Unsupervised Learning under Latent Label Shift

NeurIPS 2022


Manley Roberts*[(1)]


Pranav Mani*[(1)]


Saurabh Garg[(1)]


Zachary C. Lipton[(1)]

*equal contribution
(1) Machine Learning Department

Carnegie Mellon University

Can we find correct categories w/o labels?

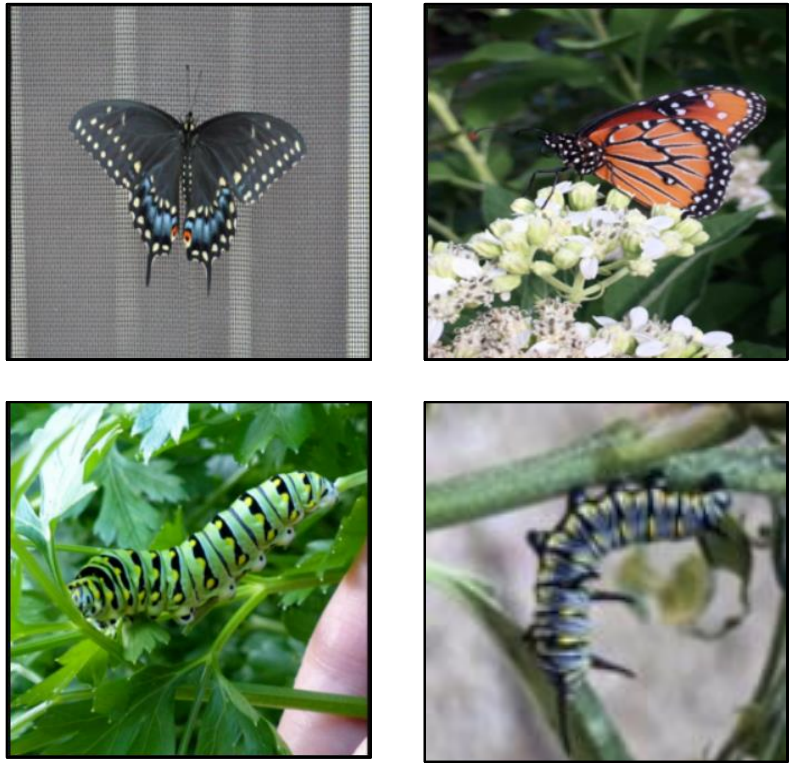
- Finding categories in unlabeled data is ill-posed.
  - Multiple ways to group the same data
- What principles can we use to determine the correct groupings?

Unlabeled examples

Can we find correct categories w/o labels?

- Finding categories in unlabeled data is ill-posed.
    - Multiple ways to group the same data
- What principles can we use to determine the correct groupings?

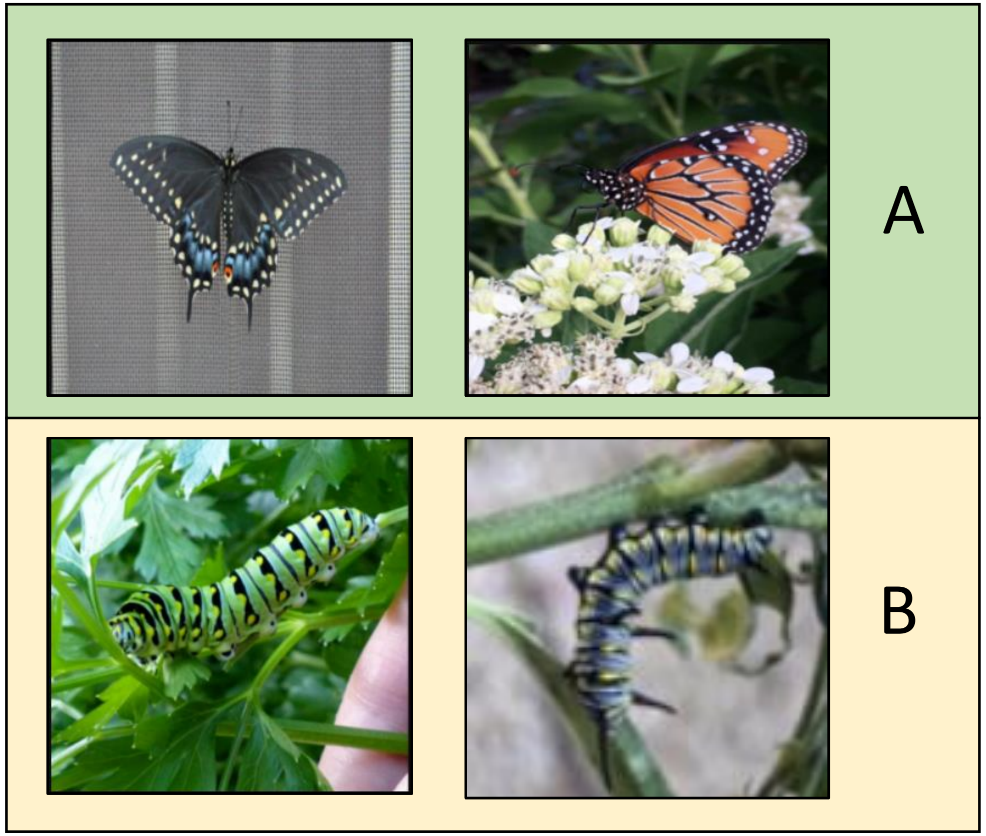
Unlabeled examples



Can we find correct categories w/o labels?

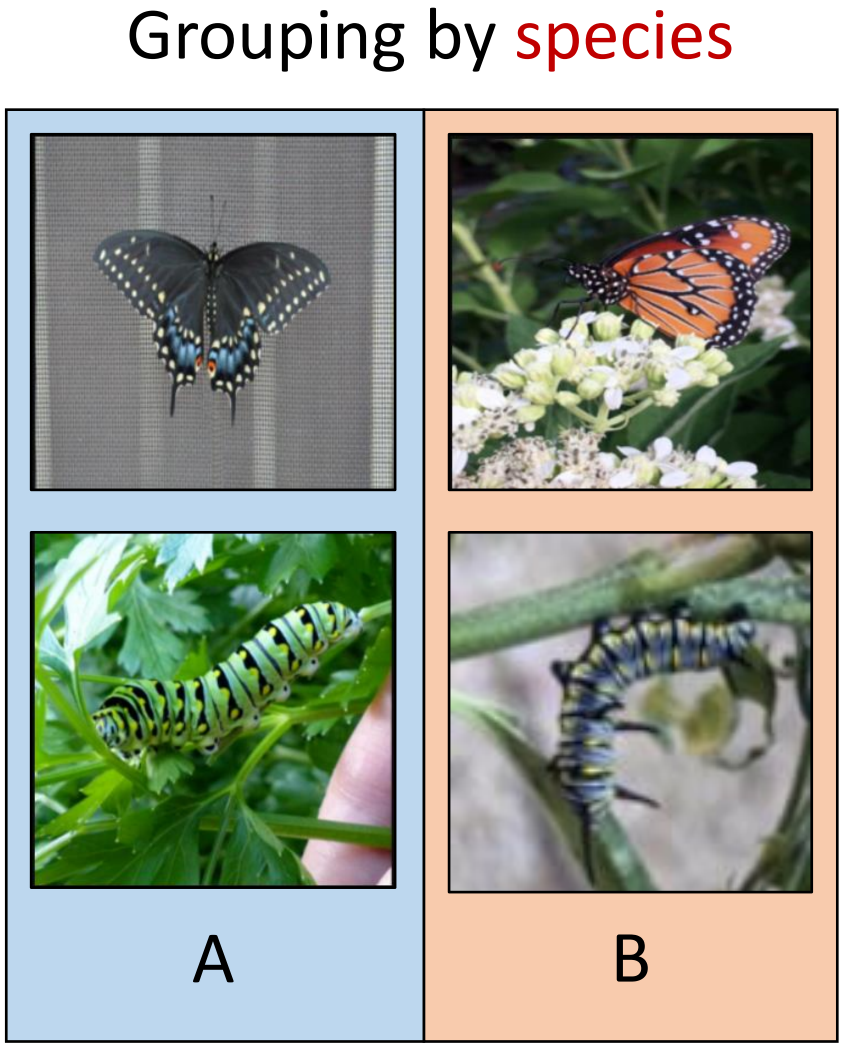
- Finding categories in unlabeled data is ill-posed.
  - Multiple ways to group the same data
- What principles can we use to determine the correct groupings?

Grouping by life stage



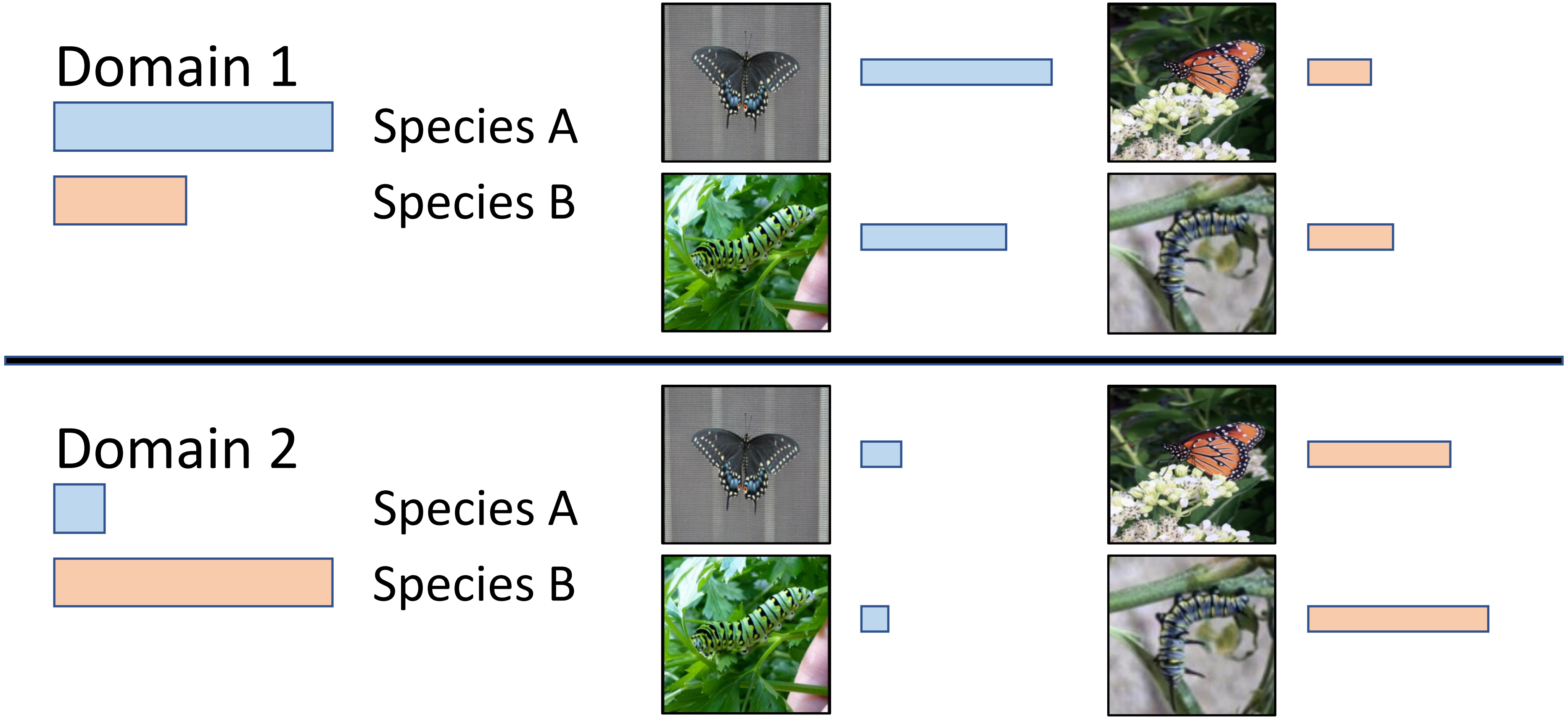
Can we find correct categories w/o labels?

- Finding categories in unlabeled data is ill-posed.
  - Multiple ways to group the same data
- What principles can we use to determine the correct groupings?



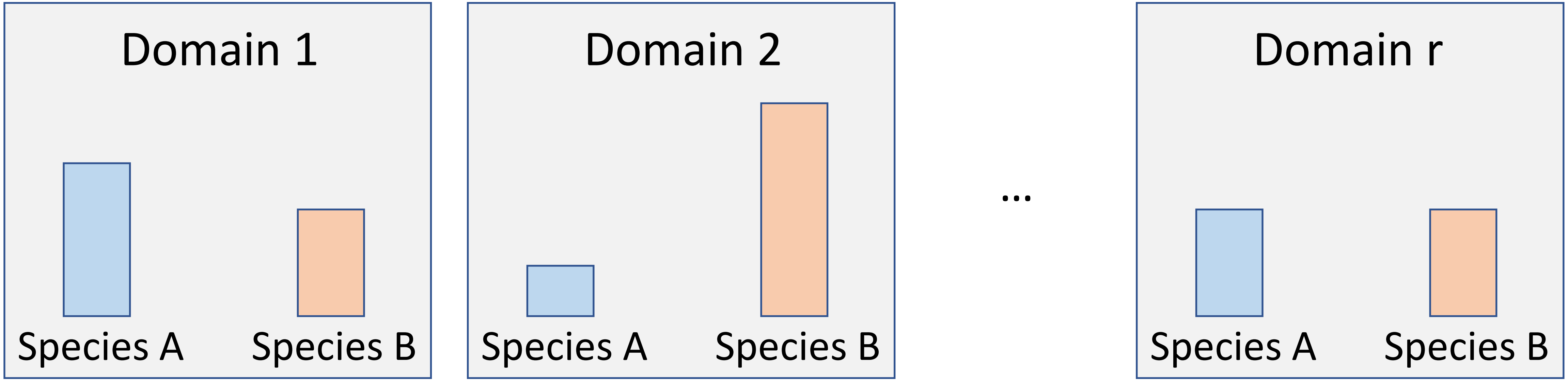
Instances that shift together group together

- Group together elements that shift together in prevalence across domains



Latent Label Shift (LLS)

- **Label Shift Assumption:** Class conditional distributions over samples remain domain invariant, while class prevalences may shift.
  - For all $d, d' \in [r], p_d(x|y) = p_{d'}(x|y)$
- **Goal:** Estimate $p_d(y)$ and $p_d(y|x)$.



Finite Inputs: an NMF model

- Consider mixing distribution Q in which domain is a random variable D.
  - Then $q(x, y|D = d) = p_d(x, y)$.
- If X takes on finite set of values $[m]$, we model the mixture as the matrix product $\boldsymbol{Q}_{X|D} = \boldsymbol{Q}_{X|Y}\boldsymbol{Q}_{Y|D}$, where
  - $\boldsymbol{Q}_{X|D}$ holds the known marginals over X in each domain
  - $\boldsymbol{Q}_{X|Y}$ holds the unknown class-conditional distributions
  - $\boldsymbol{Q}_{Y|D}$ holds the unknown marginals over Y in each domain.
- Solving for unknown matrices via Non-negative Matrix Factorization (NMF) is not identified in general.

Finite Inputs: an NMF model

- Consider mixing distribution Q in which domain is a random variable D.
  - Then $q(x, y|D = d) = p_d(x, y)$.
- If X takes on finite set of values $[m]$, we model the mixture as the matrix product $\boldsymbol{Q}_{X|D} = \boldsymbol{Q}_{X|Y}\boldsymbol{Q}_{Y|D}$, where
  - $\boldsymbol{Q}_{X|D}$ holds the known marginals over X in each domain
  - $\boldsymbol{Q}_{X|Y}$ holds the unknown class-conditional distributions
  - $\boldsymbol{Q}_{Y|D}$ holds the unknown marginals over Y in each domain.
- Solving for unknown matrices via Non-negative Matrix Factorization (NMF) is not identified in general.

Finite Inputs: an NMF model

- Consider mixing distribution Q in which domain is a random variable D.
  - Then $q(x, y|D = d) = p_d(x, y)$.
- If X takes on finite set of values $[m]$, we model the mixture as the matrix product $\boldsymbol{Q}_{X|D} = \boldsymbol{Q}_{X|Y}\boldsymbol{Q}_{Y|D}$, where
  - $\boldsymbol{Q}_{X|D}$ holds the known marginals over X in each domain
  - $\boldsymbol{Q}_{X|Y}$ holds the unknown class-conditional distributions
  - $\boldsymbol{Q}_{Y|D}$ holds the unknown marginals over Y in each domain.
- Solving for unknown matrices via Non-negative Matrix Factorization (NMF) is not identified in general.

Finite Inputs: an NMF model

- Consider mixing distribution Q in which domain is a random variable D.
  - Then $q(x, y|D = d) = p_d(x, y)$.
- If X takes on finite set of values $[m]$, we model the mixture as the matrix product $\boldsymbol{Q}_{X|D} = \boldsymbol{Q}_{X|Y}\boldsymbol{Q}_{Y|D}$, where
  - $\boldsymbol{Q}_{X|D}$ holds the known marginals over X in each domain
  - $\boldsymbol{Q}_{X|Y}$ holds the unknown class-conditional distributions
  - $\boldsymbol{Q}_{Y|D}$ holds the unknown marginals over Y in each domain.
- Solving for unknown matrices via Non-negative Matrix Factorization (NMF) is not identified in general.

Finite Inputs: an NMF model

- Consider mixing distribution Q in which domain is a random variable D.
  - Then $q(x, y|D = d) = p_d(x, y)$.
- If X takes on finite set of values $[m]$, we model the mixture as the matrix product $\boldsymbol{Q}_{X|D} = \boldsymbol{Q}_{X|Y}\boldsymbol{Q}_{Y|D}$, where
  - $\boldsymbol{Q}_{X|D}$ holds the known marginals over X in each domain
  - $\boldsymbol{Q}_{X|Y}$ holds the unknown class-conditional distributions
  - $\boldsymbol{Q}_{Y|D}$ holds the unknown marginals over Y in each domain.
- Solving for unknown matrices via Non-negative Matrix Factorization (NMF) is not identified in general.

Finite Inputs: an NMF model

- Consider mixing distribution Q in which domain is a random variable D.
  - Then $q(x, y|D = d) = p_d(x, y)$.
- If X takes on finite set of values $[m]$, we model the mixture as the matrix product $\boldsymbol{Q}_{X|D} = \boldsymbol{Q}_{X|Y}\boldsymbol{Q}_{Y|D}$, where
  - $\boldsymbol{Q}_{X|D}$ holds the known marginals over X in each domain
  - $\boldsymbol{Q}_{X|Y}$ holds the unknown class-conditional distributions
  - $\boldsymbol{Q}_{Y|D}$ holds the unknown marginals over Y in each domain.
- Solving for unknown matrices via Non-negative Matrix Factorization (NMF) is not identified in general.

Isomorphism to Topic Modeling

- Topic modeling considers documents as mixtures of topics.
  - Each topic has a word distribution (invariant over documents).
- LLS with finite set of values for X is isomorphic to topic modeling:
  - A domain is a document.
  - A label is a topic.
  - An example is a word.
- Topic modeling gives us the anchor word condition for identifiability:
  - If each label Y has some input X which occurs with nonzero probability only under that label, the solution is identifiable. [Donoho & Stodden, 2003]

Isomorphism to Topic Modeling

- Topic modeling considers documents as mixtures of topics.
  - Each topic has a word distribution (invariant over documents).
- LLS with finite set of values for X is isomorphic to topic modeling:
  - A domain is a document.
  - A label is a topic.
  - An example is a word.
- Topic modeling gives us the anchor word condition for identifiability:
  - If each label Y has some input X which occurs with nonzero probability only under that label, the solution is identifiable. [Donoho & Stodden, 2003]

Isomorphism to Topic Modeling

- Topic modeling considers documents as mixtures of topics.
  - Each topic has a word distribution (invariant over documents).
- LLS with finite set of values for X is isomorphic to topic modeling:
  - A domain is a document.
  - A label is a topic.
  - An example is a word.
- Topic modeling gives us the anchor word condition for identifiability:
  - If each label Y has some input X which occurs with nonzero probability only under that label, the solution is identifiable. [Donoho & Stodden, 2003]

Extension to Continuous Inputs

- No prior identifiability results for continuous X.
- **Our goal**: find a suitable discretization of the continuous space.
  - Resulting discrete problem will always satisfy label shift assumption.
  - If the discretized problem satisfies the anchor word assumption, we can apply discrete identifiability conditions to identify the solution.

Identifiability Result for Continuous Inputs

- In Theorem 2, we give a set of sufficient conditions to identify $p_d(y)$ and $p_d(y|x)$:
  - **Anchor subdomain condition:** for each label, there is a region of X space with nonzero support in only this label.
  - **Access to a domain discriminator:** we assume we may query a function which predicts the distribution $q(d|x)$ over domains for any value X.
  - Some other assumptions including rank assumptions on $Q_{Y|D}$.
- **Discretization strategy:**
  - Push density over X through the domain discriminator.
  - Match point masses in $q(d|x)$ space to distinct discrete values.

Identifiability Result for Continuous Inputs

- In Theorem 2, we give a set of sufficient conditions to identify $p_d(y)$ and $p_d(y|x)$:
  - **Anchor subdomain condition:** for each label, there is a region of X space with nonzero support in only this label.
  - **Access to a domain discriminator:** we assume we may query a function which predicts the distribution $q(d|x)$ over domains for any value X.
  - Some other assumptions including rank assumptions on $Q_{Y|D}$.
- **Discretization strategy:**
  - Push density over X through the domain discriminator.
  - Match point masses in $q(d|x)$ space to distinct discrete values.

Identifiability Result for Continuous Inputs

- In Theorem 2, we give a set of sufficient conditions to identify $p_d(y)$ and $p_d(y|x)$:
  - **Anchor subdomain condition:** for each label, there is a region of X space with nonzero support in only this label.
  - **Access to a domain discriminator:** we assume we may query a function which predicts the distribution $q(d|x)$ over domains for any value X.
  - Some other assumptions including rank assumptions on $Q_{Y|D}$.
- **Discretization strategy:**
  - Push density over X through the domain discriminator.
  - Match point masses in $q(d|x)$ space to distinct discrete values.

Identifiability Result for Continuous Inputs

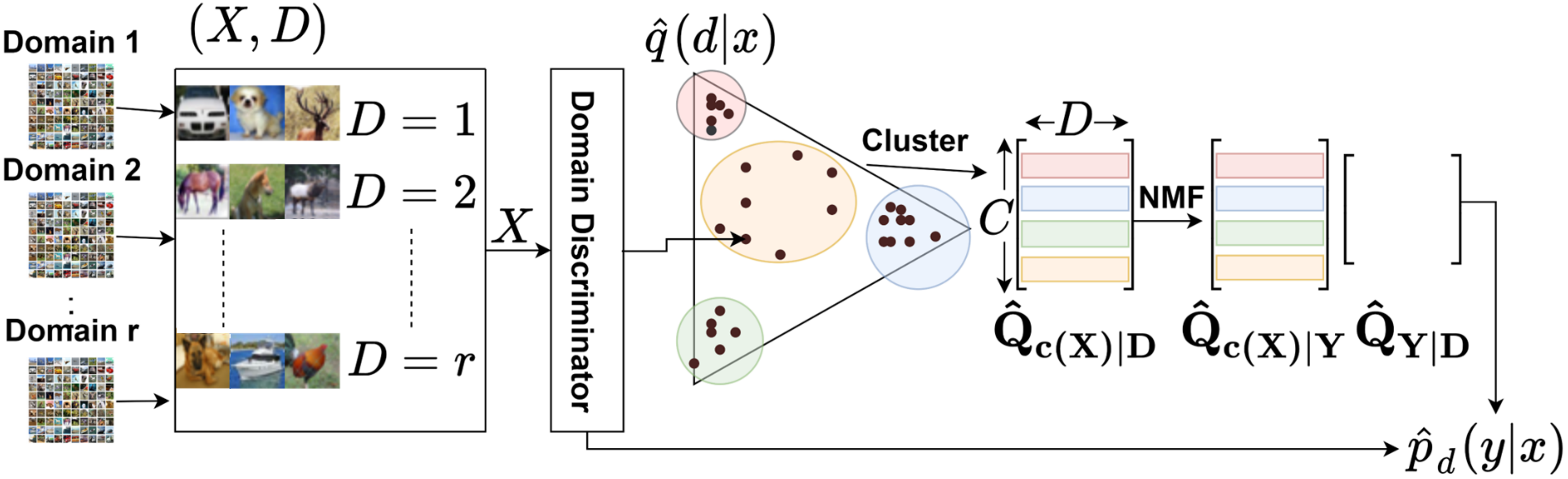
- In Theorem 2, we give a set of sufficient conditions to identify $p_d(y)$ and $p_d(y|x)$:
  - **Anchor subdomain condition:** for each label, there is a region of X space with nonzero support in only this label.
  - **Access to a domain discriminator:** we assume we may query a function which predicts the distribution $q(d|x)$ over domains for any value X.
  - Some other assumptions including rank assumptions on $Q_{Y|D}$.
- **Discretization strategy:**
  - Push density over X through the domain discriminator.
  - Match point masses in $q(d|x)$ space to distinct discrete values.

Identifiability Result for Continuous Inputs

- In Theorem 2, we give a set of sufficient conditions to identify $p_d(y)$ and $p_d(y|x)$:
  - **Anchor subdomain condition:** for each label, there is a region of X space with nonzero support in only this label.
  - **Access to a domain discriminator:** we assume we may query a function which predicts the distribution $q(d|x)$ over domains for any value X.
  - Some other assumptions including rank assumptions on $Q_{Y|D}$.
- **Discretization strategy:**
  - Push density over X through the domain discriminator.
  - Match point masses in $q(d|x)$ space to distinct discrete values.

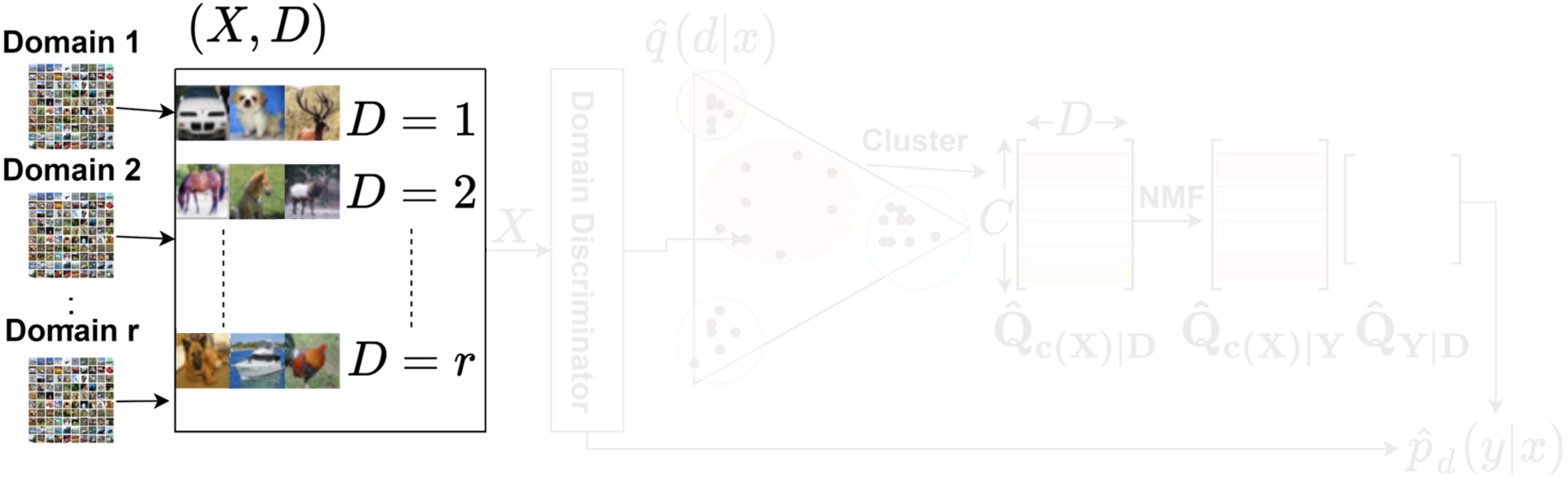
Discriminate Discretize Factorize Adjust (DDFA)

- We outline a practical algorithm to find $p_d(y)$ and $p_d(y|x)$.



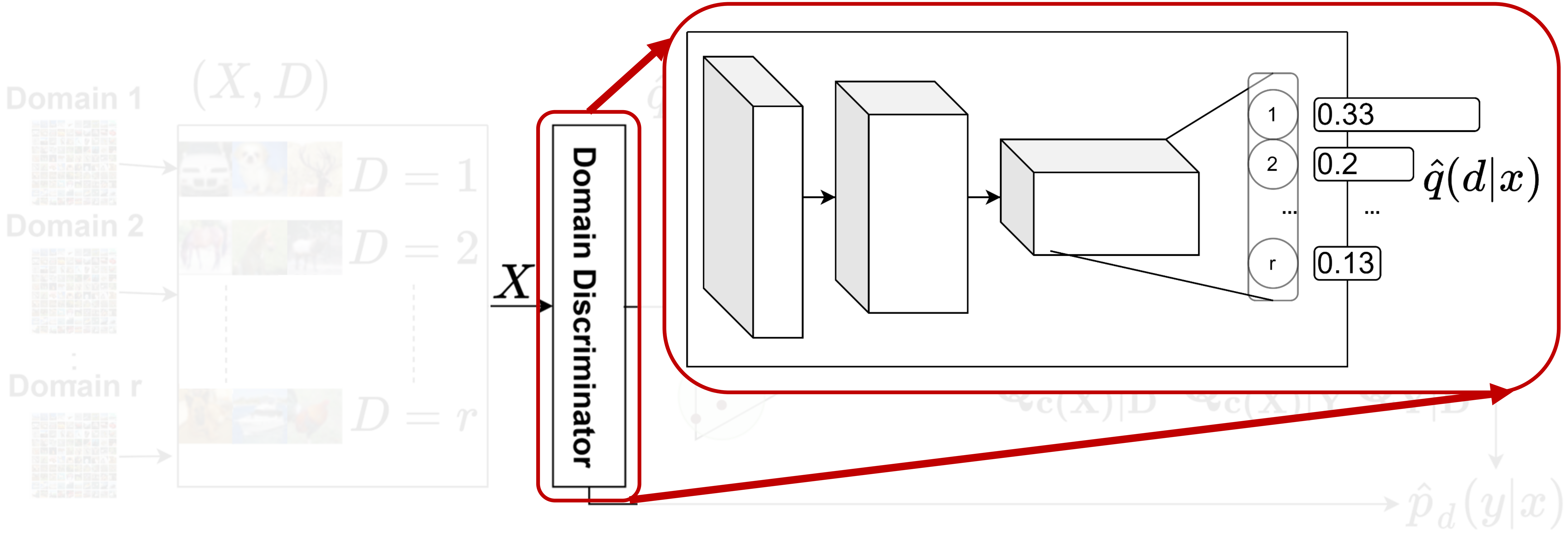
Discriminate Discretize Factorize Adjust (DDFA)

1. Sample (input x, source domain d) data pairs



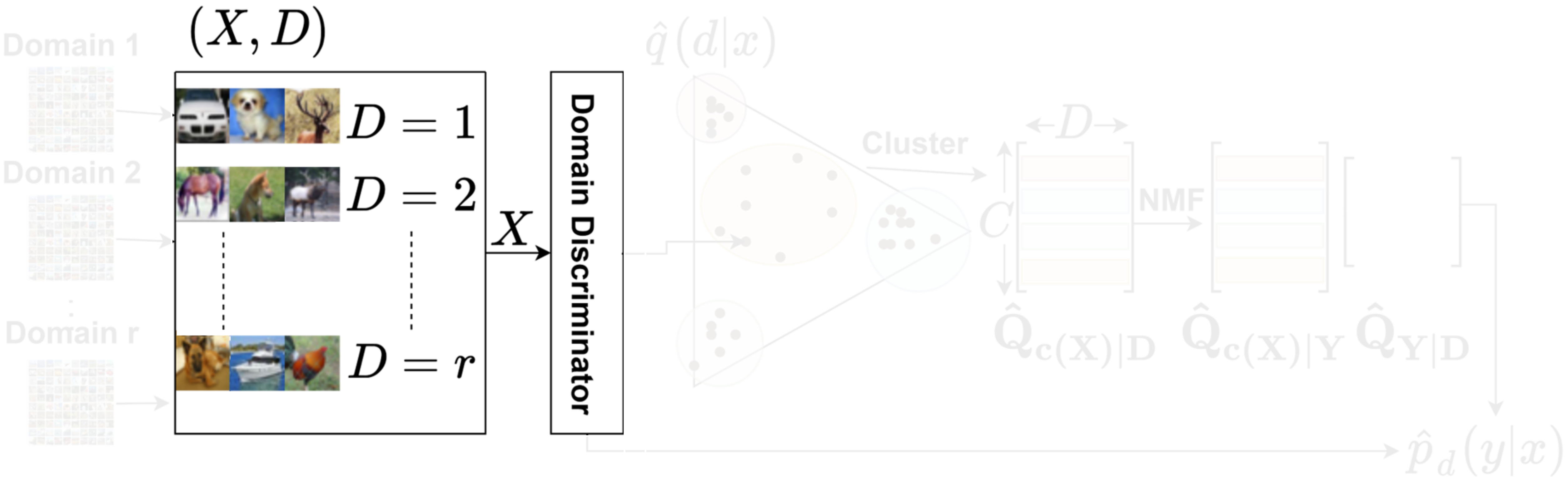
Discriminate Discretize Factorize Adjust (DDFA)

2. Train an estimate of a domain discriminator



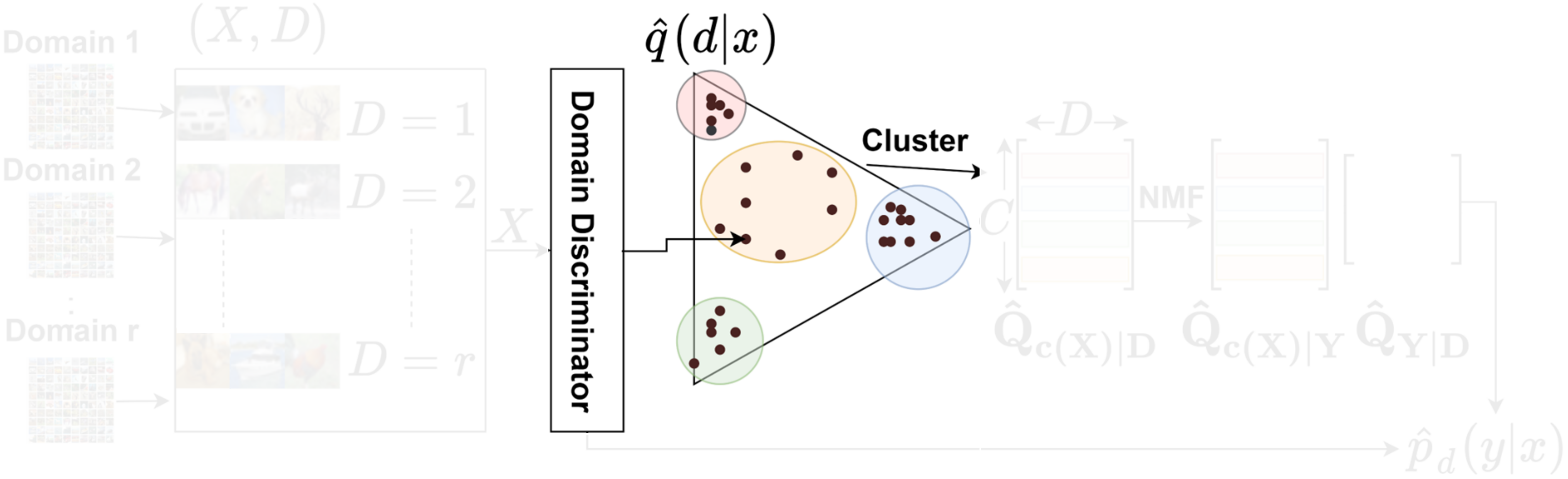
Discriminate Discretize Factorize Adjust (DDFA)

3. Push samples through learned estimate



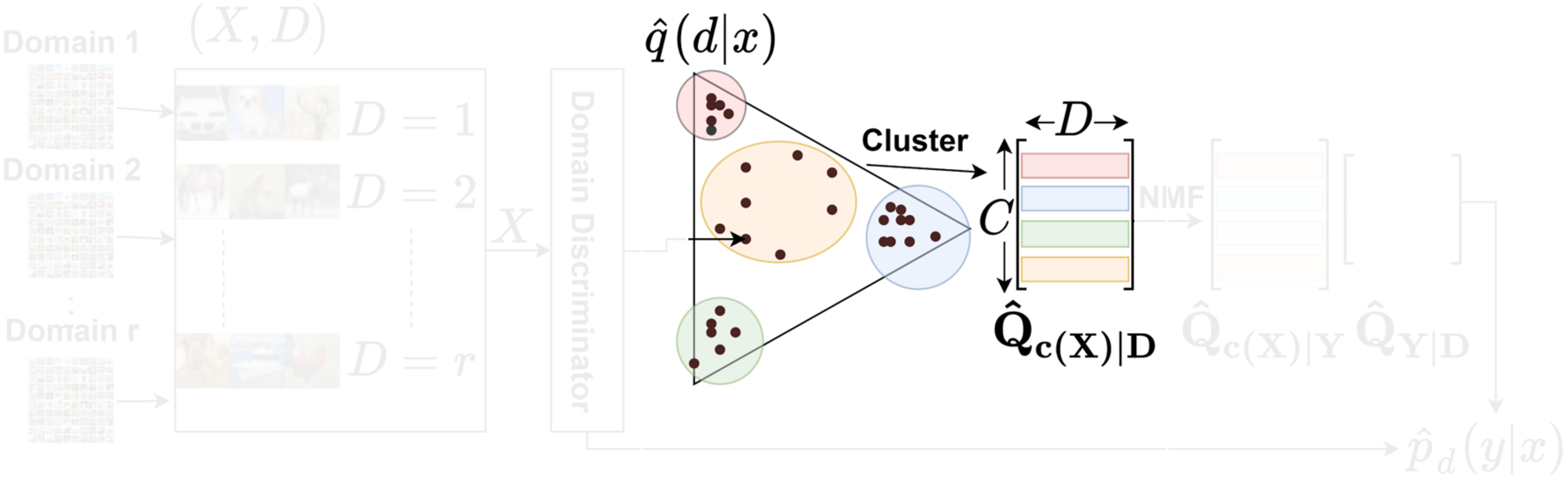
Discriminate Discretize Factorize Adjust (DDFA)

4. Cluster $\hat{q}(d|x)$ vectors into a finite number of clusters



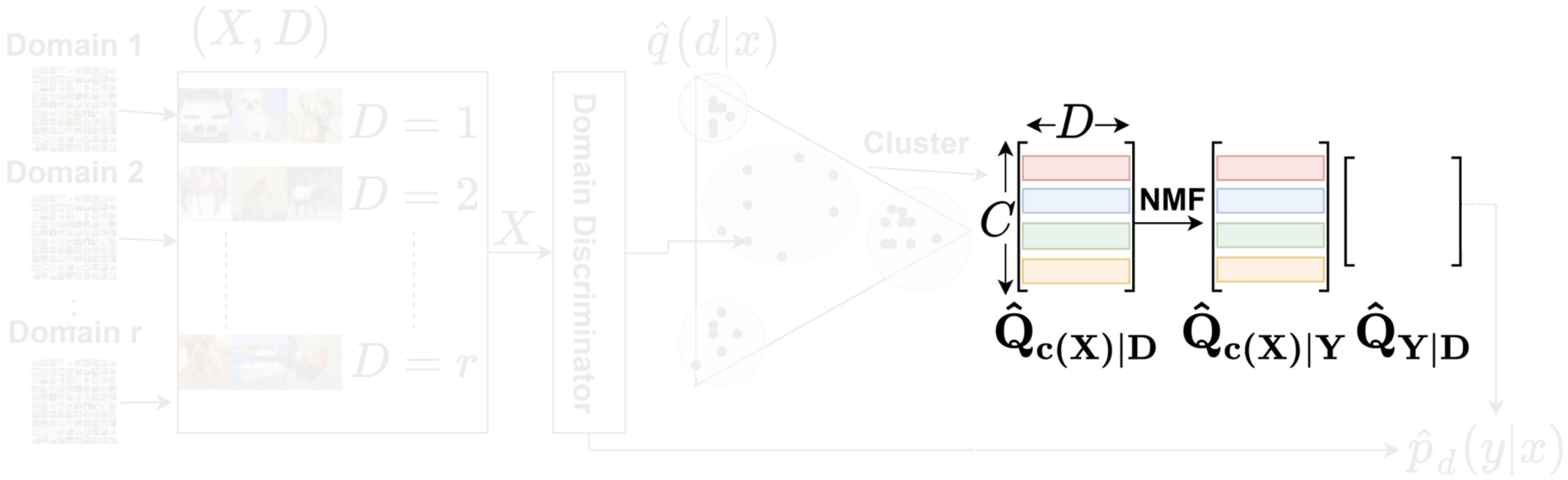
Discriminate Discretize Factorize Adjust (DDFA)

5. Discretize using clusters, build $\widehat{\boldsymbol{Q}}_{c(X)|D}$ matrix



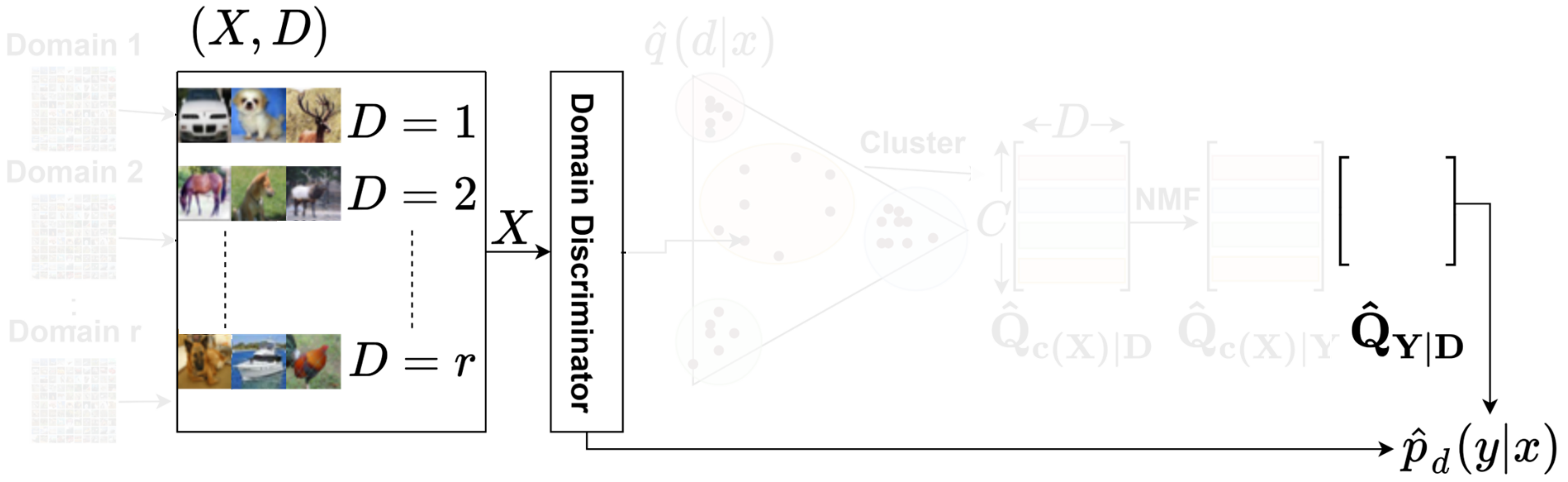
Discriminate Discretize Factorize Adjust (DDFA)

6. Using NMF algorithm, decompose matrix



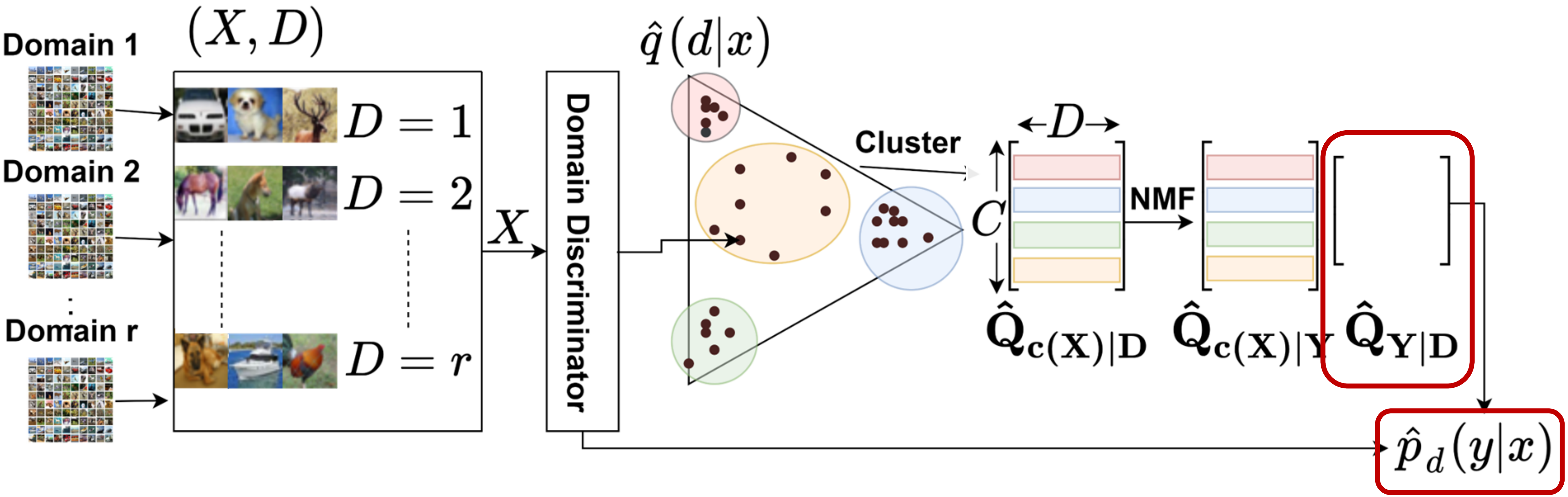
Discriminate Discretize Factorize Adjust (DDFA)

7. Estimate domain-specific classifier



Discriminate Discretize Factorize Adjust (DDFA)

Output: estimate of label-proportion matrix and domain-specific classifier.



Experiments

- Semi-synthetic experiments on CIFAR-10, CIFAR-20, ImageNet-50, FieldGuide-2, FieldGuide-28
  - Sample $\boldsymbol{Q}_{Y|D}$, assign examples to different domains according to label prevalence, train a domain discriminator and evaluate recovery of labels.
  - Can achieve higher classification accuracy and lower error in recovering $\boldsymbol{Q}_{Y|D}$ than baseline unsupervised approach SCAN, when $\boldsymbol{Q}_{Y|D}$ sufficiently sparse and in datasets with few classes.

Takeaways

- Use domain structure to uncover categories in unlabeled data
- Leverage a strong connection to topic modeling to establish sufficient set of conditions for identifiability.
- Establish experimentally that domain structure aids class discovery.