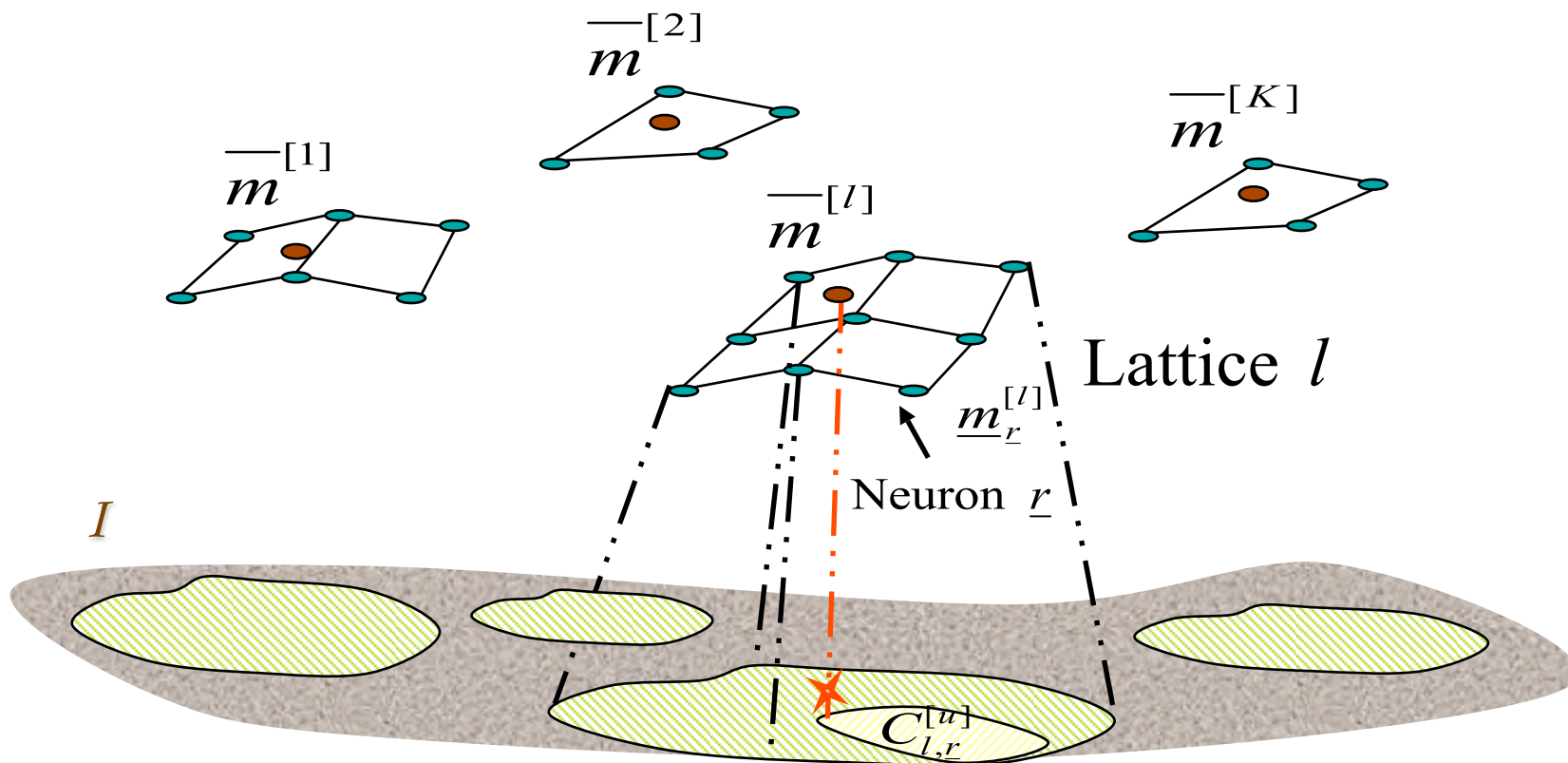
The Flexible Architecture of Self Organizing Maps

- Flexible Architecture of Self Organizing Maps (FASOM) that overcomes the Catastrophic Interference and preserves the topology of clustered data in changing environments.



The Flexible Architecture of Self Organizing Maps

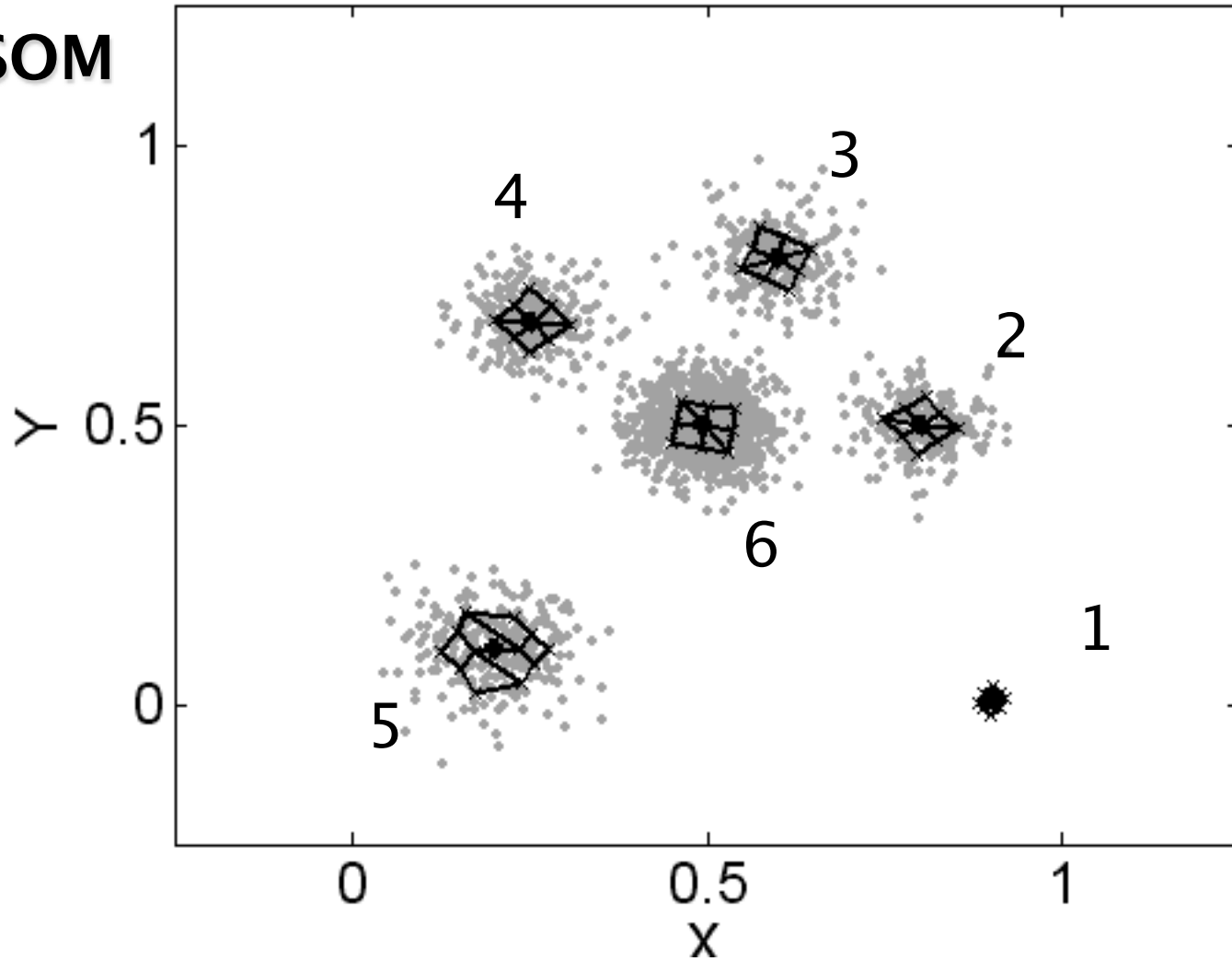
- Flexible Architecture of Self Organizing Maps (FASOM) that overcomes the Catastrophic Interference and preserves the topology of clustered data in changing environments.

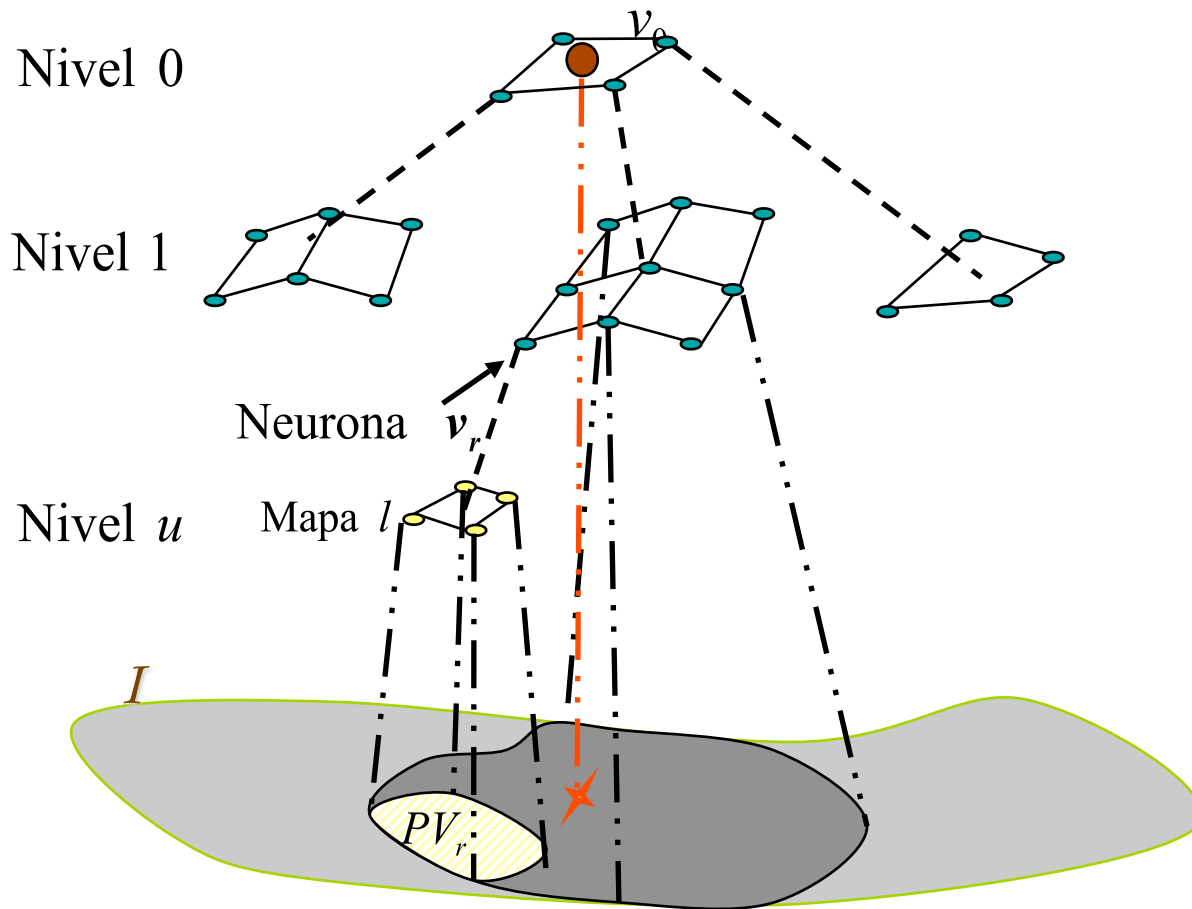


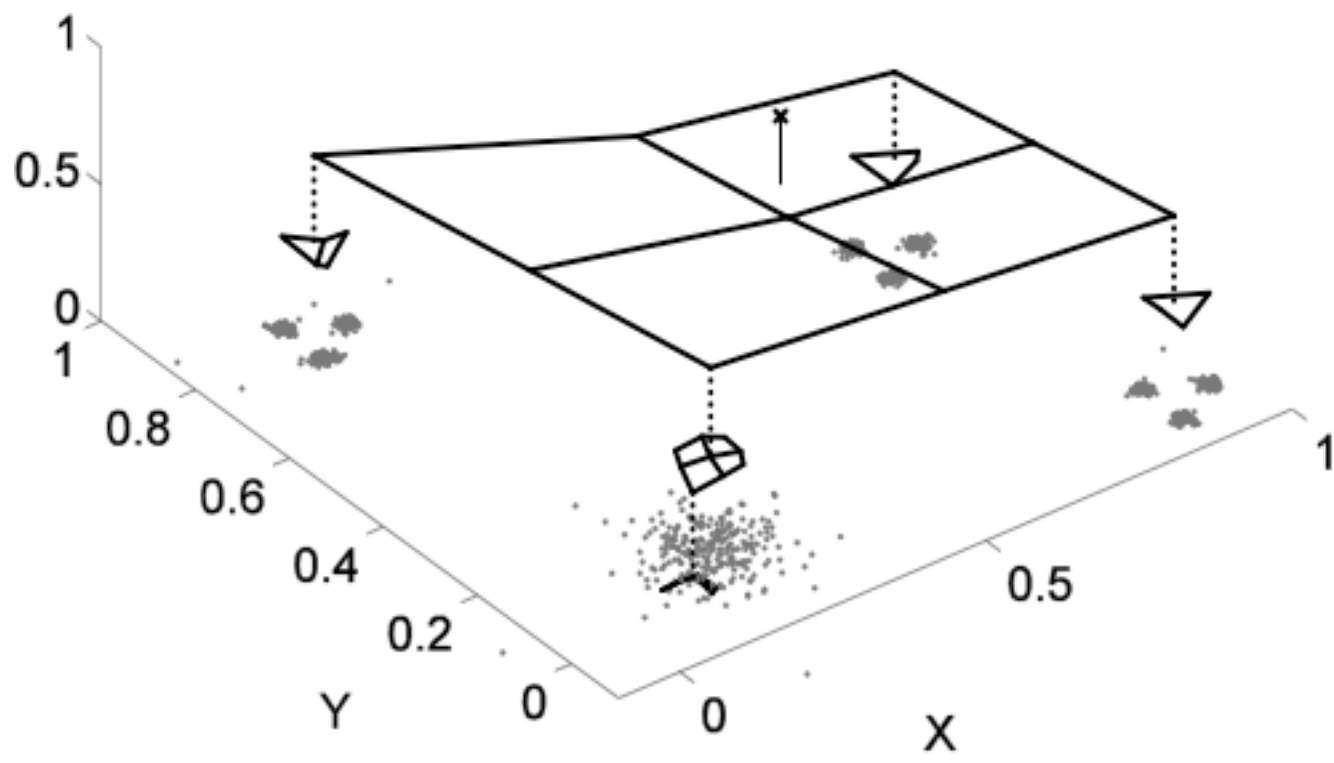


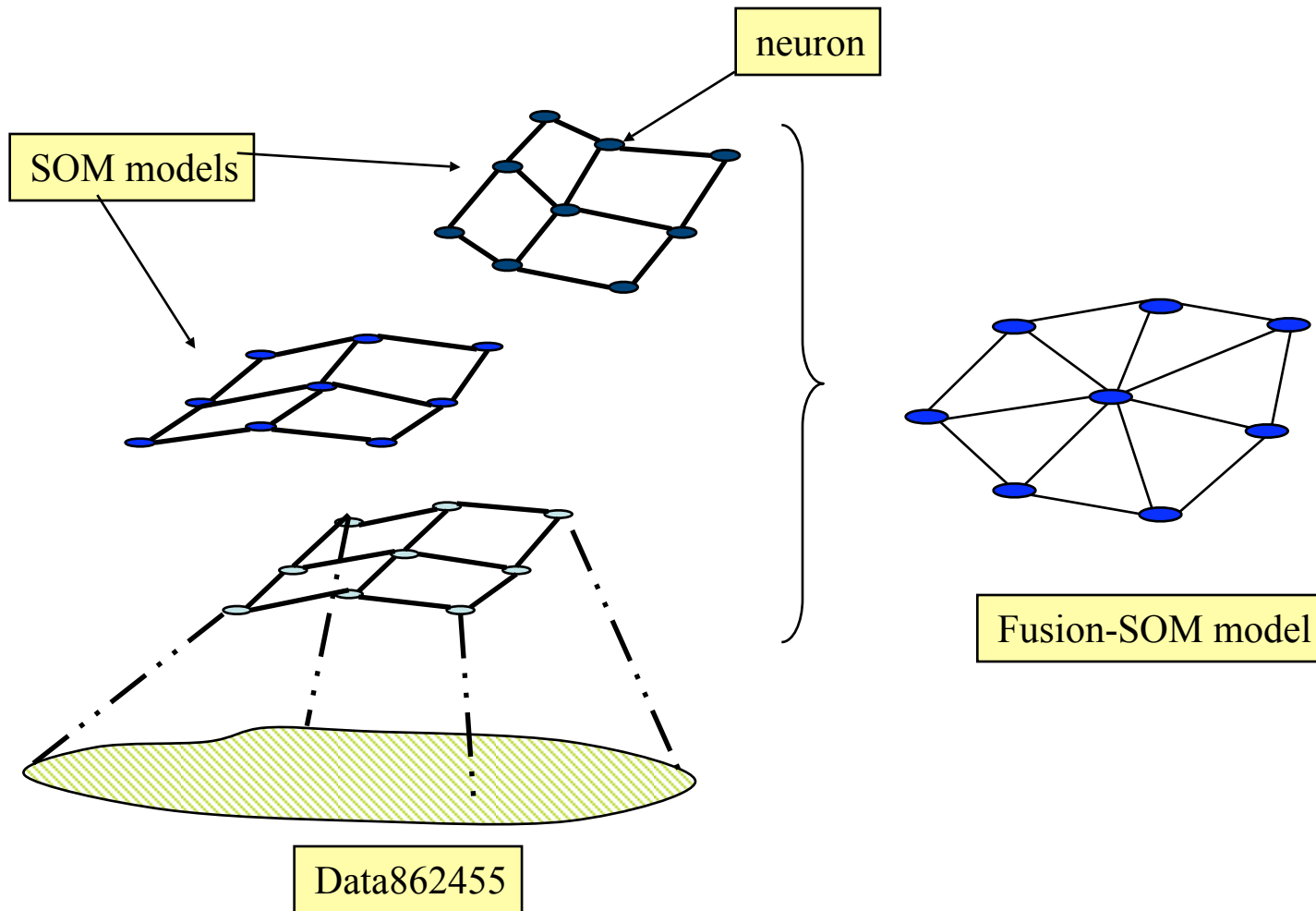
The Flexible Architecture of Self Organizing Maps

FASOM

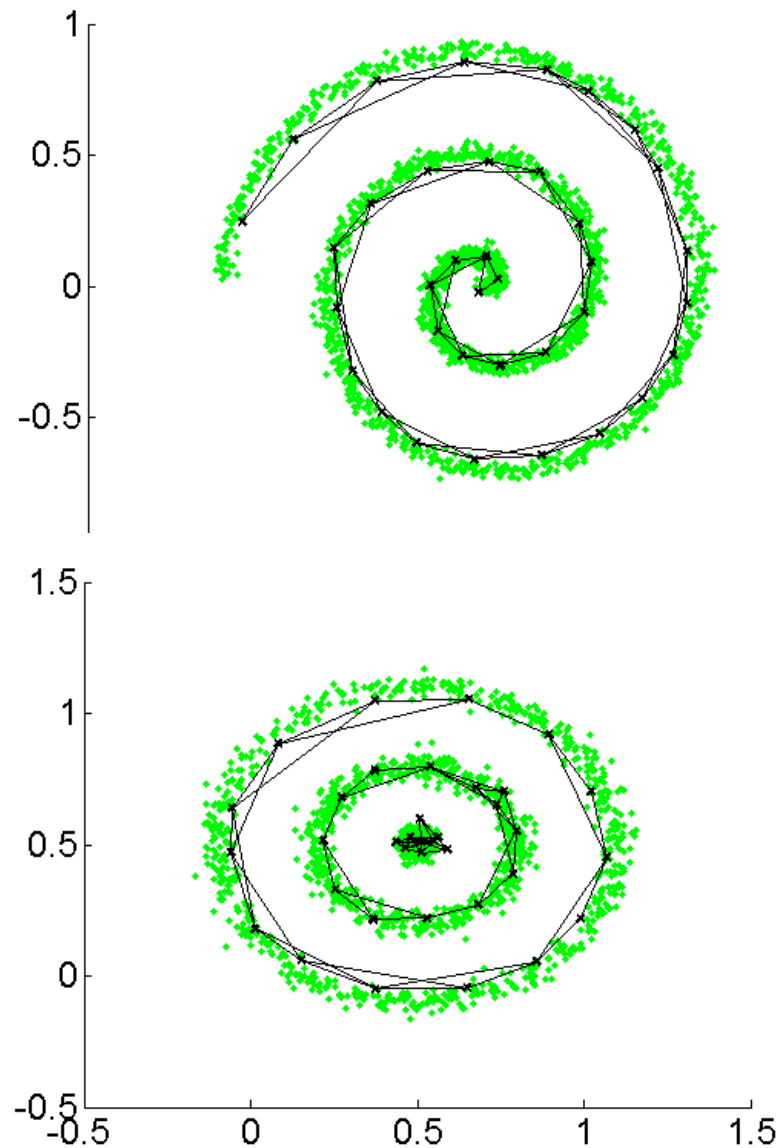
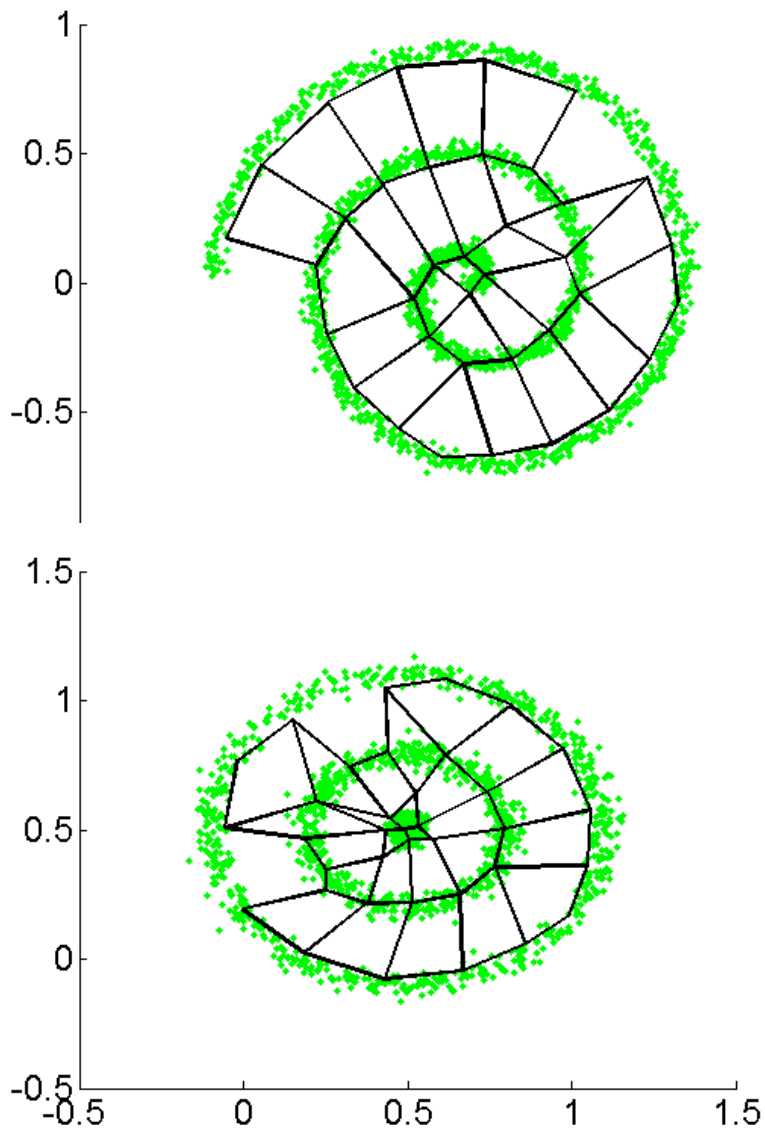


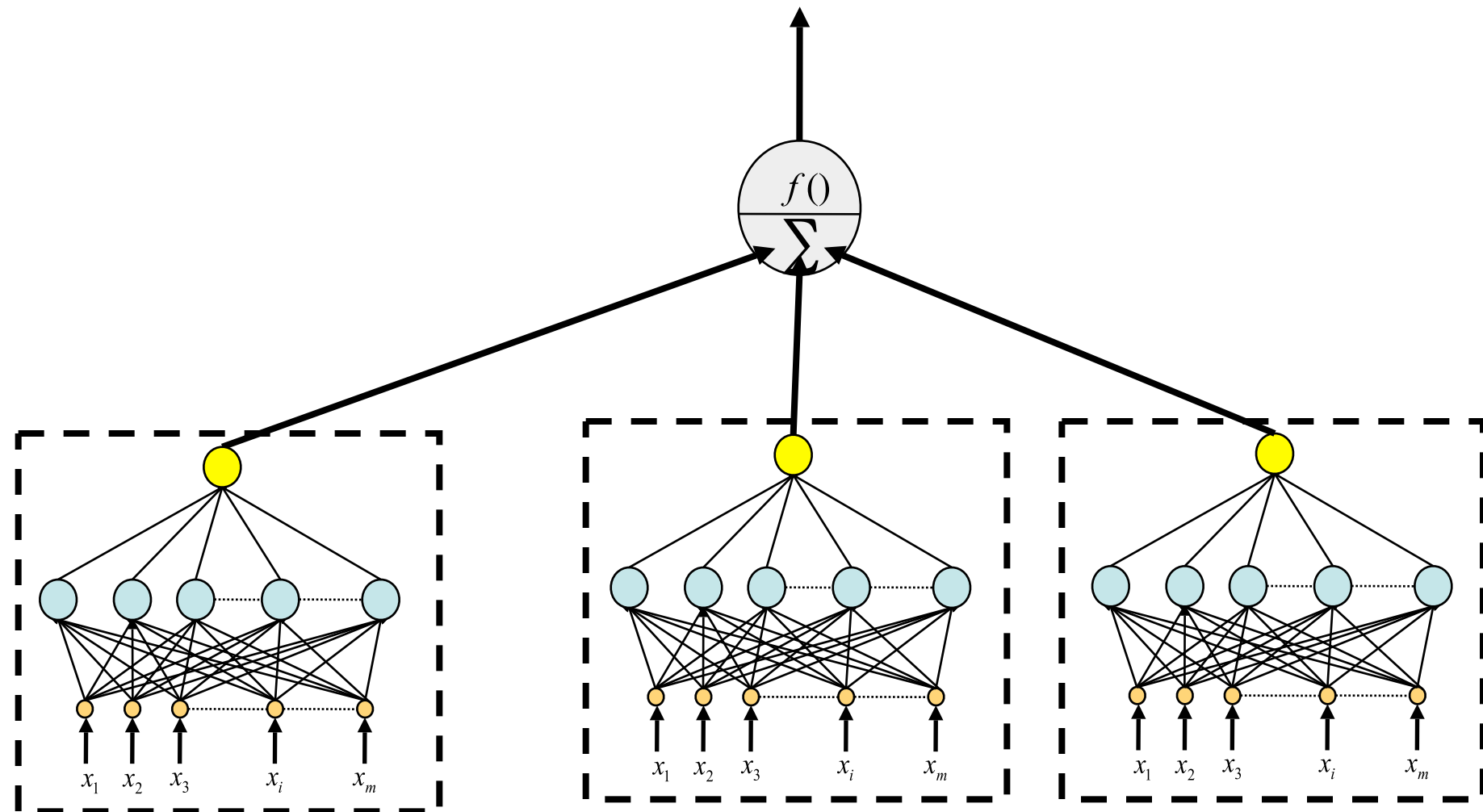






Fusion of Self Organizing Maps







Michael I. Jordan

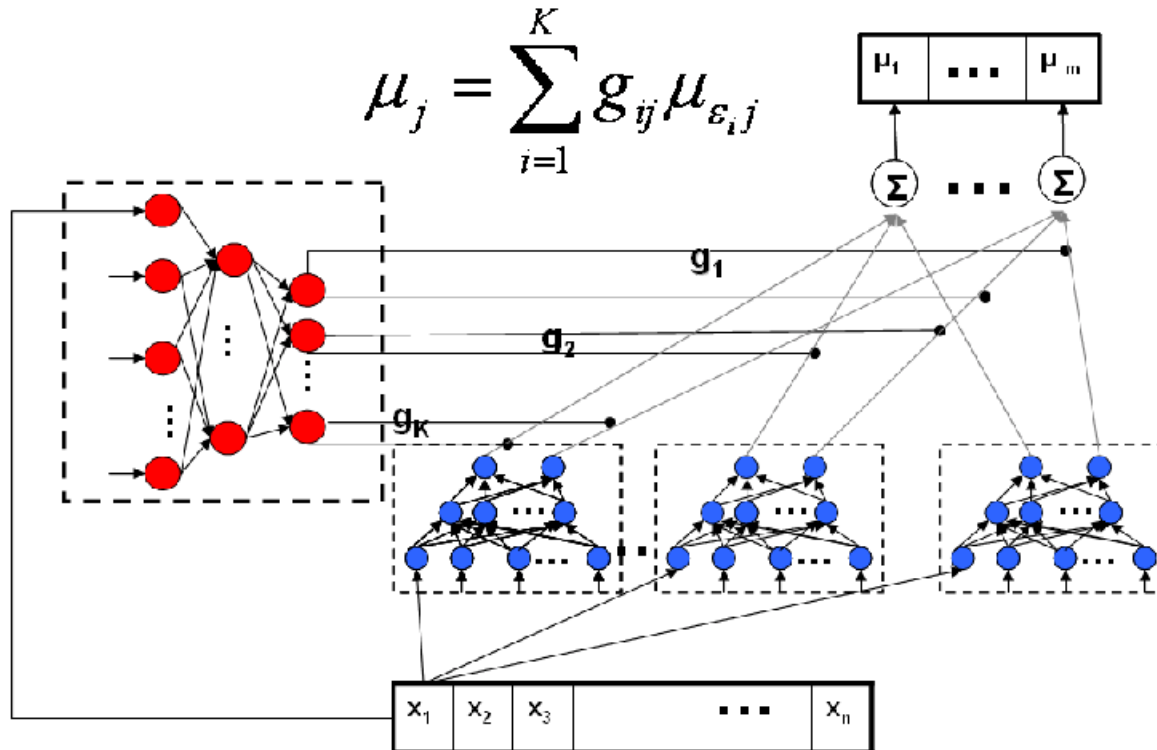
“The experts networks compete for the learning of the training data and the gating network decides which expert network is more capable to model the pattern.”

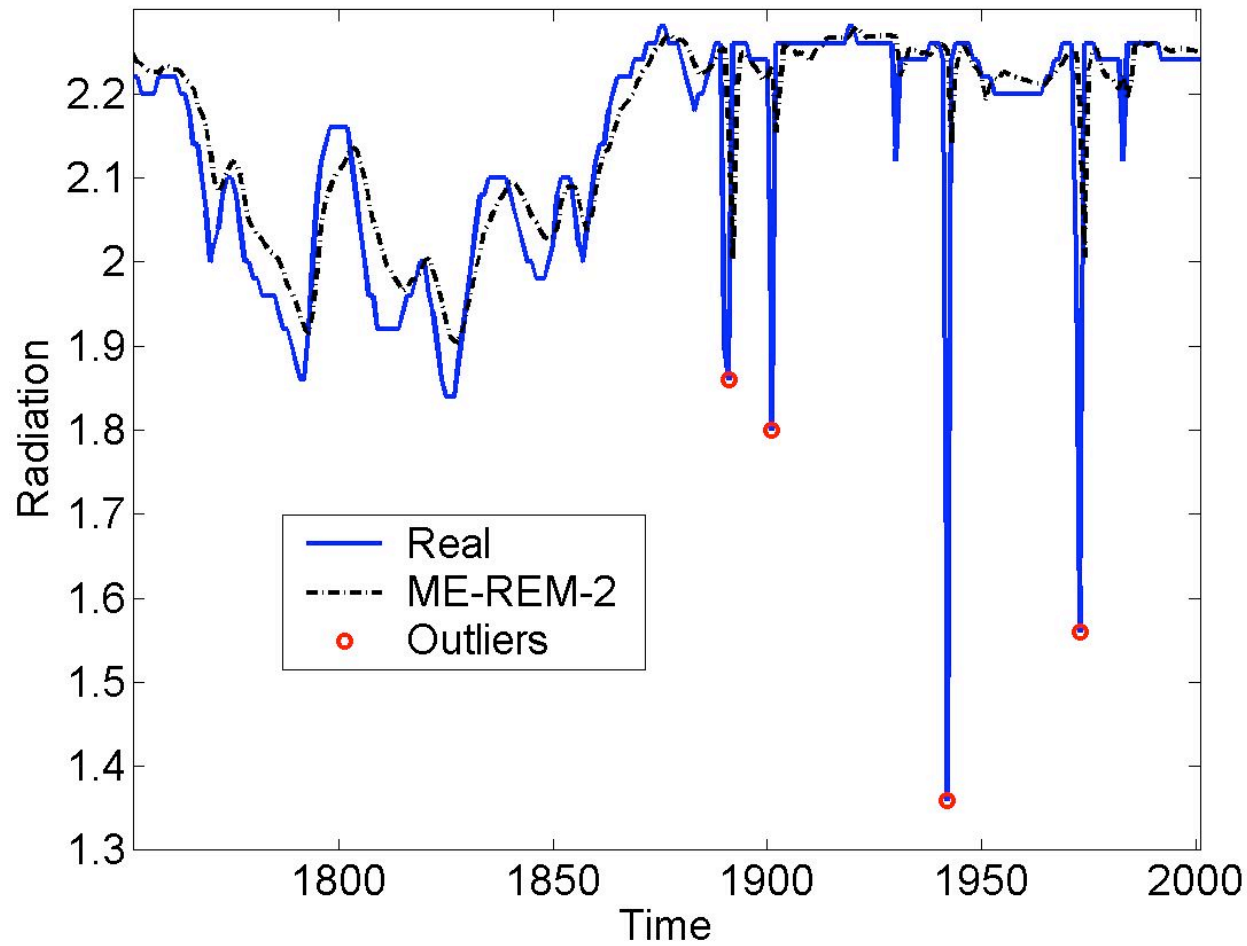


Robert Jacobs

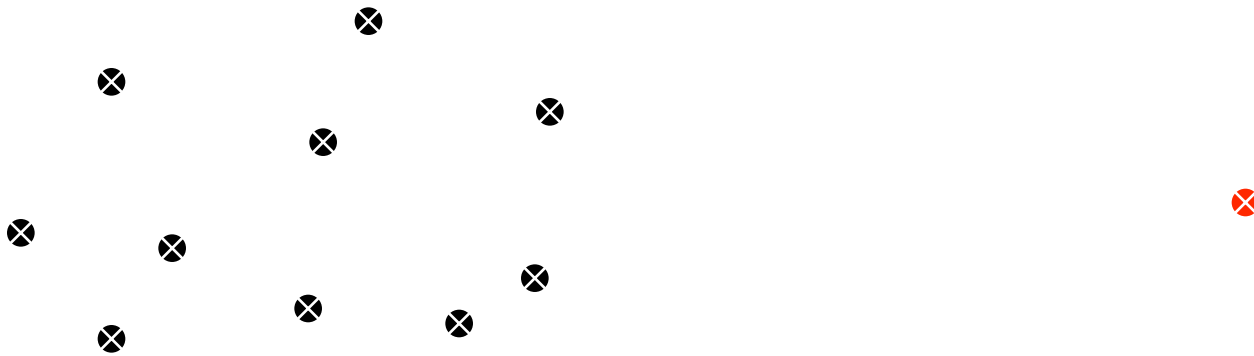


Geoffrey Hinton



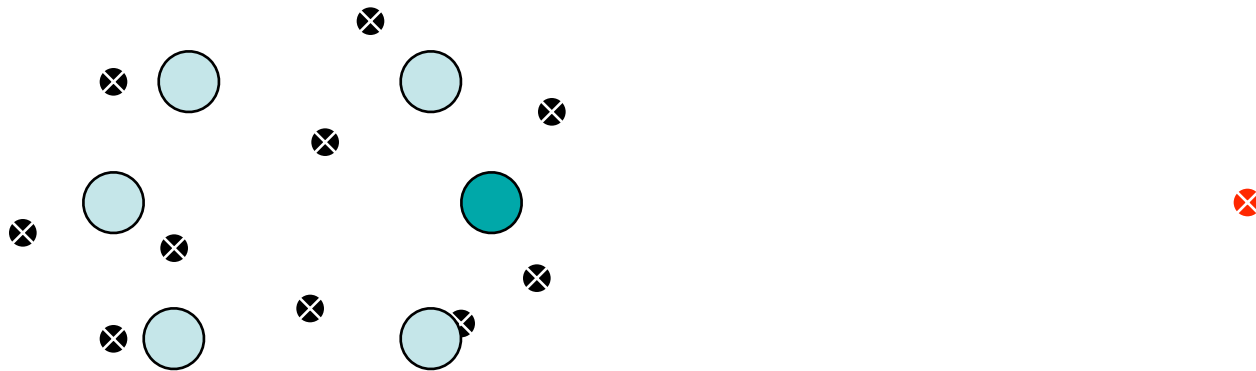


- During the learning process the outlier influences the model by moving the neurons toward the outliers.



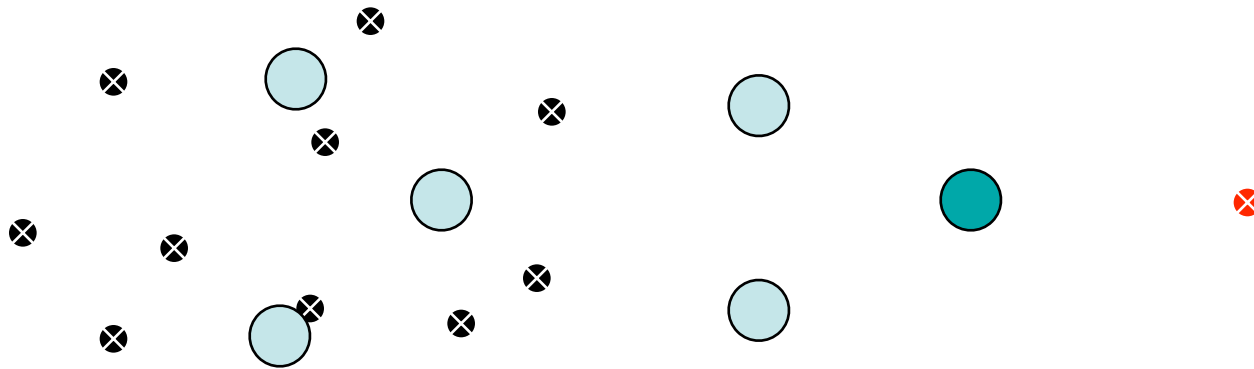
- The Robust Learning Algorithms bound the influence of samples that are likely to be regarded as outliers.

- During the learning process the outlier influences the model by moving the neurons toward the outliers.



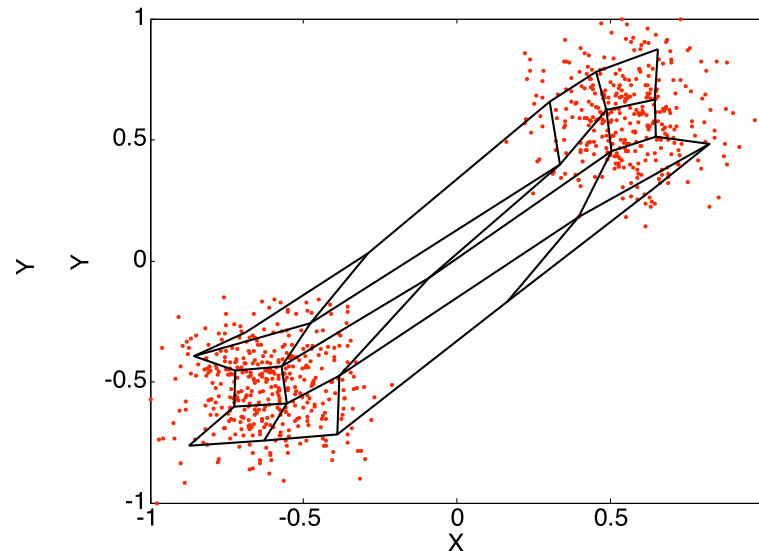
- The Robust Learning Algorithms bound the influence of samples that are likely to be regarded as outliers.

- During the learning process the outlier influences the model by moving the neurons toward the outliers.

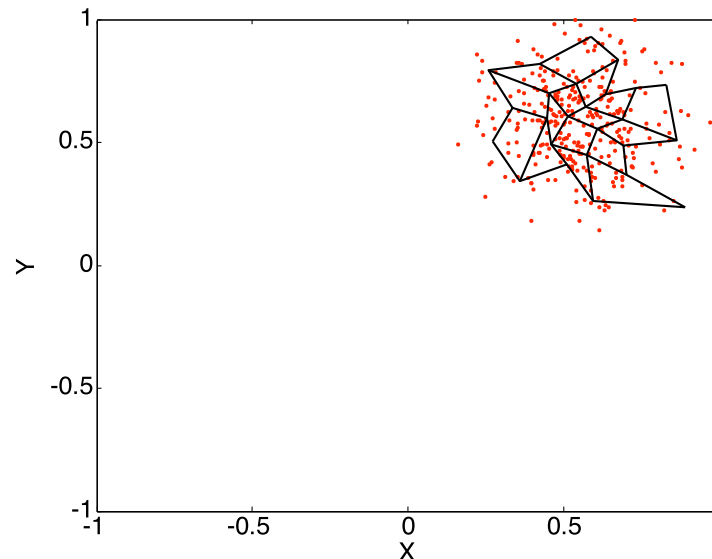


- The Robust Learning Algorithms bound the influence of samples that are likely to be regarded as outliers.

- Under non-stationary environments the classification task changes during the operation of the model.
- Catastrophic Interference is a well known problem of Artificial Neural Networks (ANN) learning algorithms where the ANN forget useful knowledge while learning from new data.



- Under non-stationary environments the classification task changes during the operation of the model.
- Catastrophic Interference is a well known problem of Artificial Neural Networks (ANN) learning algorithms where the ANN forget useful knowledge while learning from new data.





Thanks
Questions?

BIC 2007

2nd ISCV Thematic Workshop: Biologically-Inspired Computing
December 3-7, 2007 — Valparaíso Complex Systems Institute, Chile.

Ensemble methods: putting learners to work together



Ricardo Ñanculef (rnancu@inf.utfsm.cl)

INCA – Grupo de Inteligencia Computacional Aplicada
Departamento de Informática. Universidad Técnica Federico Santa María.

Rodrigo Salas F. (rodrigo.salas@uv.cl)

INCA – Grupo de Inteligencia Computacional Aplicada
IRIS – Grupo de Investigadores en Reconocimiento, Inteligencia y del Saber

