human-interpretable machine learning & computational hardness

Elaine Angelino

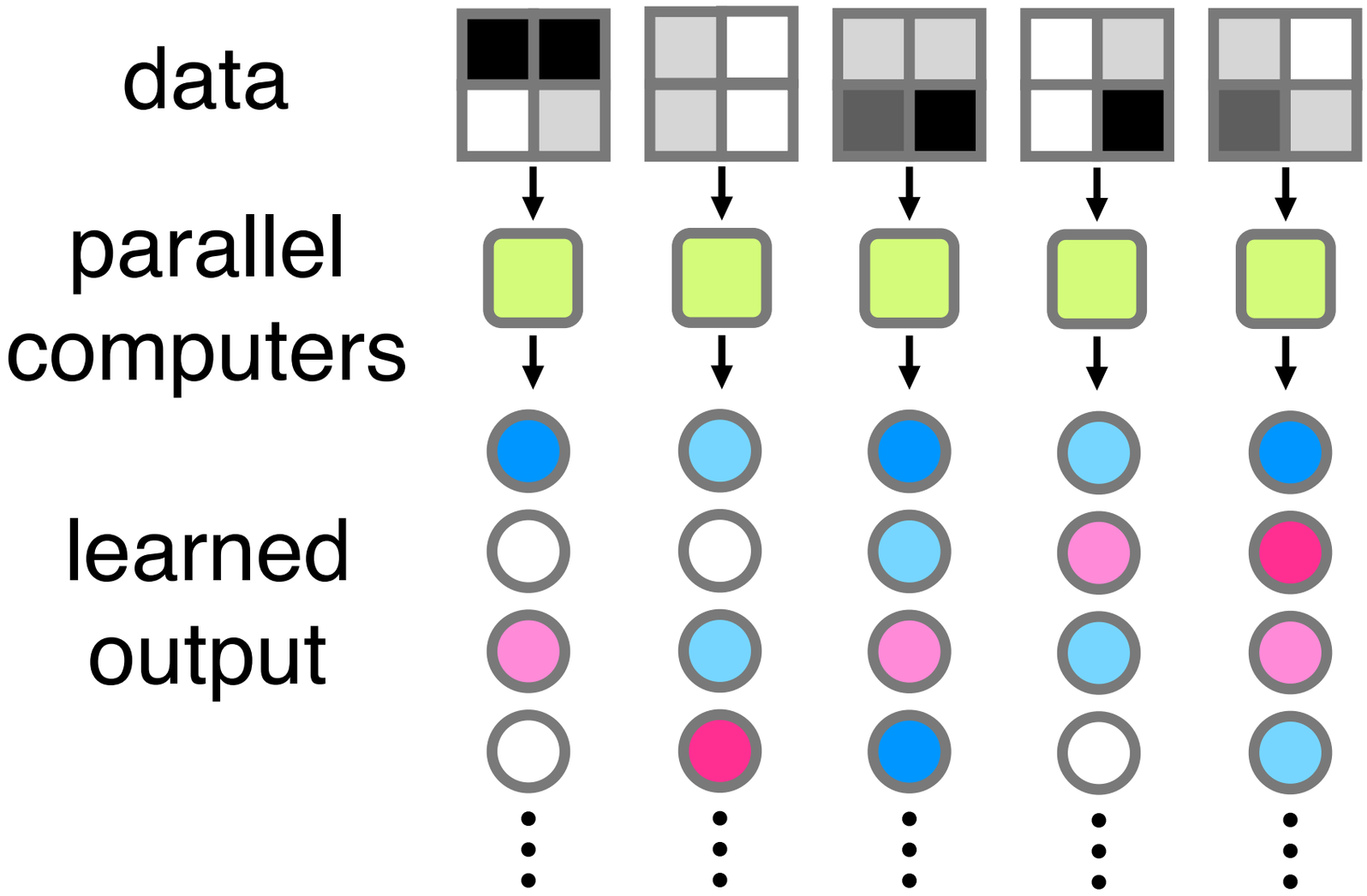
Miller Institute, UC Berkeley
April 18, 2017


machine learning is everywhere!

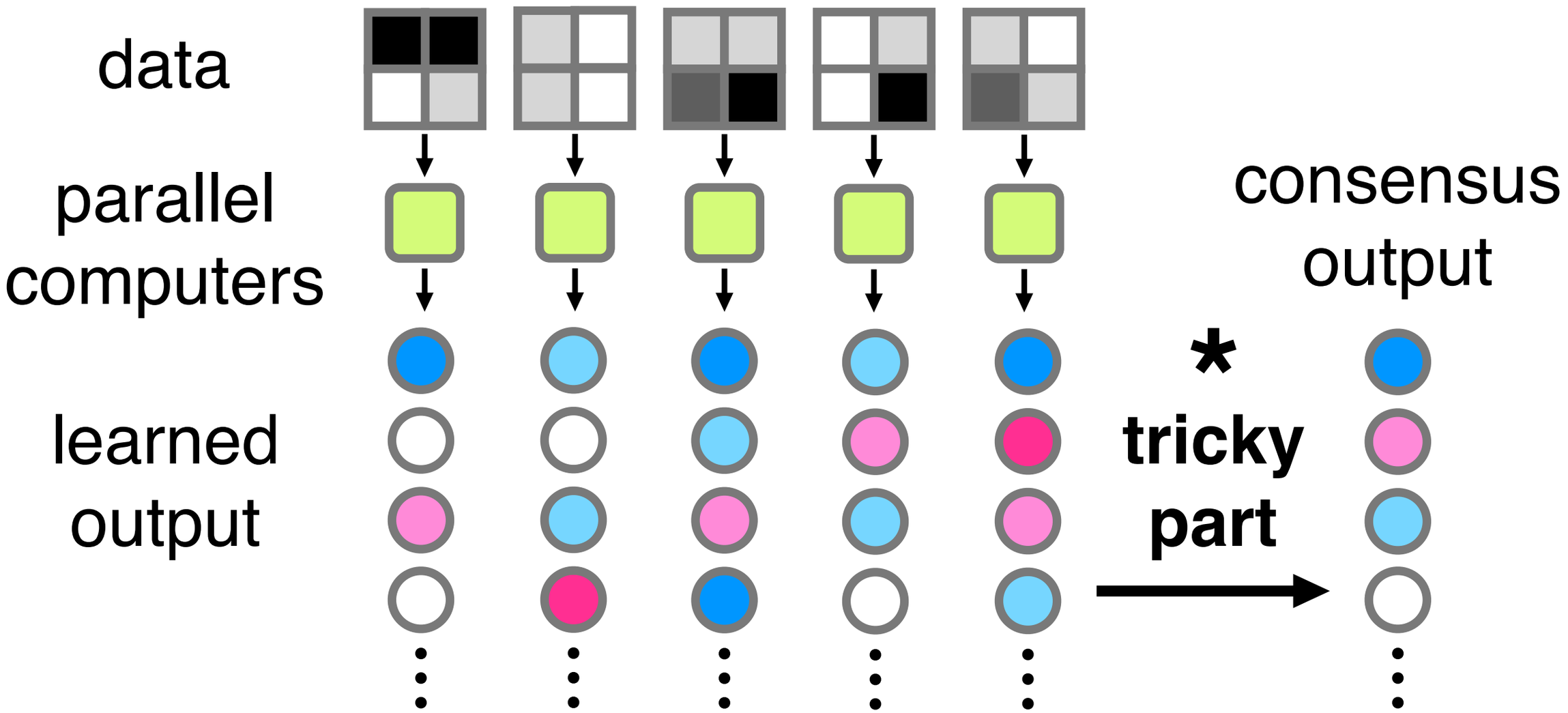
Google

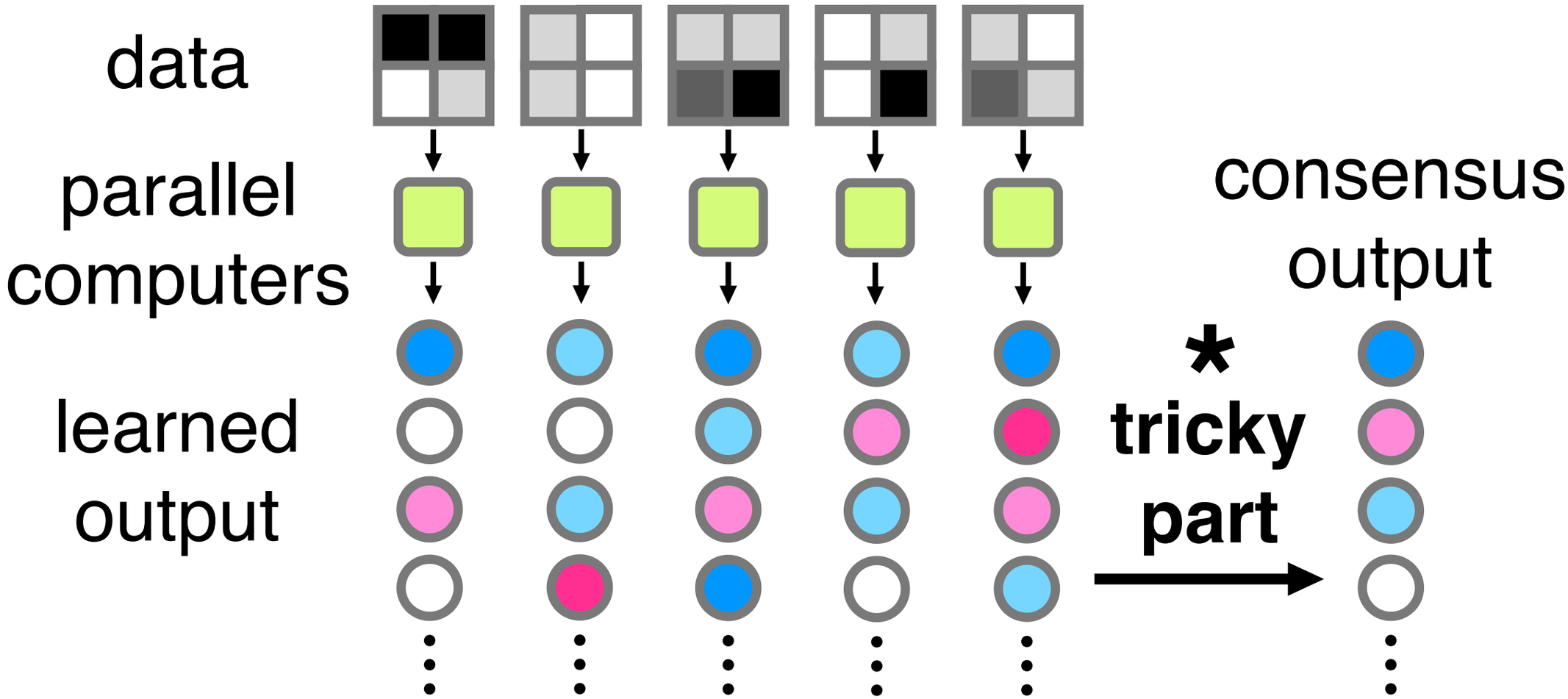
making machine learning faster

how can we exploit lots of data and lots of computers?



how can we exploit lots of data and lots of computers?





Variational consensus Monte Carlo
Neural Information Processing Systems
NIPS 2015, arXiv:1506.03074

Maxim Rabinovich

Michael I. Jordan

Patterns of scalable Bayesian inference
Foundations & Trends in Machine Learning
Now Publishers 2016, arXiv:1602.05221

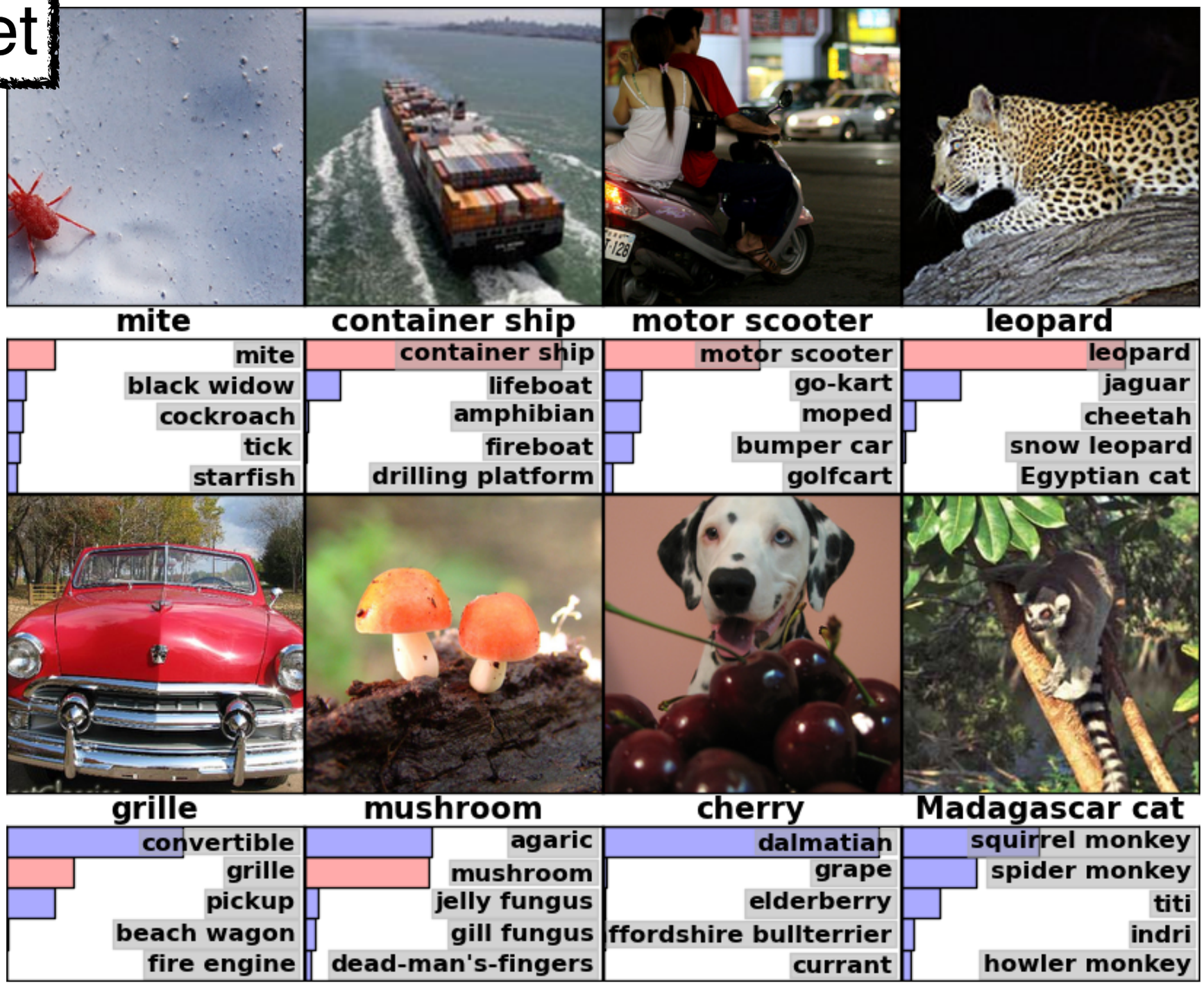
Matthew James Johnson

Ryan Adams

Google

modern machine learning is **amazing**

AlexNet

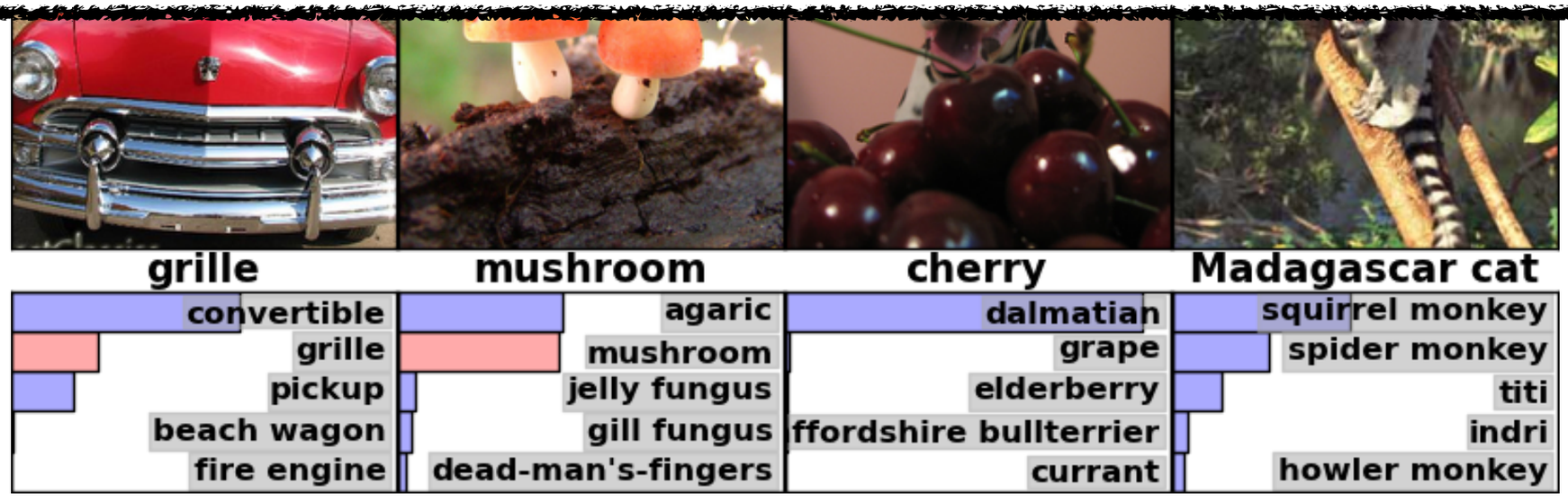


Krizhevsky et al. NIPS 2012

modern machine learning is **amazing**

AlexNet

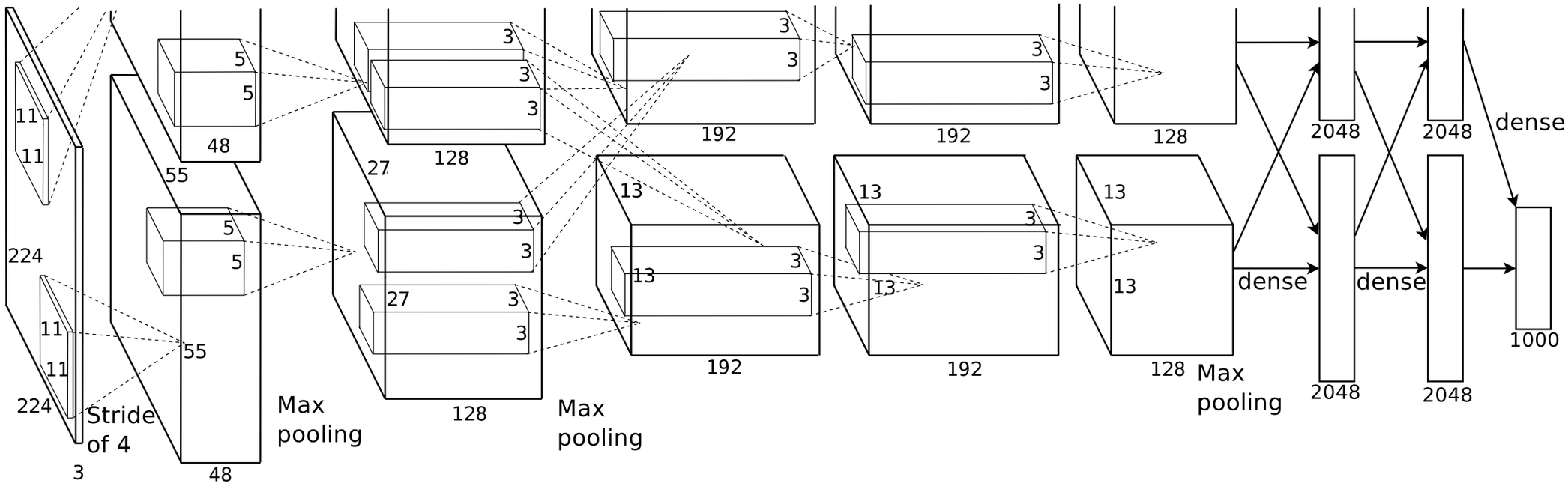
learn **function**(data) = prediction



Krizhevsky et al. NIPS 2012

modern machine learning is **amazing**

AlexNet

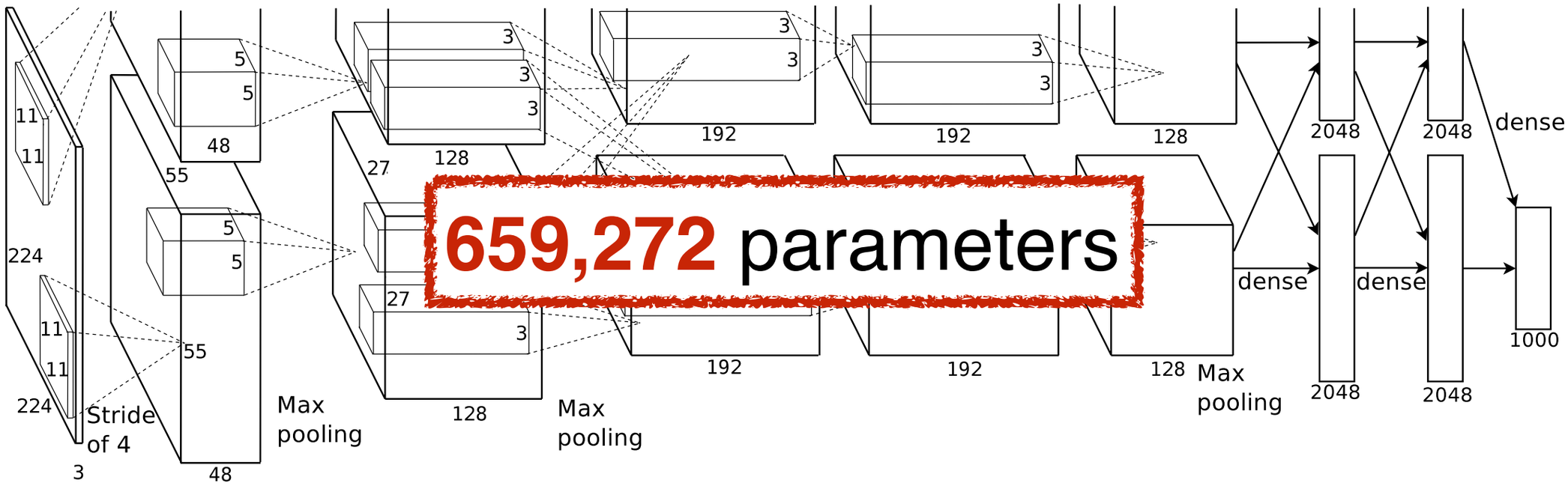


learn **function**(data) = prediction

Krizhevsky et al. NIPS 2012

modern machine learning is **amazingly complicated**

AlexNet



learn **function**(data) = prediction

Krizhevsky et al. NIPS 2012

Bank of America

NIKE

KAISER PERMANENTE®

UNDER ARMOUR

gsk

Johnson & Johnson

NETFLIX

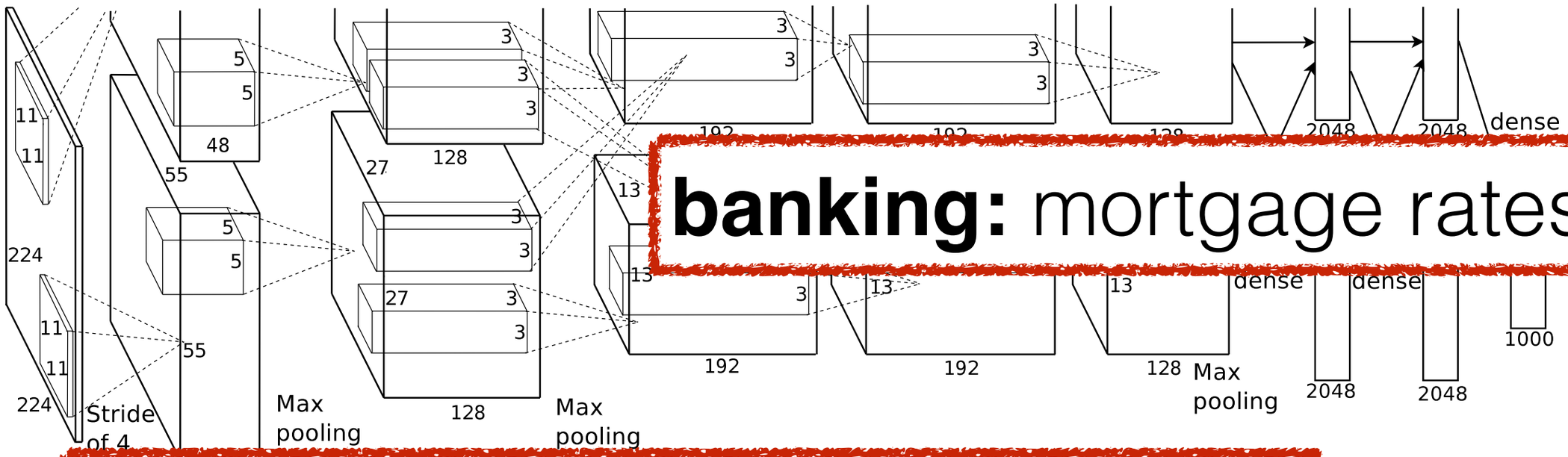
Walmart

The New York Times

CVS pharmacy™

do we want this in all areas of our lives?

medicine: treatment decisions



banking: mortgage rates

criminal justice: parole decisions

education: college admissions

human-interpretable machine learning for data-driven decision-making

human-interpretable machine learning for data-driven decision-making

for: everyone

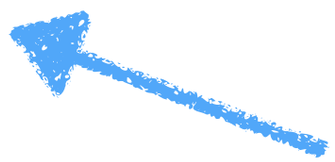
doctors, patients, scientists, educators, politicians, police, lawyers, judges, juries, credit card companies, CEOs, presidents, …

if $(age = 23 - 25)$ **and** $(priors = 2 - 3)$ **then predict** yes
else if $(age = 18 - 20)$ **then predict** yes
else if $(sex = male)$ **and** $(age = 21 - 22)$ **then predict** yes
else if $(priors > 3)$ **then predict** yes
else predict no

predict:

will the inmate reoffend within 2 years?

if $(age = 23 - 25)$ **and** $(priors = 2 - 3)$ **then predict** yes
else if $(age = 18 - 20)$ **then predict** yes
else if $(sex = male)$ **and** $(age = 21 - 22)$ **then predict** yes
else if $(priors > 3)$ **then predict** yes
else predict no



rule list

if $(age = 23 - 25)$ **and** $(priors = 2 - 3)$ **then predict** yes
else if $(age = 18 - 20)$ **then predict** yes
else if $(sex = male)$ **and** $(age = 21 - 22)$ **then predict** yes
else if $(priors > 3)$ **then predict** yes
else predict no

= most accurate

how do you find **the best rule list**
in a reasonable amount of time?

if $(age = 23 - 25)$ **and** $(priors = 2 - 3)$ **then predict** yes
else if $(age = 18 - 20)$ **then predict** yes
else if $(sex = male)$ **and** $(age = 21 - 22)$ **then predict** yes
else if $(priors > 3)$ **then predict** yes
else predict no

= most accurate

how do you find **the best rule list**
in a reasonable amount of time?

Learning certifiably optimal rule lists for categorical data
arXiv:1704.01701

Nicholas Larus-Stone

Daniel Alabi

Margo Seltzer

Cynthia Rudin

why do we care about interpretable models?

- trust in decision-making
- troubleshooting
- sharing & explaining
- can be used in the field

why do we care about interpretable models?

- trust in decision-making
- troubleshooting
- sharing & explaining
- can be used in the field

detecting & preventing model bias (racism)

why do we care about interpretable models?

- trust in decision-making
- troubleshooting
- sharing & explaining
- can be used in the field

NYC police stop-and-frisk policy

if the police frisk someone, will they find a weapon?

why do we care about interpretable models?

- trust in decision-making
- troubleshooting
- sharing & explaining
- can be used in the field

NYC police stop-and-frisk policy

if the police frisk someone, will they find a weapon?

full model **7,705** variables

$$\mathbb{P}(y_i = 1) = \text{logit}^{-1}\left(\sum_k \alpha_k x_{k,i} + \sum_{k \leq \ell} \beta_{k,\ell} x_{k,i} x_{\ell,i}\right)$$

not interpretable

Goel et al. Ann. Appl. Stat. 2016.

why do we care about interpretable models?

- trust in decision-making
- troubleshooting
- sharing & explaining
- can be used in the field

NYC police stop-and-frisk policy

if the police frisk someone, will they find a weapon?

reduced model **98** variables

$$\mathbb{P}(y_i = 1) = \text{logit}^{-1}\left(\sum_{j=1}^{18} \alpha_j a_{j,i} + \sum_{k=1}^{77} \beta_k b_{k,i} + \sum_{\ell=1}^{3} \gamma_\ell c_{\ell,i}\right)$$

still not interpretable

Goel et al. Ann. Appl. Stat. 2016.

why do we care about interpretable models?

- trust in decision-making
- troubleshooting
- sharing & explaining
- can be used in the field

NYC police stop-and-frisk policy

if the police frisk someone, will they find a weapon?

heuristic model **interpretable**

$$\sum_{j=1}^{3} \tilde{\alpha}_j a_{j,i} \geq T_r \qquad \text{where } T_r = T - \sum_{k=1}^{77} \beta_k b_{k,i} - \sum_{\ell=1}^{3} \gamma_\ell c_{\ell,i}$$

3 variables + threshold

Goel et al. Ann. Appl. Stat. 2016.

if the police frisk someone, will they find a weapon?

“… officers simply need to add at most three small, positive integers …”

3 x (suspicious object)
\+ (sights & sounds of criminal activity)
\+ (suspicious bulge)

Goel et al. Ann. Appl. Stat. 2016.

if the police frisk someone, will they find a weapon?

"… officers simply need to add at most three small, positive integers …"

{ 1 if stop circumstance noted
{ 0 otherwise

3 x (suspicious object)
\+ (sights & sounds of criminal activity)
\+ (suspicious bulge)

0, 1, 2, 3, 4, or 5 … Frisk if >= threshold

Goel et al. Ann. Appl. Stat. 2016.

if the police frisk someone, will they find a weapon?

if $(location = transit\ authority)$ **then predict** yes
else if $(stop\ reason = suspicious\ object)$ **then predict** yes
else if $(stop\ reason = suspicious\ bulge)$ **then predict** yes
else predict no

Angelino et al. 2017.

3 x (suspicious object)
\+ (sights & sounds of criminal activity)
\+ (suspicious bulge)

0, 1, 2, 3, 4, or 5 ... Frisk if >= threshold

Goel et al. Ann. Appl. Stat. 2016.

rule lists are simple! what's computationally hard?

rule lists are simple! what's computationally hard?

- the number of possible rule lists is **huge**
- traditional algorithms are either **greedy** or **randomly search** the space
- they **can't** find the best rule list

- finding the best rule list is a difficult combinatorial problem
- but sometimes we want optimality

- finding the best rule list is a difficult combinatorial problem
- but sometimes we want optimality
    - societal implications
    - medical outcomes
    - legal requirements

e.g., the European Union's regulations on algorithmic decision-making on the **"right to an explanation"**

the number of possible rule lists is **yuuuge**

→ recidivism problem: start with **155** possible rules

… naïve algorithm considers rule lists up to **length 10**

the number of possible rule lists is **yuuuge**

recidivism problem: start with **155** possible rules

… naïve algorithm considers rule lists up to **length 10**

→ … naïve search space ~ **6×10^{21}** rule lists

the number of possible rule lists is **yuuuge**

recidivism problem: start with **155** possible rules

… naïve algorithm considers rule lists up to **length 10**

… naïve search space ~ **6 x 10^{21}** rule lists

… ~ **2 microseconds** each = **500,000 / second**

the number of possible rule lists is **yuuuge**

recidivism problem: start with **155** possible rules

… naïve algorithm considers rule lists up to **length 10**

… naïve search space ~ **6 x 10^{21}** rule lists

… ~ **2 microseconds** each = **500,000 / second**

… that's ~ **300 million years**

the number of possible rule lists is **yuuuge**

recidivism problem: start with **155** possible rules

… naïve algorithm considers rule lists up to **length 10**

… naïve search space ~ **6 x 10^{21}** rule lists

… ~ **2 microseconds** each = **500,000 / second**

… that's ~ **300 million years**

spoiler alert: our algorithm is 33 trillion X faster

Angelino et al. 2017.

the number of possible rule lists is **yuuuge**

recidivism problem: start with **155** possible rules

… naïve algorithm considers rule lists up to **length 10**

… naïve search space ~ **6 x 10^{21}** rule lists

… ~ **2 microseconds** each = **500,000 / second**

… that's ~ **300 million years**

spoiler alert: our algorithm is 33 trillion X faster

… we only have to evaluate ~ **180 million** rule lists

… we find an optimal solution **< 10 seconds**

… and certify optimality in ~ **6 minutes**

Angelino et al. 2017.

how do we outperform brute-force search?

- prove exploitable facts (theorems) about the space of rule lists
- design efficient data structures
- write fast code

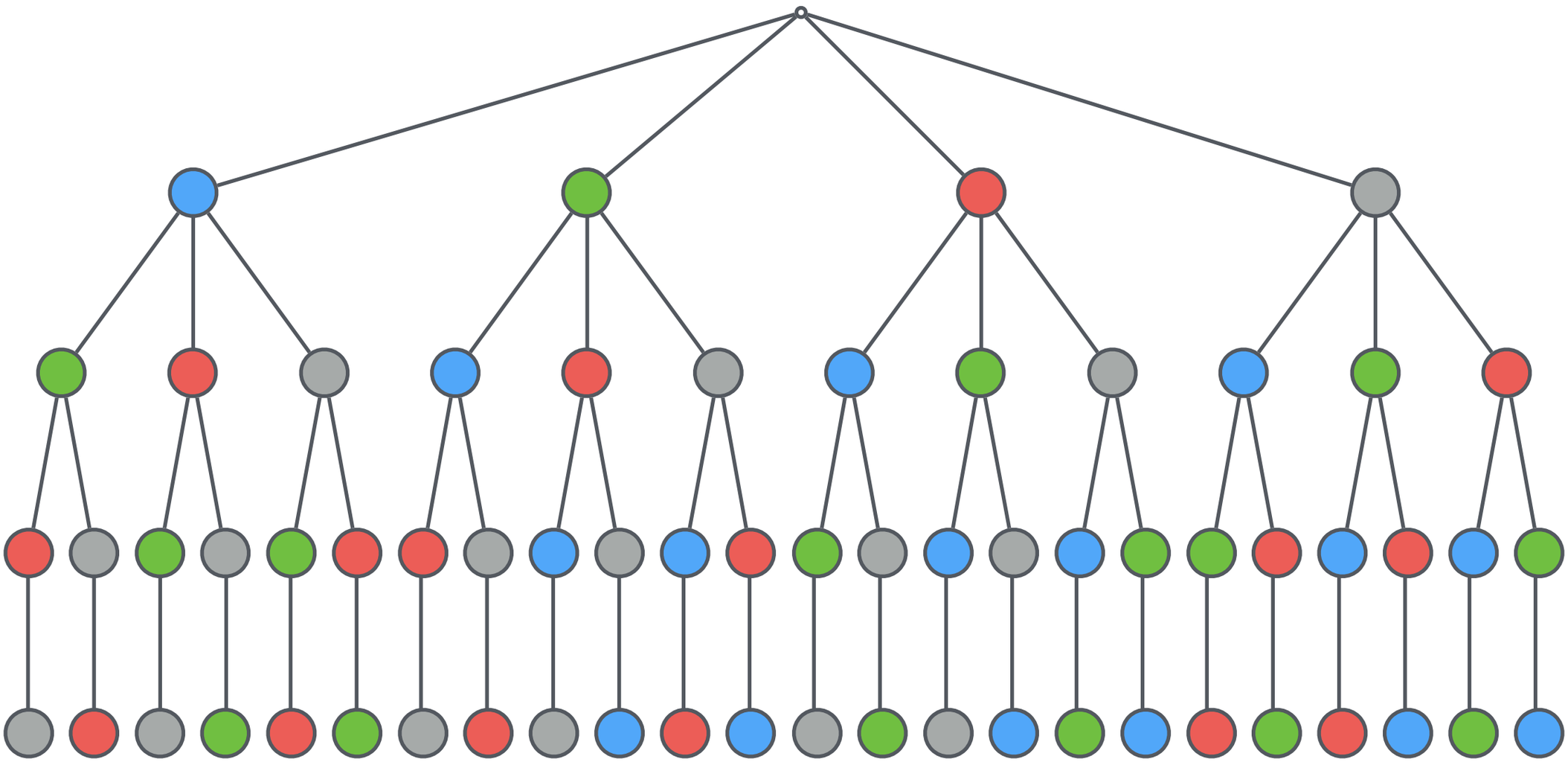
how do we outperform brute-force search?

- **prove exploitable facts (theorems) about the space of rule lists**
- design efficient data structures
- write fast code

the space of rule lists has rich structure

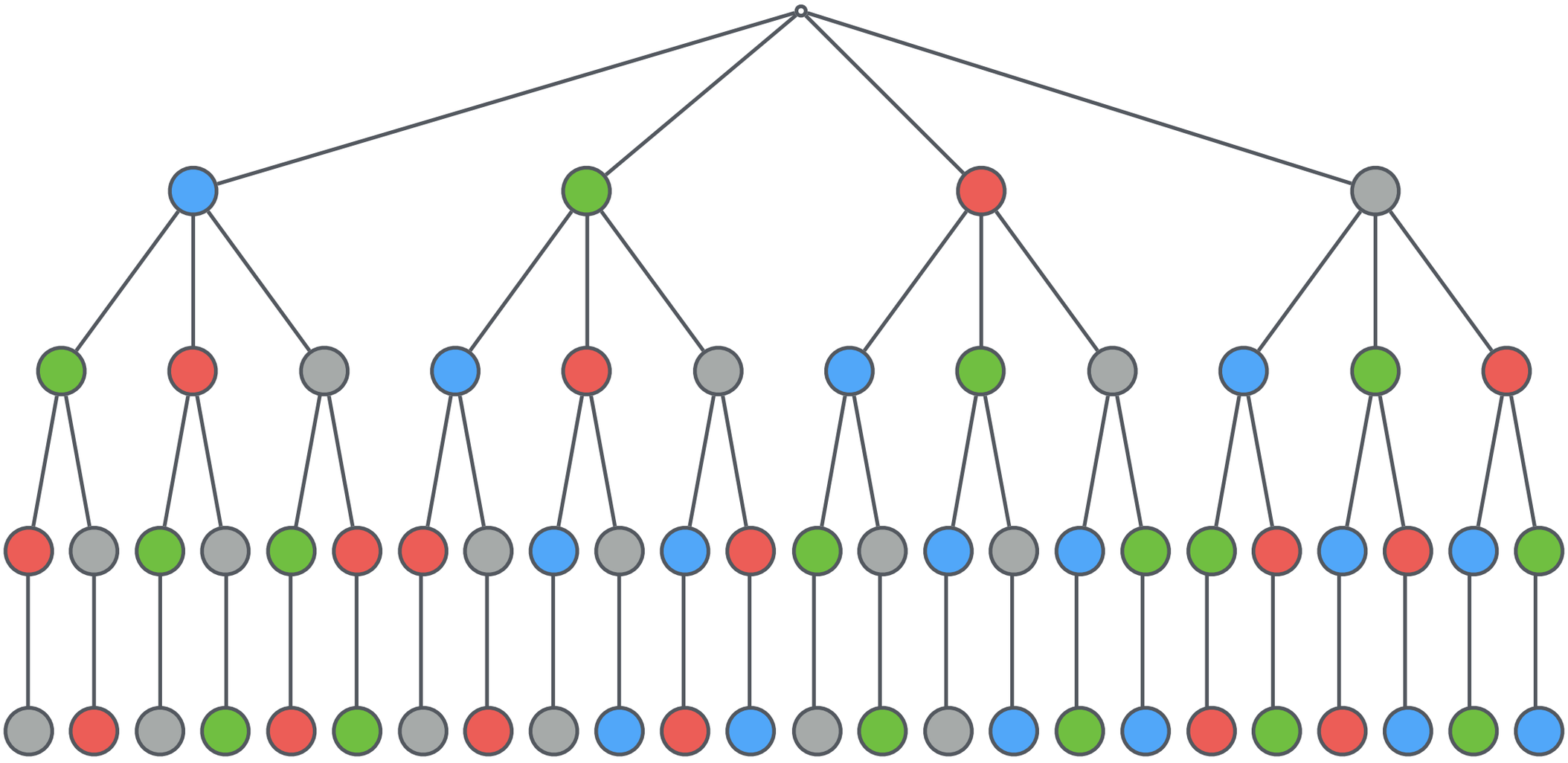
view the space of all rule lists as a tree

(each node is a rule)



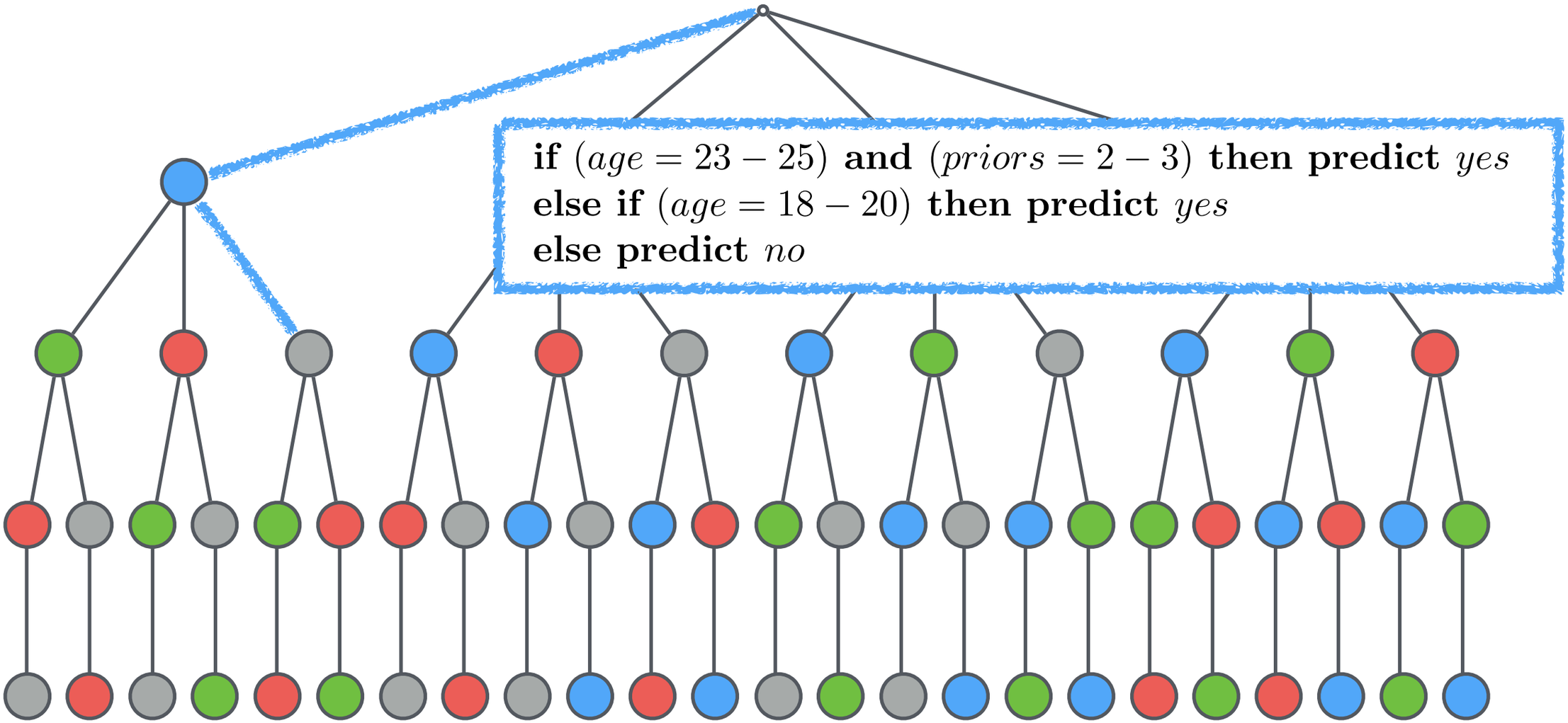
view the space of all rule lists as a tree

→ **each path encodes a rule list**



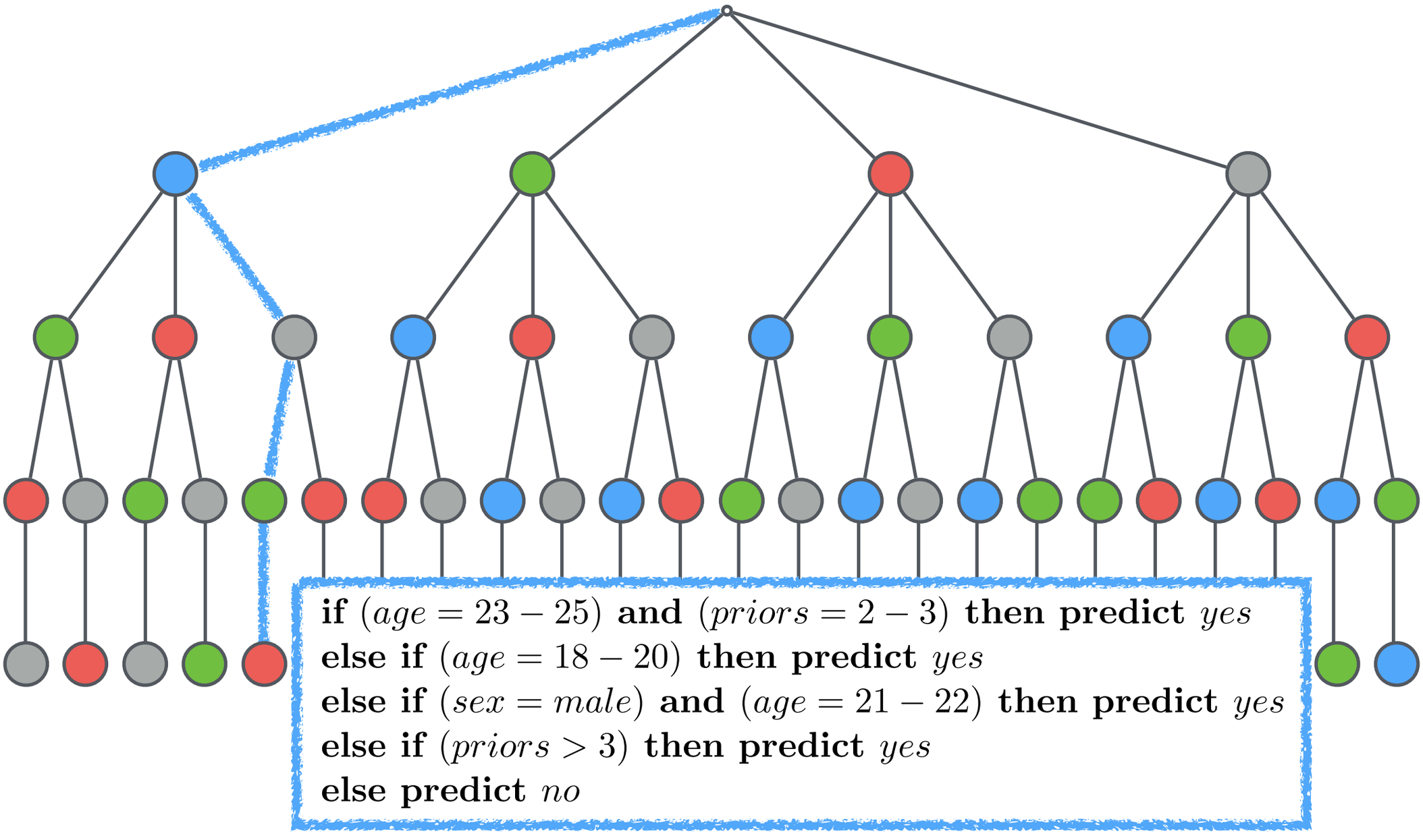
view the space of all rule lists as a tree

→ each path encodes a rule list



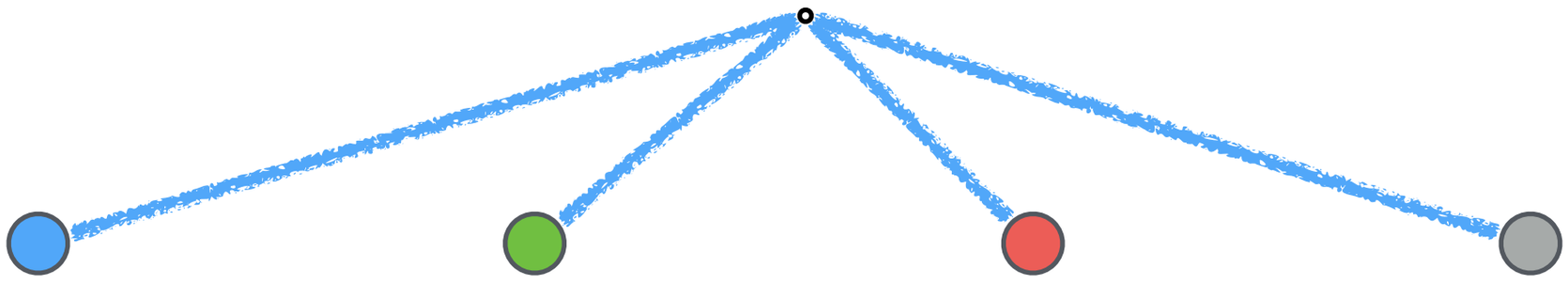
view the space of all rule lists as a tree

→ each path encodes a rule list



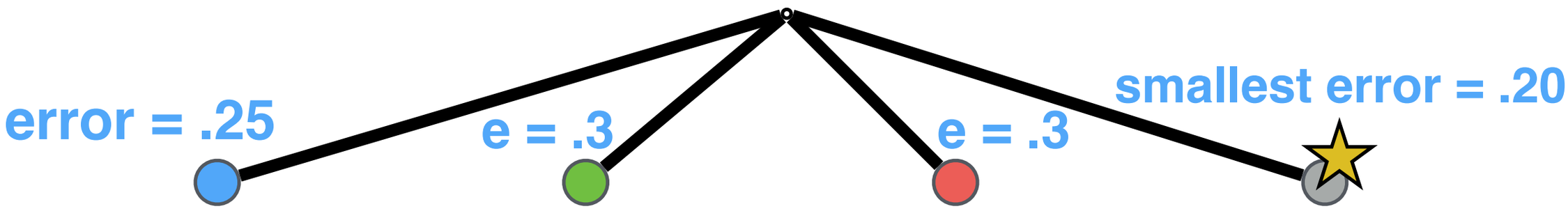
if $(age = 23 - 25)$ **and** $(priors = 2 - 3)$ **then predict** yes
else if $(age = 18 - 20)$ **then predict** yes
else if $(sex = male)$ **and** $(age = 21 - 22)$ **then predict** yes
else if $(priors > 3)$ **then predict** yes
else predict no

algorithm: systematically explore the tree, eliminating large regions along the way



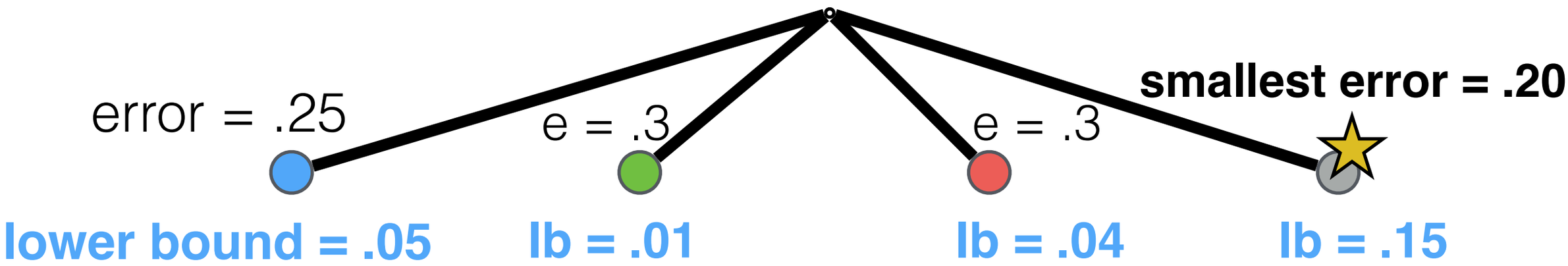
algorithm: systematically explore the tree, eliminating large regions along the way

step 1A: calculate error for all rule lists of length 1 & note best

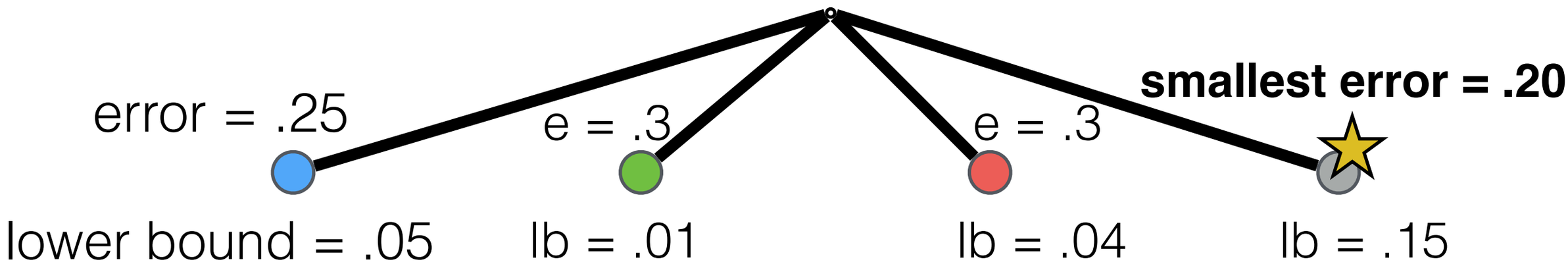


algorithm: systematically explore the tree, eliminating large regions along the way

step 1B: calculate a **lower bound** on longer rule lists

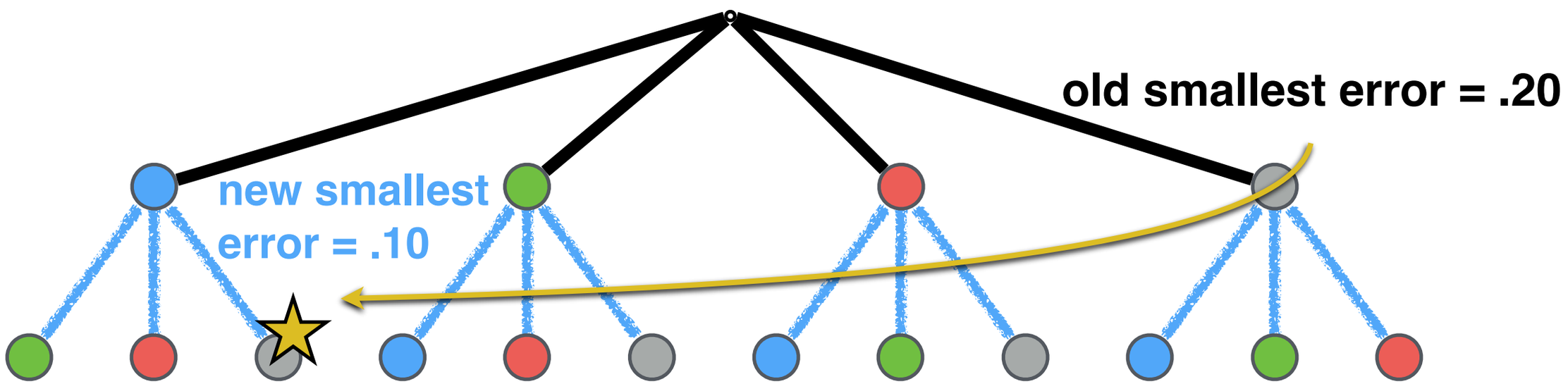


step 2: delete rule lists with lower bound >= smallest error



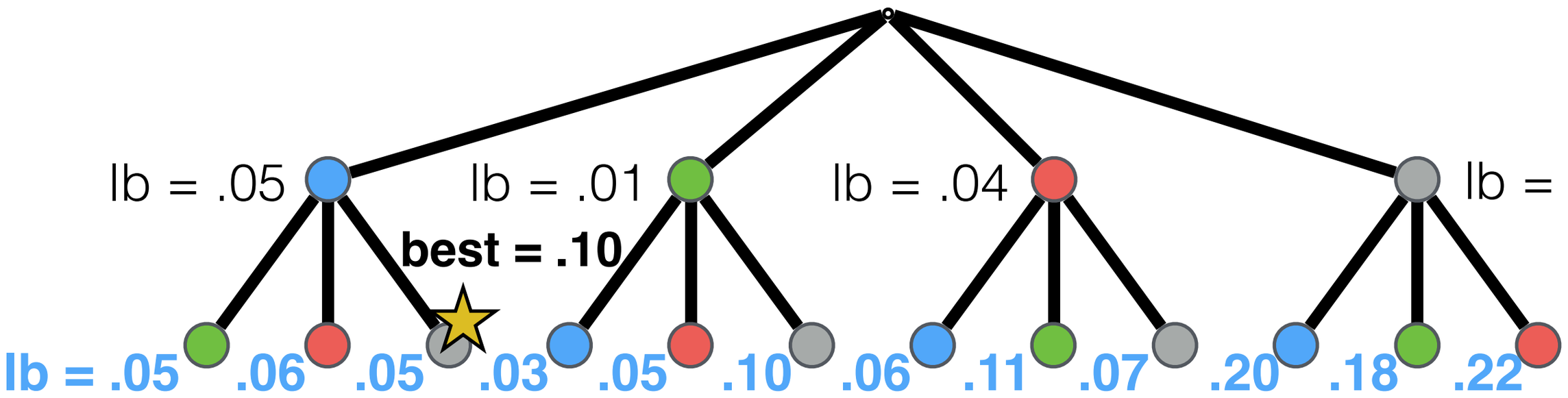
theorem: no extension of a rule list **R** can have error smaller than the lower bound of **R**

step 3A: incrementally* grow to all rule lists of length 2

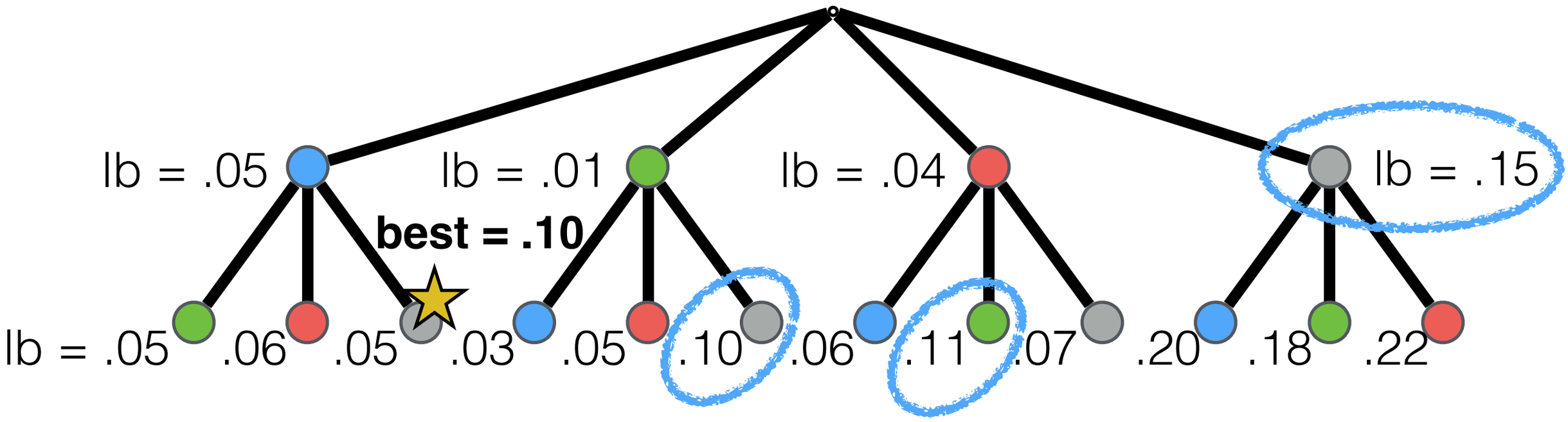


*requires efficient method to reuse old computations

step 3B: calculate lower bounds

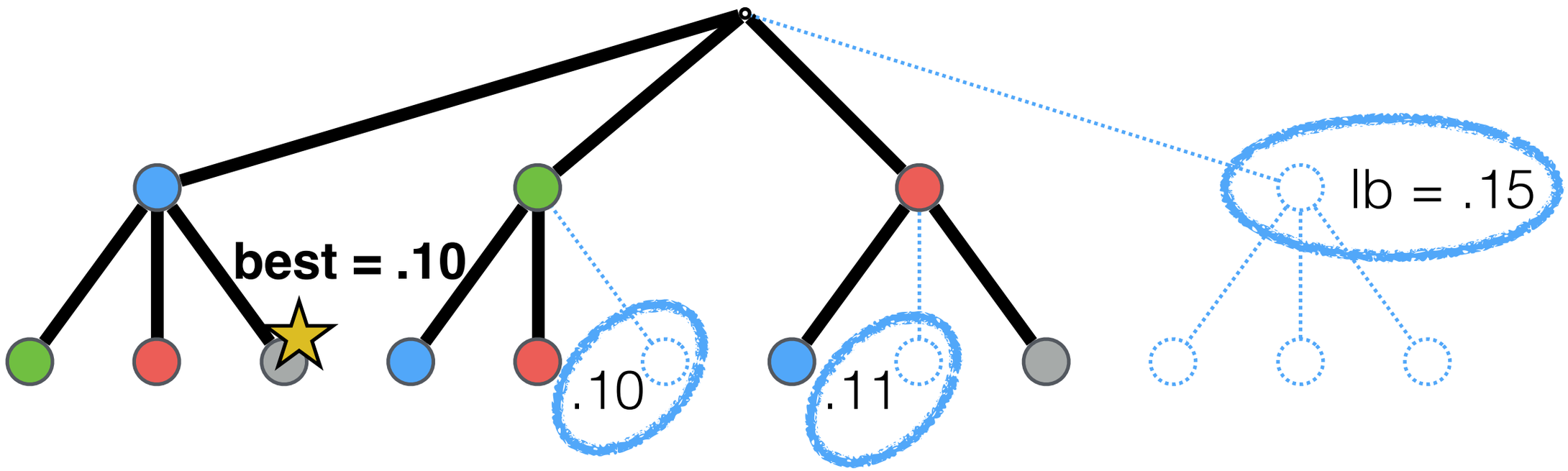


step 4: delete rule lists with lower bound >= smallest error

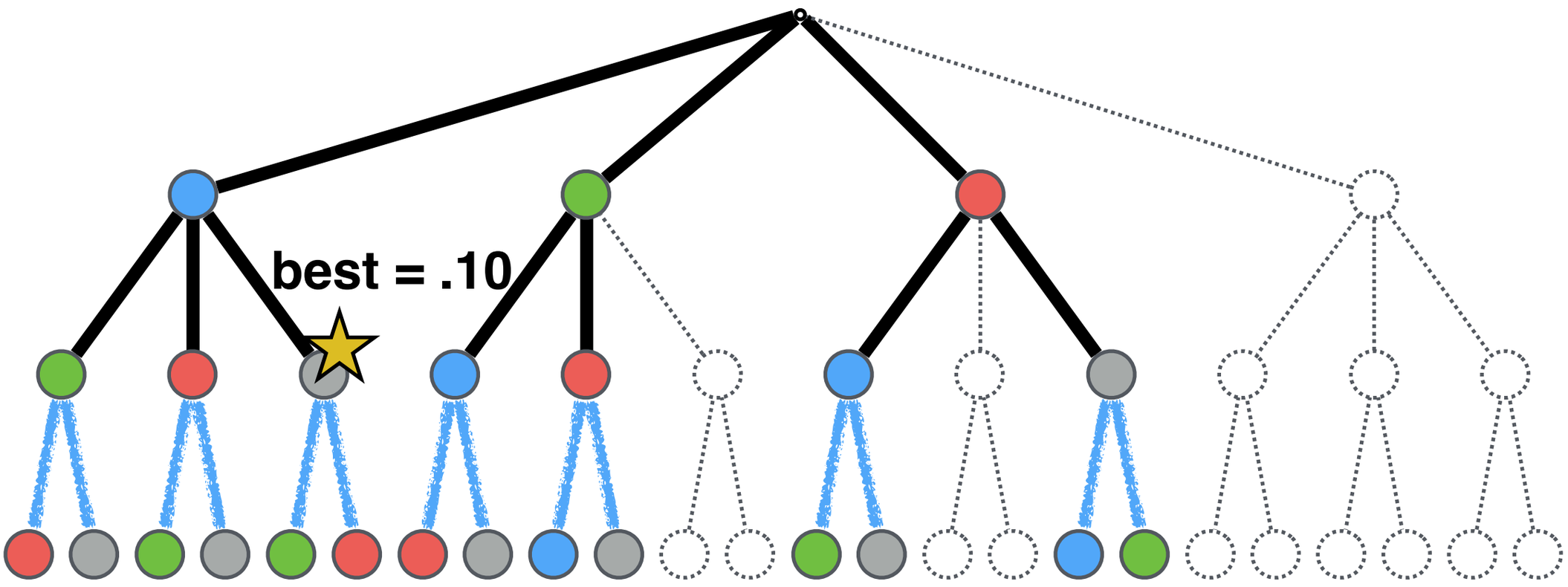


theorem: no extension of a rule list **R** can have error smaller than the lower bound of **R**

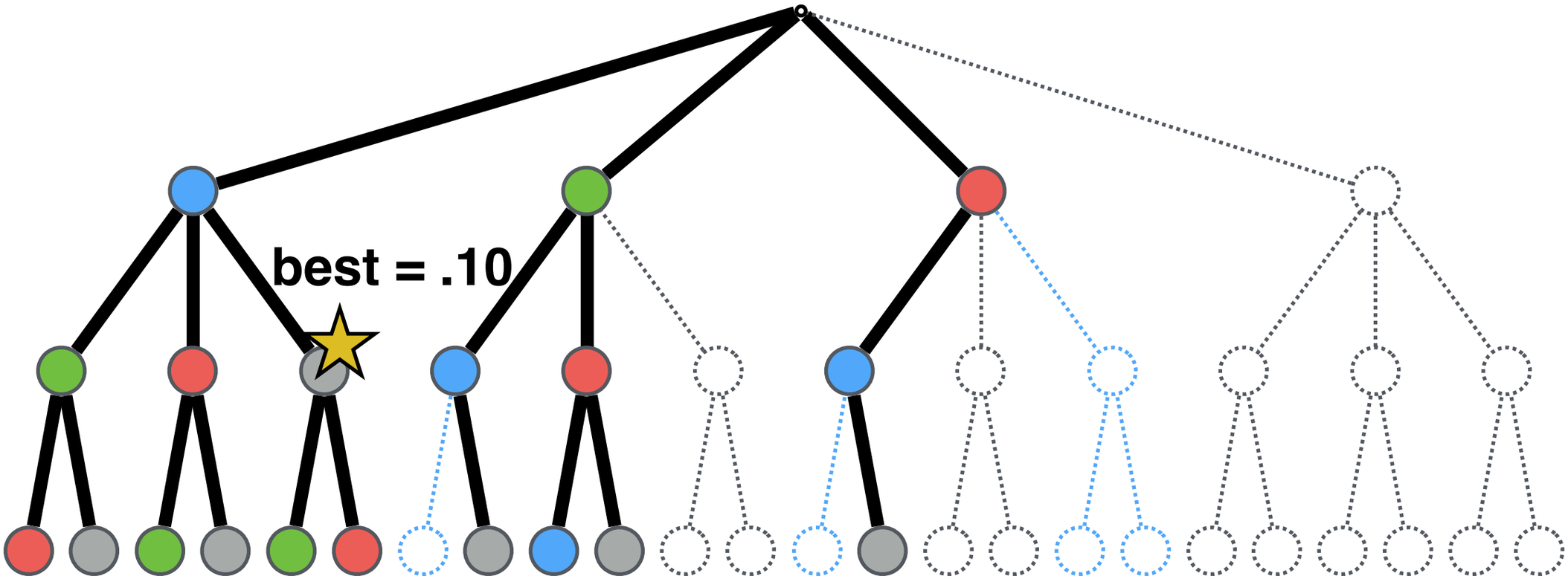
lower bounds let us prune our search space



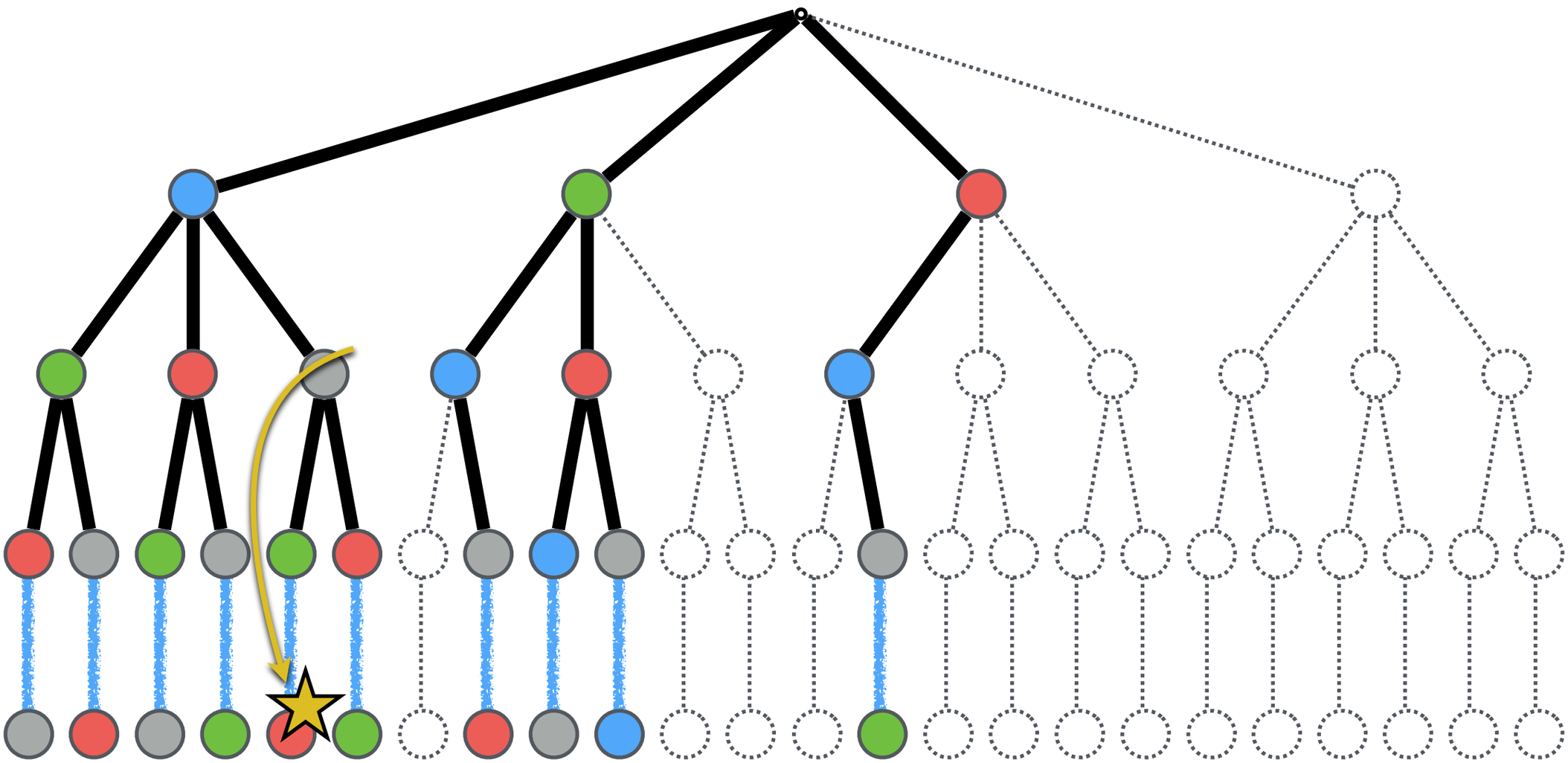
step 5: incrementally grow to all rule lists of length 3



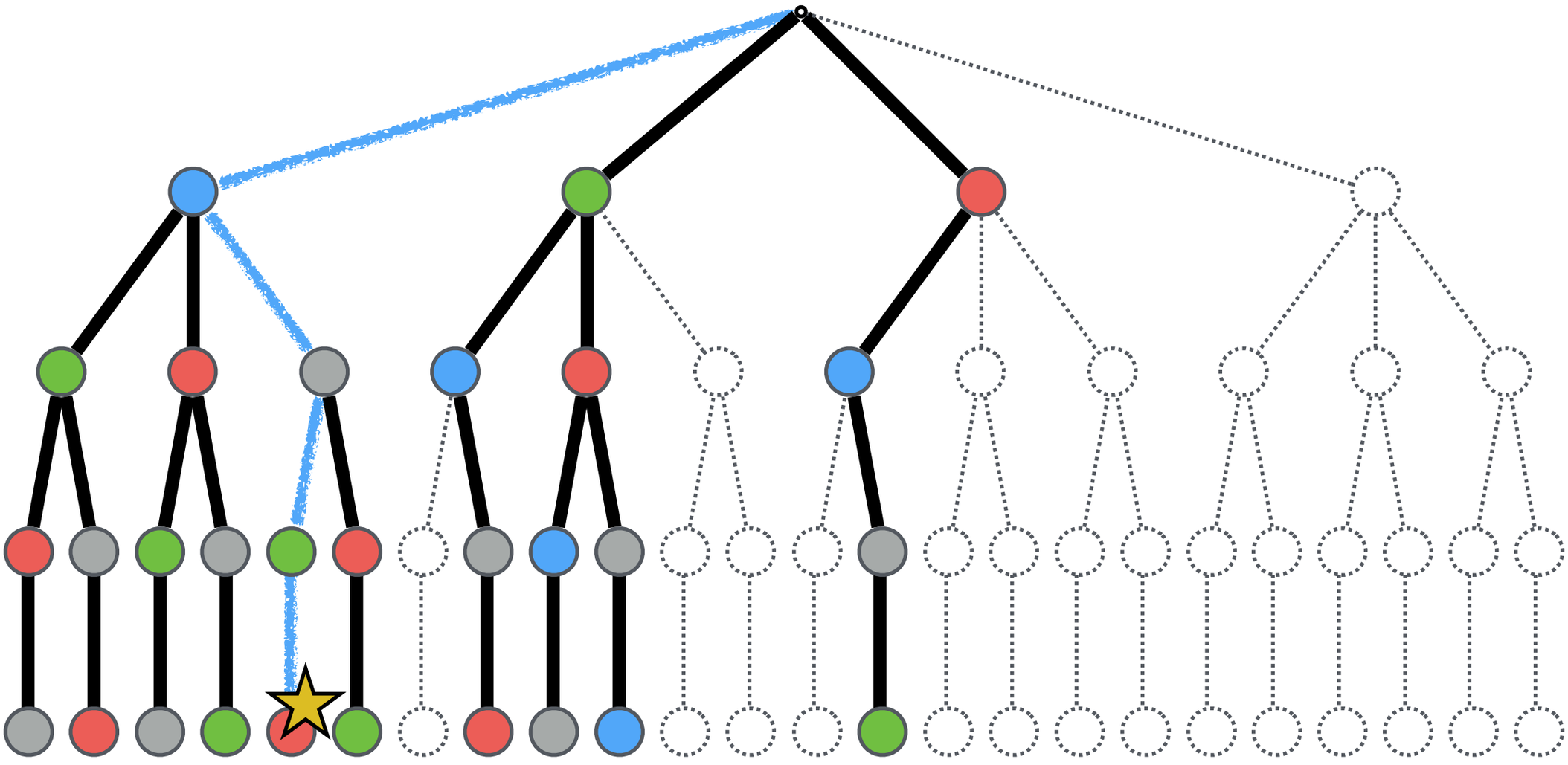
step 6: use lower bounds to prune



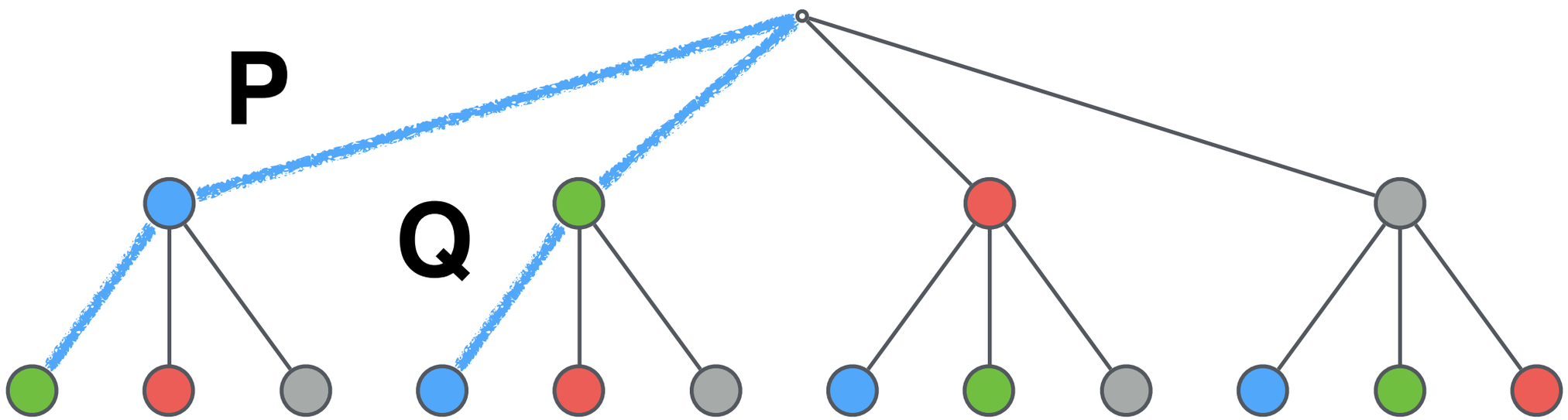
step 7: incrementally grow to prefixes of length 4



global optimization complete

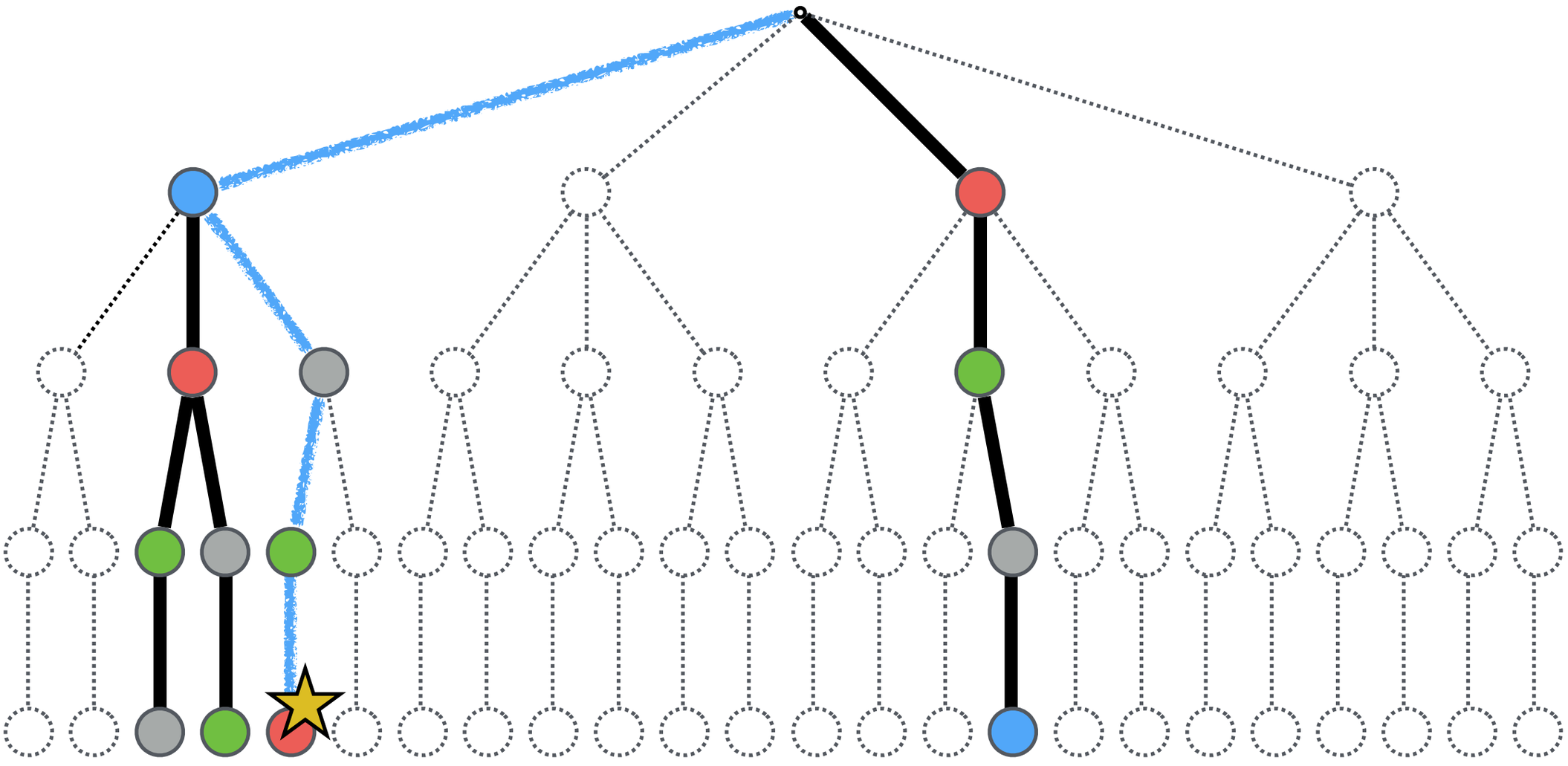


the complete algorithm combines multiple theorems.
here's one more…



theorem: if **P** and **Q** are equivalent up to a permutation, then we only extend the better one

symmetry arguments lead to more aggressive pruning



summary

- rule lists are **human-interpretable** machine learning models with many applications
- learning rule lists is computationally hard
- developed a novel algorithm for efficiently learning optimal rule lists
- **enormous speedup** via theoretical bounds that dramatically prune the search space

Thanks!

Miller Institute Staff!
Kathy, Erin, Emily, Donata

Miller Fellow Coaches!
Alejandro, Rebecca, Ryan

Miller Faculty Host & Lab!
Prof. Michael I. Jordan @ AMPLab → RISELab

Miller Institute & UC Berkeley EECS!

Bank of America

NIKE

KAISER PERMANENTE®

UNDER ARMOUR

gsk

Johnson & Johnson

NETFLIX

Walmart

The New York Times

cvs pharmacy™

"**Machine Bias:** There's software used across the country to predict future criminals. And it's biased against blacks."

www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

"**Machine Bias:** There's software used across the country to predict future criminals. And it's biased against blacks."

Sold the government an expensive, proprietary, **black box** algorithm

COMPAS = Correctional Offender Management Profiling for Alternative Sanctions

Used to predict risk of committing a future crime

www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

"**Machine Bias:** There's software used across the country to predict future criminals. And it's biased against blacks."

Sold the government an expensive, proprietary, **black box** algorithm

COMPAS = Correctional Offender Management Profiling for Alternative Sanctions

Used to predict risk of committing a future crime

- Whether to free someone
- Setting bond amounts
- Given to judges during sentencing
- Funds for rehabilitation

www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Two Petty Theft Arrests

Prior Offenses
2 armed robberies,
1 attempted armed robbery

Subsequent Offenses
1 grand theft

White Male, 41

RISK: **3**

Black Female, 18

RISK: **8**

Prior Offenses
4 juvenile misdemeanors

Subsequent Offenses
None

She *was rated high risk for future crime after she and a friend took a kid's bike and scooter that were sitting outside. She did not reoffend.*

www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Why do we care about interpretable models?

- Trust in decision-making
- Troubleshooting
- Sharing & explaining **(doctors & patients)**
- Can be used in the field

Stroke prediction model

if hemiplegia **and** age $>$ 60 **then** *stroke risk* 58.9% (53.8%–63.8%)
else if cerebrovascular disorder **then** *stroke risk* 47.8% (44.8%–50.7%)
else if transient ischaemic attack **then** *stroke risk* 23.8% (19.5%–28.4%)
else if occlusion and stenosis of carotid artery without infarction **then** *stroke risk* 15.8% (12.2%–19.6%)
else if altered state of consciousness **and** age $>$ 60 **then** *stroke risk* 16.0% (12.2%–20.2%)
else if age $\leq$ 70 **then** *stroke risk* 4.6% (3.9%–5.4%)
else *stroke risk* 8.7% (7.9%–9.6%)

Letham et al. Ann. Appl. Stat. 2015.

the number of possible rule lists is **yuuuge**

suppose there are **1000** possible rules and an oracle tells you the best rule list has at most length **10** …

… then there are ~ $\mathbf{10^{29}}$ possible rule lists

… if we take ~ **4 microseconds** to evaluate each

… that's ~ $\mathbf{10^{17}}$ **years**

Recidivism problem: start with **155** possible rules, end up looking at rule lists up to length **5** …

… naïve search space has ~ **80 billion** rule lists

… we take ~ **4 microseconds** to evaluate each

… that would require ~ **4 days**

Spoiler alert: Our algorithm is 1000x faster

… we only have to evaluate ~ **80 million** rule lists

… we find an optimal solution **< 10 seconds**

… and certify optimality in ~ **5.5 minutes**