A Framework for Learnig Predictive Structures from Multiple Tasks and Unlabeled Data

Rie Kubota Ando and Tong Zhang

IBM Watson Research Center
Yahoo Research

Nov. 20th, 2006

Semi-supervised Learning

- Large amount of unlabeled data, while labeled data are very costly
- Various methods: transductive inference, co-training (basically label propagation), fails when noise is introduced into classification through non-perfect classification.
- Another direction: define a good functional structures using unlabeled data. (what is a structure? distance, kernel, manifold) But a graph structure might not be predictive.
- Can we learn a predictive structure?
- Yes, if we have multiple related tasks.

Semi-supervised Learning

- Large amount of unlabeled data, while labeled data are very costly
- Various methods: transductive inference, co-training (basically label propagation), fails when noise is introduced into classification through non-perfect classification.
- Another direction: define a good functional structures using unlabeled data. (what is a structure? distance, kernel, manifold) But a graph structure might not be predictive.
- Can we learn a predictive structure?
- Yes, if we have multiple related tasks.

Semi-supervised Learning

- Large amount of unlabeled data, while labeled data are very costly
- Various methods: transductive inference, co-training (basically label propagation), fails when noise is introduced into classification through non-perfect classification.
- Another direction: define a good functional structures using unlabeled data. (what is a structure? distance, kernel, manifold) But a graph structure might not be predictive.
- Can we learn a predictive structure?
- Yes, if we have multiple related tasks.

Semi-supervised Learning

- Large amount of unlabeled data, while labeled data are very costly
- Various methods: transductive inference, co-training (basically label propagation), fails when noise is introduced into classification through non-perfect classification.
- Another direction: define a good functional structures using unlabeled data. (what is a structure? distance, kernel, manifold) But a graph structure might not be predictive.
- Can we learn a predictive structure?
- Yes, if we have multiple related tasks.

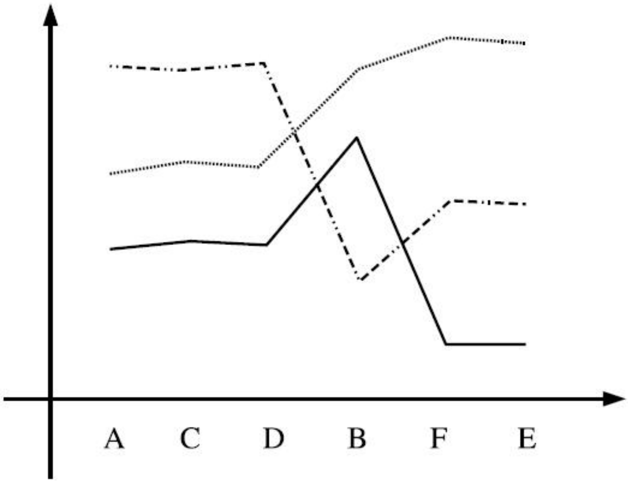
Semi-supervised Learning

- Large amount of unlabeled data, while labeled data are very costly
- Various methods: transductive inference, co-training (basically label propagation), fails when noise is introduced into classification through non-perfect classification.
- Another direction: define a good functional structures using unlabeled data. (what is a structure? distance, kernel, manifold) But a graph structure might not be predictive.
- Can we learn a predictive structure?
- Yes, if we have multiple related tasks.

Learning Predictive Structures

1. Structural learning from multiple tasks
2. Use unlabeled data to generate auxiliary(related) tasks.

A toy example



The intrinsic distance metric should force A, C, D “close” to each other, and F and E to each other.

Connection to Hypothesis Space

Supervised Learning

Find a predictor in the hypothesis space.

- Estimation error: The smaller the space is, the easier to learn a best predictor given limited samples.
- Approximation error: caused by a restricted size of hypothesis
- Need a trade-off of these two types of errors (model selection)

Model Selection

- Cross validation
- Can achieve better result if we have multiple problems on the same underlying domain.

Connection to Hypothesis Space

Supervised Learning

Find a predictor in the hypothesis space.

- Estimation error: The smaller the space is, the easier to learn a best predictor given limited samples.
- Approximation error: caused by a restricted size of hypothesis
- Need a trade-off of these two types of errors (model selection)

Model Selection

- Cross validation
- Can achieve better result if we have multiple problems on the same underlying domain.

Empirical Risk Minimization(ERM)

Supervised Learning

Find a predictor f such that

$$R(f) = E_{\mathbf{X},Y} L(f(\mathbf{X}), Y))$$

Empirically, we use the loss on training data as an indicator.

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \sum_{i=1}^{n} L(f(X_i), Y_i)$$

To avoid over-fitting, usually some regularization term is added

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \sum_{i=1}^{n} L(f(X_i), Y_i) + \underbrace{g(f)}_{\text{Regularization term}}$$

Joint Empirical Risk Minimization

In STL, the hypothesis space (bias) is fixed.

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \sum_{i=1}^{n} L(f(X_i), Y_i) + g(f)$$

Use parameter θ to represent the hypothesis space, then

$$\hat{f}_\theta = \arg\min_{f \in \mathcal{H}(\theta)} \sum_{i=1}^{n} L(f(X_i), Y_i) + g(f)$$

For multiple related tasks, we want to find the hypothesis shared by all these tasks. (To determine a proper θ)

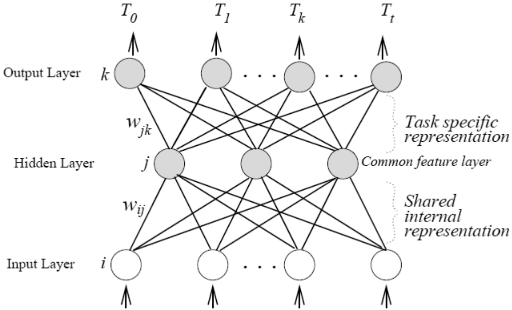
$$[\hat{f}_l, \hat{\theta}] = \arg\min_{f_l, \theta} \left[\underbrace{r(\theta)}_{\text{regularization}} + \sum_{l=1}^{m} \left(g(f_l(\theta)) + \frac{1}{n_l} \sum_{l=1}^{n_l} L(f_l(\theta), X_i^l, Y_i^l) \right) \right]$$

Structural Learning with Linear Predictors

$$f(x) = w^T \cdot \underbrace{\phi(x)}_{\text{task specific features}} + v^T \cdot \underbrace{\psi_\theta(x)}_{\text{internal dimensions}}$$

How to represent θ? A matrix(can be considered as a transformation matrix to find new dimensions)

$$f_\theta(w, v; x) = w^T\phi(x) + v^T\theta\psi(x)$$

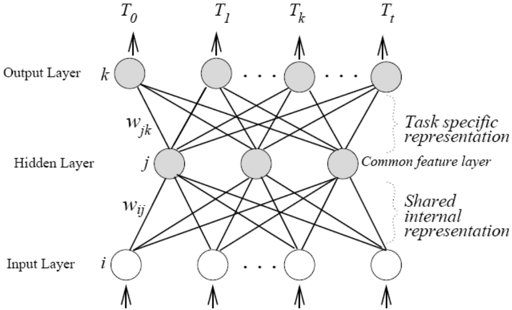


Structural Learning with Linear Predictors

$$f(x) = w^T \cdot \underbrace{\phi(x)}_{\text{task specific features}} + v^T \cdot \underbrace{\psi_\theta(x)}_{\text{internal dimensions}}$$

How to represent θ? A matrix(can be considered as a transformation matrix to find new dimensions)

$$f_\theta(w, v; x) = w^T \phi(x) + v^T \theta\psi(x)$$



Alternating structure optimization(1)

Assume $\phi(x) = \psi(x) = x$, it follows that

$$[\{\hat{w}_l, \hat{v}_l\}, \hat{\theta}] =$$

$$\arg\min_{\{w_l, v_l\}, \theta} \sum_{l=1}^{m} \left(\frac{1}{n_l} \sum_{i=1}^{n_l} L((w_l + \theta^T v_l)^T X_i^l, Y_i^l) + \lambda_l ||w_l||_2^2 \right)$$

$$s.t. \qquad \underbrace{\theta\theta^T = I}_{\text{equivalent to regularization}}$$

Let $u = w + v\theta^T$, then $f(x) = u^T x$.

$$\min \sum_{l=1}^{m} \left(\frac{1}{n_l} \sum_{i=1}^{n_l} L(u_l^T X_i^l, Y_i^l) + \lambda_l ||u_l - \theta^T v_l||_2^2 \right)$$

$$s.t. \qquad \theta\theta^T = I$$

Alternating structure optimization(1)

Assume $\phi(x) = \psi(x) = x$, it follows that

$$[\{\hat{w}_l, \hat{v}_l\}, \hat{\theta}] =$$

$$\arg\min_{\{w_l, v_l\}, \theta} \sum_{l=1}^{m} \left(\frac{1}{n_l} \sum_{i=1}^{n_l} L((w_l + \theta^T v_l)^T X_i^l, Y_i^l) + \lambda_l ||w_l||_2^2 \right)$$

$$s.t. \qquad \underbrace{\theta\theta^T = I}_{\text{equivalent to regularization}}$$

Let $u = w + v\theta^T$, then $f(x) = u^T x$.

$$\min \textstyle\sum_{l=1}^{m} \left(\frac{1}{n_l} \sum_{i=1}^{n_l} L(u_l^T X_i^l, Y_i^l) + \lambda_l ||u_l - \theta^T v_l||_2^2 \right)$$

$$s.t. \qquad \theta\theta^T = I$$

Alternating structure optimization (2)

Algorithm

1. Fix (θ, v), optimize with respect to u (a convex optimization problem)
2. Fix u, optimize with respect to (θ, v). It turns out θ are the top left eigenvectors for the SVD of a matrix

$$U = [\sqrt{\lambda_1} u_1, \sqrt{\lambda_2} u_2, \cdots, \sqrt{\lambda_m} u_m]$$

3. Iterate until convergence.
4. Usually one iteration is enough.

Connection to PCA

- PCA find the "principal components" of data points.
- u_l is actually the predictor for task l. It is finding the "principal components" of the predictors.
- Each predictor is considered a point in the predictor space.

Alternating structure optimization (2)

Algorithm

1. Fix (θ, v), optimize with respect to u (a convex optimization problem)
2. Fix u, optimize with respect to (θ, v). It turns out θ are the top left eigenvectors for the SVD of a matrix

$$U = [\sqrt{\lambda_1}u_1, \sqrt{\lambda_2}u_2, \cdots, \sqrt{\lambda_m}u_m]$$

3. Iterate until convergence.
4. Usually one iteration is enough.

Connection to PCA

- PCA find the "principal components" of data points.
- u_l is actually the predictor for task l. It is finding the "principal components" of the predictors.
- Each predictor is considered a point in the predictor space.

Semi-supervised learning

1. Learn structure parameter θ by joint empirical risk minimization.
2. Learn a predictor based on θ

How to generate auxiliary problems?

- Automatic labeling.
- Relevancy.

Two strategies:

- Unsupervised
- Semi-supervised

1. Learn structure parameter θ by joint empirical risk minimization.
2. Learn a predictor based on θ

How to generate auxiliary problems?

- Automatic labeling.
- Relevancy.

Two strategies:

- Unsupervised
- Semi-supervised

Semi-supervised learning

1. Learn structure parameter θ by joint empirical risk minimization.
2. Learn a predictor based on θ

How to generate auxiliary problems?

- Automatic labeling.
- Relevancy.

Two strategies:

- Unsupervised
- Semi-supervised

Auxiliary Problem Generation(unsupervised)

Two problems: text categorization, word tagging.

Predicting observable sub-structure

Mask some features as unobserved, learn classifiers to predict these "masked" features.

$$W_1 = \{\text{"}stadium\text{"}, \text{"}scientist\text{"}, \text{"}stock\text{"}\};$$
$$W_2 = \{\text{"}baseball\text{"}, \text{"}basketball\text{"}, \text{"}physics\text{"}, \text{"}marker\text{"}\}$$

- Let W_1 be unobserved. predict whether "stadium" occurs more than other two words in the document.
- Predict the words at current position given the words on the left and right.

Predicting the behavior of target classifier(semi-supervised)

1. Train a classifier T_1 with labeled data for the target task, using feature map ϕ_1.
2. Propogate the labels to unlabeled data.
3. Learn structural parameter θ by joint ERM on the auxiliary problems using feature map ϕ_2.
4. Train a final classifier based on θ and some appropriate feature map ϕ_3.

Several examples

- Predict the prediction of classifier T_1
- Predict the top-k choices of the classifier
- Predict the range of confidence values produced by the classifier (whether the confidence value is larger than a threshold)

Predicting the behavior of target classifier(semi-supervised)

1. Train a classifier T_1 with labeled data for the target task, using feature map ϕ_1.
2. Propogate the labels to unlabeled data.
3. Learn structural parameter θ by joint ERM on the auxiliary problems using feature map ϕ_2.
4. Train a final classifier based on θ and some appropriate feature map ϕ_3.

Several examples

- Predict the prediction of classifier T_1
- Predict the top-k choices of the classifier
- Predict the range of confidence values produced by the classifier (whether the confidence value is larger than a threshold)

Experiments

Data sets

- Text categorization (20-newsgroup, RCV1)
- named entity chunking experiment (CoNLL'03 corpora)
- Part-of-Speech tagging (Brown corpus)
- Hand-written digit image classification (MNIST)

- supervised learning based on Huber's robust loss

$$L(p, y) = \begin{cases} max(0, 1 - py)^2 & \textit{if } py \geq -1 \\ -4py & \textit{otherwise} \end{cases}$$

- semi-supervised learning proposed by this work with different auxiliary problems
- Co-training
- One manifold learning method (See *Semi-supervised learning on Riemannian manifolds*)

Experiments

Data sets

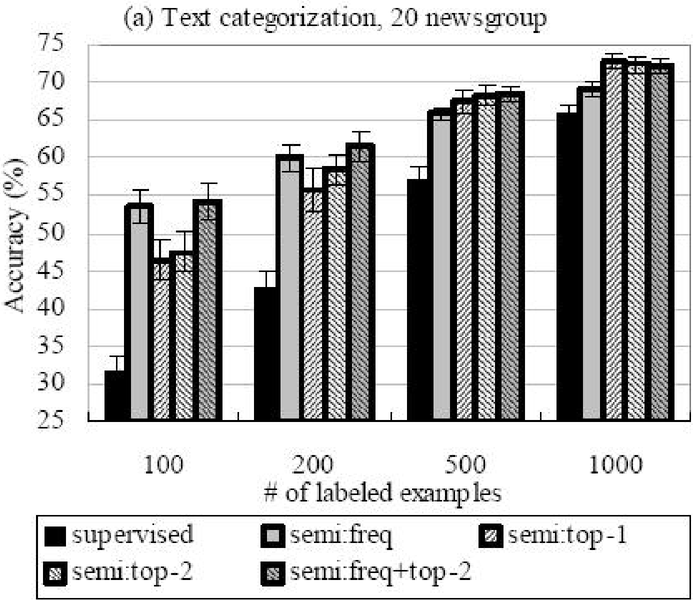
- Text categorization (20-newsgroup, RCV1)
- named entity chunking experiment (CoNLL'03 corpora)
- Part-of-Speech tagging (Brown corpus)
- Hand-written digit image classification (MNIST)

- supervised learning based on Huber's robust loss

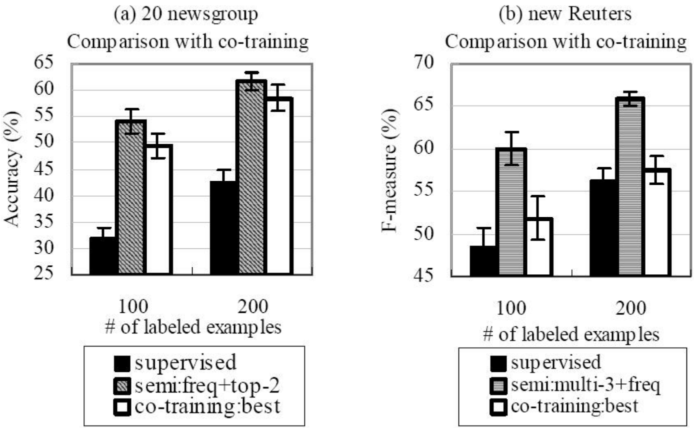
$$L(p, y) = \begin{cases} max(0, 1 - py)^2 & if\ py \geq -1 \\ -4py & otherwise \end{cases}$$

- semi-supervised learning proposed by this work with different auxiliary problems
- Co-training
- One manifold learning method (See *Semi-supervised learning on Riemannian manifolds*)

Accuracy on Text



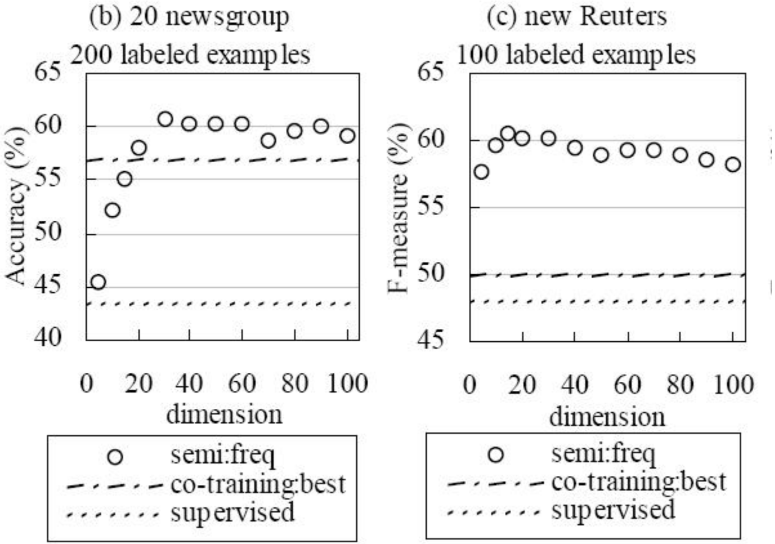
Comparison to Co-training



Comparison to Manifold learning

# of labeled examples	BN04 best (manifold)	ASO-semi
100	39.8	54.1
200	–	61.6
500	59.9	68.5
1000	64.0	72.3

Sensitivity to internal dimensions



Interpretations of Internal dimensions

row#		features
2	+	pc, vesa, ibm, boards
	−	god, christian, bible, exist, doctrine, nature, worship, athos.rutgers.edu
3	+	team, detroit, series, leafs, play, cup, playoffs, played, penguins, devils
	−	israel, peace, jewish, lebanese, israelis, land, gaza, civilians, palestine, syria
4	+	files, jpeg, pov, utility, ms-windows, icon
	−	eisa, nubus, agents, attorney
5	+	oil, bikes, front, brake, rear, transmission, owner, driving, dogs, highway
	−	printer, hp, ink, appreciate, bj-200, toner, printing, bubblejet, laserjet, gcc

Computer vs religion

Sports vs Middle east issues

No silver bulletin

- This method seems way too good. But actually it's not.
- I tried Information Gain to select 2000 features and run NBC on 20 newsgroup, and it performs comparable to their method, sometimes a significant improvement.
- I think this method is basically adding some features to the original feature space. Unfortunately, no comparison with PCA+supervised learning.
- Why this method works is still not clear to me? The authors argue that "adding irrelevant features won't hurt, but adding relevant features will yield a huge gain". Why?? Can we inject 1000 random features to the data set? Still work?
- They provide a theory to show MTL's perform gain is guaranteed. But actually, we only care about the target task. What if on average it improves, but target task's performance decreases? MTL $\neq$ target task!!!
- Only works on high dimensional data?
- Currently, no MTL method compared to this work.

No silver bulletin

- This method seems way too good. But actually it's not.
- I tried Information Gain to select 2000 features and run NBC on 20 newsgroup, and it performs comparable to their method, sometimes a significant improvement.
- I think this method is basically adding some features to the original feature space. Unfortunately, no comparison with PCA+supervised learning.
- Why this method works is still not clear to me? The authors argue that "adding irrelevant features won't hurt, but adding relevant features will yield a huge gain". Why?? Can we inject 1000 random features to the data set? Still work?
- They provide a theory to show MTL's perform gain is guaranteed. But actually, we only care about the target task. What if on average it improves, but target task's performance decreases? MTL $\neq$ target task!!!
- Only works on high dimensional data?
- Currently, no MTL method compared to this work.

No silver bulletin

- This method seems way too good. But actually it's not.
- I tried Information Gain to select 2000 features and run NBC on 20 newsgroup, and it performs comparable to their method, sometimes a significant improvement.
- I think this method is basically adding some features to the original feature space. Unfortunately, no comparison with PCA+supervised learning.
- Why this method works is still not clear to me? The authors argue that "adding irrelevant features won't hurt, but adding relevant features will yield a huge gain". Why?? Can we inject 1000 random features to the data set? Still work?
- They provide a theory to show MTL's perform gain is guaranteed. But actually, we only care about the target task. What if on average it improves, but target task's performance decreases? MTL $\neq$ target task!!!
- Only works on high dimensional data?
- Currently, no MTL method compared to this work.

No silver bulletin

- This method seems way too good. But actually it's not.
- I tried Information Gain to select 2000 features and run NBC on 20 newsgroup, and it performs comparable to their method, sometimes a significant improvement.
- I think this method is basically adding some features to the original feature space. Unfortunately, no comparison with PCA+supervised learning.
- Why this method works is still not clear to me? The authors argue that "adding irrelevant features won't hurt, but adding relevant features will yield a huge gain". Why?? Can we inject 1000 random features to the data set? Still work?
- They provide a theory to show MTL's perform gain is guaranteed. But actually, we only care about the target task. What if on average it improves, but target task's performance decreases? MTL $\neq$ target task!!!
- Only works on high dimensional data?
- Currently, no MTL method compared to this work.

No silver bulletin

- This method seems way too good. But actually it's not.
- I tried Information Gain to select 2000 features and run NBC on 20 newsgroup, and it performs comparable to their method, sometimes a significant improvement.
- I think this method is basically adding some features to the original feature space. Unfortunately, no comparison with PCA+supervised learning.
- Why this method works is still not clear to me? The authors argue that "adding irrelevant features won't hurt, but adding relevant features will yield a huge gain". Why?? Can we inject 1000 random features to the data set? Still work?
- They provide a theory to show MTL's perform gain is guaranteed. But actually, we only care about the target task. What if on average it improves, but target task's performance decreases? MTL $\neq$ target task!!!
- Only works on high dimensional data?
- Currently, no MTL method compared to this work.

No silver bulletin

- This method seems way too good. But actually it's not.
- I tried Information Gain to select 2000 features and run NBC on 20 newsgroup, and it performs comparable to their method, sometimes a significant improvement.
- I think this method is basically adding some features to the original feature space. Unfortunately, no comparison with PCA+supervised learning.
- Why this method works is still not clear to me? The authors argue that "adding irrelevant features won't hurt, but adding relevant features will yield a huge gain". Why?? Can we inject 1000 random features to the data set? Still work?
- They provide a theory to show MTL's perform gain is guaranteed. But actually, we only care about the target task. What if on average it improves, but target task's performance decreases? MTL $\neq$ target task!!!
- Only works on high dimensional data?
- Currently, no MTL method compared to this work.

No silver bulletin

- This method seems way too good. But actually it's not.
- I tried Information Gain to select 2000 features and run NBC on 20 newsgroup, and it performs comparable to their method, sometimes a significant improvement.
- I think this method is basically adding some features to the original feature space. Unfortunately, no comparison with PCA+supervised learning.
- Why this method works is still not clear to me? The authors argue that "adding irrelevant features won't hurt, but adding relevant features will yield a huge gain". Why?? Can we inject 1000 random features to the data set? Still work?
- They provide a theory to show MTL's perform gain is guaranteed. But actually, we only care about the target task. What if on average it improves, but target task's performance decreases? MTL $\neq$ target task!!!
- Only works on high dimensional data?
- Currently, no MTL method compared to this work.

Conclusions

Contributions

- A framework for MTL(seems robust to unrelated tasks)
- Automatically generate auxiliary problems

	data-mining	structural-mining
space of interest	data space	predictor space
instances	data-points	predictors from multiple tasks
uncertainty	measurement error	estimation error
goal	find patterns in data	find structures of the predictors
predictive power	maybe	yes
duality	a data point is a predictor of points in the predictor-space	

Figure 16: Data mining versus structural mining

Conclusions

Contributions

- A framework for MTL(seems robust to unrelated tasks)
- Automatically generate auxiliary problems

	data-mining	structural-mining
space of interest	data space	predictor space
instances	data-points	predictors from multiple tasks
uncertainty	measurement error	estimation error
goal	find patterns in data	find structures of the predictors
predictive power	maybe	yes
duality	a data point is a predictor of points in the predictor-space	

Figure 16: Data mining versus structural mining