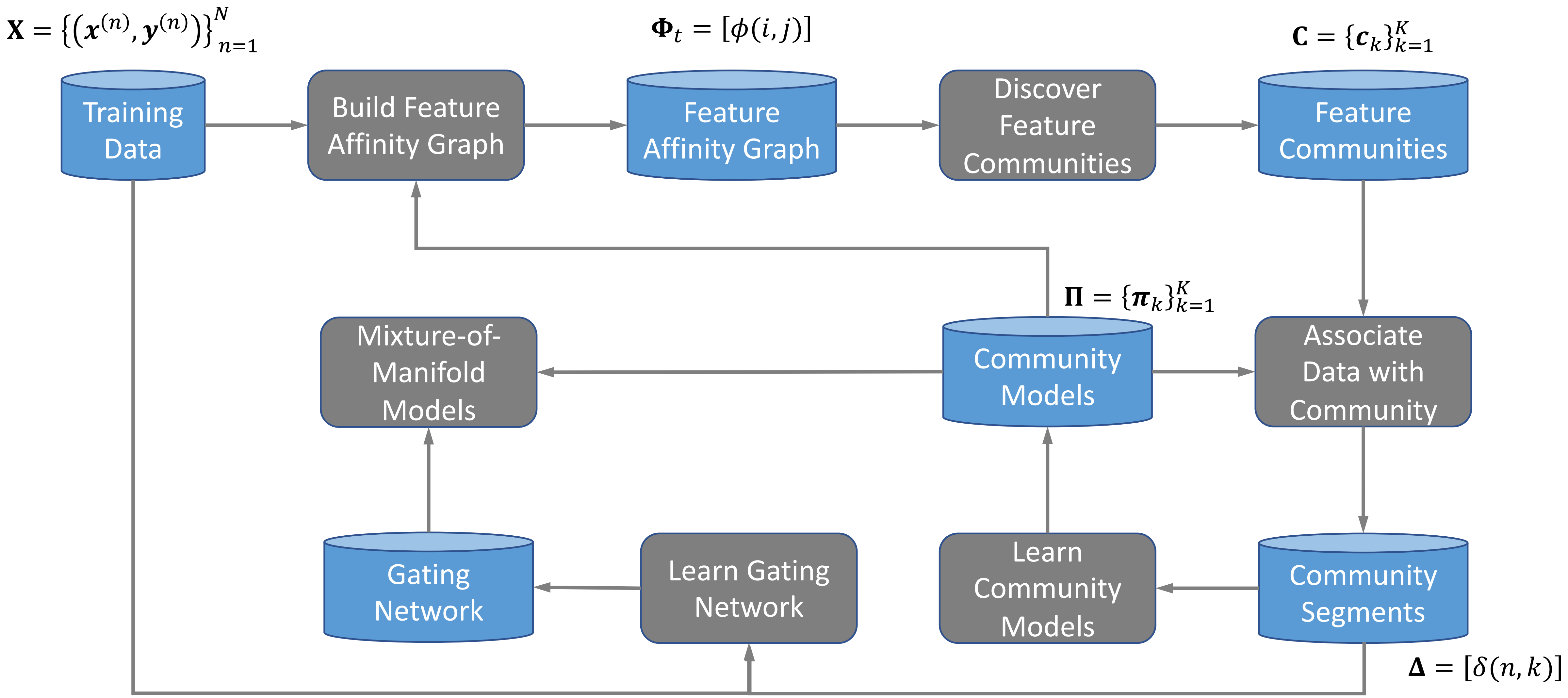
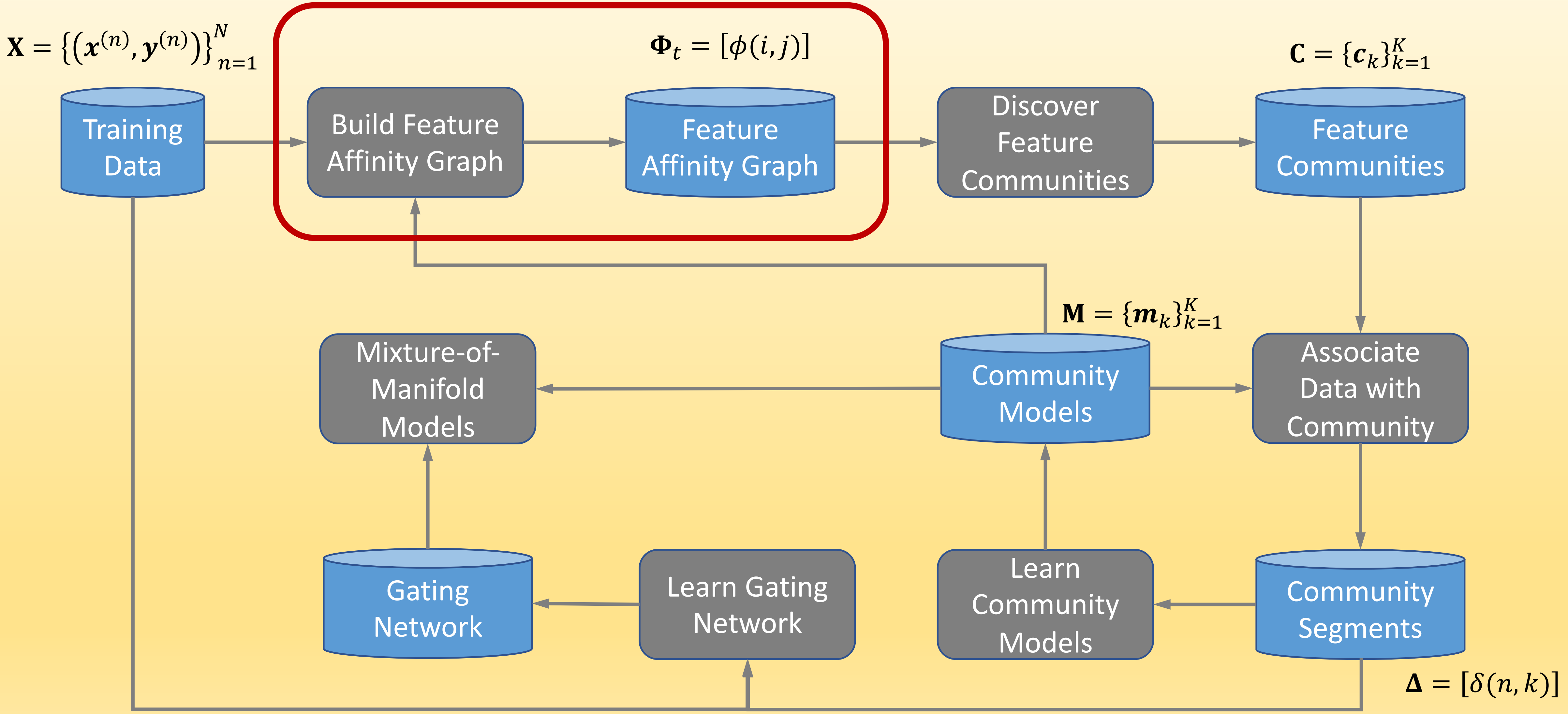
Mixture-of-Manifold Experts

Shailesh Kumar | Anurag Sahoo

Overall Architecture



Build Feature Affinity Graph: Φ_0



1. Build Feature Affinity Graph: Φ_0

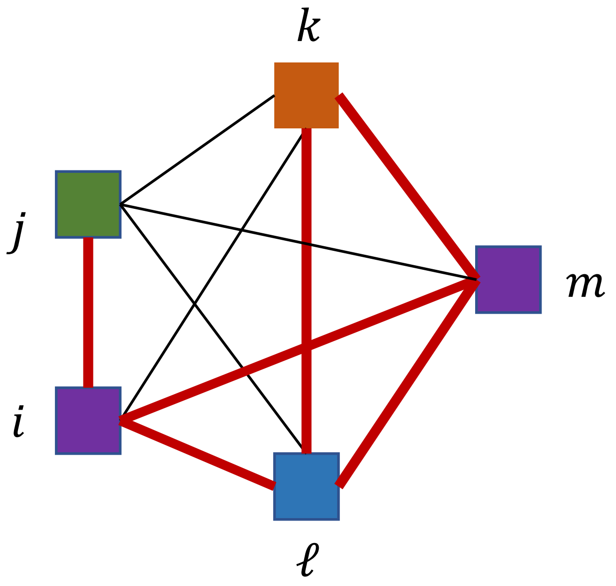
Two features have a high affinity with each other the if model accuracy is significantly higher when they are together than independently.

$\theta_0(i) =$ Goodness of model built with feature i only

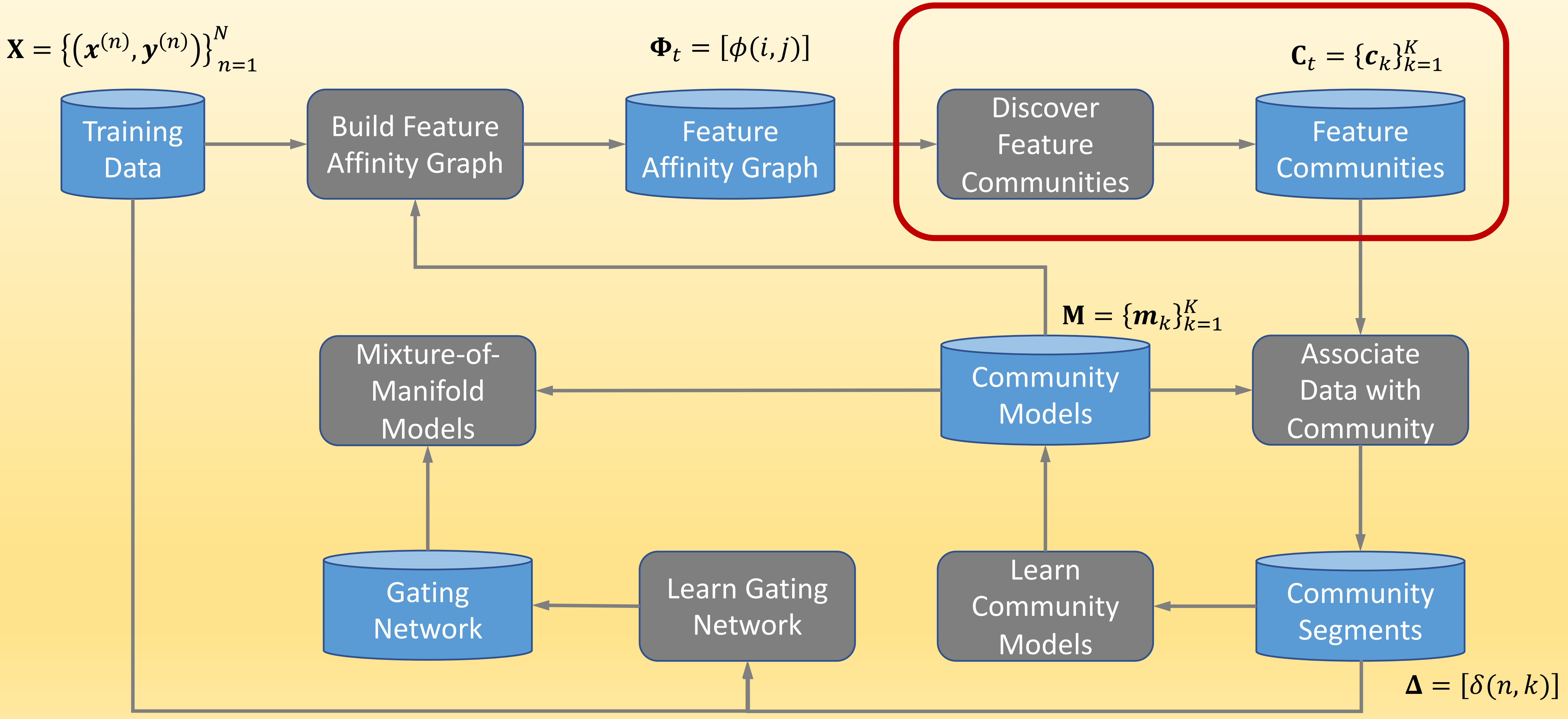
$\theta_0(i, j) =$ Goodness of model built with feature pair $\{i, j\}$

$\phi_0(i, j)$= Pair-wise affinity between features $\{i, j\}$

$$\phi_0(i, j) = \frac{\theta_0(i, j)}{\sqrt{\theta_0(i)\theta_0(j)}}$$



Discover Feature Communities: $\mathbf{C}_t$



2. Discover Feature Communities: $\mathbf{C}_t$

A community (soft maximal clique) of features is such that removing a feature from it or adding a feature to it decreases its coherence.

$\boldsymbol{c}$ = Any community (set) of features (possibly overlapping with others)

$\pi(\boldsymbol{c})$ = Coherence of community $\boldsymbol{c} = \lambda_1(\boldsymbol{c}) \times \min\{\boldsymbol{v}_1(\boldsymbol{c})\}$

$\lambda_1(\boldsymbol{c})$ = First Eigen Value of the Affinity Sub-matrix $\boldsymbol{\Phi}(\boldsymbol{c})$

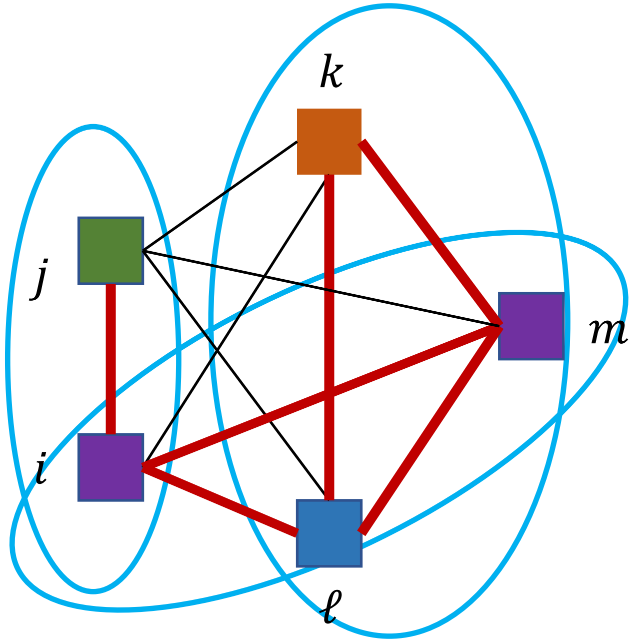
$\boldsymbol{v}_1(\boldsymbol{c})$ = First Eigen Vector of the Affinity Sub-matrix $\boldsymbol{\Phi}(\boldsymbol{c})$

$\boldsymbol{c}^*$ = A locally optimal (soft maximal clique) community such that:

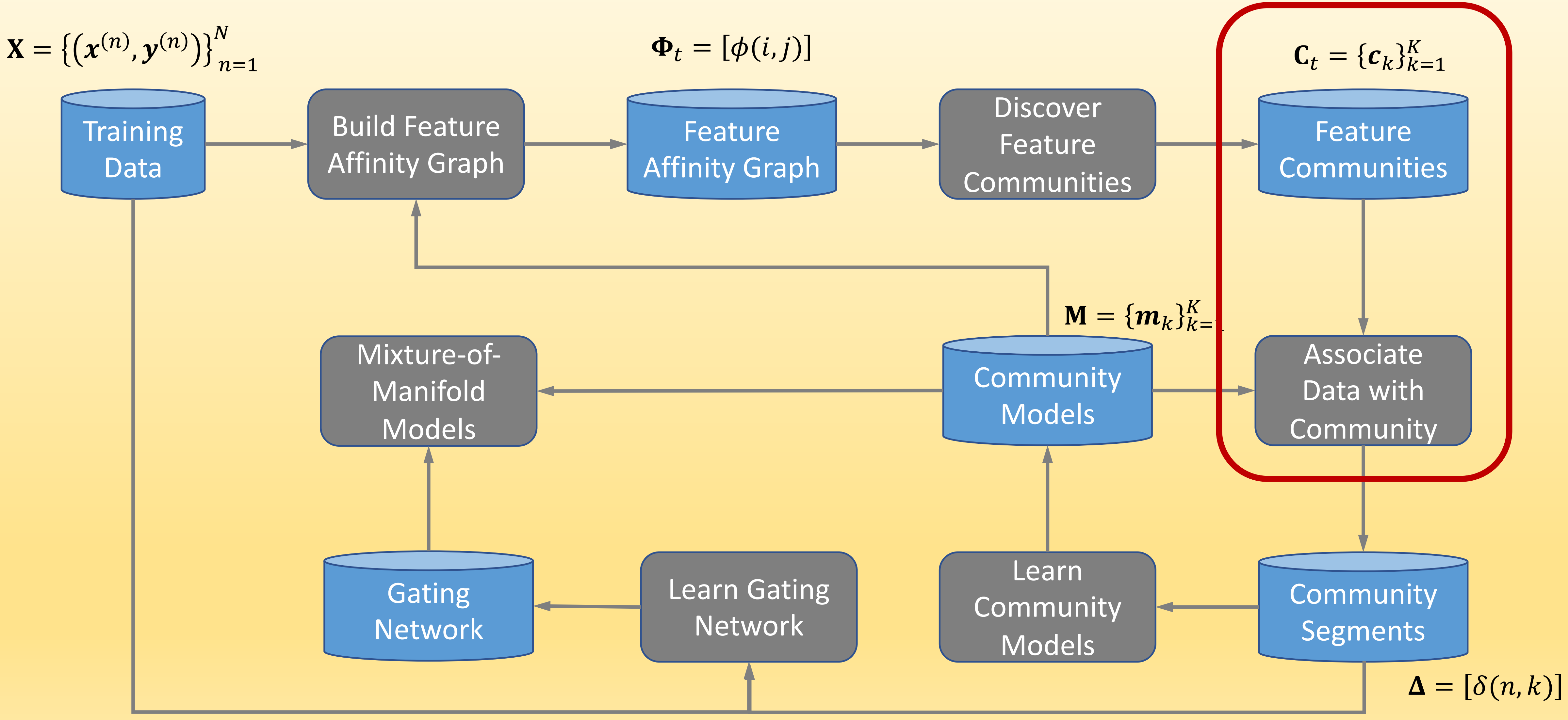
$\pi(\boldsymbol{c}^*) \geq \pi(\boldsymbol{c}^* \ominus i), \forall i \in \boldsymbol{c}^*$ (i.e. removing a feature reduces coherence)

$\pi(\boldsymbol{c}^*) \geq \pi(\boldsymbol{c}^* \oplus i), \forall i \notin \boldsymbol{c}^*$ (i.e. adding a feature also reduces coherence)

$\mathbf{C}_t$ = Set of all soft maximal communities in the feature affinity graph



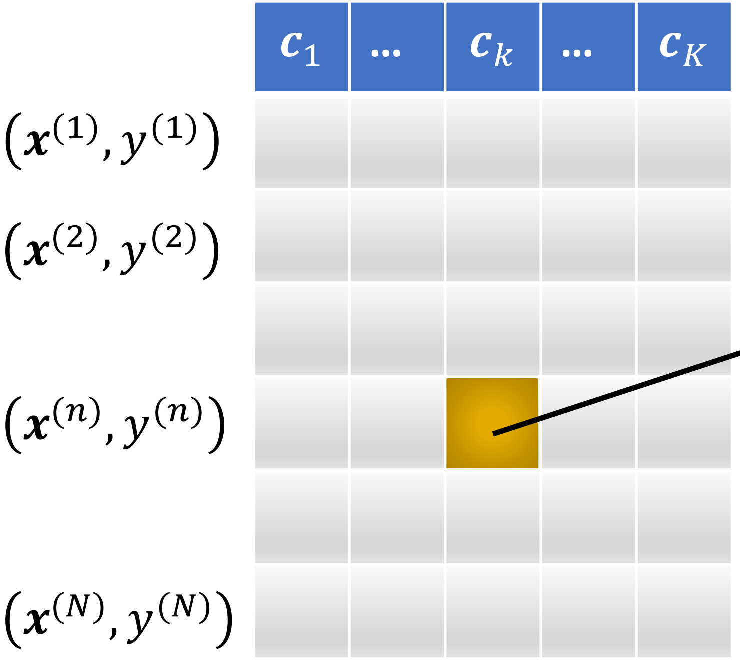
Initialize Data Association with Communities



Initialize Data Association with Communities

Each Feature Community represents a “Sub-Space Manifold”. We now need to associate each data point with one of the sub-space manifold.

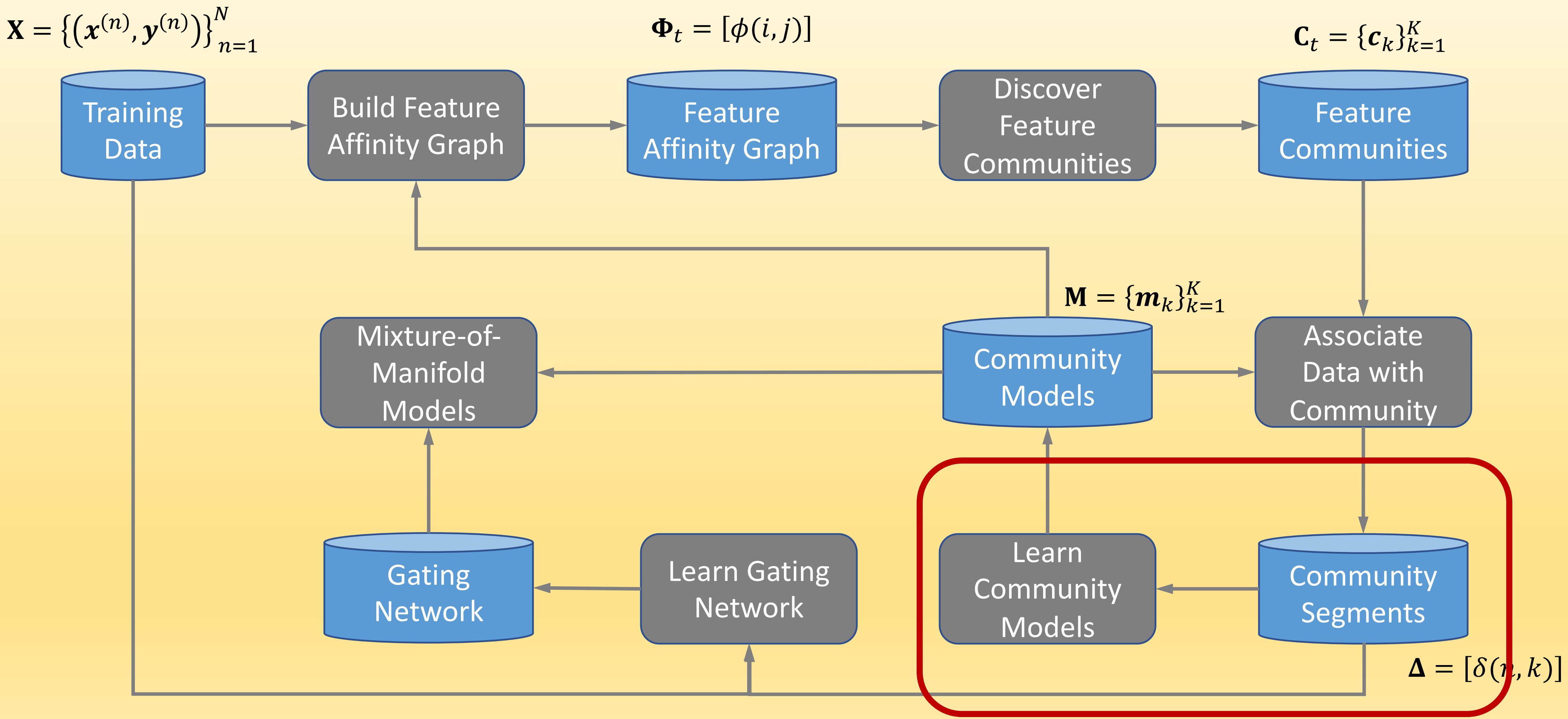
$\mathbf{\Delta} = [\delta(n,k)] =$ is the association of $\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)$ with community $\boldsymbol{c}_k$



$$\delta_0(n,k) = \frac{1}{K}$$

Initially all data points are equally associated with all communities

Learn Community Models



Learn Community Models

Given a modelling technique (expert type), association of a data point with a community, and the community features, learn a model.

$$\mathbf{M}_t = \left\{\boldsymbol{m}_k^{(t)}\right\}_{k=1}^{K} = \text{Set of models trained in iteration } t \text{ of association loop.}$$

	$\boldsymbol{c}_1$	…	$\boldsymbol{c}_k$	…	$\boldsymbol{c}_K$
$\left(\boldsymbol{x}^{(1)}, y^{(1)}\right)$					
$\left(\boldsymbol{x}^{(2)}, y^{(2)}\right)$					
$\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)$					
$\left(\boldsymbol{x}^{(N)}, y^{(N)}\right)$					

$\boldsymbol{\delta}_t(*, k)$

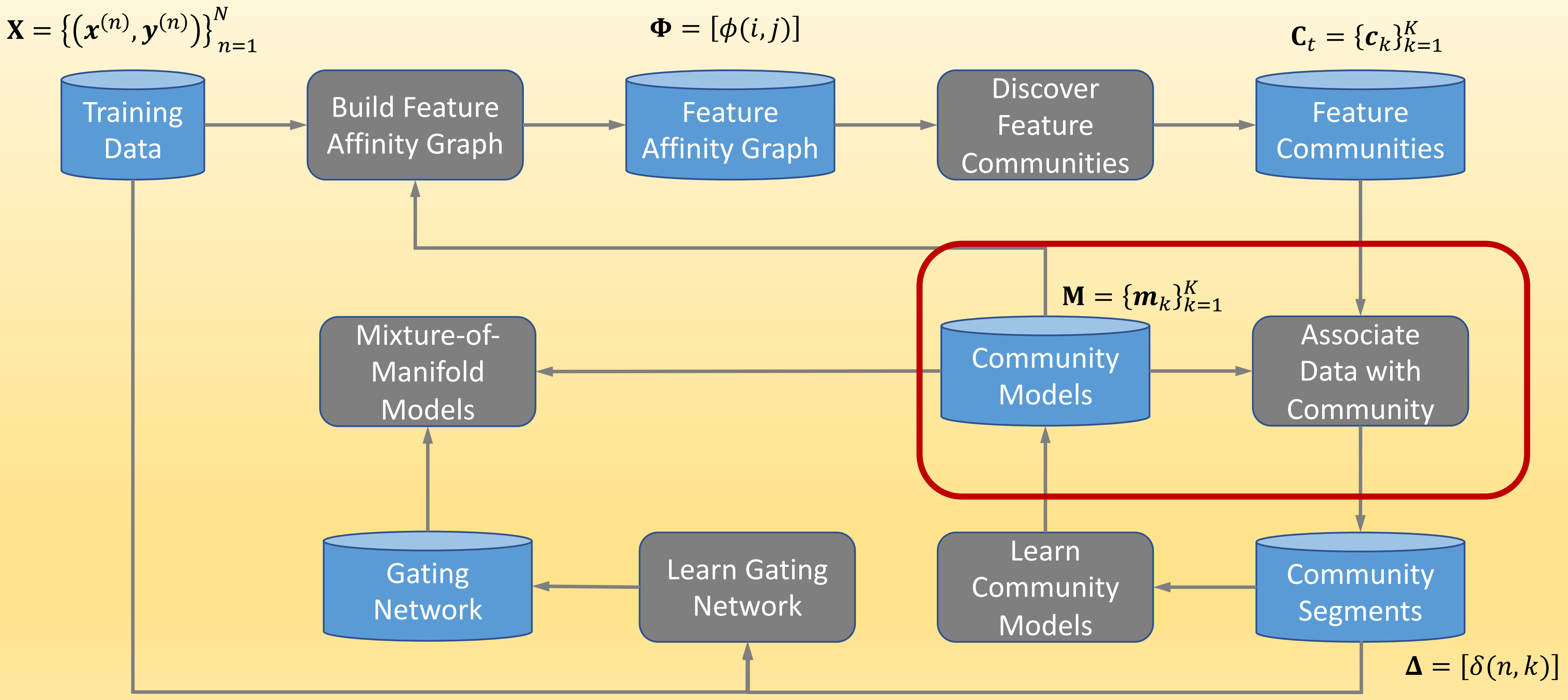
$$\boldsymbol{m}_k^{(t)} = \text{Train}\left(\mathbf{X}, \boldsymbol{\delta}_t(*, k)\right), \forall k = 1 \ldots K$$

$\delta_t(n, k) =$ Weight of the nth data point for kth model.

In the HARD version, assign each data point to max weighted.

In the SOFT version, assign each data point proportional to weigh.

Update association of data with community



Update Association of data with Community

A data point should be associated with that community whose current model gives highest confidence, correct label for this data point

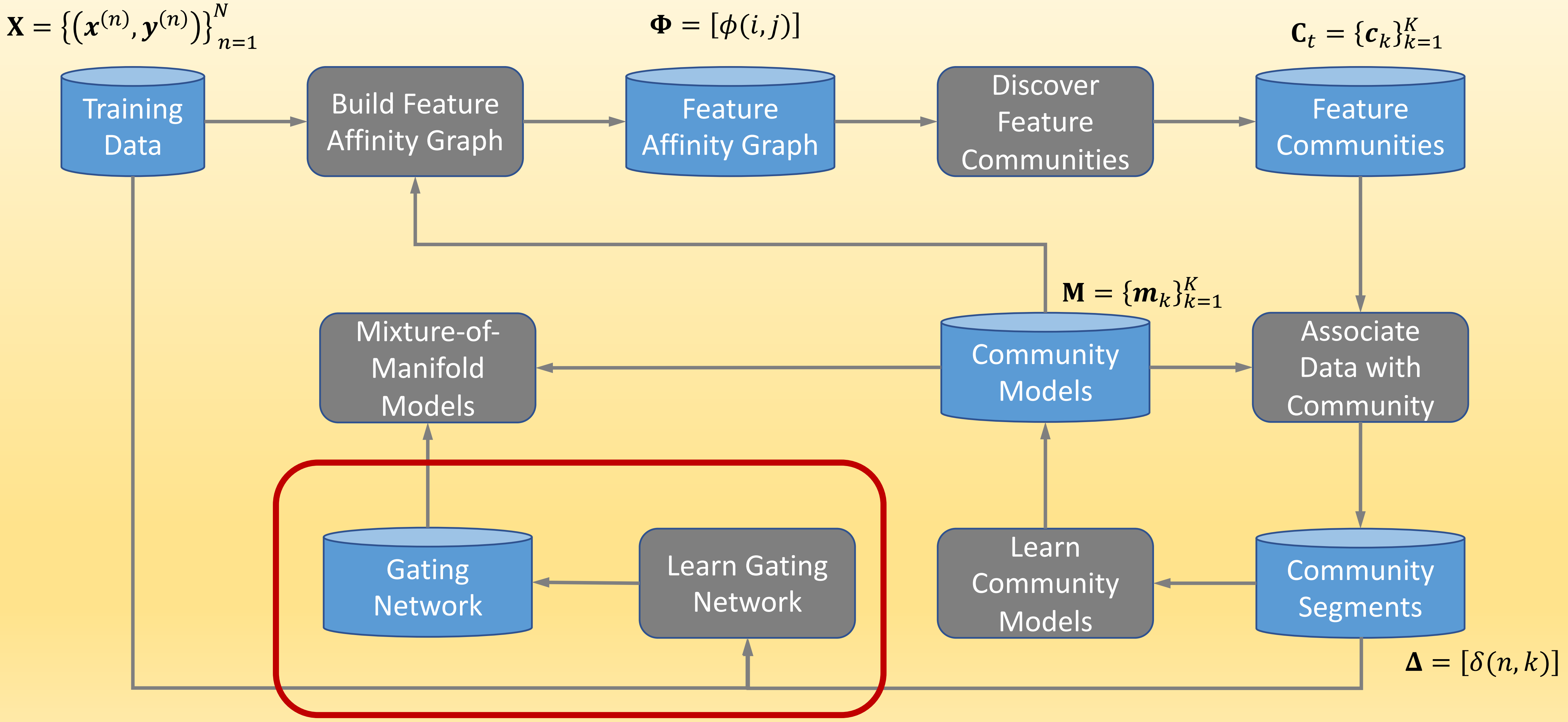
$(\ell_k^{(n)}, f_k^{(n)}) = Predict\left(\boldsymbol{m}_k^{(t)}, \boldsymbol{x}^{(n)}\right)$ = Predict label and confidence for each data point w.r.t. each model

$\ell_k^{(n)} \in \{-1,1\}$ class label (two class problem)

$f_k^{(n)} = P\left(\ell_k^{(n)} | \boldsymbol{x}_k^{(n)}\right), \forall k = 1 \ldots K$ = confidence score in the predicted class label

$\delta_{n,k}^{(t+1)} = \frac{\exp\left(\beta_t \times y_k^{(n)} \times \ell_k^{(n)} \times f_k^{(n)}\right)}{\sum_j \exp\left(\beta_t \times y_j^{(n)} \times \ell_j^{(n)} \times f_j^{(n)}\right)}$ = Updated association of $\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)$ with community k

Learn Gating Network



Learn Gating Network

When a new data point arrives, the gating network first determines the weightage of each of the experts and combines their outputs.

$\boldsymbol{x}^{(n)}$ = Training Input to the gating network

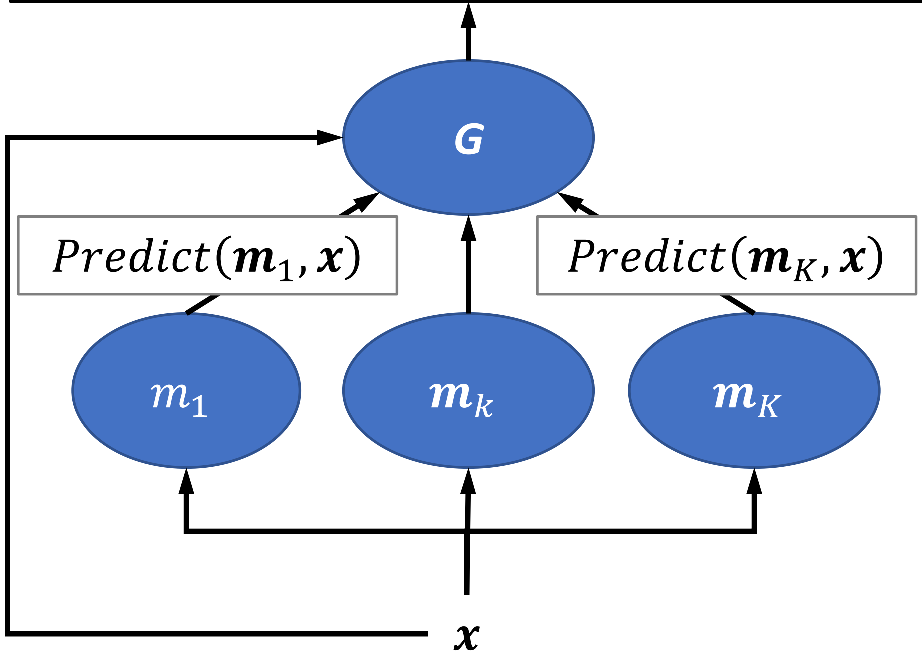
$\delta(n,k)$ = Training Output of the gating network

$G(\boldsymbol{x})$ = Output of the trained gating network

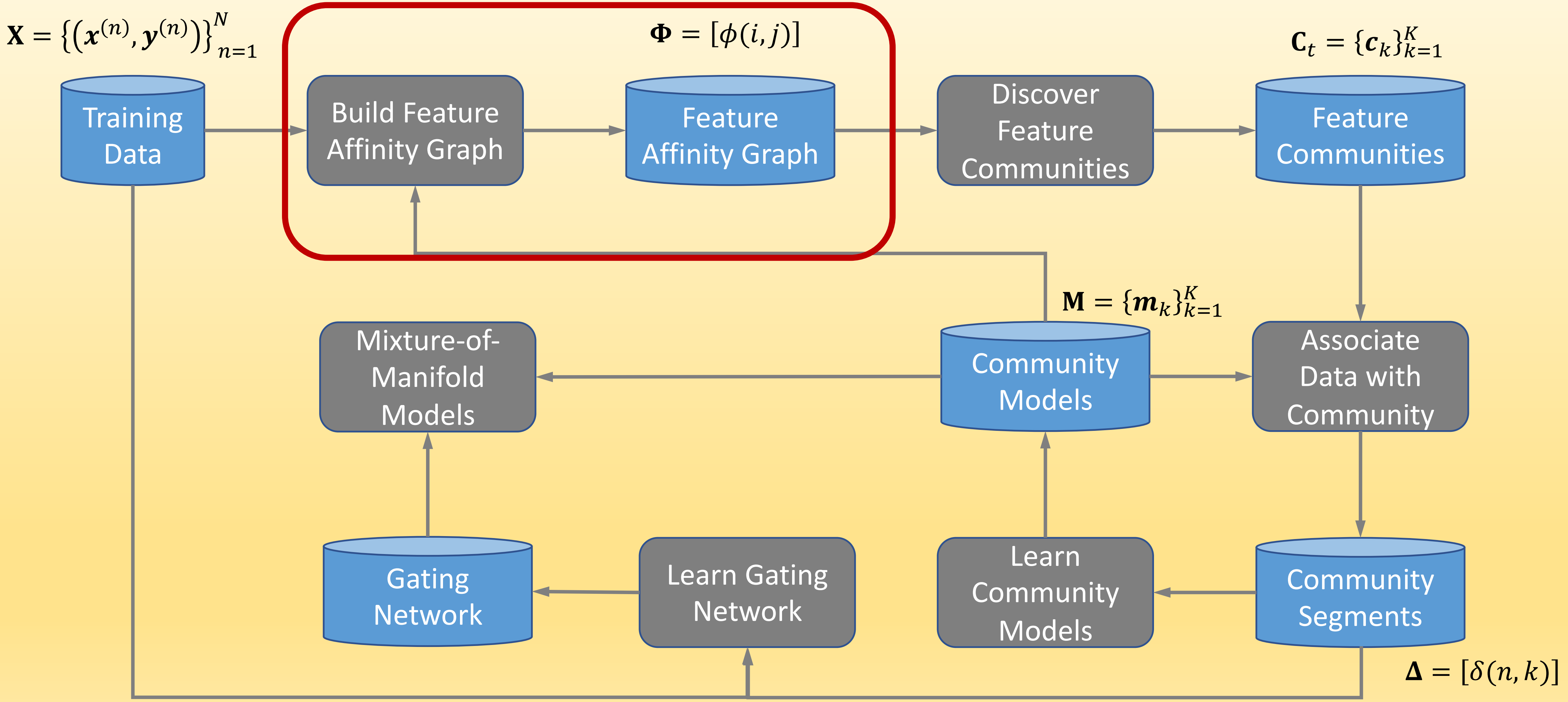
$$G(\boldsymbol{x}) = (P(1|\boldsymbol{x}), P(2|\boldsymbol{x}), \ldots, P(K|\boldsymbol{x}))$$

$P(k|\boldsymbol{x})$ = probability that gate k is best for input $\mathbf{x}$

$$P(y|\boldsymbol{x}) = \sum_{k=1}^{K} P(k|\boldsymbol{x}) \times Predict(\boldsymbol{m}_k, \boldsymbol{x})$$



Update Feature Affinity Matrix



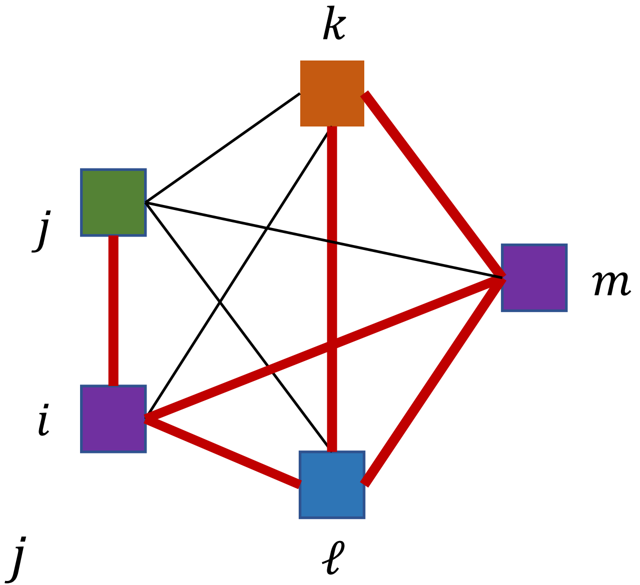
Update Feature Affinity Matrix

The initial Affinity Matrix was created based on all data points and only on pair-wise combinations. Now can we do better?

$w(i|k) = \text{Importance of feature } i \text{ in model } k$
$w(i|k) = 0 \text{ if feature } i \text{ i is not present in model } k$

$A(k) = \text{Accuracy/AUC (some metric) of model } k$

$\theta_t(i) = \text{Expected Goodness of feature } i$
$$\theta_t(i) = \frac{\sum_{k=1}^{K} w(i|k)A(k)}{\sum_{k=1}^{K} w(i|k)}$$



$\theta_t(i,j) = \text{Expected Goodness of feature pair } i, j$
$$\theta_t(i,j) = \frac{\sum_{k=1}^{K} w(i|k)w(j|k)A(k)}{\sum_{k=1}^{K} w(i|k)w(j|k)}$$

$\phi_0(i,j) = \text{Pair-wise affinity of feature pair } i, j$
$$\phi_t(i,j) = \frac{\theta_t(i,j)}{\sqrt{\theta_t(i)\theta_t(j)}}$$