Regularization and Optimal Multiclass Learning

Julian Asilis, Siddartha Devic, Shaddin Dughmi, Vatsal Sharan, and Shang-Hua Teng

University of Southern California

Binary & Multiclass Classification

Slides made by Julian and Sid, based on some slides by Shaddin

Binary classification: simplest type of learning problem.

- Domain $\mathcal{X}$, label set $\mathcal{Y} = \{0, 1\}$.

Binary & Multiclass Classification

Slides made by Julian and Sid, based on some slides by Shaddin

Binary classification: simplest type of learning problem.

- Domain $\mathcal{X}$, label set $\mathcal{Y} = \{0, 1\}$.
- What is learnable?
- How to learn?

Binary & Multiclass Classification

Slides made by Julian and Sid, based on some slides by Shaddin

Binary classification: simplest type of learning problem.

- Domain $\mathcal{X}$, label set $\mathcal{Y} = \{0, 1\}$.
- What is learnable? Classes of finite VC dimension.
  - Known since the '80's, [Blu+89].
- How to learn? Empirical risk minimization (ERM).

Binary & Multiclass Classification

Slides made by Julian and Sid, based on some slides by Shaddin

Binary classification: simplest type of learning problem.

- Domain $\mathcal{X}$, label set $\mathcal{Y} = \{0, 1\}$.
- What is learnable? Classes of finite VC dimension.
  - Known since the '80's, [Blu+89].
- How to learn? Empirical risk minimization (ERM).

Multiclass classification:

- Domain $\mathcal{X}$, arbitrary label set $\mathcal{Y}$ (perhaps infinite).

Binary & Multiclass Classification

Slides made by Julian and Sid, based on some slides by Shaddin

Binary classification: simplest type of learning problem.

- Domain $\mathcal{X}$, label set $\mathcal{Y} = \{0, 1\}$.
- What is learnable? Classes of finite VC dimension.
  - Known since the '80's, [Blu+89].
- How to learn? Empirical risk minimization (ERM).

Multiclass classification:

- Domain $\mathcal{X}$, arbitrary label set $\mathcal{Y}$ (perhaps infinite).
- What is learnable?
- How to learn?

Binary & Multiclass Classification

Slides made by Julian and Sid, based on some slides by Shaddin

Binary classification: simplest type of learning problem.

- Domain $\mathcal{X}$, label set $\mathcal{Y} = \{0, 1\}$.
- What is learnable? Classes of finite VC dimension.
  - Known since the '80's, [Blu+89].
- How to learn? Empirical risk minimization (ERM).

Multiclass classification:

- Domain $\mathcal{X}$, arbitrary label set $\mathcal{Y}$ (perhaps infinite).
- What is learnable? Classes of finite DS dimension [Bru+22].
  - Proven just last year (at FOCS)!
- How to learn? Not clear: ERM fails for learnable problems [DS14].

Binary & Multiclass Classification

Slides made by Julian and Sid, based on some slides by Shaddin

Binary classification: simplest type of learning problem.

- Domain $\mathcal{X}$, label set $\mathcal{Y} = \{0, 1\}$.
- What is learnable? Classes of finite VC dimension.
  - Known since the '80's, [Blu+89].
- How to learn? Empirical risk minimization (ERM).

Multiclass classification: Not so simple...

- Domain $\mathcal{X}$, arbitrary label set $\mathcal{Y}$ (perhaps infinite).
- What is learnable? Classes of finite DS dimension [Bru+22].
  - Proven just last year (at FOCS)!
- How to learn? Not clear: ERM fails for learnable problems [DS14].

PAC Learning

Learning problem defined by a **hypothesis class** $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$.

Each $h \in \mathcal{H}$ is simply a function $h : \mathcal{X} \to \mathcal{Y}$ (think a neural network, where $\mathcal{H}$ is the class of all NNs).

PAC Learning

Learning problem defined by a **hypothesis class** $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$.

Each $h \in \mathcal{H}$ is simply a function $h : \mathcal{X} \to \mathcal{Y}$ (think a neural network, where $\mathcal{H}$ is the class of all NNs).

1. Nature selects "ground truth" $h^* \in \mathcal{H}$ and distribution D over $\mathcal{X}$.

PAC Learning

Learning problem defined by a **hypothesis class** $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$.

Each $h \in \mathcal{H}$ is simply a function $h : \mathcal{X} \to \mathcal{Y}$ (think a neural network, where $\mathcal{H}$ is the class of all NNs).

1. Nature selects "ground truth" $h^* \in \mathcal{H}$ and distribution D over $\mathcal{X}$.
2. Learner receives sample $S = \big((x_1, h^*(x_1)), \ldots (x_n, h^*(x_n))\big)$, for $x_i \sim D$.

PAC Learning

Learning problem defined by a **hypothesis class** $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$.

Each $h \in \mathcal{H}$ is simply a function $h : \mathcal{X} \to \mathcal{Y}$ (think a neural network, where $\mathcal{H}$ is the class of all NNs).

1. Nature selects "ground truth" $h^* \in \mathcal{H}$ and distribution D over $\mathcal{X}$.
2. Learner receives sample $S = \big((x_1, h^*(x_1)), \ldots (x_n, h^*(x_n))\big)$, for $x_i \sim D$.
3. Learner outputs function f with small **error**

$$L_D(f) = \mathbb{P}_{x \sim D}\big(f(x) \neq h^*(x)\big).$$

PAC Learning

Learning problem defined by a **hypothesis class** $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$.

Each $h \in \mathcal{H}$ is simply a function $h : \mathcal{X} \to \mathcal{Y}$ (think a neural network, where $\mathcal{H}$ is the class of all NNs).

1. Nature selects "ground truth" $h^* \in \mathcal{H}$ and distribution D over $\mathcal{X}$.
2. Learner receives sample $S = \big((x_1, h^*(x_1)), \ldots (x_n, h^*(x_n))\big)$, for $x_i \sim D$.
3. Learner outputs function f with small **error**

$$L_D(f) = \mathbb{P}_{x \sim D}\big(f(x) \neq h^*(x)\big).$$

PAC Learning

How many samples are needed to output a function of error $\leq \epsilon$ with probability $\geq 1 - \delta$ over the randomness of S?

Empirical risk minimization

Quintessential learning algorithm: **empirical risk minimization** (ERM).

- Learner A such that $A(S) \in \mathcal{H}$ and $A(S)$ has perfect performance on sample $S = (x_i, y_i)_{i \in [n]}$ (realizability assumption).

Empirical risk minimization

Quintessential learning algorithm: **empirical risk minimization** (ERM).

- Learner A such that $A(S) \in \mathcal{H}$ and $A(S)$ has perfect performance on sample $S = (x_i, y_i)_{i \in [n]}$ (realizability assumption).
- Intuition: true error $L_D(f)$ is unknowable to A. Natural proxy is empirical risk

$$L_S(f) = \frac{1}{n} \sum_{i=1}^{n} |f(x_i) - y_i|.$$

Empirical risk minimization

Quintessential learning algorithm: **empirical risk minimization** (ERM).

- Learner A such that $A(S) \in \mathcal{H}$ and $A(S)$ has perfect performance on sample $S = (x_i, y_i)_{i \in [n]}$ (realizability assumption).
- Intuition: true error $L_D(f)$ is unknowable to A. Natural proxy is empirical risk
$$L_S(f) = \frac{1}{n} \sum_{i=1}^{n} |f(x_i) - y_i|.$$
- One kind of **proper learner**: learner that only outputs hypotheses in $\mathcal{H}$.

Some Relevant Background

- Binary classification: ERM characterizes learning.
  - I.e., ERM is near-optimal for all learnable problems.

Some Relevant Background

- Binary classification: ERM characterizes learning.
  - I.e., ERM is near-optimal for all learnable problems.
- Multiclass classification: there are learnable problems where ERM fails. In fact, any proper learner fails [DS14]!
  - To learn $\mathcal{H}$, we *need* to use functions outside of $\mathcal{H}$.

Some Relevant Background

- Binary classification: ERM characterizes learning.
  - I.e., ERM is near-optimal for all learnable problems.
- Multiclass classification: there are learnable problems where ERM fails. In fact, any proper learner fails [DS14]!
  - To learn $\mathcal{H}$, we *need* to use functions outside of $\mathcal{H}$.

High-level question

What is the "simplest" learning algorithm that learns all multiclass problems possible?

Launching Point: "Vanilla" SRM

Structural Risk Minimization

- Choose a regularizer $\psi : \mathcal{H} \to \mathbb{R}$ quantifying hypothesis complexity.
- Given labeled training data S, output $h \in \mathcal{H}$ minimizing

$$L_S(h) + \psi(h).$$

- Generalizes ERM with inductive bias for "simplicity" (user defined).
- Like ERM, gives a proper learner.

Launching Point: “Vanilla” SRM

Structural Risk Minimization

- Choose a regularizer $\psi : \mathcal{H} \to \mathbb{R}$ quantifying hypothesis complexity.
- Given labeled training data S, output $h \in \mathcal{H}$ minimizing

$$L_S(h) + \psi(h).$$

- Generalizes ERM with inductive bias for “simplicity” (user defined).
- Like ERM, gives a proper learner.
- By [DS14], must fail for some learnable classification problems.

Launching Point: "Vanilla" SRM

Structural Risk Minimization

- Choose a regularizer $\psi : \mathcal{H} \to \mathbb{R}$ quantifying hypothesis complexity.
- Given labeled training data S, output $h \in \mathcal{H}$ minimizing

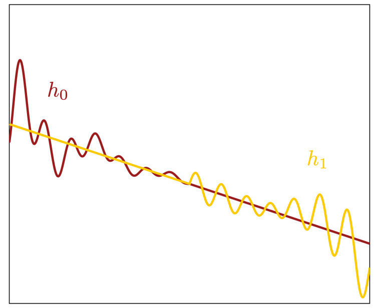
$$L_S(h) + \psi(h).$$

- Generalizes ERM with inductive bias for "simplicity" (user defined).
- Like ERM, gives a proper learner.
- By [DS14], must fail for some learnable classification problems.

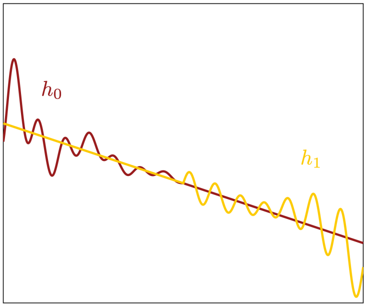
Question

What is the minimal augmentation of SRM that allows it to learn all (learnable) multiclass problems?

Relaxation 1: Local Regularization



Relaxation 1: Local Regularization



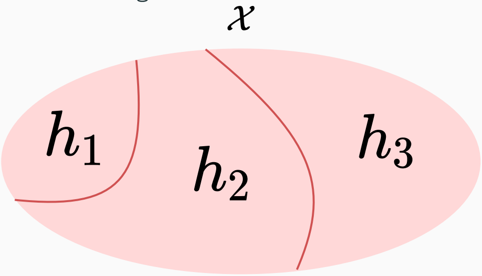
- Geometrically: $h \in \mathcal{H}$ can be "complex" at places, "simple" at others.
- Local Regularizer $\psi(h, x)$: "complexity" of h at x.

Relaxation 1: Local Regularization

- Key obstruction: SRM is inherently proper, phrased as an optimization proper over $\mathcal{H}$.
  - How to be improper while still optimizing over $\mathcal{H}$?

Relaxation 1: Local Regularization

- Key obstruction: SRM is inherently proper, phrased as an optimization proper over $\mathcal{H}$.
  - How to be improper while still optimizing over $\mathcal{H}$?
- Solution: allow regularizer to depend on test point.
  - $A(S)$ "glues" actions of different $h \in \mathcal{H}$ across $\mathcal{X}$.
  - We call this a "local regularizer."



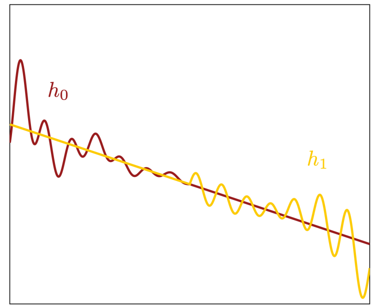
Relaxation 1: Local Regularization

- Key obstruction: SRM is inherently proper, phrased as an optimization proper over $\mathcal{H}$.
    - How to be improper while still optimizing over $\mathcal{H}$?
- Solution: allow regularizer to depend on test point.
    - $A(S)$ "glues" actions of different $h \in \mathcal{H}$ across $\mathcal{X}$.
    - We call this a "local regularizer."
- Formally, $\psi : \mathcal{H} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$,

$$A(S)(x) \in \{h(x) : h \in \operatorname{argmin}_{\mathcal{H}} L_S(h) + \psi(h, x)\}.$$

- Intuition: ψ is a collection of *local* preferences on $\mathcal{H}$, rather than a single *global* preference on $\mathcal{H}$.

Relaxation 1: Local Regularization



Theorem (Informal)

Even local regularization fails on learnable multiclass problems.

Relaxation 2: Unsupervised Learning of Regularizer

- Model complexity can be "distribution dependent"
  - h_1 varies simply over A, but with complexity over B.
  - h_2 does the opposite.
  - For $x \in A \cap B$, which of h_1 and h_2 is simpler?

Relaxation 2: Unsupervised Learning of Regularizer

- Model complexity can be "distribution dependent"
  - h_1 varies simply over A, but with complexity over B.
  - h_2 does the opposite.
  - For $x \in A \cap B$, which of h_1 and h_2 is simpler?
  - Depends on whether data distribution supported on A or B...

Relaxation 2: Unsupervised Learning of Regularizer

- Model complexity can be "distribution dependent"
  - h_1 varies simply over A, but with complexity over B.
  - h_2 does the opposite.
  - For $x \in A \cap B$, which of h_1 and h_2 is simpler?
  - Depends on whether data distribution supported on A or B...
- Unsupervised learning stage: derive ψ from unlabeled examples.

Unsupervised Local SRM

Structural Risk Minimization (Unsupervised, Local)

- Given unlabeled data S_X, learn local regularizer $\psi : \mathcal{H} \times \mathcal{X} \to \mathbb{R}$

Unsupervised Local SRM

Structural Risk Minimization (Unsupervised, Local)

- Given unlabeled data S_X, learn local regularizer $\psi : \mathcal{H} \times \mathcal{X} \to \mathbb{R}$
- Given labeled data S, test point x, find $h \in \mathcal{H}$ minimizing

$$L_S(h) + \psi(h, x)$$

and output prediction $h(x)$.

Unsupervised Local SRM

Structural Risk Minimization (Unsupervised, Local)

- Given unlabeled data S_X, learn local regularizer $\psi : \mathcal{H} \times \mathcal{X} \to \mathbb{R}$
- Given labeled data S, test point x, find $h \in \mathcal{H}$ minimizing

$$L_S(h) + \psi(h, x)$$

  and output prediction $h(x)$.

Theorem

Every realizable classification problem with countably many labels admits near optimal (factor 2) local unsupervised SRM (deterministic).

Unsupervised Local SRM

Structural Risk Minimization (Unsupervised, Local)

- Given unlabeled data S_X, learn local regularizer $\psi : \mathcal{H} \times \mathcal{X} \to \mathbb{R}$
- Given labeled data S, test point x, find $h \in \mathcal{H}$ minimizing

$$L_S(h) + \psi(h, x)$$

and output prediction $h(x)$.

Theorem

Every realizable classification problem with countably many labels admits near optimal (factor 2) local unsupervised SRM (deterministic).

BUT loses factor 2, somewhat hard to interpret, no extension to agnostic.

Result: Randomized SRM

Theorem

Every (realizable or agnostic) classification problem with finitely many labels admits exactly optimal local unsupervised SRM (randomized).

Result: Randomized SRM

Theorem

Every (realizable or agnostic) classification problem with finitely many labels admits exactly optimal local unsupervised SRM (randomized).

We suspect this extends to countably infinite $\mathcal{Y}$, maybe by compactness...

Result: Randomized SRM

Theorem

Every (realizable or agnostic) classification problem with finitely many labels admits exactly optimal local unsupervised SRM (randomized).

Admits three related interpretations:

Result: Randomized SRM

Theorem

Every (realizable or agnostic) classification problem with finitely many labels admits exactly optimal local unsupervised SRM (randomized).

Admits three related interpretations:

Interpretation 1: Bayesian

- Learns prior on hypotheses ρ from unlabeled data.
- Given labels, Bayes updates posterior ρ' on consistent hypotheses.
- Sample $h \sim \rho'$, output $h(x_{\text{test}})$.

Result: Randomized SRM

Theorem

Every (realizable or agnostic) classification problem with finitely many labels admits exactly optimal local unsupervised SRM (randomized).

Admits three related interpretations:

Interpretation 2: SRM

- Local Unsupervised SRM on randomized hypotheses.
- Learns prior $\rho = \rho(x_{\text{test}})$ from unlabeled data.
- Regularize by KL-divergence of (random) hypothesis from ρ.

Result: Randomized SRM

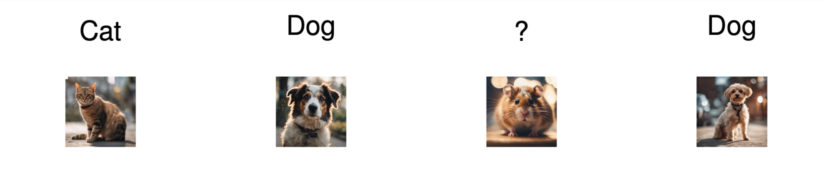
Theorem

Every (realizable or agnostic) classification problem with finitely many labels admits exactly optimal local unsupervised SRM (randomized).

Admits three related interpretations:

Interpretation 3: Maximum Entropy Principle

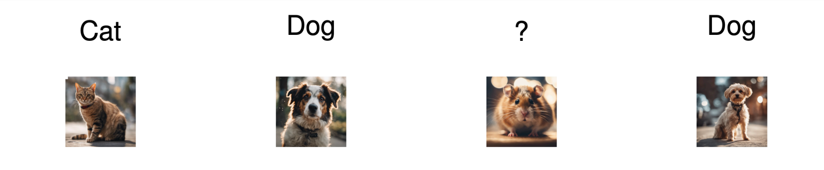
- Subject to consistency with data, choose distribution with max entropy.
    - Retains as much randomness as possible from learned prior ρ over predictions, subject to consistency with data.



Transductive Learning Model

- n adversarially chosen examples
- Exactly one label missing chosen uniformly at random (test point)
- "Fill in the blank"

Building Block: One-inclusion Graph (OIG)

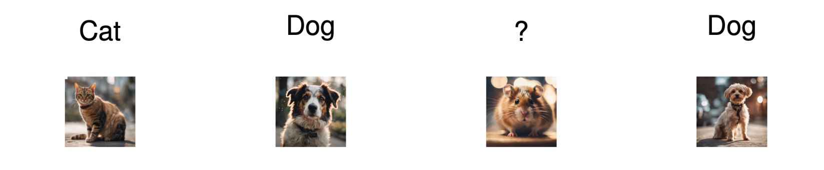


Transductive Learning Model

- n adversarially chosen examples
- Exactly one label missing chosen uniformly at random (test point)
- “Fill in the blank”

- Appears more fine grained than i.i.d. model (sample by sample)
- However, essentially equivalent to PAC model
  - $\log(1/\epsilon)$ difference in sample complexity

Building Block: One-inclusion Graph (OIG)



Transductive Learning Model

- n adversarially chosen examples
- Exactly one label missing chosen uniformly at random (test point)
- “Fill in the blank”

- Appears more fine grained than i.i.d. model (sample by sample)
- However, essentially equivalent to PAC model
  - $\log(1/\epsilon)$ difference in sample complexity

OIGs provide a mathematical handle on transductively learning $\mathcal{H}$.

One-inclusion Graph

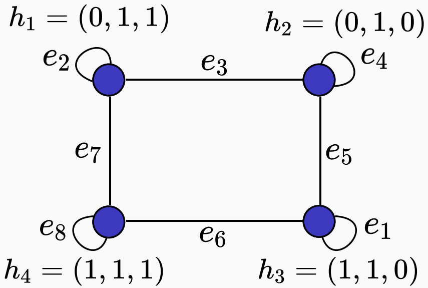
The **one-inclusion graph** of $\mathcal{H}$ on $S \in \mathcal{X}^n$ has:

- Vertex set $\mathcal{H}|_S$.
- Hyperedges group hypotheses that agree on $n - 1$ points.

One-inclusion Graph

The **one-inclusion graph** of $\mathcal{H}$ on $S \in \mathcal{X}^n$ has:

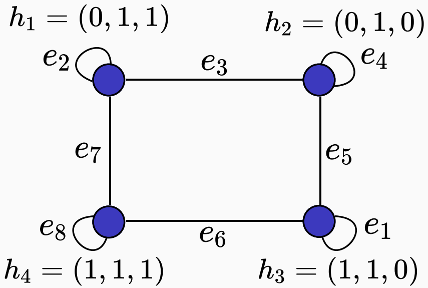
- Vertex set $\mathcal{H}|_S$.
- Hyperedges group hypotheses that agree on $n - 1$ points.



One-inclusion Graph

The **one-inclusion graph** of $\mathcal{H}$ on $S \in \mathcal{X}^n$ has:

- Vertex set $\mathcal{H}|_S$.
- Hyperedges group hypotheses that agree on $n-1$ points.

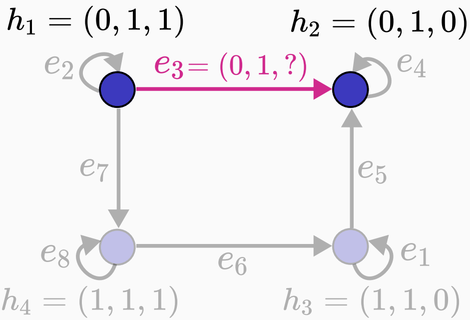


Key observation 1: learner $\equiv$ orientation of edges.

- For each observation, pick consistent hypothesis

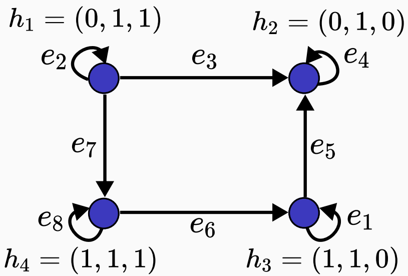
Learner $\equiv$ orientation of the graph.

- For a given test point, various hypotheses consistent with the data.
- Choose one by directing the edge.

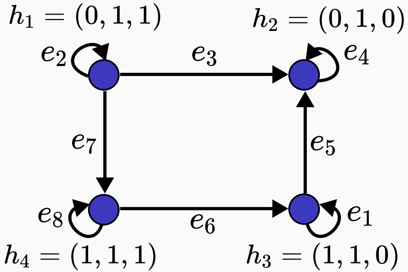


Learners and OIGs

Thus, a learner corresponds exactly to an orientation of the entire OIG (for each possible $S \in \mathcal{X}^n$).



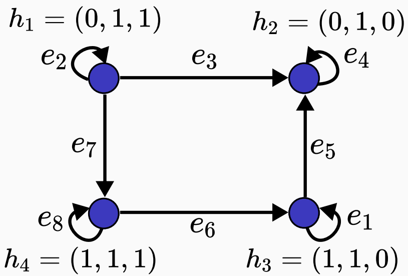
Thus, a learner corresponds exactly to an orientation of the entire OIG (for each possible $S \in \mathcal{X}^n$).



Key observation 2:

$$\text{Good learner (error } \leq \epsilon) \iff \text{outdegrees } \leq n\epsilon$$

Thus, a learner corresponds exactly to an orientation of the entire OIG (for each possible $S \in \mathcal{X}^n$).

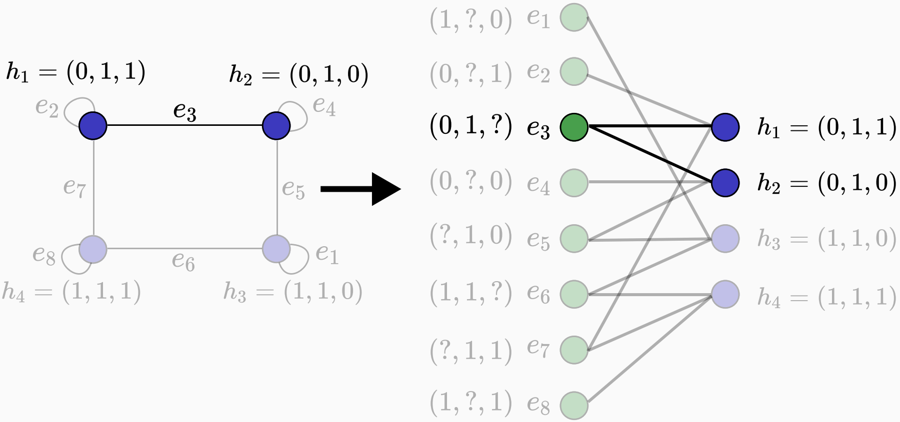


Key observation 2:

$$\text{Good learner (error } \leq \epsilon) \iff \text{outdegrees } \leq n\epsilon$$

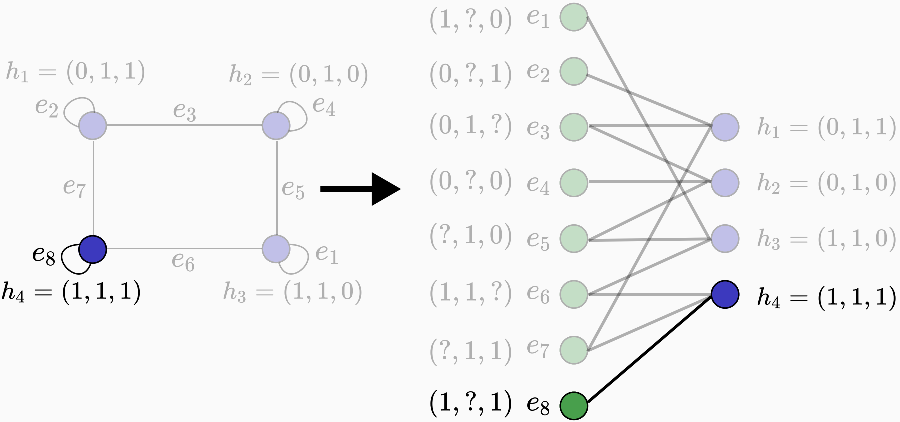
$$\iff \text{indegrees } \geq n \cdot (1 - \epsilon)$$

From the OIG G, we can derive a bipartite variant.

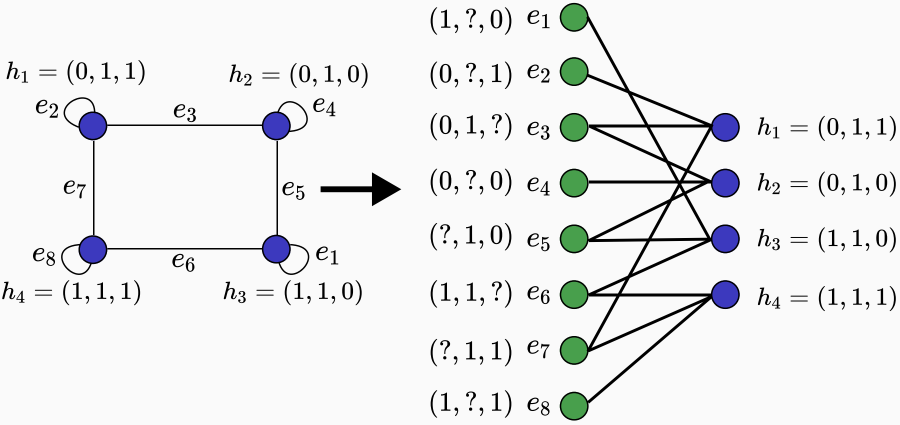


Bipartite Perspective

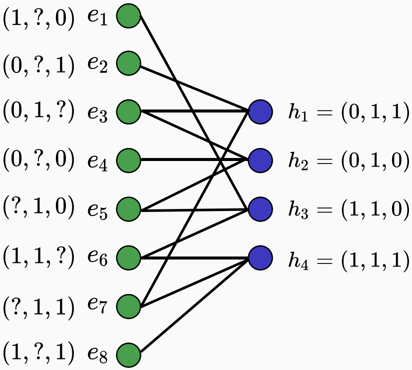
From the OIG G, we can derive a bipartite variant.



From the OIG G, we can derive a bipartite variant.

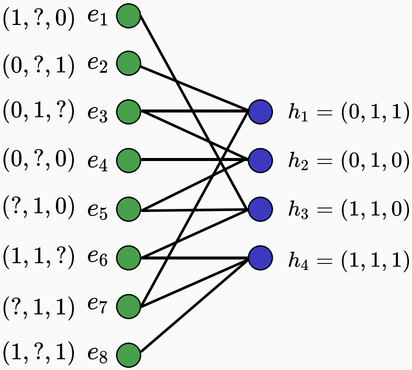


Bipartite OIGs



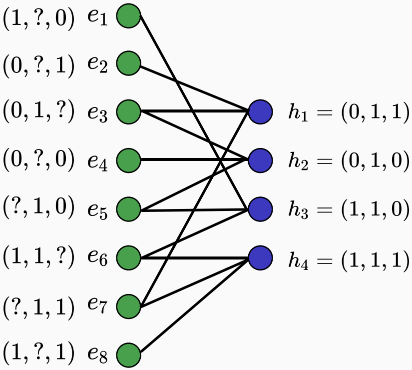
- Hyperedges (observations) on left, hypotheses on right

Bipartite OIGs



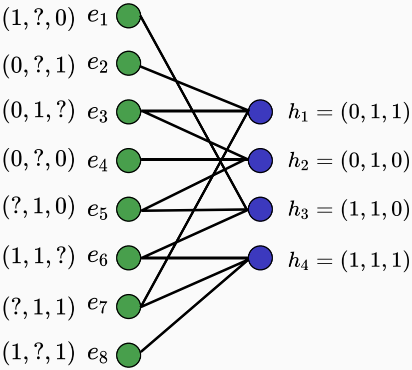
- Hyperedges (observations) on left, hypotheses on right
- Edge if hypothesis is consistent with observation

Bipartite OIGs



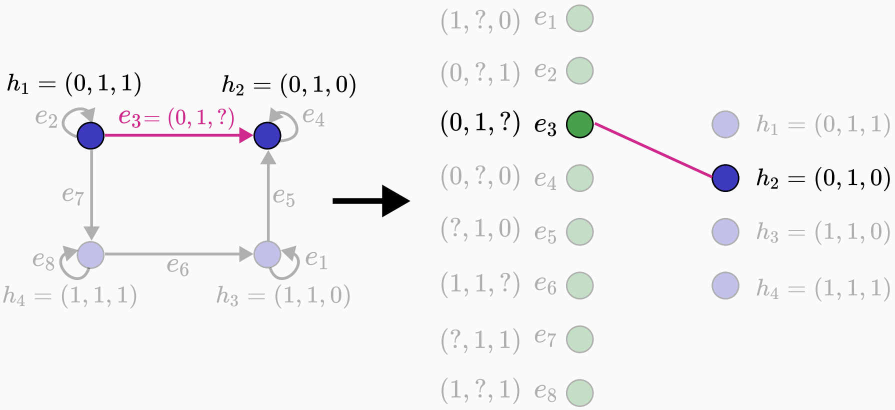
- Hyperedges (observations) on left, hypotheses on right
- Edge if hypothesis is consistent with observation
- Degrees: n on RHS, up to $|\mathcal{Y}|$ on left

Bipartite OIGs

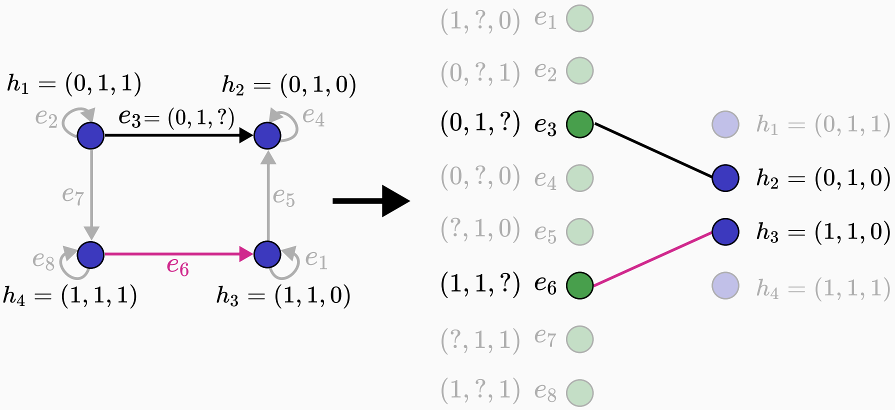


- Hyperedges (observations) on left, hypotheses on right
- Edge if hypothesis is consistent with observation
- Degrees: n on RHS, up to $|\mathcal{Y}|$ on left
- Learner $\equiv$ assignment of LHS (matching each LHS node)

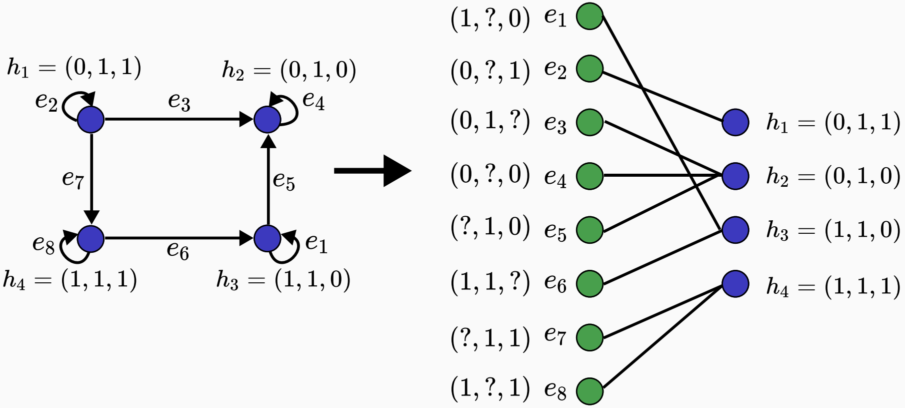
Bipartite Perspective: Learner ≡ Assignment



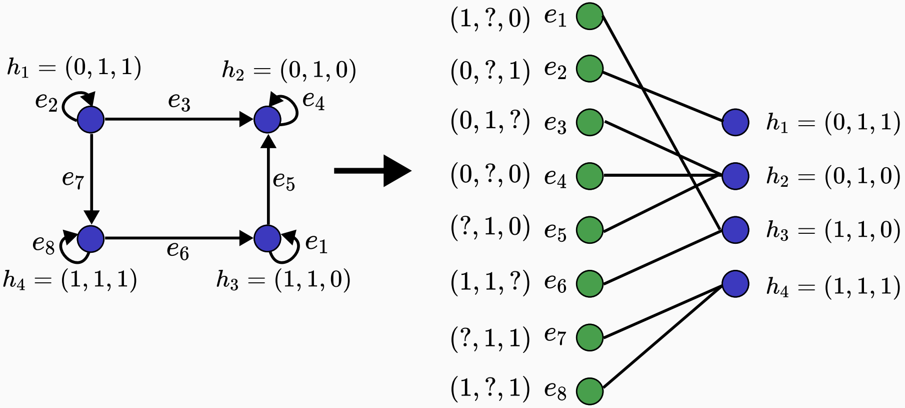
Bipartite Perspective: Learner ≡ Assignment



Bipartite Perspective: Learner $\equiv$ Assignment

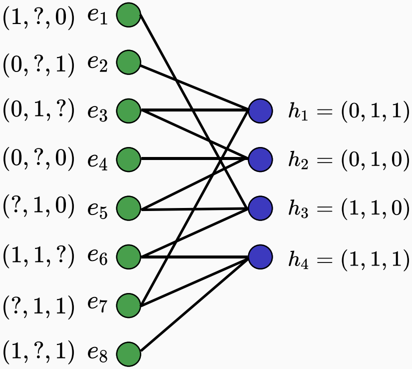


Bipartite Perspective: Learner ≡ Assignment



- Error $\leq \epsilon \iff$ each node on RHS matched $\geq n \cdot (1-\epsilon)$ times

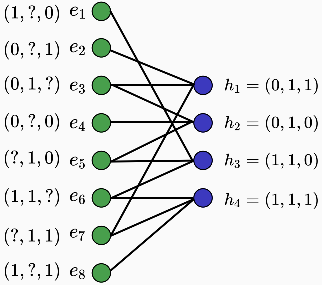
Bipartite OIGs



Progress!

Essentially converts learning into a matching problem, in the wheelhouse of graph theory / combinatorial optimization!

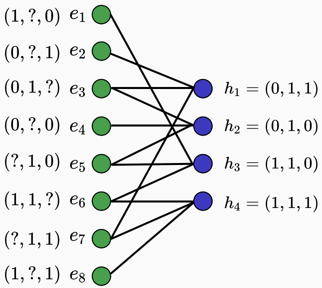
Deterministic SRM from OIGs



What does an SRM look like in terms of the bipartite OIG?

- Complexity measure ψ defines total order on the right
- Each left node picks its smallest (simplest) neighbor in terms of ψ
- Unsupervised, local: baked into OIG

Deterministic SRM from OIGs



What does an SRM look like in terms of the bipartite OIG?

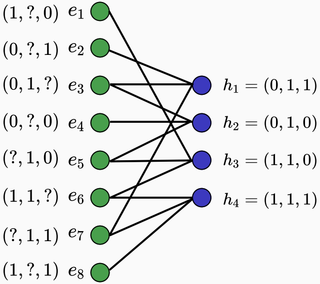
- Complexity measure ψ defines total order on the right
- Each left node picks its smallest (simplest) neighbor in terms of ψ
- Unsupervised, local: baked into OIG

Such "greedy" policies typically do well for matching, up to factor 2.

Learner implicit in [DS14] is such a "greedy" learner, hence SRM.

Randomized SRM from Bipartite OIGs

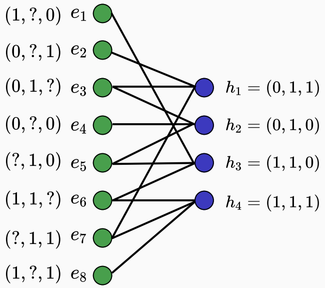
Something else we know about matching: dual variables guide you to the optimum! But for which primal?



$$\begin{aligned} \max \quad & \text{entropy}(p_M) \\ \text{s.t.} \quad & M : \text{LHS} \to \text{RHS} \\ & \deg_M(h) \geq (1-\epsilon^*)n \quad \forall h \in \mathcal{H} \end{aligned}$$

- Max entropy programs well-studied in statistical physics, optimization, approximation algorithms.
- Hard-coded to have optimal misclassification rate ϵ^*.

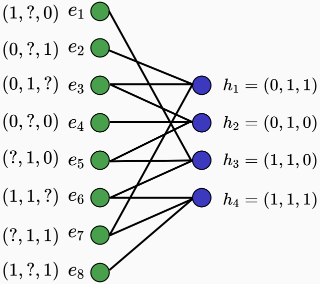
Randomized SRM from Bipartite OIGs



$$\begin{aligned} \max \quad & \text{entropy}(p_M) \\ \text{s.t.} \quad & M : \text{LHS} \to \text{RHS} \\ & \deg_M(h) \geq (1-\epsilon^*)n \quad \forall h \in \mathcal{H} \end{aligned}$$

- Max entropy programs well-studied in statistical physics, optimization, approximation algorithms.
- Dual exhibits product structure: $\Pr[\text{solution}] \propto \prod$ duals

Randomized SRM from Bipartite OIGs



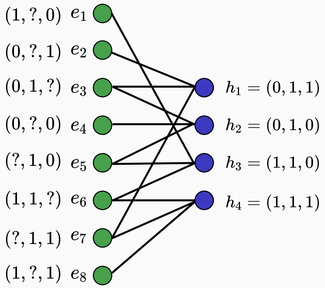
$$\begin{aligned} \max \quad & \text{entropy}(p_M) \\ \text{s.t.} \quad & M : \text{LHS} \to \text{RHS} \\ & \deg_M(h) \geq (1 - \epsilon^*)n \quad \forall h \in \mathcal{H} \end{aligned}$$

- Max entropy programs well-studied in statistical physics, optimization, approximation algorithms.
- Dual exhibits product structure: $\Pr[\text{solution}] \propto \prod \text{duals}$

Interpretation 1: Bayesian

- Normalize duals to form a prior distribution ρ on $\mathcal{H}$.
- Each $e \in \text{LHS}$ independently picks neighbor h w.p. $\propto \rho_h$

Randomized SRM from Bipartite OIGs



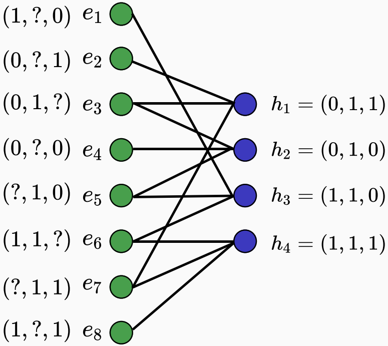
$$\begin{aligned} \max \quad & \text{entropy}(p_M) \\ \text{s.t.} \quad & M : \text{LHS} \to \text{RHS} \\ & \deg_M(h) \geq (1-\epsilon^*)n \quad \forall h \in \mathcal{H} \end{aligned}$$

- Max entropy programs well-studied in statistical physics, optimization, approximation algorithms.
- Dual exhibits product structure: $\Pr[\text{solution}] \propto \prod$ duals

Interpretations 2 and 3: SRM and Max Entropy

- Normalize duals of max entropy program to form prior ρ on $\mathcal{H}$.
- Each $e \in$ LHS chooses ρ' over neighbors minimizing $D_{KL}(\rho'|\rho)$.
    - Retain as much of the entropy of ρ as possible.

Companion Result: Hall Complexity



- Bipartite perspective allows us to characterize optimal error rate $\epsilon(n)$ using Hall's theorem [Philip Hall '35].
- Wrinkle: Hall's theorem fails for infinite graphs.
- But holds when the side you want to match has finite degrees [Marshall Hall '48], which is true for us!

Companion Result: Agnostic OIGs

- OIGs model realizable learning, we extend to agnostic.
- Extend RHS to “Hamming cube”, i.e., $\mathcal{Y}^n$.
  - Edges group strings agreeing in $n - 1$ places.
- Assignments correspond to agnostic learners.
- Discount matching requirements by Hamming distance from $\mathcal{H}$.
- Hall complexity extends naturally, as does our randomized learner.

Conclusion

- We should that our relaxation of SRM is “minimal.”
  - Removing locality gives rise to proper learners, which must fail.
  - We show some dependence of ψ on training data is necessary.

Conclusion

- We should that our relaxation of SRM is "minimal."
  - Removing locality gives rise to proper learners, which must fail.
  - We show some dependence of ψ on training data is necessary.
- BUT can't rule out that size of training data suffices.

Conjecture

Local regularizers which depend only on the size of the training data cannot learn all learnable classification problems.

Future Work

- Extend from ℓ_{0-1} to more general loss functions.
- Understand gaps between deterministic and randomized learners (realizable and agnostic).
- Resolve conjecture on local size-based regularizers.