Neural network model for gene function prediction (GFP)

Corso di Bioinformatica

Anno accademico 2016/2017

Università degli Studi di Milano

Dipartimento di Informatica

Outline

- Hopfield neural network model
- The Gene Function Prediction (GFP) problem
- Related approach for GFP
- COSNet
- COSNet extensions
- Possible developments

Hopfield Networks

- A paper by John Hopfield in 1982 was the catalyst in attracting the attention of many physicists to "Neural Networks".

- His aim:

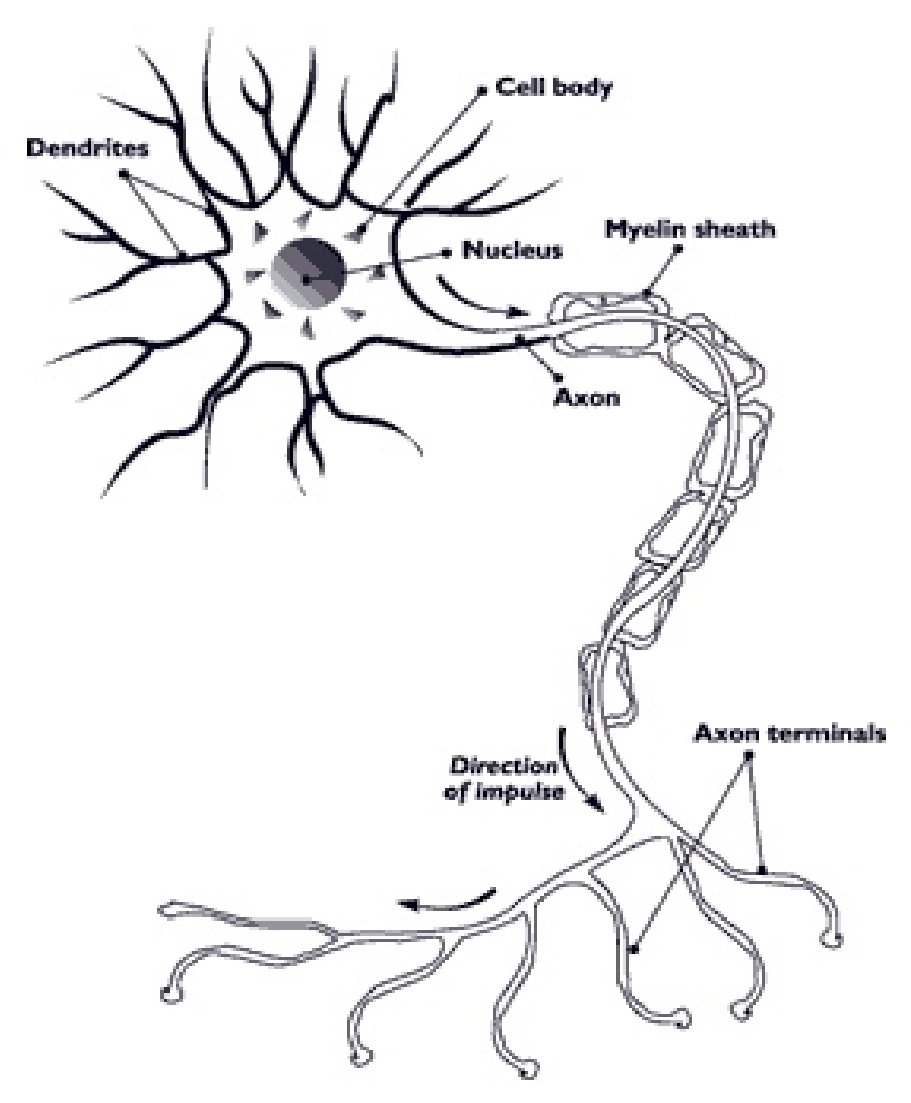
  How is one to understand the incredible effectiveness of a brain in tasks such as recognizing a particular face in a complex scene?

- ***Like all computers, a brain is a dynamical system that carries out its computations by the change of its 'state' with time.***

Hebb's rule

- A Hopfield network (HN) is based on the Hebbian rule
  - *Hebb's rule states that if neuron i is near enough to excite neuron j and repeatedly participates in its activation, the synaptic connection between these two neurons is strengthened and neuron j becomes more sensitive to stimuli from neuron i.*
  - *If two neurons on either side of a connection are activated synchronously, then the weight of that connection is increased*
  - *If two neurons on either side of a connection are activated asynchronously, then the weight of that connection is decreased*

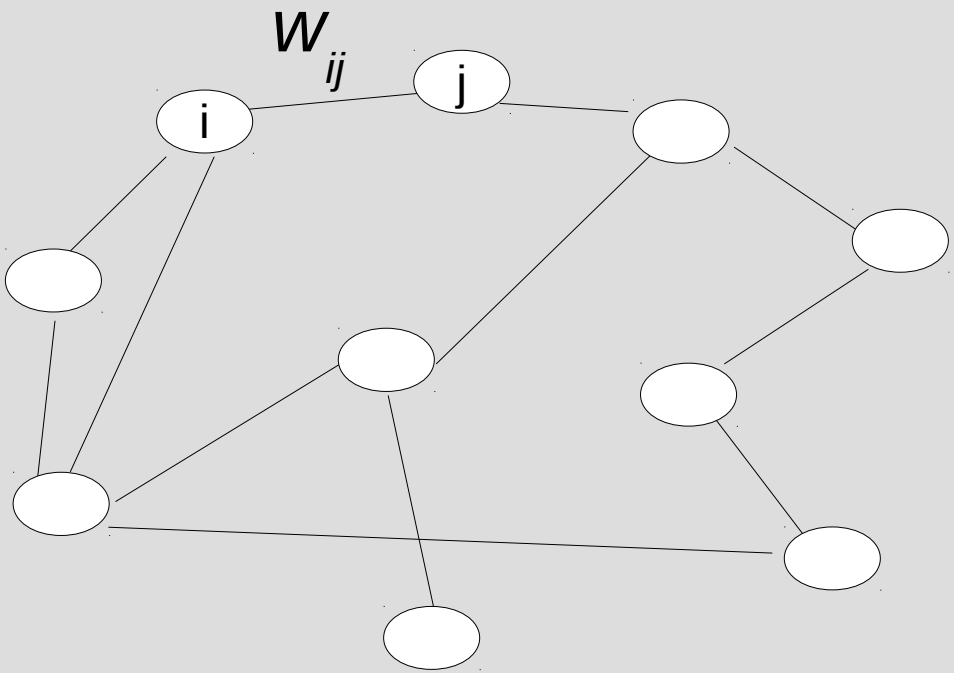
Neuron

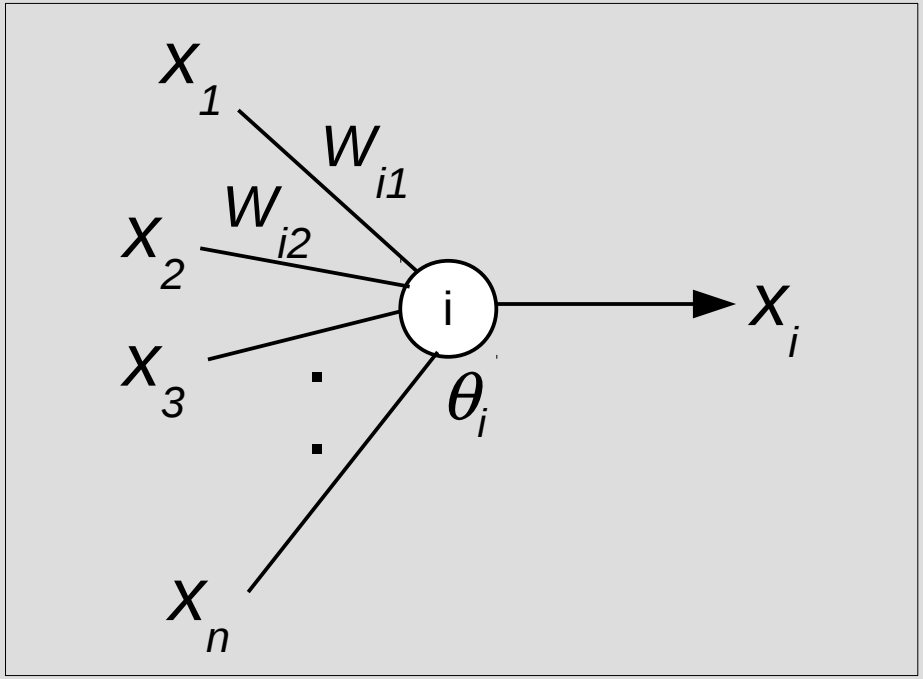

Cell body
Dendrites
Myelin sheath
Nucleus
Axon
Axon terminals
Direction of impulse

Hopfield Networks

- Dynamic model in which at each time *t* each neuron *i* has an activation value (state) $x_i \in \{1, 0 (-1)\}$

and activation threshold θ_i.





$$x_i = sgn\left(\sum_{j=1}^{n} w_{ij} x_j - \theta_i\right)$$

$$sgn(a) = \begin{cases} 1 \; if \; a \geq 0 \\ 0 \; if \; a < 0 \end{cases}$$

Update rule

- The state of the network $\underline{x}(t)=(x_1(t), x_2(t), \ldots, x_n(t))$ at each time *t* is the vector of the neuron activation values at time *t.*
- The neurons are subject to the asynchronous rule for updating one neuron at a time:

  Pick a unit *i* at random and set

$$x_i(t+1)=Sgn\left(\sum_{j=1}^{i-1} w_{ij}\, x_j(t+1)+\sum_{k=i+1}^{n} w_{ik}\, x_k(t)-\theta_i\right)$$

  If the input at neuron ***i*** is greater than θ_i, turn it on otherwise turn it off

- Moreover, Hopfield assumes symmetric weights: $w_{ij} = w_{ji}$

Energy function

- Hopfield defined the state function called "energy function":

$$E(\underline{\boldsymbol{x}}) = -\tfrac{1}{2} \sum_{ij} x_i x_j w_{ij} + \sum_{i} x_i \theta_i$$

- If we pick unit *i* and the firing rule (previous slide) does not change its state x_i, it will not change E

- **Theorem**: the dynamics from the initial state follows a trajectory to an equilibrium state, which is (local) minimum of the energy function

Convergence

- x_i: 0 to 1 transition
  - It means x_i initially equals 0, and $\sum_j w_{ij} x_j \geq \theta_i$
  - The corresponding change in E is

$$\Delta E = (1-0)\left(-\tfrac{1}{2} \sum_j (w_{ij} x_j + w_{ji} x_j) + \theta_i\right)$$

$$= -\left(\sum_j w_{ij} x_j - \theta_i\right) \qquad \text{(by symmetry)}$$

$$\leq 0 \quad \text{(since the neuron passed from state 0 to state 1)}$$

Convergence

- x_i: 1 to 0 transition
  - It means x_i initially equals 1, and $\sum_j w_{ij} x_j < \theta_i$
  - The corresponding change in E is

    $\Delta E = (0-1)\,(-\tfrac{1}{2} \sum_j (w_{ij} x_j + w_{ji} x_j) + \theta_i) = (\sum_j w_{ij} x_j - \theta_i) \leq 0$

    On every updating we have $\Delta E \leq 0$
  - Hence the dynamics of the net tends to move ***E*** toward a minimum
  - We stress that there may be different such states — they are *local minima*. Global minimization is not guaranteed.

Convergence

- The symmetry condition $w_{ij} = w_{ji}$ is crucial for $\Delta E \leq 0$
- Without this condition $\frac{1}{2} \Sigma_j (w_{ij} + w_{ji}) s_j - \theta_i$ cannot be reduced to $(\Sigma_j w_{ij} s_j - \theta_i)$, so that Hopfield's updating rule cannot be guaranteed to yield a passage to energy minimum
  - It might instead yield a **limit cycle**

HN as local optimizer

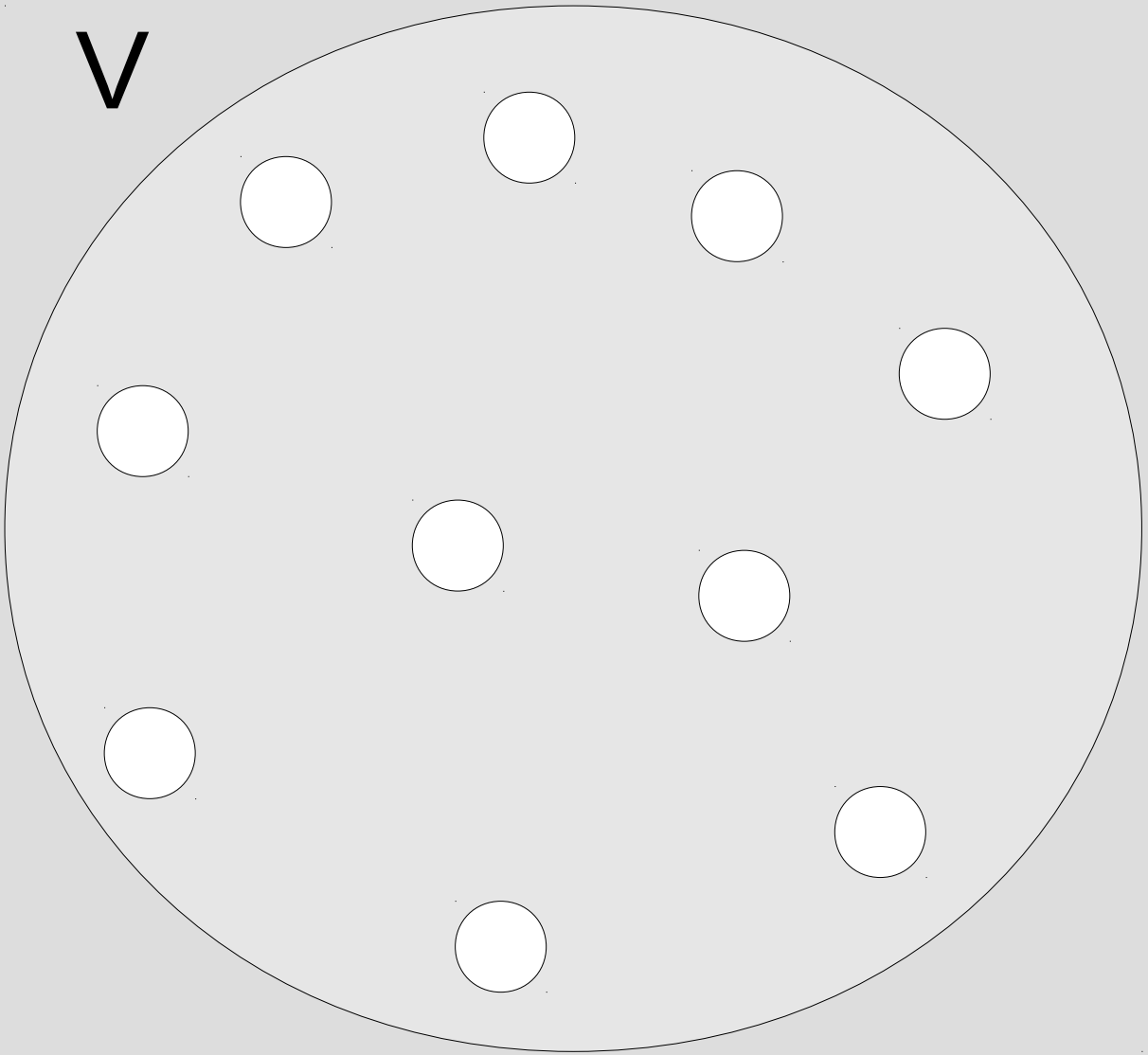
- To design Hopfield nets to solve optimization problems:
  - choose weights for the network so that ***E*** is a measure of the overall constraint violation.
  - A famous example is the traveling salesman problem.
    - [HBTNN articles:Neural Optimization; Constrained Optimization and the Elastic Net. See also TMB2 Section 8.2.]

Gene Function Prediction

Genome sequencing

- Main problem: understanding biological functions of new genes
- Taxonomy: hierarchical definition of gene properties
  - Gene Ontology(GO), FunCat
- Annotation: established involvement of a gene in the biological mechanism represented by a functional class (term)
  - Classes are often highly unbalanced

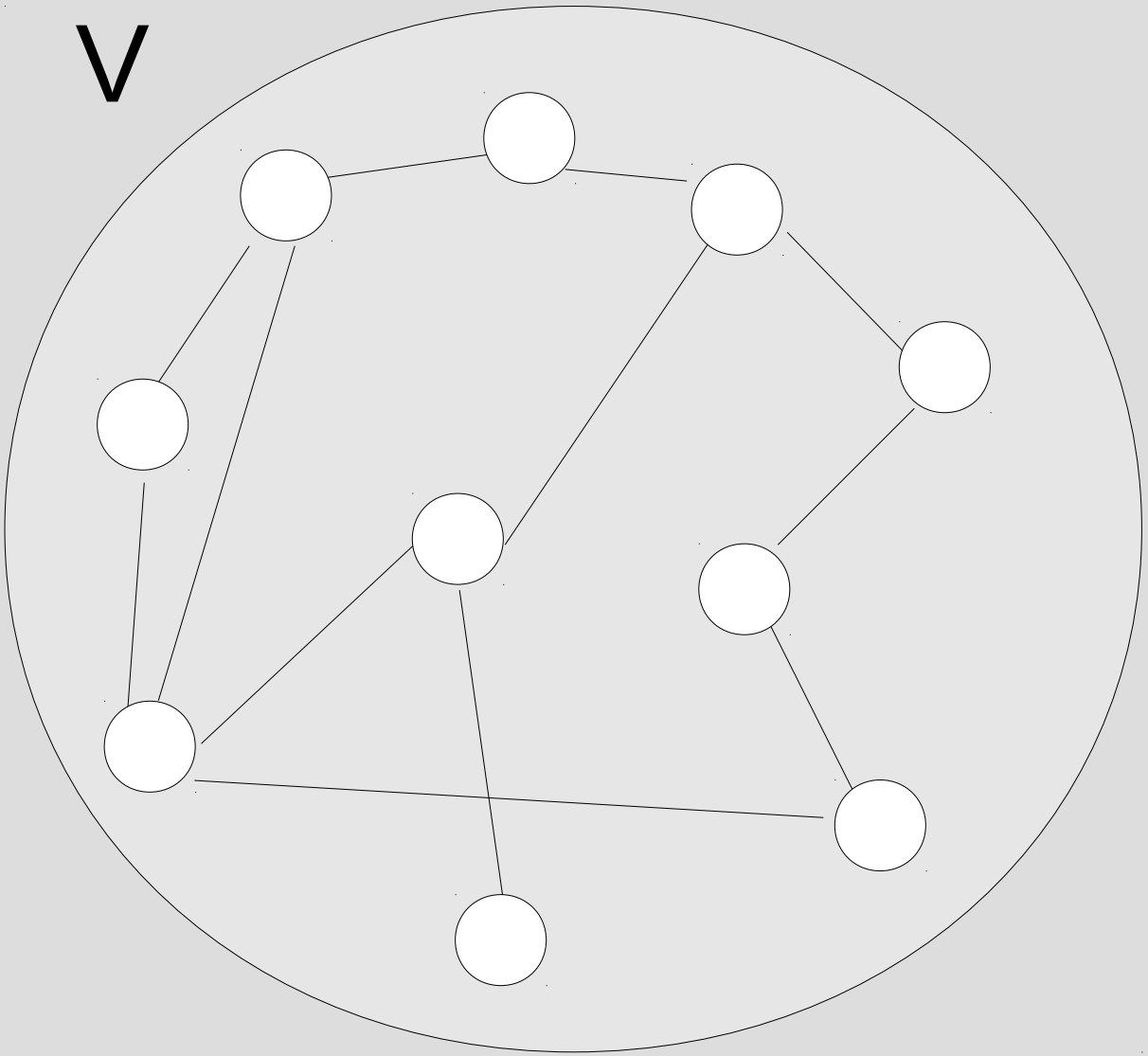
Gene Function Prediction Problem



Input:

- *V* genes

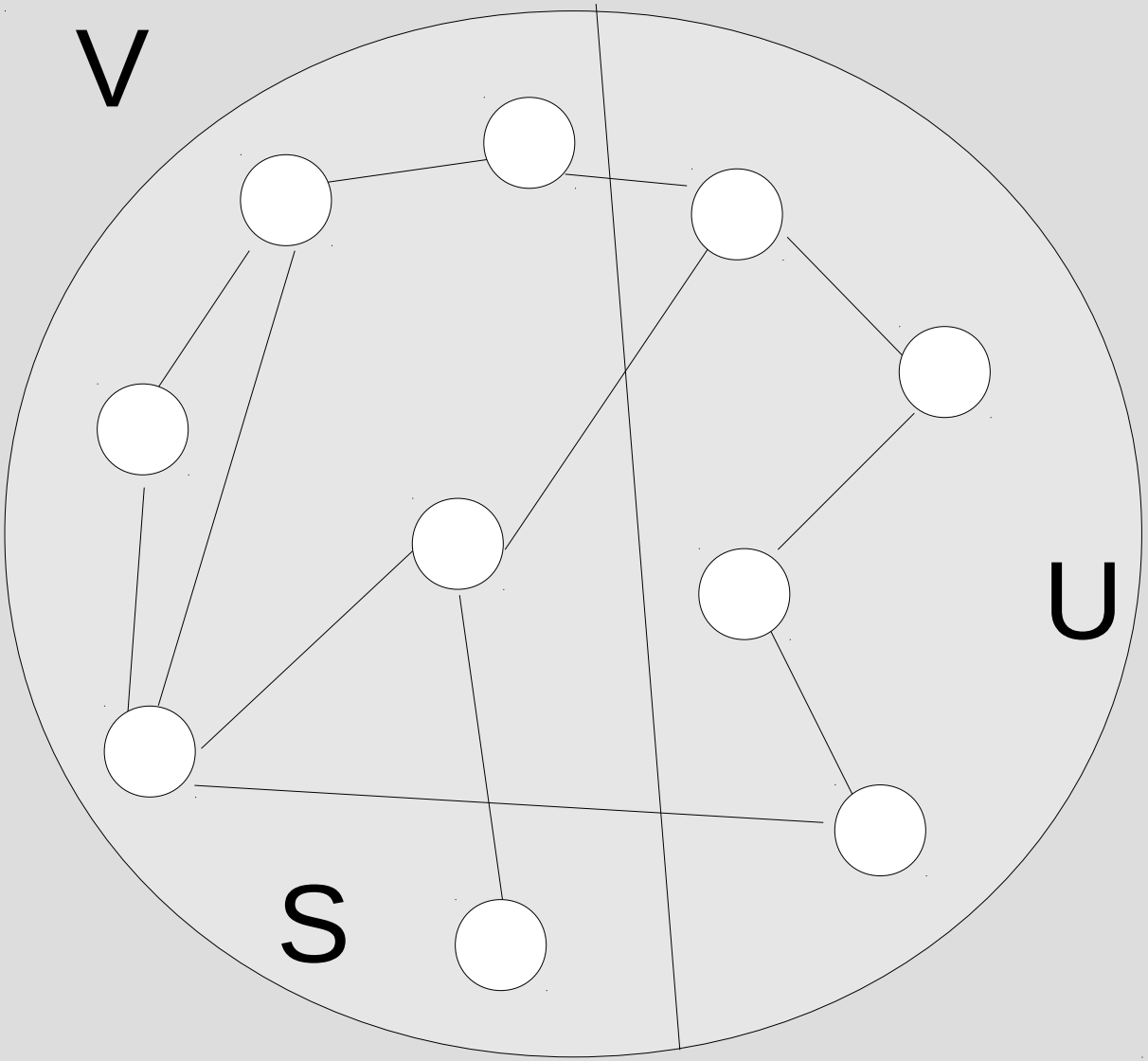
Gene Function Prediction Problem



Input:

- *V* genes
- *W* symmetric matrix

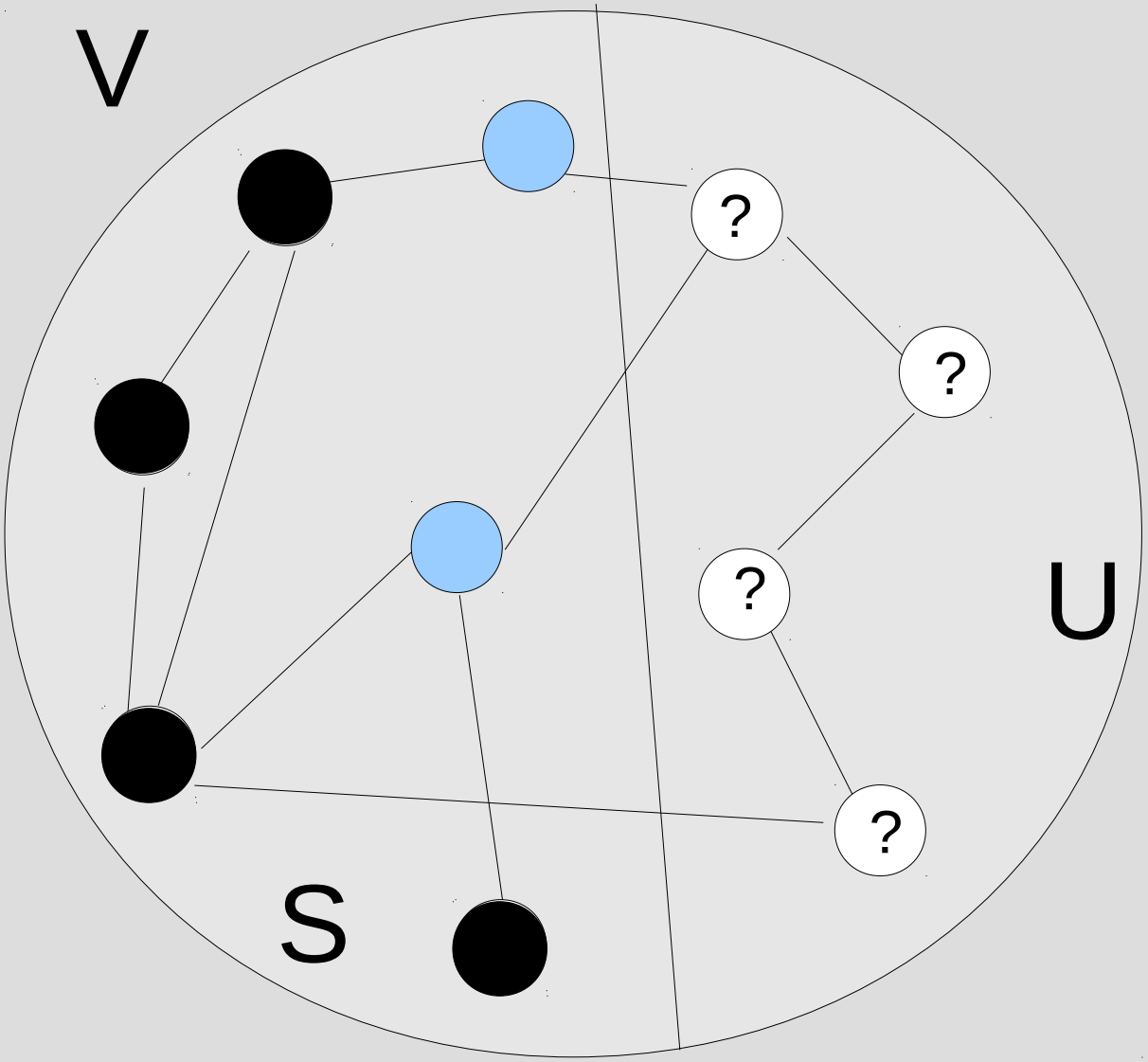
Gene Function Prediction Problem



Input:

- *V* genes
- *W* symmetric matrix
- *S*, *U* bipartition of *V*
  - *S* labeled genes
  - *U* unlabeled genes

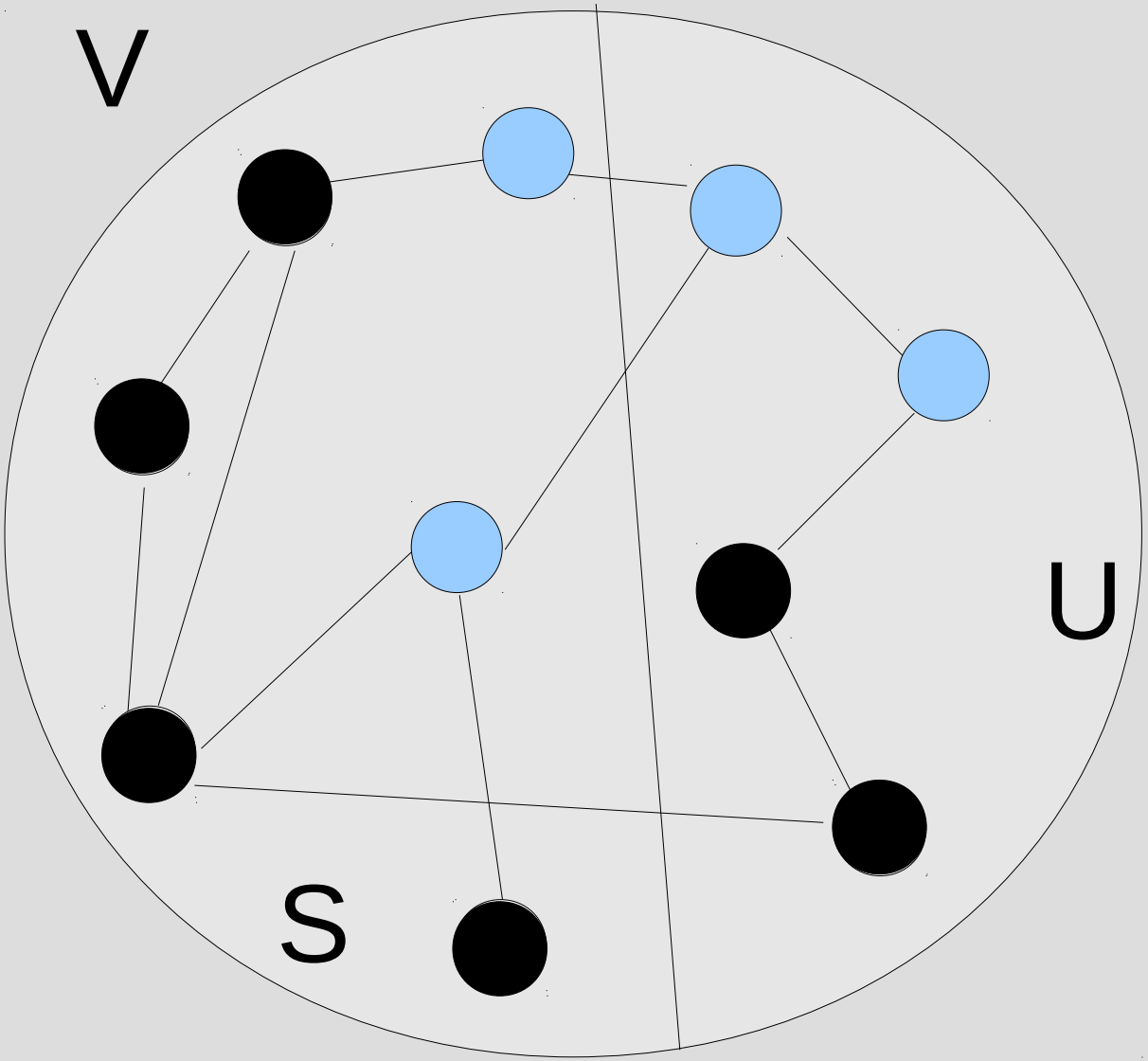
Gene Function Prediction Problem



Input:

- V genes
- W symmetric matrix
- S, U bipartition of V
  - S labeled genes
  - U unlabeled genes
- S^p, S^n bipartition of S

Gene Function Prediction Problem



Input:

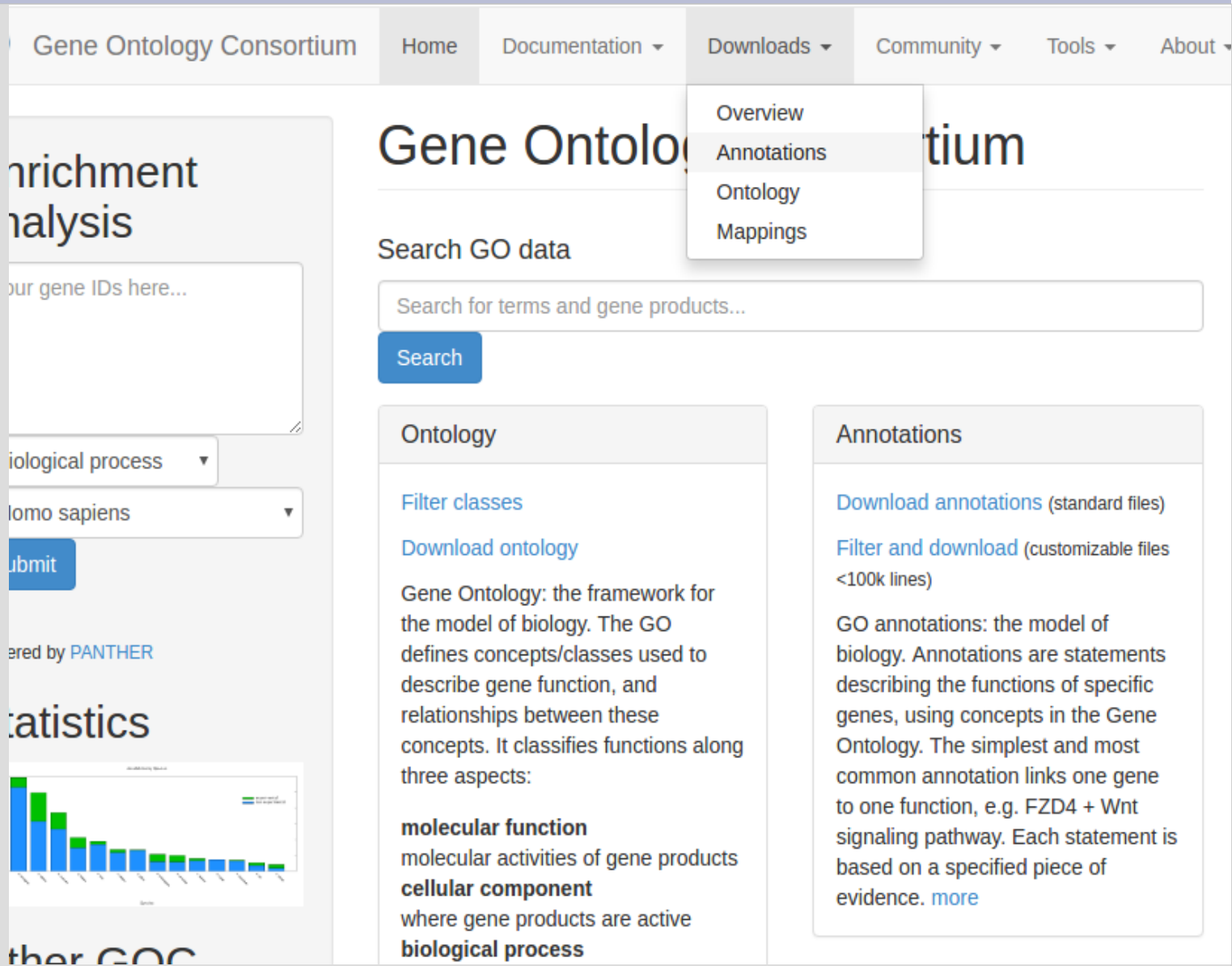
- V genes
- W symmetric matrix
- S, U bipartition of V
  - S labeled genes
  - U unlabeled genes
- S^p, S^n bipartition of S

Output:

- U^p,U^n bipartition of U

Data bank for annotations: the Gene Ontology (http://www.geneontology.org/)

Downloading annotations



Data bank for annotations: the Gene Ontology (http://www.geneontology.org/)

Downloading annotations for S.cerevisiae organism (yeast)

Saccharomyces cerevisiae SGD Stanford University	6448	111356 (60174 non-IEA)	1/14/2017	README	gene_association.sgd.gz (1 mb)
Solanaceae	867	1457 (1457 non-IEA)	9/17/2015	README	gene_association.sgn.gz (32 kb)

Opening the file

```
!
SGD   S000007287  15S_RRNA    GO:0005763  SGD_REF:S000073641|PMID:6262728  IDA
s rRNA|15S_RRNA_2 gene  taxon:559292   20150612 SGD
SGD   S000007287  15S_RRNA    GO:0032543  SGD_REF:S000073641|PMID:6262728  IC
020|14s rRNA|15S_RRNA_2 gene  taxon:559292   20150612 SGD
SGD   S000007287  15S_RRNA    GO:0003735  SGD_REF:S000073641|PMID:6262728  IC
020|14s rRNA|15S_RRNA_2 gene  taxon:559292   20150612 SGD
```

Data bank for gene-gene interactions/similarities

- BioGRID : protein-protein interactions

- Pfam, InterPro : protein domain data

- STRING : interaction networks including several source of Information about genes and their products

Etc.

Machine learning methods for GFP

- Inductive methods
  - Learn a model to infer functions for all genes
    - Support Vector Machines [Lanckriet et al 2004]
- Transductive methods
  - Infer functional predictions only for genes in test set
    - MRF [Deng et al 2002],
    - Neural networks [Karaoz et al 2003],
    - Functional Linkage Networks [Marcotte 1999]
    - Label propagation [Zhu et al 2003, Mostafavi 2008-2010].

Gene Annotation using Integrated Networks (GAIN)

- Karaoz et al. (2003)
- Discrete Hopfield network

Genes → Neurons

Similarity → Weights

(thresholds = 0)

Bipartitions → Initial state

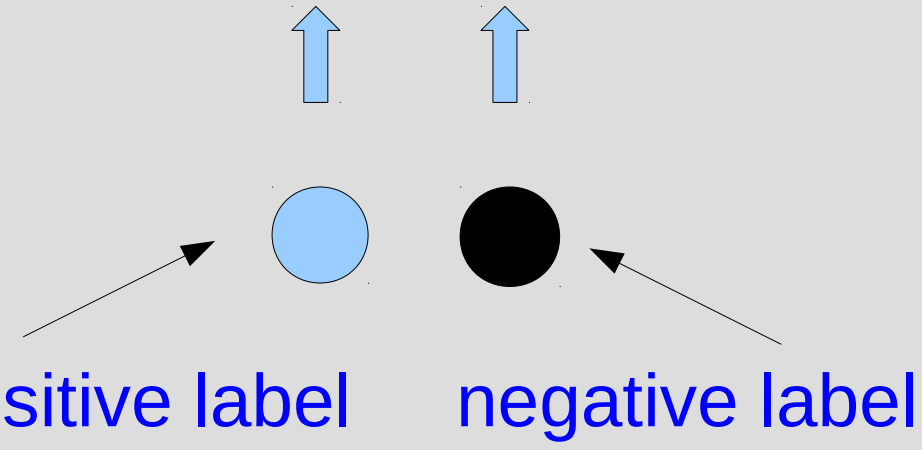
↓ Dynamics

Inference ← Equilibrium state

GAIN

- Initial state for each neuron *i* :

$x_i(0) =$ *1*, *-1*, *0* ⟸ ?, Unlabeled



$$x_i(t+1) = Sgn\left(\sum_{j=1}^{i-1} w_{ij}\, x_j(t+1) + \sum_{k=i+1}^{n} w_{ik}\, x_k(t)\right)$$

GAIN

- Energy function

$$E(x) = -\frac{1}{2} \cdot \sum_{i=1}^{n} x_i \left(\sum_{j=1}^{n} x_j w_{ij} \right)$$

- Minimizing *E* means maximizing the weighted sum of consistent edges (i.e. connecting nodes at the same state)

- The equilibrium state $\tilde{\tilde{x}} = (\tilde{\tilde{s}}, \tilde{\tilde{u}})$ characterizes the bipartition of *U*

$$U^p = \{i \in U \mid \tilde{\tilde{u}}_i = 1\}$$

$$U^n = \{i \in U \mid \tilde{\tilde{u}}_i = -1\}$$

Drawbacks of GAIN

- ”Same relevance” for positive and negative examples
  - Data imbalance not managed
- GAIN tries to find a global minimum $\tilde{\tilde{x}}$ of E assuming that the initial state $\bar{s}$ of labeled nodes is a part of $\tilde{\tilde{x}}$, *i.e.*

$$\tilde{\tilde{x}} = (\bar{s}, \tilde{\tilde{u}})$$

- In many cases $\bar{s}$ is not a part of a minimum
  - No coherence with the prior knowledge

COSNet [1,2]

GAIN:

- *Positive labels* := *1*
- *Nagative labels* := *-1*
- *Thresholds* := *0*

COSNet:

- *Positive labels* := $\sin\alpha$
- *Nagative labels* := $-\cos\alpha$
- *Thresholds* := γ

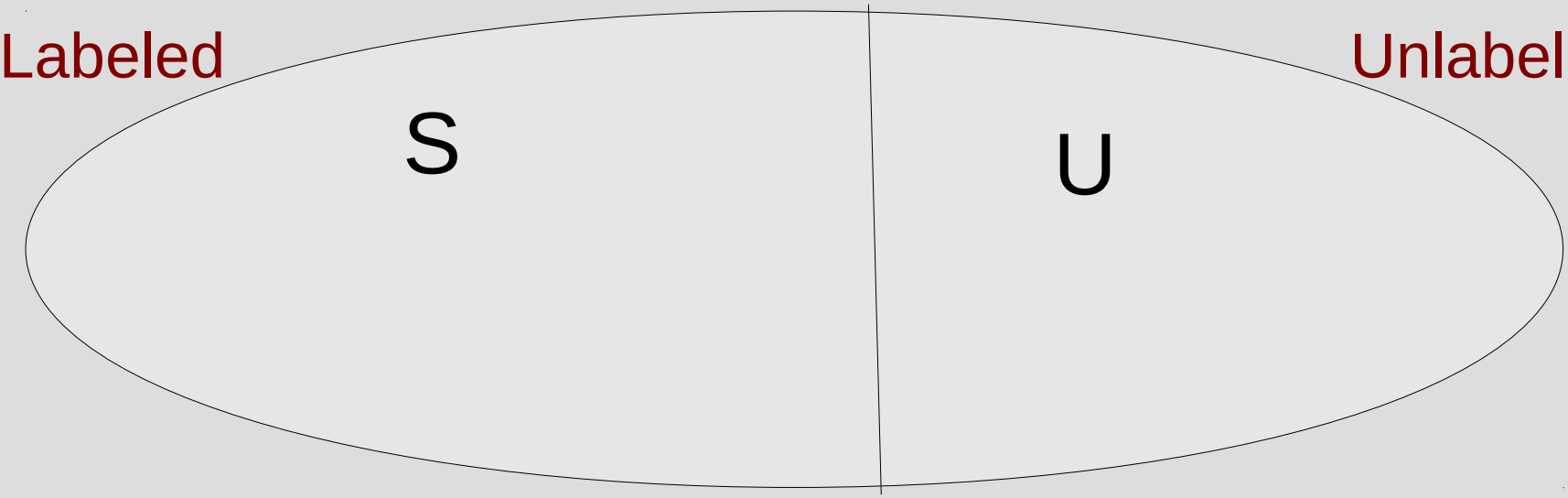
parameters to be learned!

Parametrized DHN: $< W, \gamma, \alpha >$

COSNet

- $H = < W, \underline{\gamma}, \alpha >$ DHN on nodes in V
  - W connection matrix
  - $\underline{\gamma} = (\gamma_1 , \gamma_2 , ..., \gamma_n)$ vector of activation thresholds
  - $\alpha \in]\, 0, \pi/2 \,[$, neuron values are $\sin\alpha$, $-\cos\alpha$
- In GAIN
  - $\underline{\gamma} = \underline{0}$
  - $\alpha = \pi/4$

Sub Network



Two Subnetworks: $H|_{S,U^p}$ and $H|_{U,S^p}$

Deal with Data Imbalance and prior knowledge ”coherence”

Preserve prior knowledge

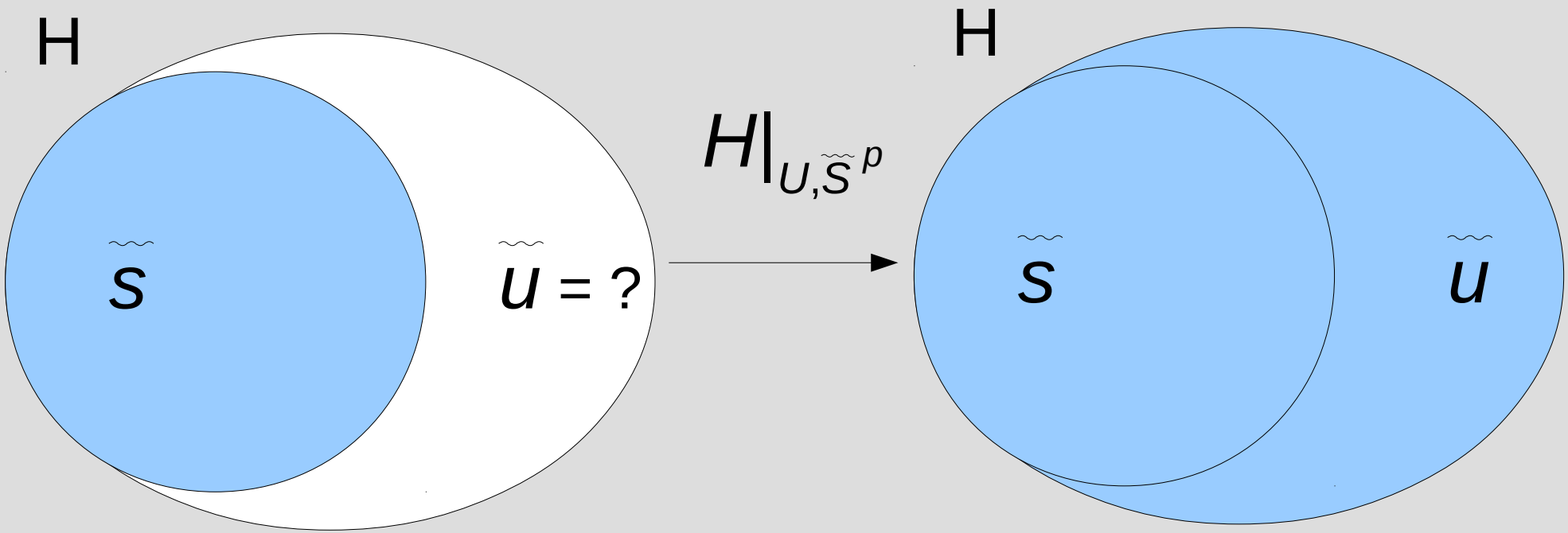
Sub-network property

- Given
  - DHN $H < W, \chi, \alpha >$ with neurons V
  - S, U bipartition of V
  - S^p, S^n bipartition of S
  - U^p, U^n bipartition of U

It holds: if $\tilde{\tilde{x}} = (\tilde{\tilde{s}}, \tilde{\tilde{u}})$ is an energy global minimum H, then $\tilde{\tilde{u}}$ is an energy global minimum of $H|_{U,S^p}$

Sub-network property

- Having a part $\tilde{\tilde{s}}$ of a minimum of energy of H, it's possible to discover the hidden part $\tilde{\tilde{u}}$ by minimizing the energy of $H|_{U,\tilde{\tilde{s}}^p}$

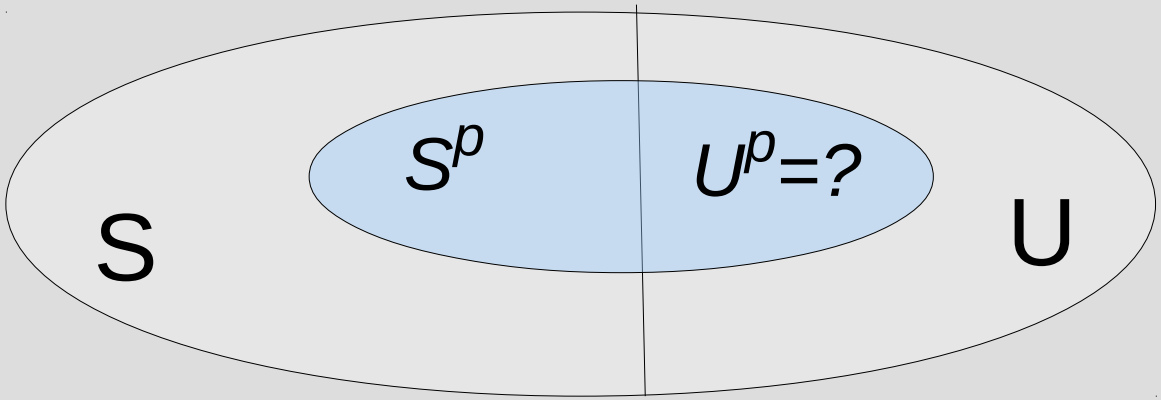


Sketch of COSNet

- INPUT: W similarity matrix; S, U bipartiton of V; S^p, S^n bipartiton of S
- OUTPUT: U^p ,U^n bipartition of U

1. Generate a temporary solution U^p ,U^n
2. Find the couple (α , γ) such that the initial state of the network $H|_{S,U^p}$ is as close as possible to an equilibrium state
    - Extend the parameters (α , γ) to the network $H|_{U,S^p}$
3. Run the network $H|_{U,S^p}$

Step 1: generating a temporary solution



p_s positive rate in S

p_u positive rate in U

Procedure:

- Generate k according to binomial distribution $B(|U|, \frac{|S^p|}{|S|})$
- U^p := k elements randomly chosen in U
- $U^n := U \setminus U^p$

FACT:

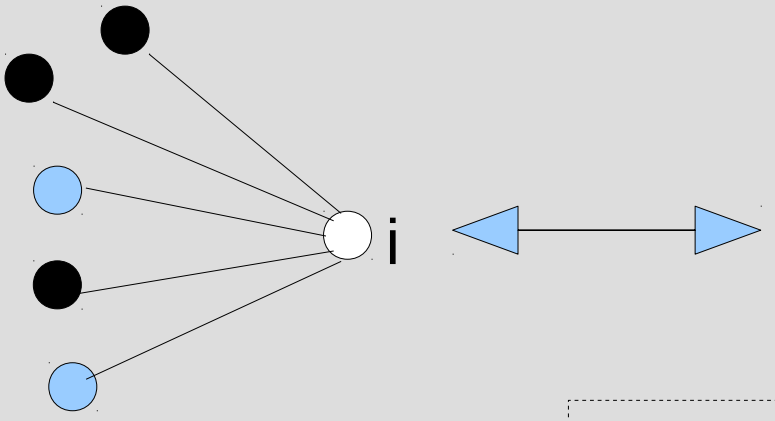
$$\frac{|S^p|}{|S|} = \underset{x}{argmax}\ Prob\left\{ p_u = x \,|\, p_s = \frac{|S^p|}{|S|} \right\}$$

Step2: finding the optimal parameters

S → Set of labeled points in R^2

L^p : positive points

L^n : negative points



$$P_i \equiv \left(\sum_{j \in S^p \cup U^p} w_{ij} \, , \sum_{j \in S^n \cup U^n} w_{ij} \right)$$

Label of P_i = label of i

AIM: ”optimal” separation of L^p from L^n by a straight line

$y = \tan\alpha \, x + q$ according to the F-score criterion

F-score

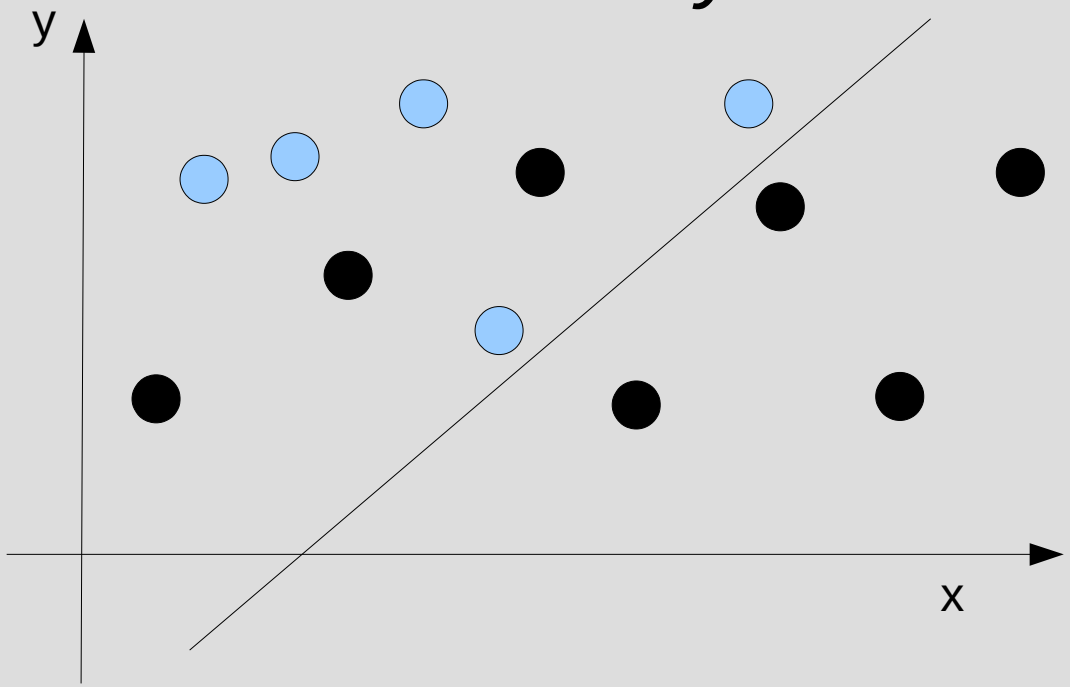
TP := Positive predicted as positive

FP := Negative predicted as positive

FN := Positive predicted as negative

- F-score := 2TP / (2TP + FN + FP)

Step2: finding the optimal parameters

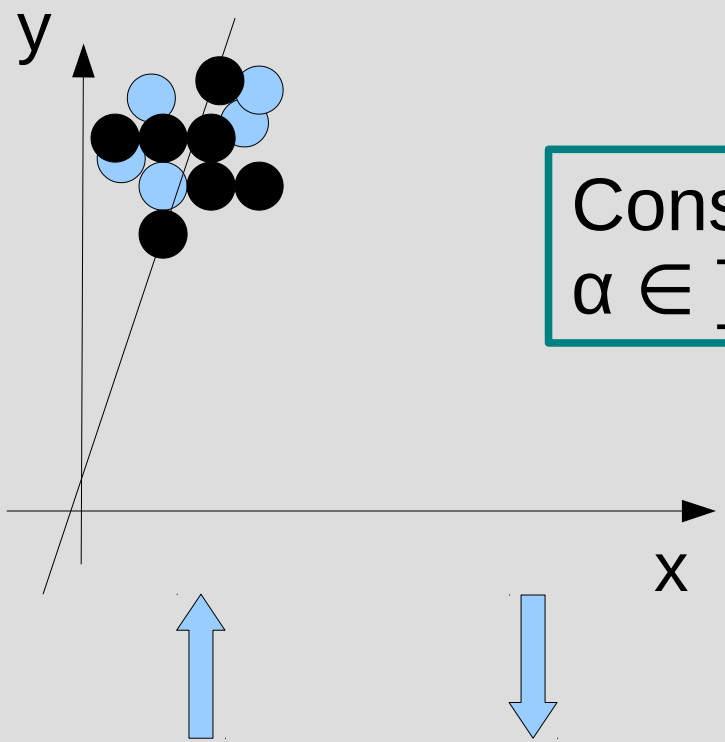


$\gamma = -q \cos\alpha$

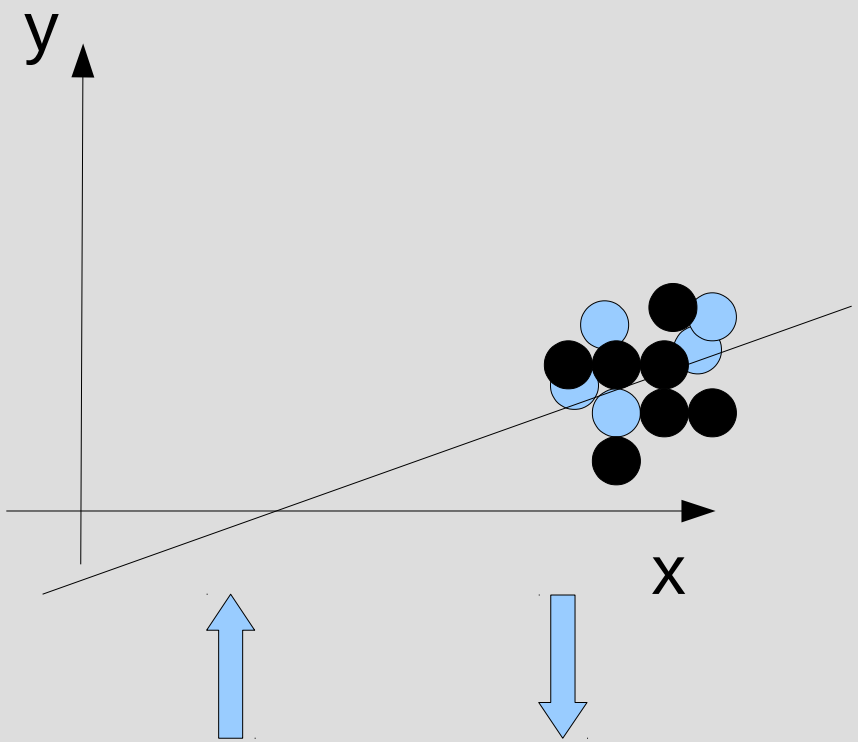
we chose the same y for each neuron

Fact: Fscore (opt) = 1 $\Leftrightarrow$ the corresponding state of $H|_{S,U^p}$ is an *equilibrium point*

Data imbalance management



Constraint:
$\alpha \in]0, \pi/2[$



$|S^p| << |S^n|$, $\alpha > \pi/4$

$|\sin\alpha| > |\cos\alpha|$

$|S^p| >> |S^n|$, $\alpha < \pi/4$

$|\sin\alpha| < |\cos\alpha|$

Step 3: finding the final solution

- Dynamics of the sub-network $H|_{U,S^p}$ with the found parameters until fixed point $\tilde{\tilde{u}}$ is reached
- Infer bipartition of U as follows:
  - $U^p = \{i \in U \mid \tilde{\tilde{u}}_i = \sin\alpha\}$
  - $U^n = \{i \in U \mid \tilde{\tilde{u}}_i = -\cos\alpha\}$

Extending the number of parameters [3]

- In COSNet all neurons have the same activation values: in principle many types of neurons may be adopted

- We consider now neurons of two types:

  Type 1: activation values $\{\sin\alpha_1, -\cos\alpha_1\}$, threshold 0

  Type 2: activation values $\{\sin\alpha_2, -\cos\alpha_2\}$, threshold 0

Set of parameters

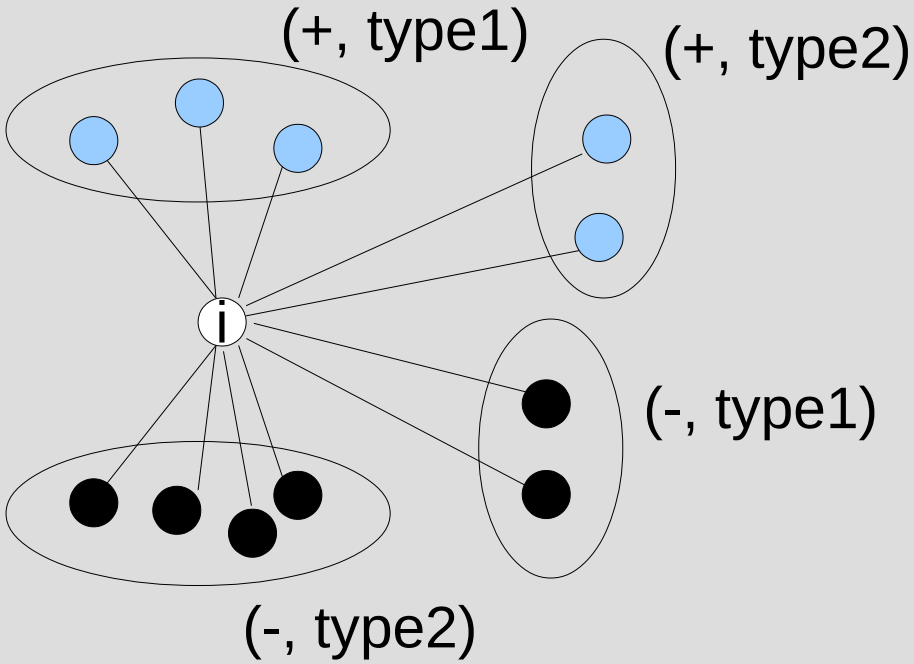
- Parameters to be learned:
  - Bipartition (G_1, G_2) of V, where
    - G_1 set of neurons of type 1
    - G_2 set of neurons of type 2

    The bipartition is described by $\mathbf{b} \in \{0, 1\}^{|V|}$, the characteristic vector of G_1

    NOTE: this partition may be passed as input to the model, hence in such case we do not learn the parameter **b**
  - α_1 and α_2

Neuron internal energy

- Network of labeled neurons (subset *S*)



Internal energy A_i:

$A_i := \mathbf{f_i}(\alpha_1, \alpha_2, b^s)$

non linear function

Learning: F-score

- Fixed α_1 , α_2 and $\boldsymbol{b}^s$:
  - TP(α_1 , α_2, $\boldsymbol{b}^s$) := {i | $A_i > 0$, i positive }
  - FN(α_1 , α_2, $\boldsymbol{b}^s$) := {i | $A_i \leq 0$, i positive }
  - FP(α_1 , α_2, $\boldsymbol{b}^s$) := {i | $A_i > 0$, i negative }
- $F_{score}(\alpha_1 , \alpha_2, \boldsymbol{b}^s) = \dfrac{2TP}{2TP + FP + FN}$
- **FACT**: $F_{score}(\alpha_1 , \alpha_2, \boldsymbol{b}^s) = 1$ sse Network of labeled neurons with parameters (α_1 , α_2, $\boldsymbol{b}^s$) is in an **equilibrium state**

Learning parameters

- Our Problem

  $\text{argmax}_{\alpha_1, \alpha_2, \boldsymbol{b}^s} \; F_{\text{score}}(\alpha_1, \alpha_2, \boldsymbol{b}^s)$

- Strategy: continuous parameters are optimized separately by the discrete ones
  - Fixed $\boldsymbol{b}^s \in \{0, 1\}^{|S|}$, compute

    $$\tilde{\tilde{\boldsymbol{\alpha}}}_1, \tilde{\tilde{\boldsymbol{\alpha}}}_2 = \text{argmax}_{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2} F_{\text{score}}(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{b}^s)$$

  - Fixed $\alpha_1 = \tilde{\tilde{\boldsymbol{\alpha}}}_1$, $\alpha_2 = \tilde{\tilde{\boldsymbol{\alpha}}}_2$, optimize $\boldsymbol{b}^s$ by local search procedure on hypercube $\{0,1\}^{|S|}$

Extending parameters to Subnetwork H_U

- **Extending the bipartition type 1 and type 2 to U**
  - Learning two **bivariate normal distributions** $N_2(\mu_1,\Sigma_1)$, $N_2(\mu_2,\Sigma_2)$ where, for $j = 1, 2$, μ_j and Σ_j sample mean vector and covariance matrix neurons of type j
  - Each sample P_r, with $r \in S$, is a point in the plane given by the sum of positive and negative connections in its lebeled neighborhood

- If $k \in U$, we set $b_k = 1$ **iif** the probability of P_k, according to $N_2(\mu_1,\Sigma_1)$, is greater than the probability of P_k, according to $N_2(\mu_2,\Sigma_2)$

Inferring the solution

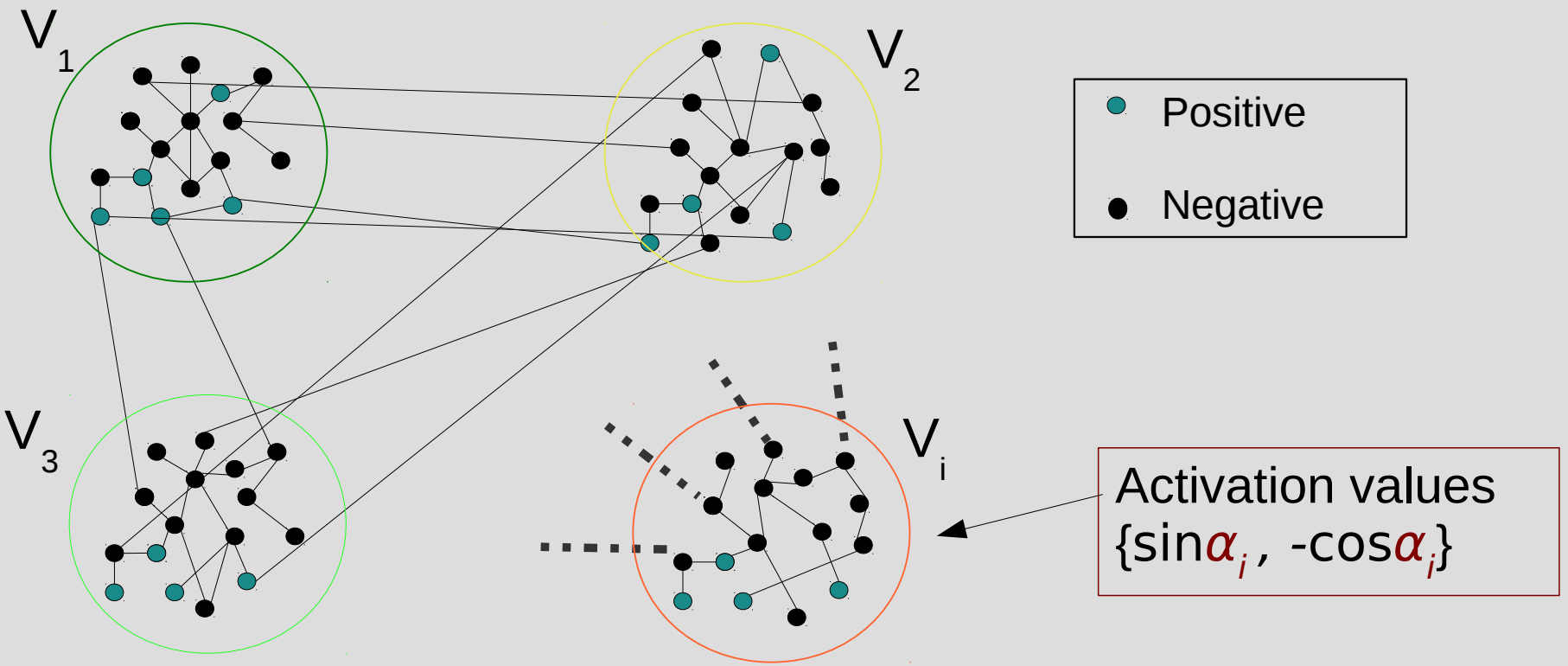
- **After extended the bipartition type 1 and type 2 to U**
  - Run the subnetwork of the unlabeled nodes with the learned parameters until the equilibrium state $\tilde{\tilde{u}}$ is reached
- The equilibrium $\tilde{\tilde{u}}$ characterizes the classification of U in positive U^p and negative neurons U^n:

$$U^p = \{i \in U \mid \tilde{\tilde{u}}_i > 0\}$$

$$U^n = \{i \in U \mid \tilde{\tilde{u}}_i \leq 0\}$$

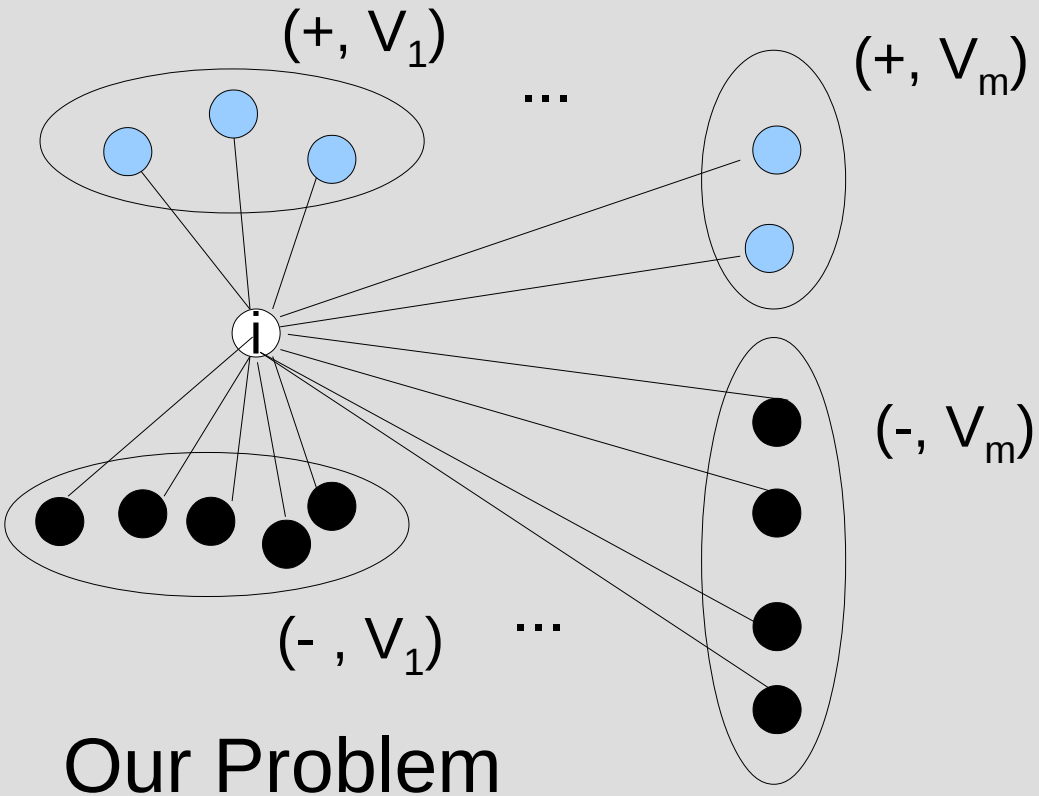
More categories: HoMCat (Hopfield Multi-Category) [4]

- With *m>1* categories we have *m* different couples of activation values $\{\sin\alpha_i, -\cos\alpha_i\}$
- The partion in categories is given in input



Homcat: neuron internal energy

- Network of labeled neurons (subset S)



Internal energy:

$A_i := \mathbf{f_i}(\alpha_1, \ldots, \alpha_m, \gamma, \boldsymbol{b})$

non linear function

Our Problem

$\text{argmax}_{\alpha_1, \ldots, \alpha_m, \gamma} \; F_{score}(\alpha_1, \ldots, \alpha_m, \gamma, \underline{b}^s)$

Homcat and multi-species protein function prediction

- A possible application of *HoMCat* is in predicting the protein functions in multi-species protein networks
  - The network contains proteins from different species
  - Proteins in different species are connected through homology
  - Each category of the model contains the proteins in one species
  - Intra-species and inter-species connections are retrieved from different data banks

Conclusions

- We studied:
  - A Cost-Sensitive method based on neural network for predicting labels in graph
- Better performance w.r.t. the state-of-the-art methods
- The time complexity $O(|S|*\log|S| + |W|)$ allows the application to nets with thousands of nodes
- We increased the number of parameters by considering two or more categories of neurons
  - Learned by the model or received in input as argument

Possible developments

- Increase the number of parameters
  - Different threshlolds for neurons or different slopes
  - Find optimal number of parameters
- Multi task extension
  - Use hierarchical relationship between terms

References

- [1] A. Bertoni, M. Frasca, G. Valentini. COSNet: a Cost Sensitive Neural Network for Semi-supervised Learning in Graphs, In: "European Conference on Machine Learning, ECML PKDD, 2011, Athens, Proceedings, Part I, Lecture Notes on Artificial Intelligence, vol. 6911, pp.219- 234, Springer.

- [2] M. Frasca, A. Bertoni, M. Re, and G. Valentini. A neural network algorithm for semi-supervised node label learning from unbalanced data. *Neural Networks*, 43:84 – 98, 2013.

- [3] M. Frasca, A. Bertoni, A. Sion. A neural procedure for Gene Function Prediction, *Neural Nets and Surroundings,Smart Innovation, Systems and Technologies*. Volume 19, 2013, pp 179-188, WIRN 2012.

- [4] Marco Frasca, Simone Bassis, and Giorgio Valentini. Learning node labels with multi-category hopfield networks. *Neural Computing and Applications*, 27(6):1677–1692, 2016.