How to do Predictive Analytics with Limited Data

Ulrich Rueckert

Agenda

- Introduction and Motivation
- Semi-Supervised Learning
  - Generative Models
  - Large Margin Approaches
  - Similarity Based Methods
- Conclusion

Predictive Modeling

- Traditional Supervised Learning
  - Promotion on bookseller's web page
  - Customers can rate books.
  - Will a new customer like this book?
  - Training set: observations on previous customers
  - Test set: new customers
- What happens if only few customers rate a book?

Attributes

Target Label

Age	Income	LikesBook
24	60K	+
65	80K	-
60	95K	-
35	52K	+
20	45K	+
43	75K	+
26	51K	+
52	47K	-
47	38K	-
25	22K	-
33	47K	+

Training Data

Test Data

Age	Income	LikesBook
22	67K	?
39	41K	?

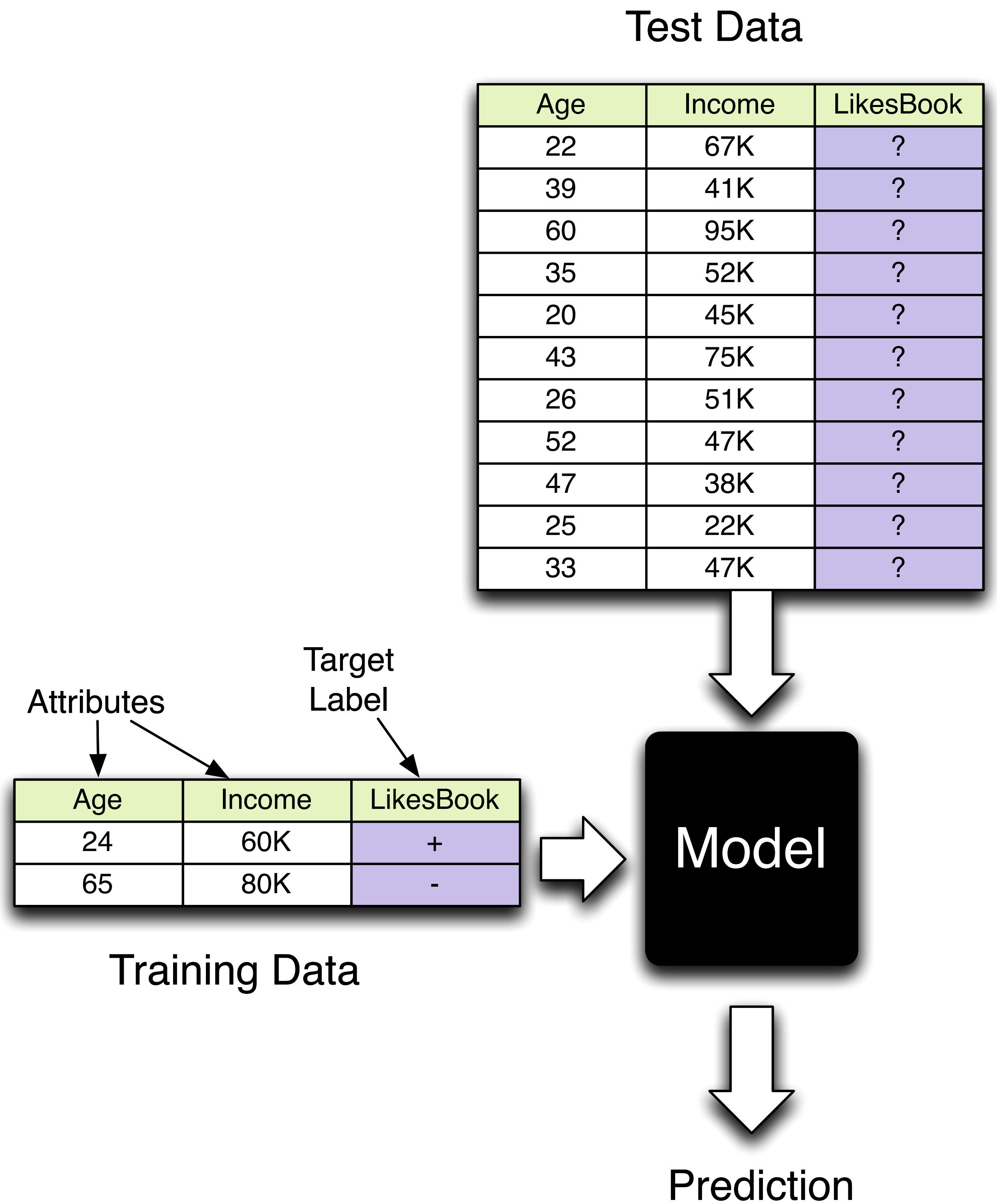
Model

Age	Income	LikesBook
22	67K	+
39	41K	-

Prediction

Predictive Modeling

- Traditional Supervised Learning
  - Promotion on bookseller's web page
  - Customers can rate books.
  - Will a new customer like this book?
  - Training set: observations on previous customers
  - Test set: new customers
- What happens if only few customers rate a book?



Test Data

Age	Income	LikesBook
22	67K	?
39	41K	?
60	95K	?
35	52K	?
20	45K	?
43	75K	?
26	51K	?
52	47K	?
47	38K	?
25	22K	?
33	47K	?

Training Data

Age	Income	LikesBook
24	60K	+
65	80K	-

Predictive Modeling

- Traditional Supervised Learning
  - Promotion on bookseller's web page
  - Customers can rate books.
  - Will a new customer like this book?
  - Training set: observations on previous customers
  - Test set: new customers
- What happens if only few customers rate a book?

Attributes

Target Label

Age	Income	LikesBook
24	60K	+
65	80K	-
60	95K	?
35	52K	?
20	45K	?
43	75K	+
26	51K	?
52	47K	-
47	38K	?
25	22K	?
33	47K	?

Training Data

Test Data

Age	Income	LikesBook
22	67K	?
39	41K	?

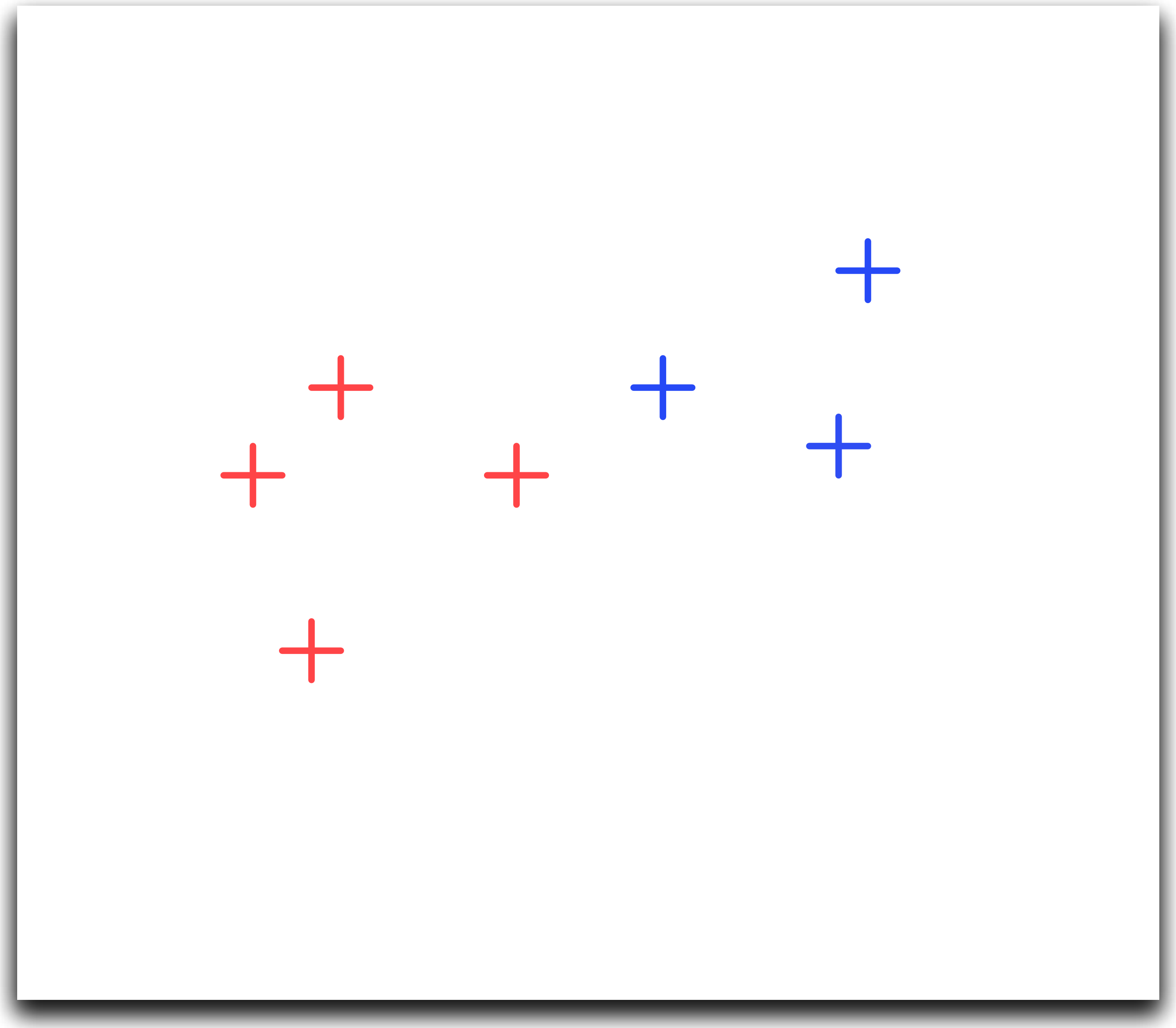
Model

Age	Income	LikesBook
22	67K	+
39	41K	-

Prediction

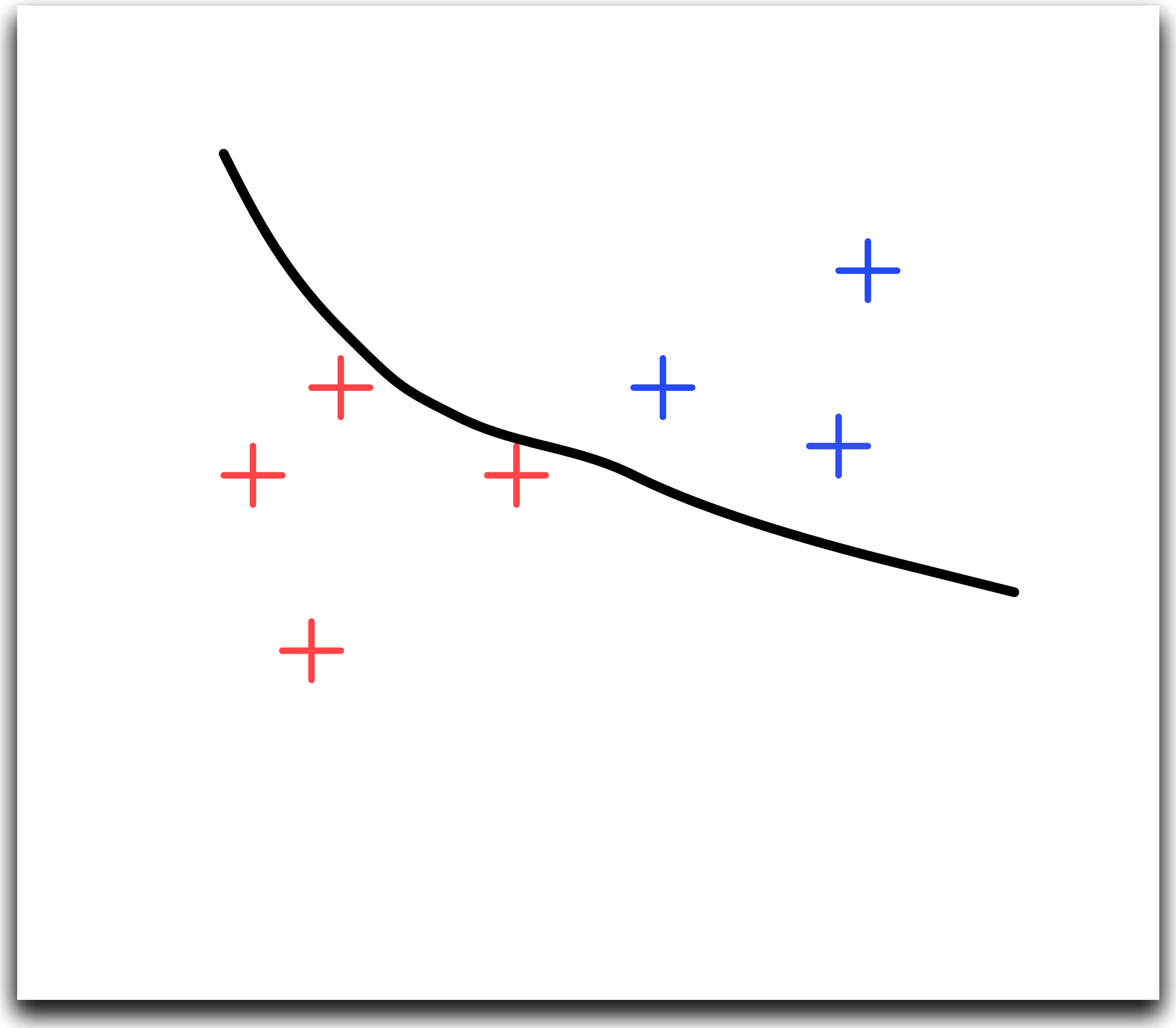
Supervised Learning

- Classification
  - For now: each data instance is a point in a 2D coordinate system
  - Color denotes target label
  - Model is given as decision boundary
- What's the correct model?
  - In theory: no way to tell
  - Smoothness assumption: similar instances have similar labels
  - All learning systems have underlying assumptions



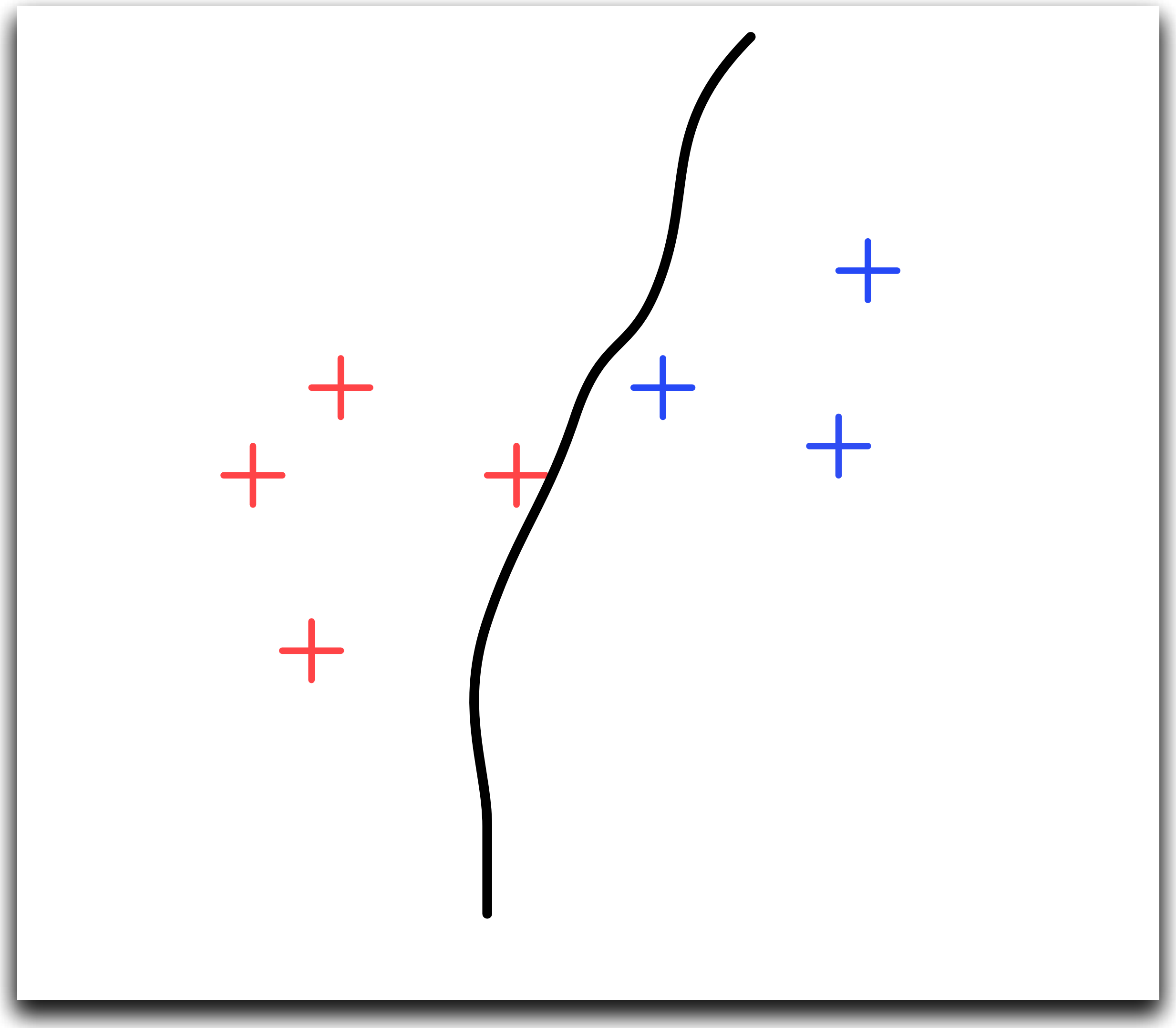
Supervised Learning

- Classification
  - For now: each data instance is a point in a 2D coordinate system
  - Color denotes target label
  - Model is given as decision boundary
- What's the correct model?
  - In theory: no way to tell
  - Smoothness assumption: similar instances have similar labels
  - All learning systems have underlying assumptions



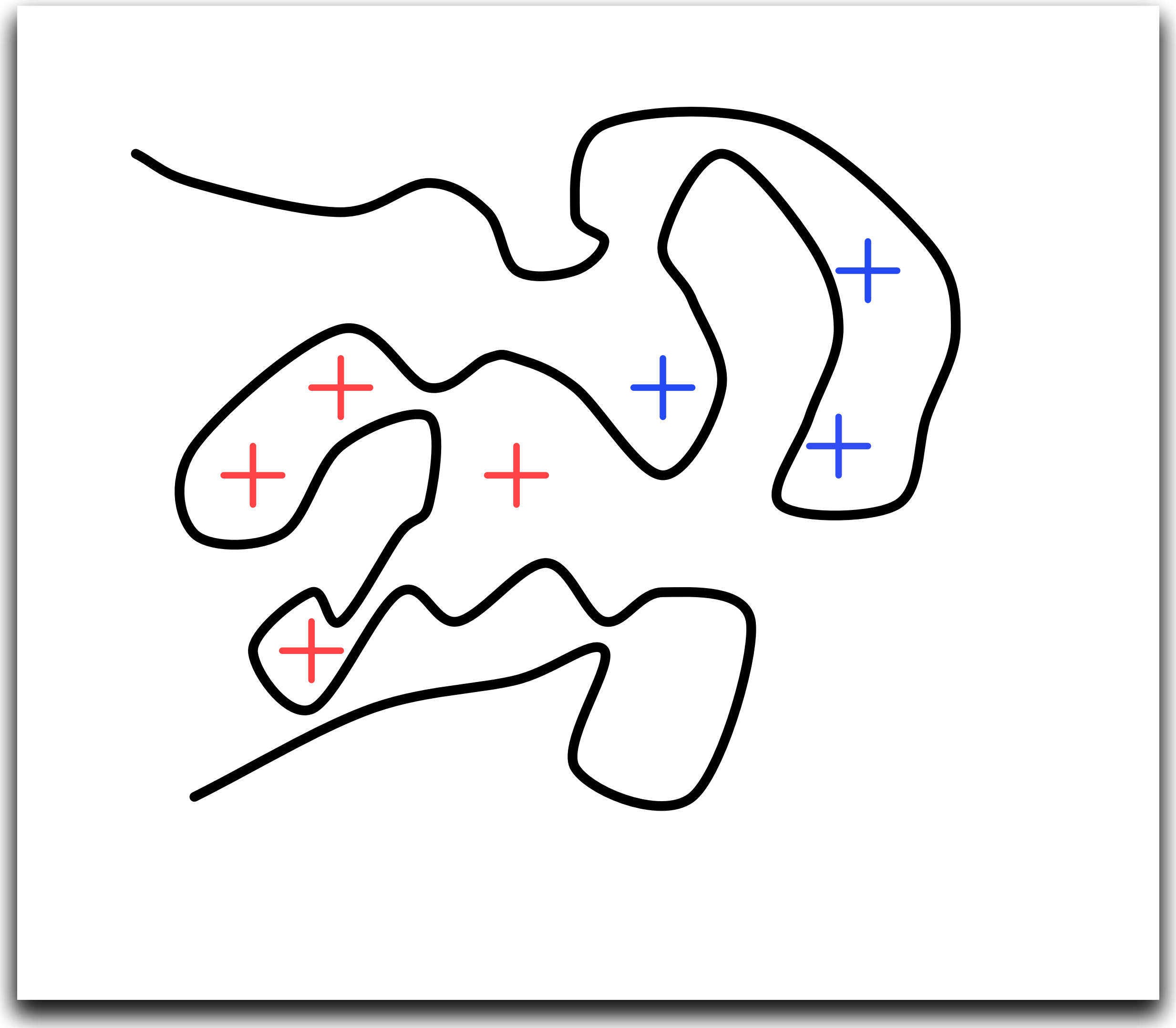
Supervised Learning

- Classification
  - For now: each data instance is a point in a 2D coordinate system
  - Color denotes target label
  - Model is given as decision boundary
- What's the correct model?
  - In theory: no way to tell
  - Smoothness assumption: similar instances have similar labels
  - All learning systems have underlying assumptions



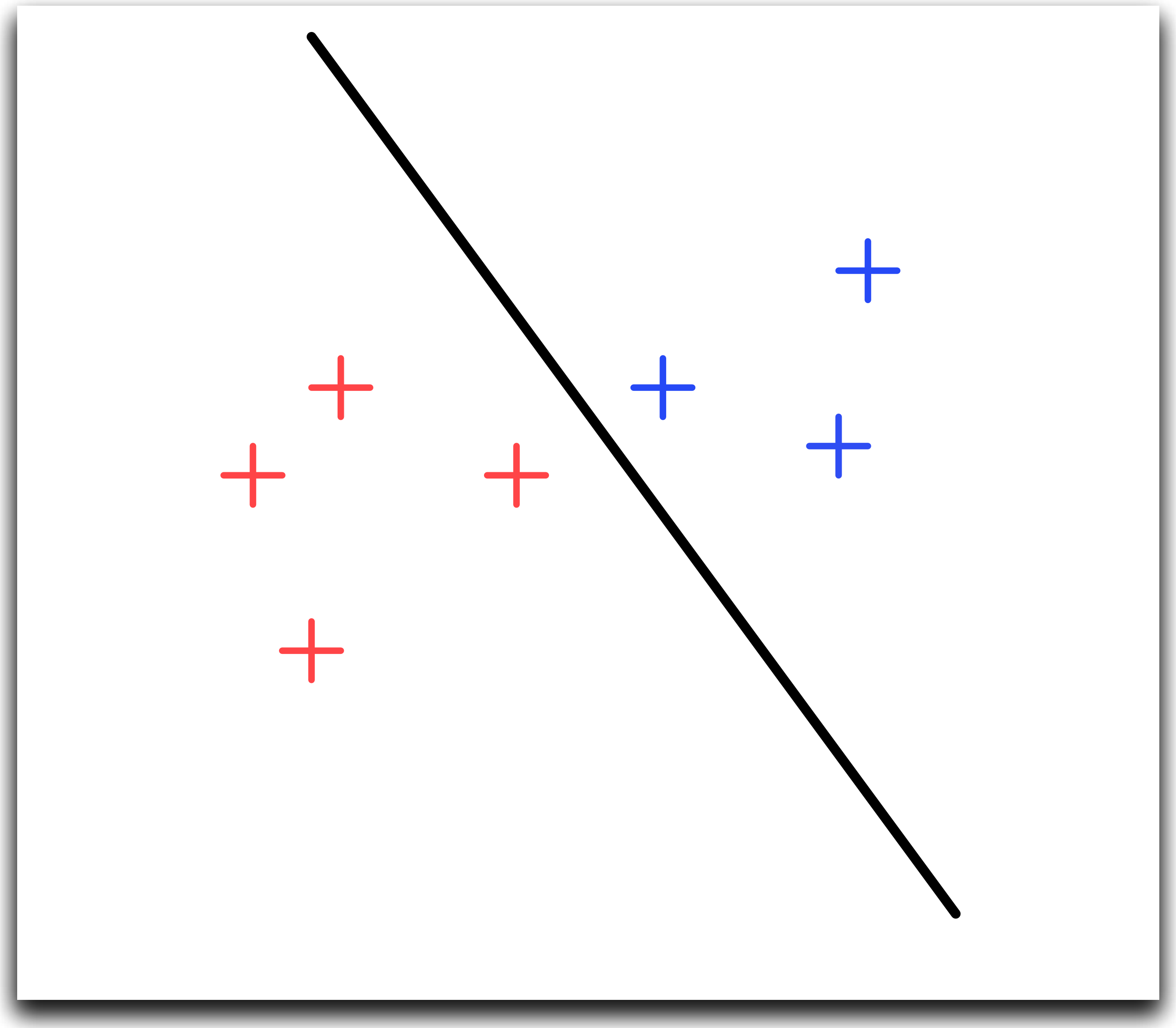
Supervised Learning

- Classification
  - For now: each data instance is a point in a 2D coordinate system
  - Color denotes target label
  - Model is given as decision boundary
- What's the correct model?
  - In theory: no way to tell
  - Smoothness assumption: similar instances have similar labels
  - All learning systems have underlying assumptions



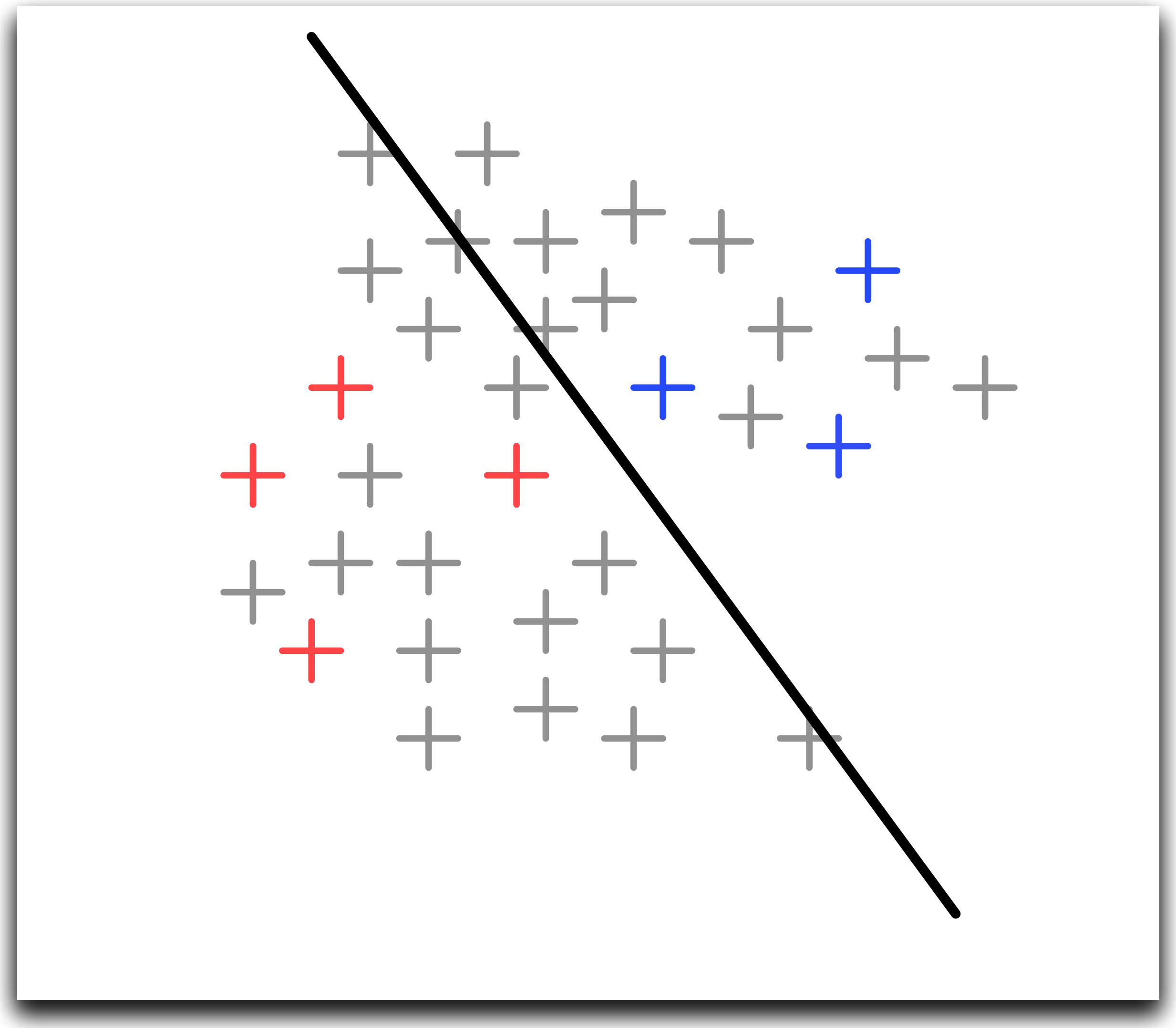
Supervised Learning

- Classification
  - For now: each data instance is a point in a 2D coordinate system
  - Color denotes target label
  - Model is given as decision boundary
- What's the correct model?
  - In theory: no way to tell
  - Smoothness assumption: similar instances have similar labels
  - All learning systems have underlying assumptions



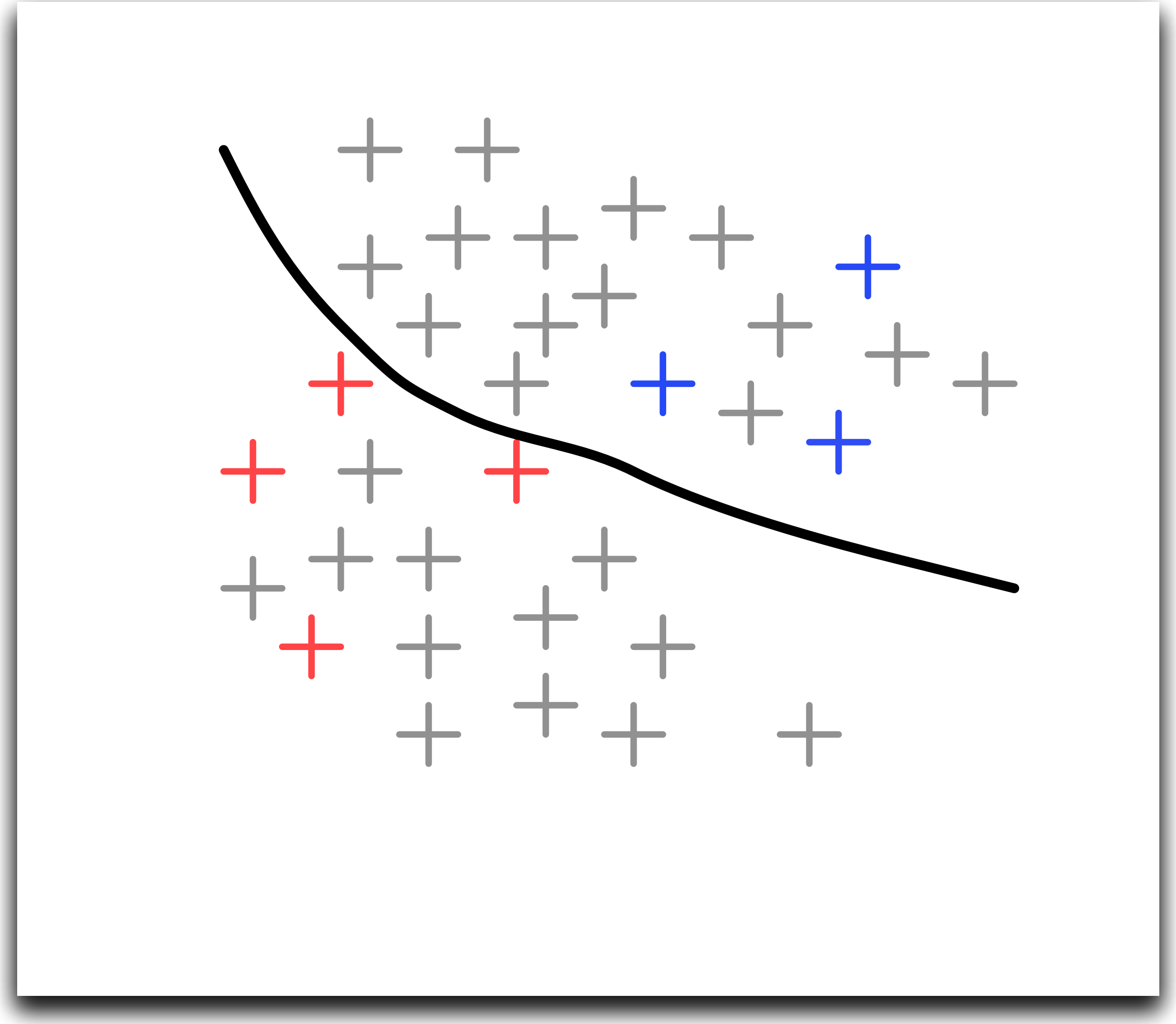
Semi-Supervised Learning

- Can we make use of the unlabeled data?
  - In theory: no
  - ... but we can make assumptions
- Popular Assumptions
  - Clustering assumption
  - Low density assumption
  - Manifold assumption



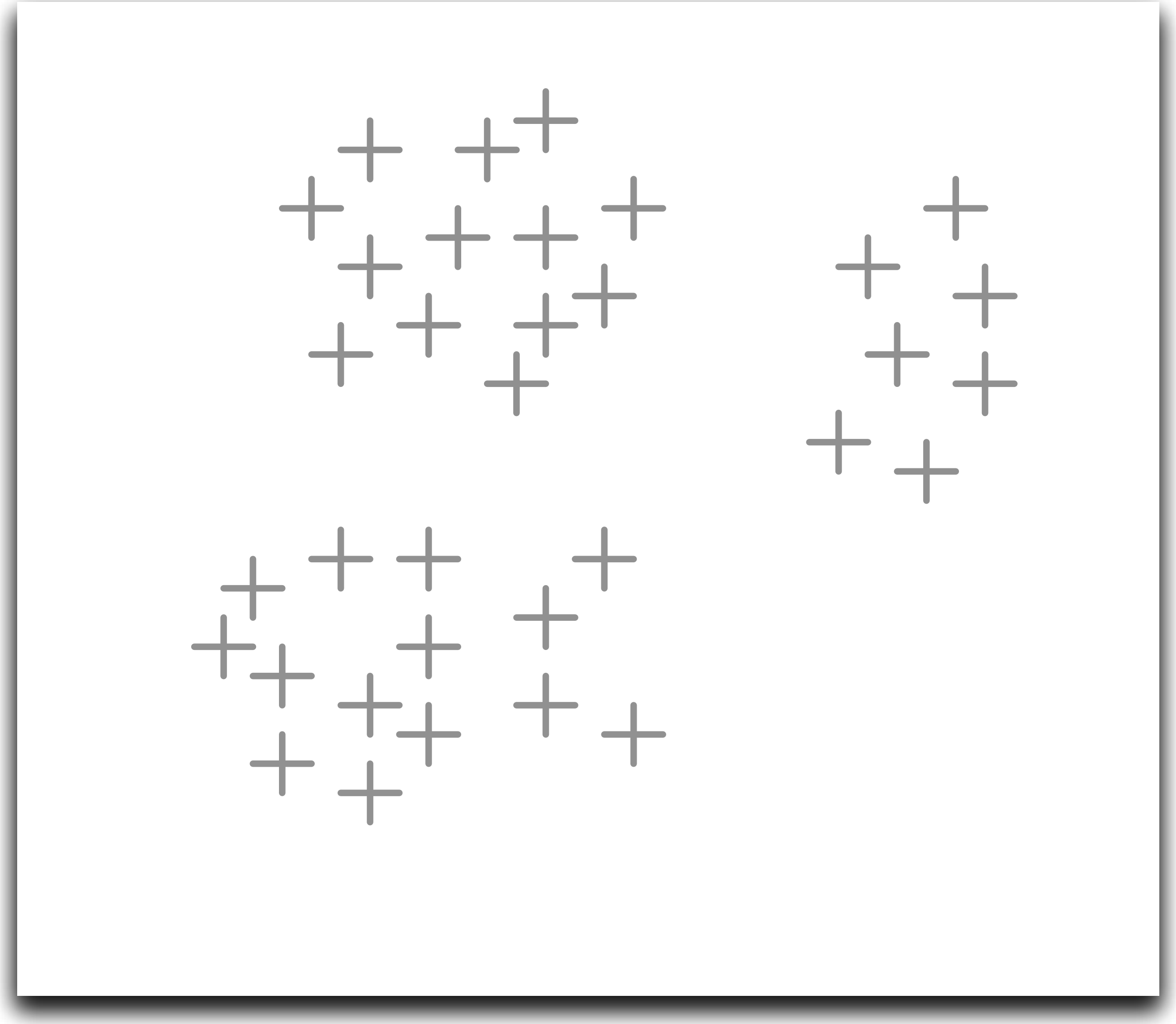
Semi-Supervised Learning

- Can we make use of the unlabeled data?
  - In theory: no
  - ... but we can make assumptions
- Popular Assumptions
  - Clustering assumption
  - Low density assumption
  - Manifold assumption



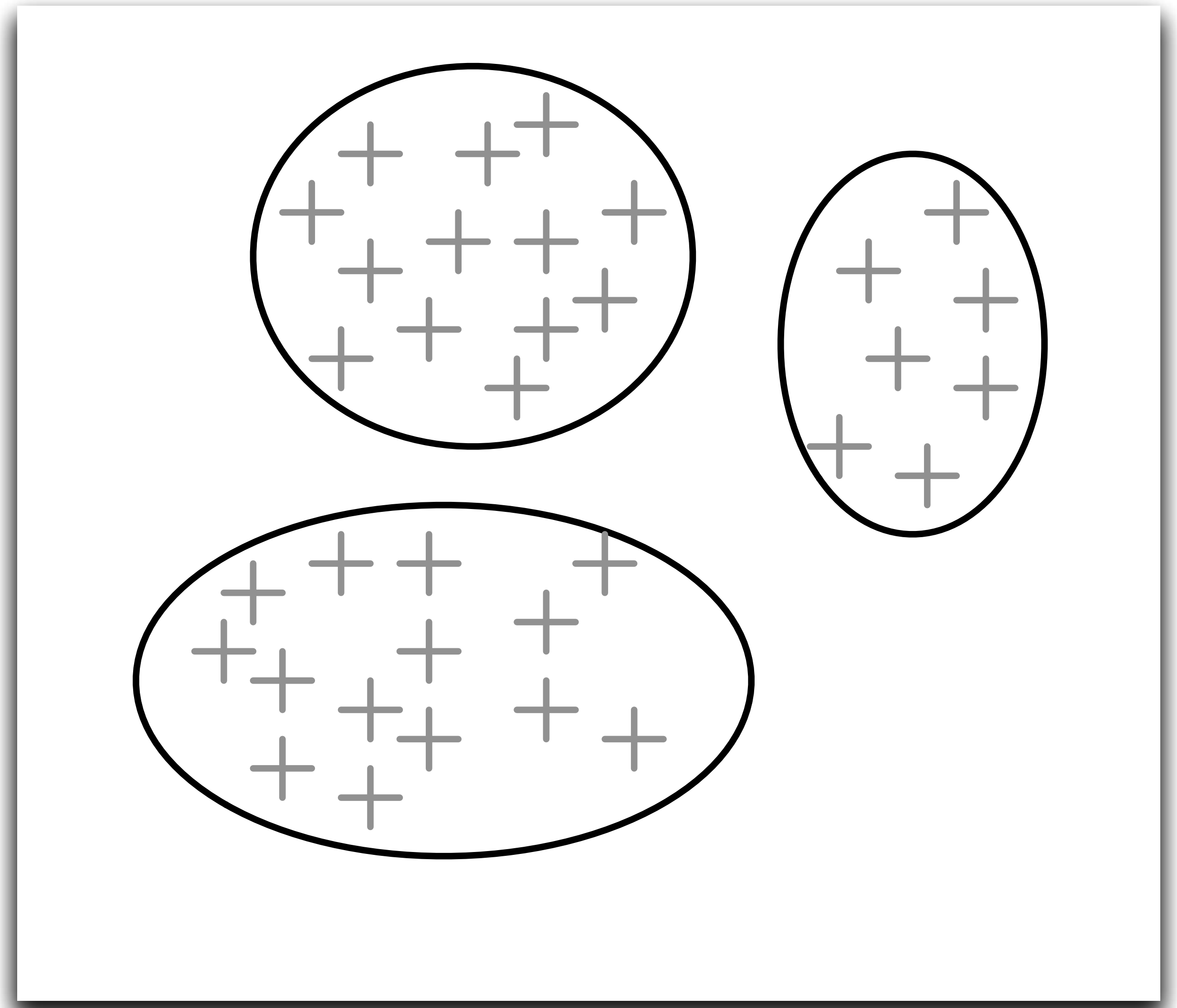
The Clustering Assumption

- Clustering
  - Partition instances into groups (clusters) of similar instances
  - Many different algorithms: k-Means, EM, DBSCAN, etc.
  - Available e.g. on Mahout
- Clustering Assumption
  - The two classification targets are distinct clusters
  - Simple semi-supervised learning: cluster, then perform majority vote



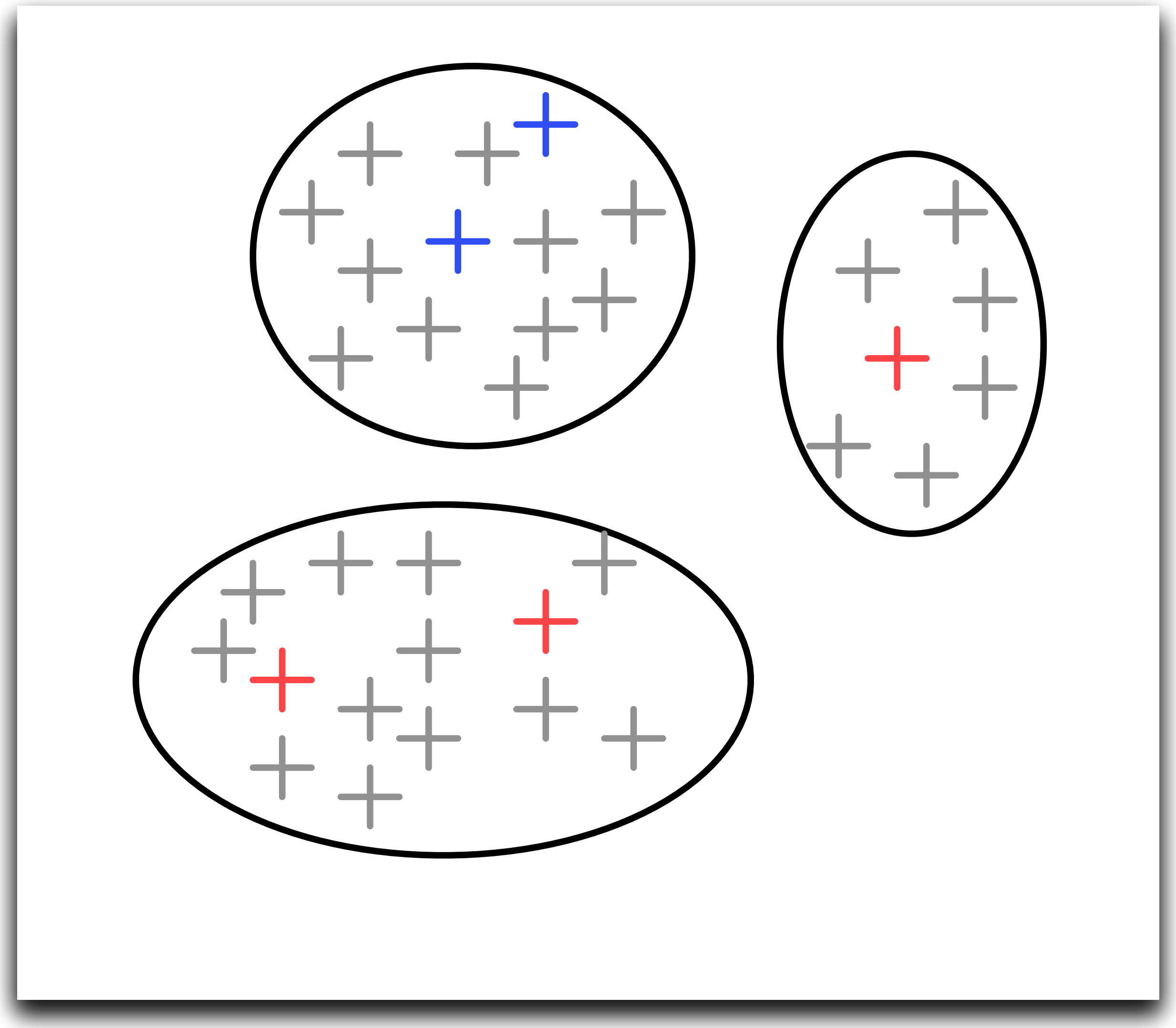
The Clustering Assumption

- Clustering
    - Partition instances into groups (clusters) of similar instances
    - Many different algorithms: k-Means, EM, DBSCAN, etc.
    - Available e.g. on Mahout
- Clustering Assumption
    - The two classification targets are distinct clusters
    - Simple semi-supervised learning: cluster, then perform majority vote



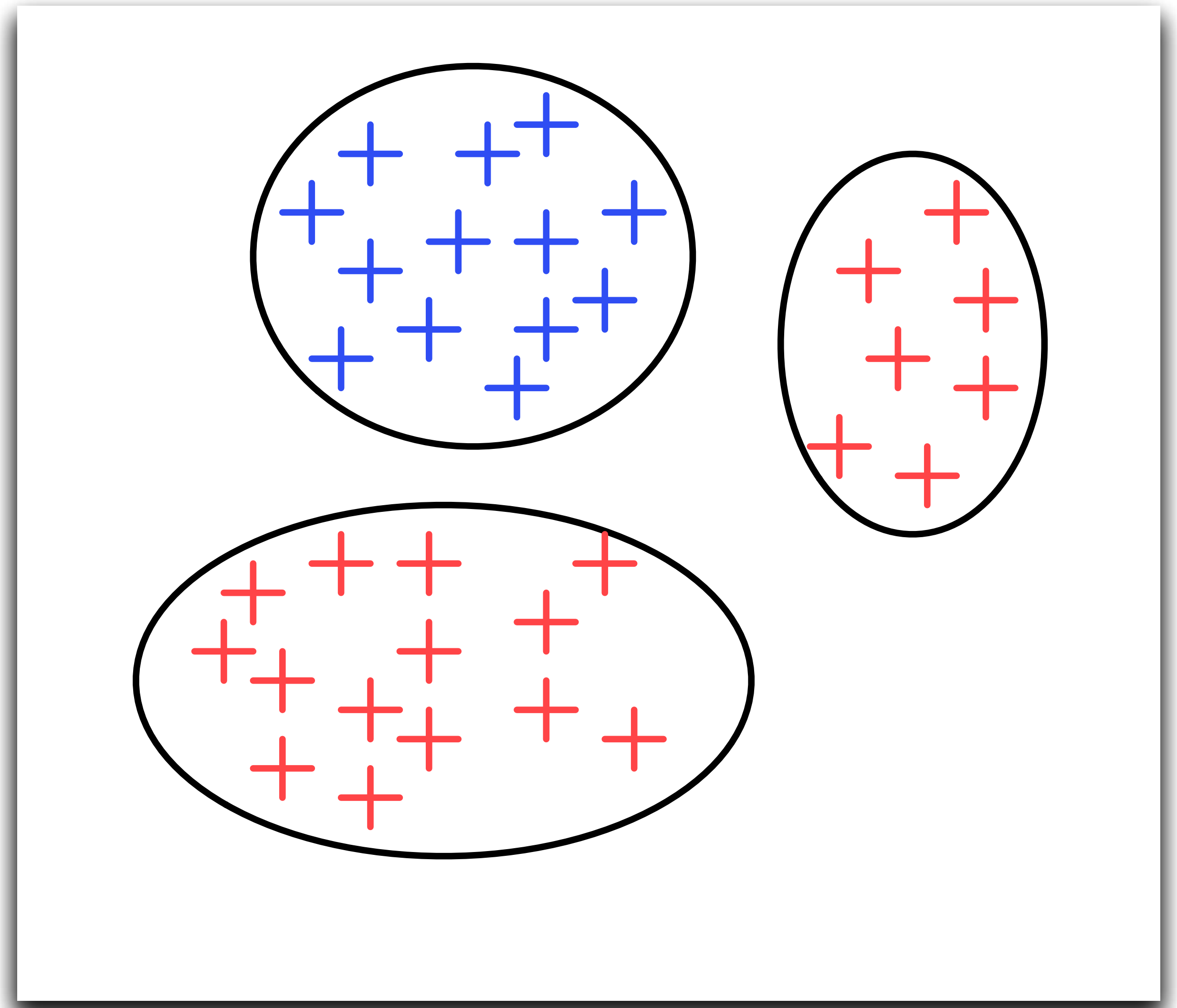
The Clustering Assumption

- Clustering
  - Partition instances into groups (clusters) of similar instances
  - Many different algorithms: k-Means, EM, DBSCAN, etc.
  - Available e.g. on Mahout
- Clustering Assumption
  - The two classification targets are distinct clusters
  - Simple semi-supervised learning: cluster, then perform majority vote



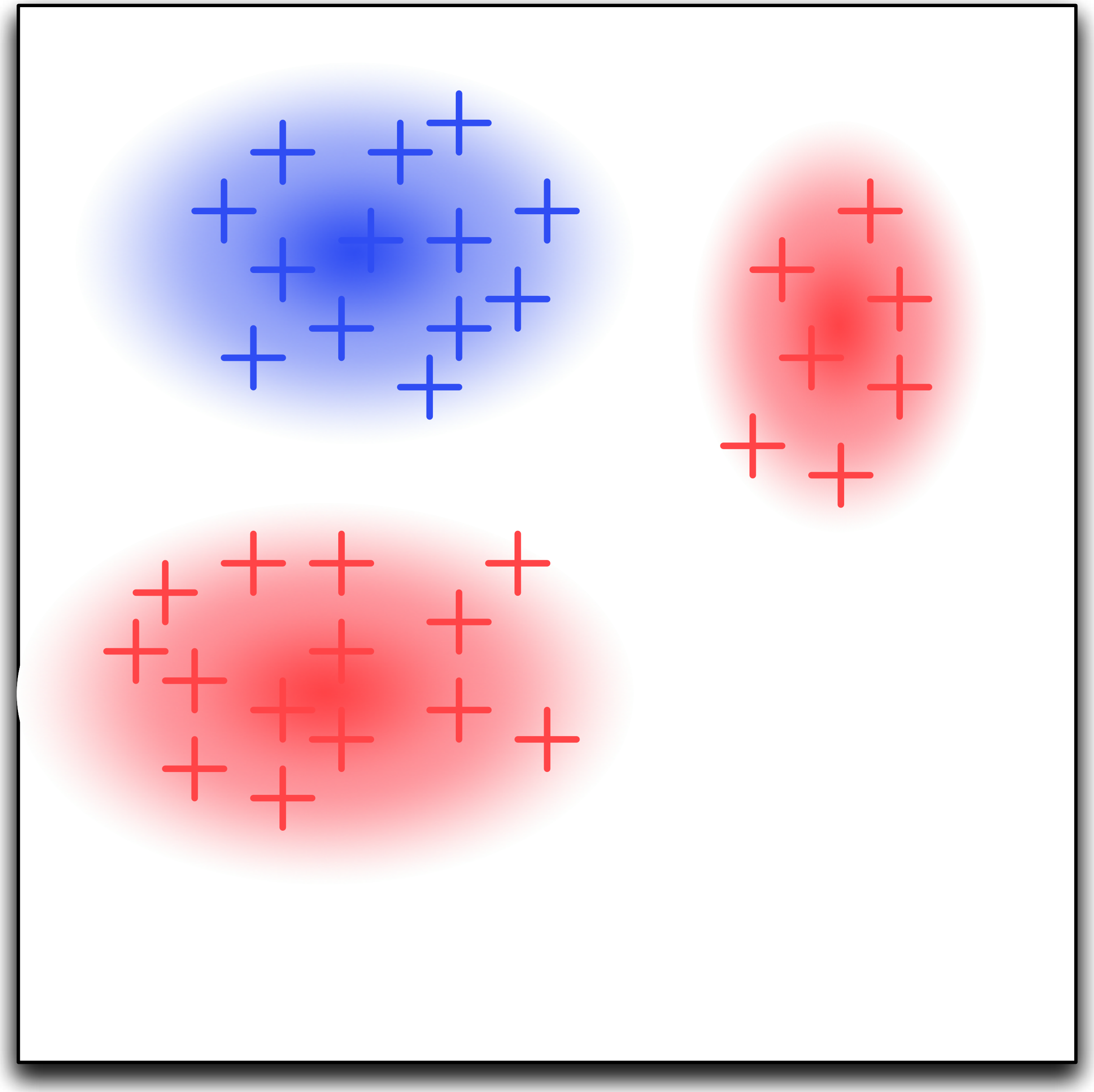
The Clustering Assumption

- Clustering
  - Partition instances into groups (clusters) of similar instances
  - Many different algorithms: k-Means, EM, DBSCAN, etc.
  - Available e.g. on Mahout
- Clustering Assumption
  - The two classification targets are distinct clusters
  - Simple semi-supervised learning: cluster, then perform majority vote



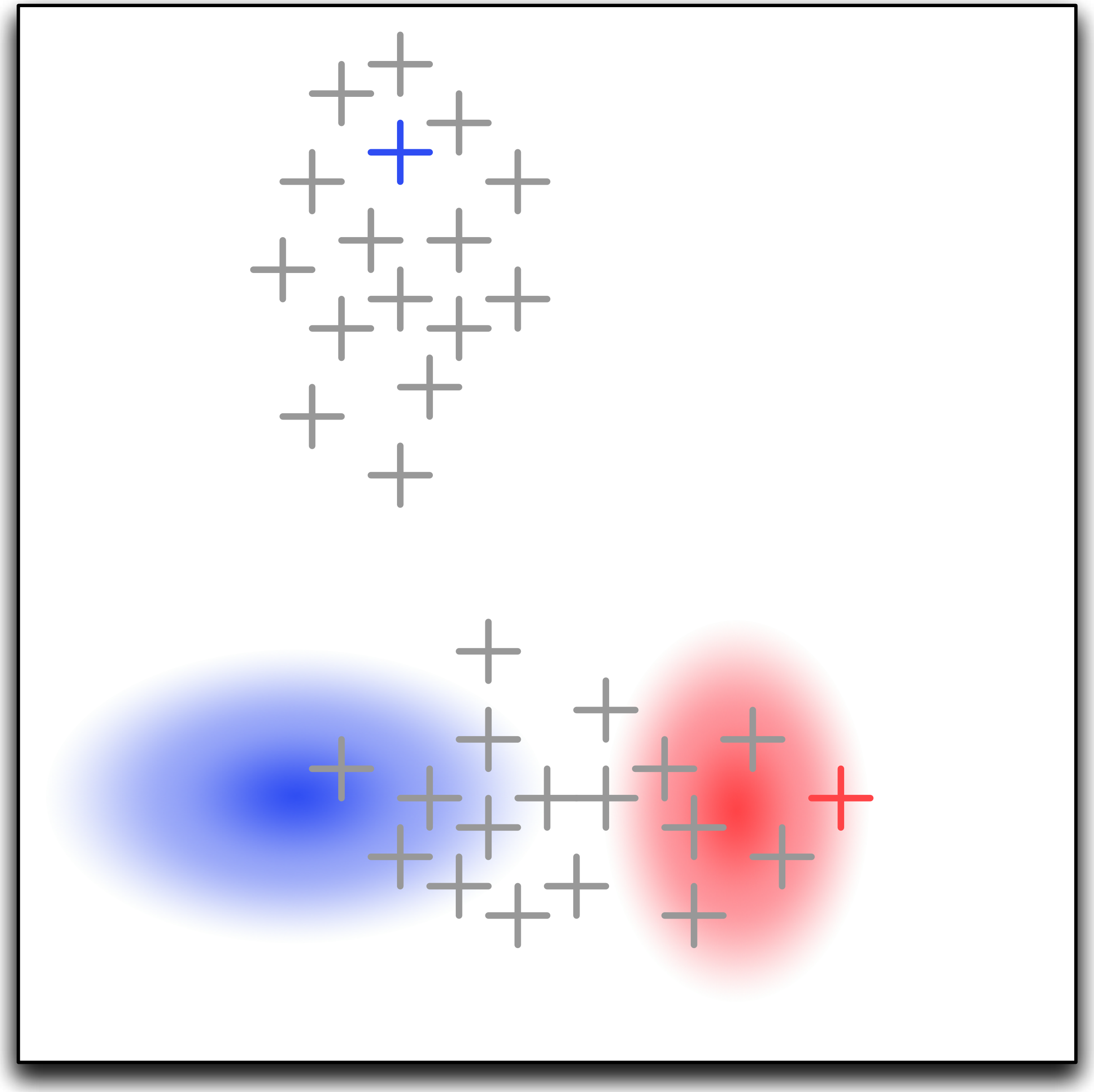
Generative Models

- Mixture of Gaussians
  - Assumption: the data in each cluster is generated by a normal distribution
  - Find most probable location and shape of clusters given data
- Expectation-Maximization
  - Two step optimization procedure
  - Keeps estimates of cluster assignment probabilities for each instance
  - Each step is one MapReduce job
  - Might converge to local optimum



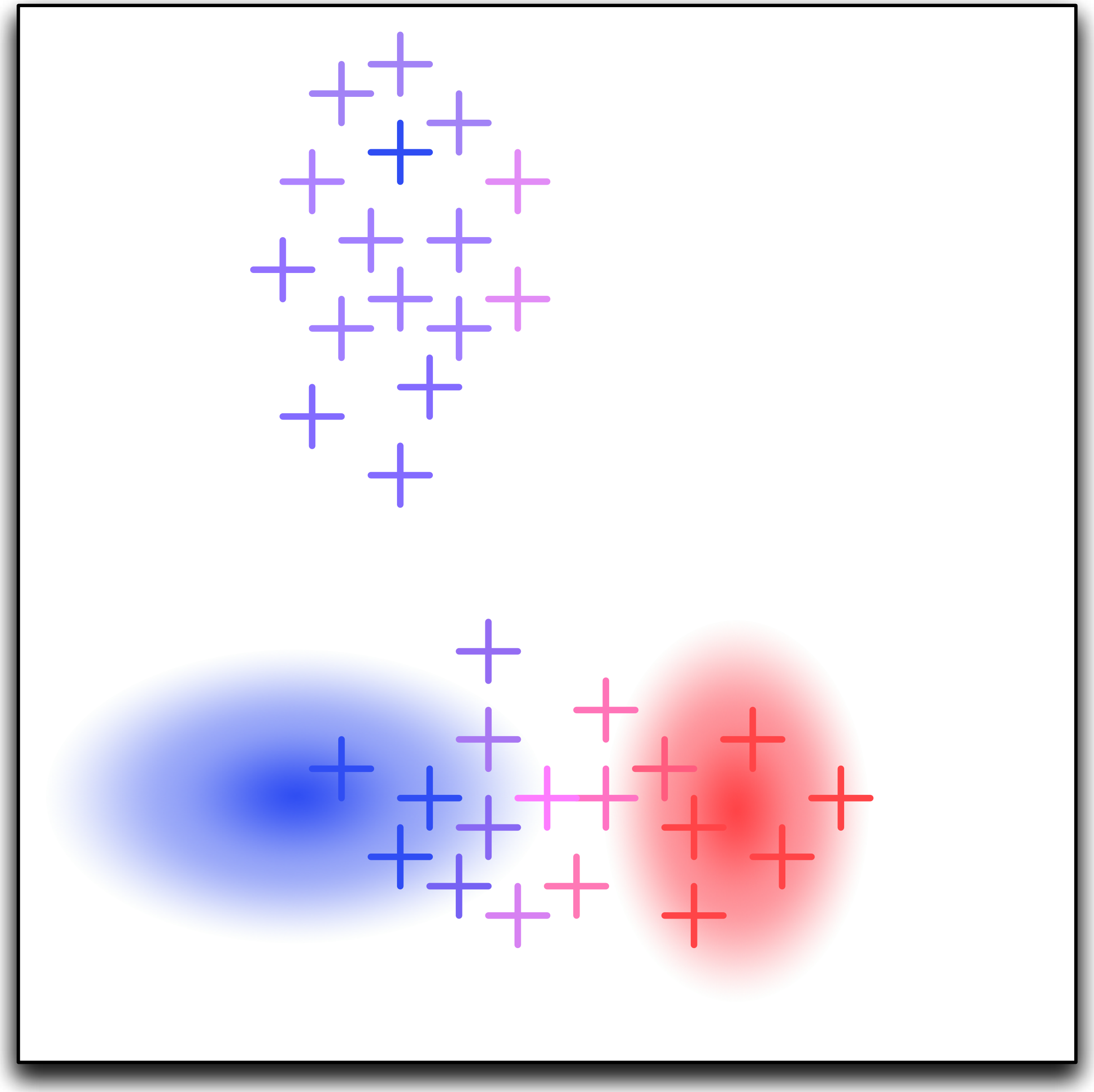
Generative Models

- Mixture of Gaussians
  - Assumption: the data in each cluster is generated by a normal distribution
  - Find most probable location and shape of clusters given data
- Expectation-Maximization
  - Two step optimization procedure
  - Keeps estimates of cluster assignment probabilities for each instance
  - Each step is one MapReduce job
  - Might converge to local optimum



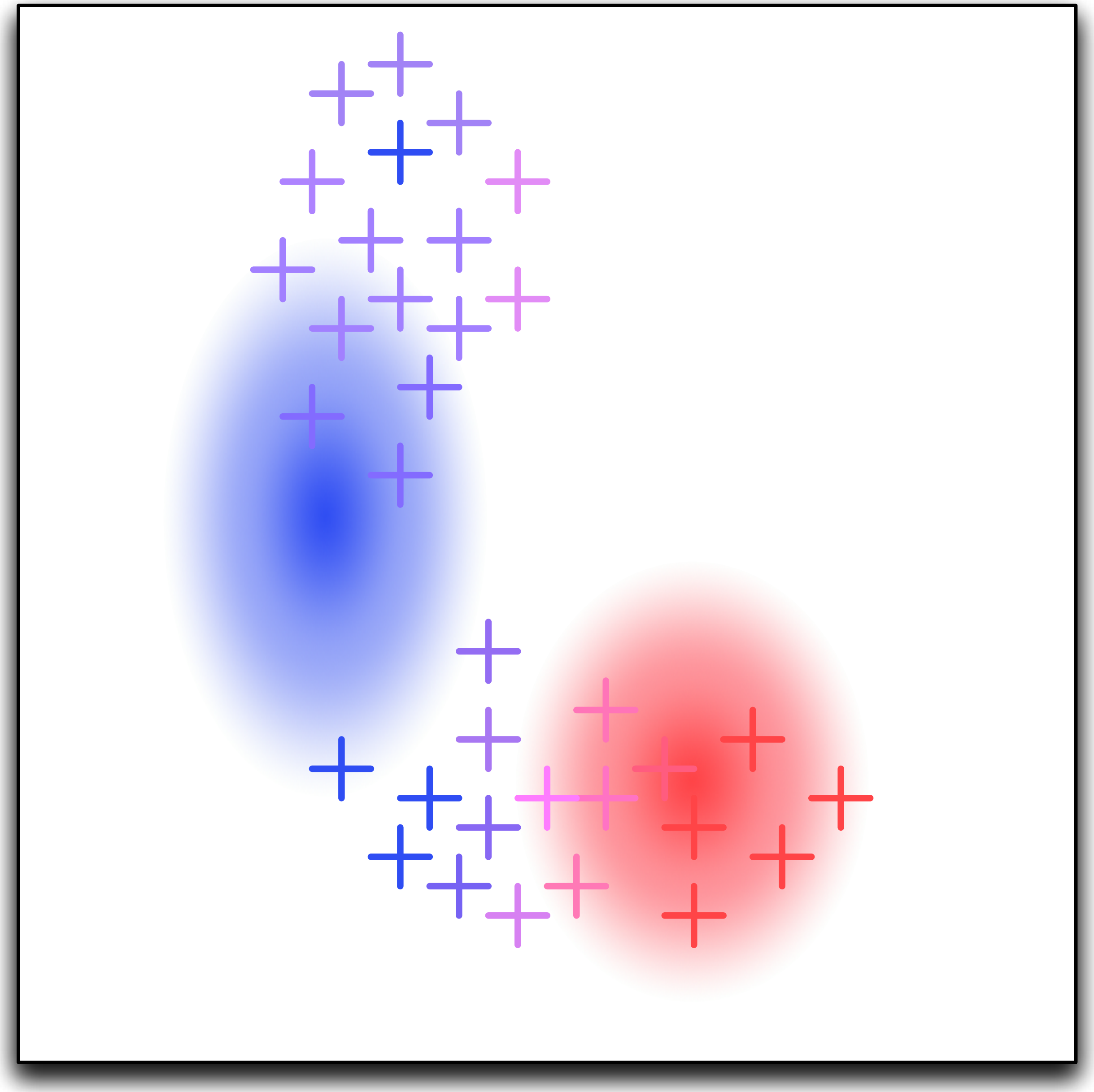
Generative Models

- Mixture of Gaussians
  - Assumption: the data in each cluster is generated by a normal distribution
  - Find most probable location and shape of clusters given data
- Expectation-Maximization
  - Two step optimization procedure
  - Keeps estimates of cluster assignment probabilities for each instance
  - Each step is one MapReduce job
  - Might converge to local optimum



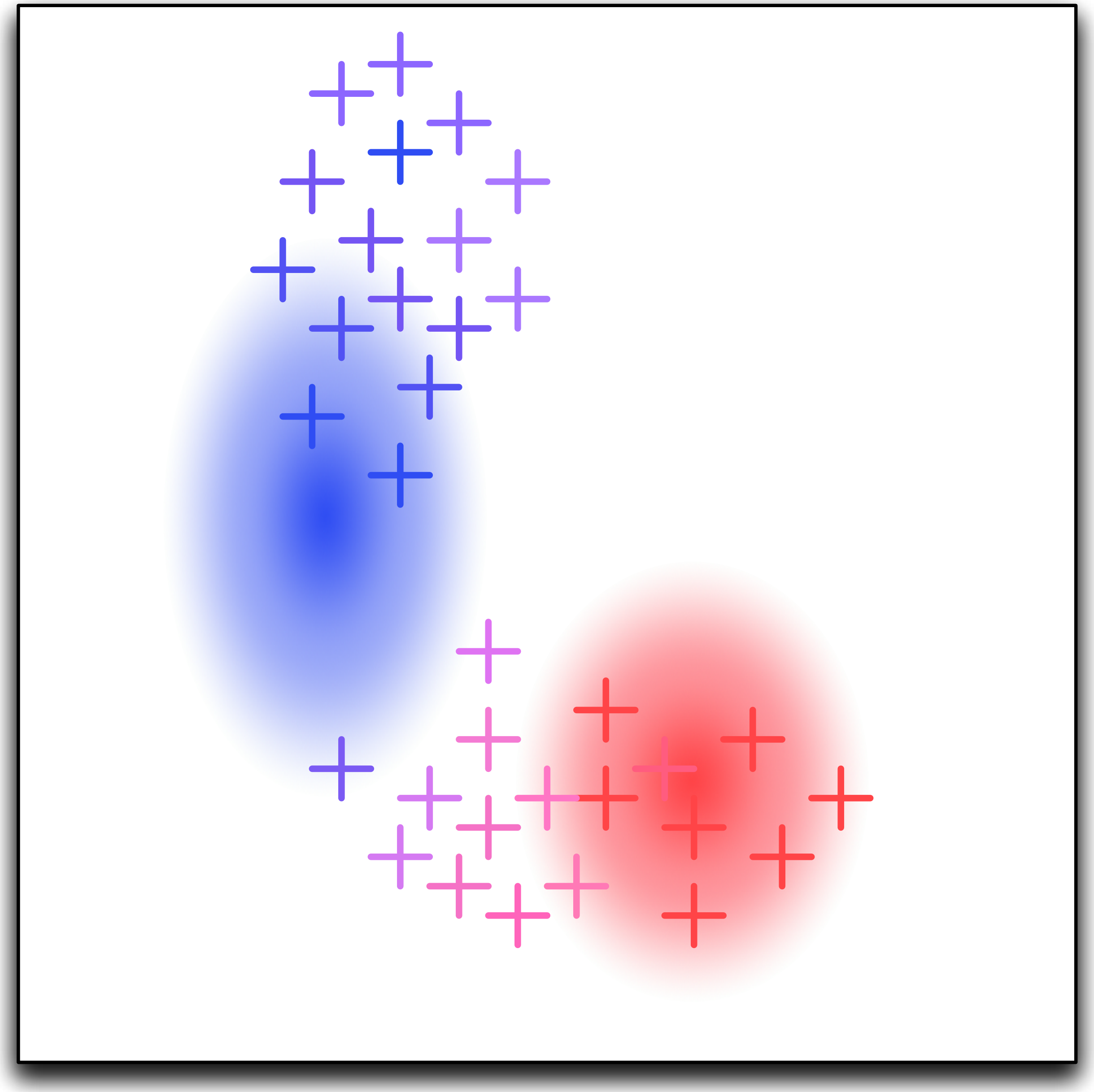
Generative Models

- Mixture of Gaussians
  - Assumption: the data in each cluster is generated by a normal distribution
  - Find most probable location and shape of clusters given data
- Expectation-Maximization
  - Two step optimization procedure
  - Keeps estimates of cluster assignment probabilities for each instance
  - Each step is one MapReduce job
  - Might converge to local optimum



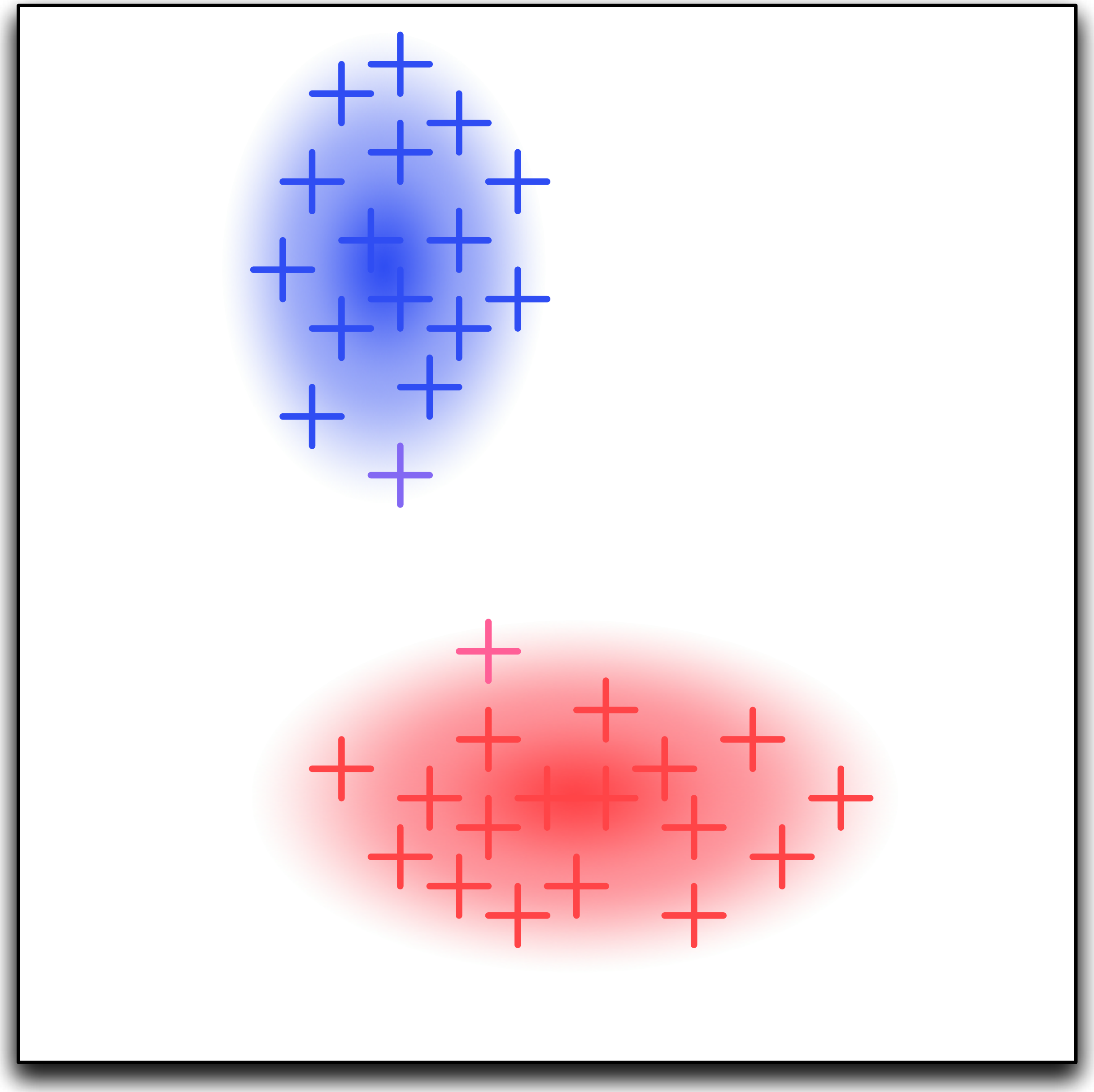
Generative Models

- Mixture of Gaussians
  - Assumption: the data in each cluster is generated by a normal distribution
  - Find most probable location and shape of clusters given data
- Expectation-Maximization
  - Two step optimization procedure
  - Keeps estimates of cluster assignment probabilities for each instance
  - Each step is one MapReduce job
  - Might converge to local optimum



Generative Models

- Mixture of Gaussians
  - Assumption: the data in each cluster is generated by a normal distribution
  - Find most probable location and shape of clusters given data
- Expectation-Maximization
  - Two step optimization procedure
  - Keeps estimates of cluster assignment probabilities for each instance
  - Each step is one MapReduce job
  - Might converge to local optimum

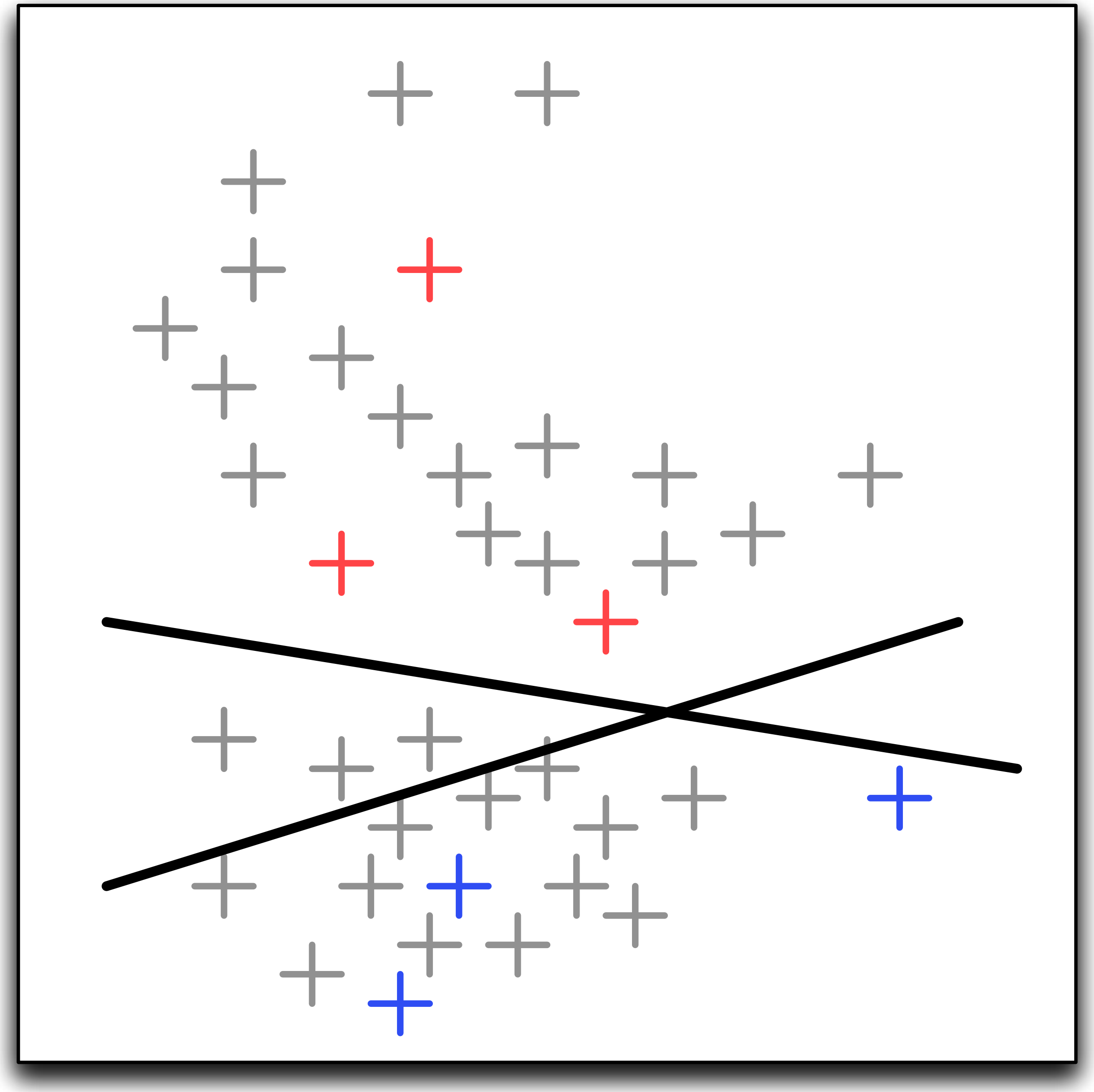


Beyond Mixtures of Gaussians

- Expectation-Maximization
  - Can be adjusted to all kinds of mixture models
  - E.g. use Naive Bayes as mixture model for text classification
- Self-Training
  - Learn model on labeled instances only
  - Apply model to unlabeled instances
  - Learn new model on all instances
  - Repeat until convergence

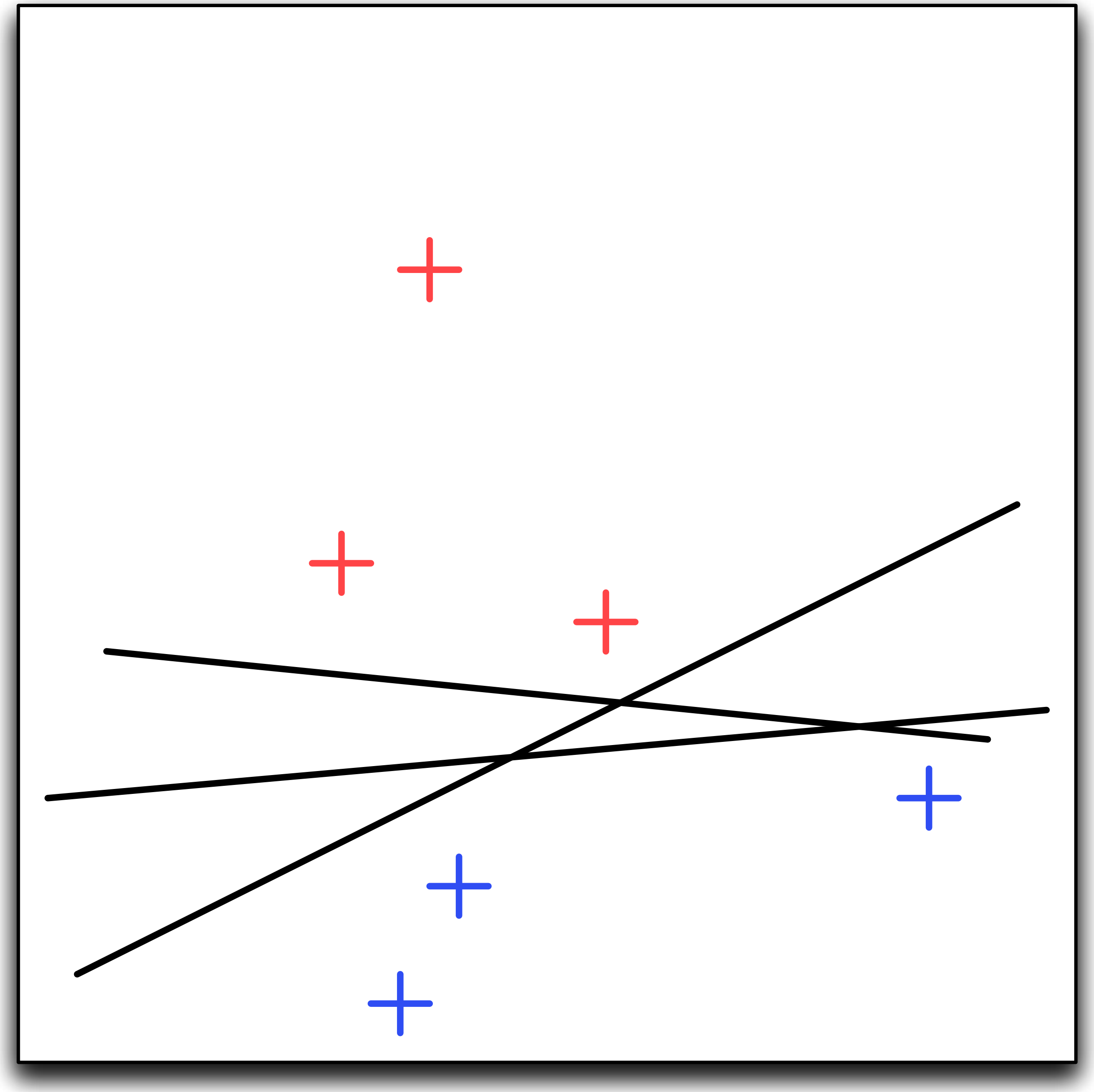
The Low Density Assumption

- Assumption
  - The area between the two classes has low density
  - Does not assume any specific form of cluster
- Support Vector Machine
  - Decision boundary is linear
  - Maximizes margin to closest instances
  - Can be learned in one Map-Reduce step (Stochastic Gradient Descent)



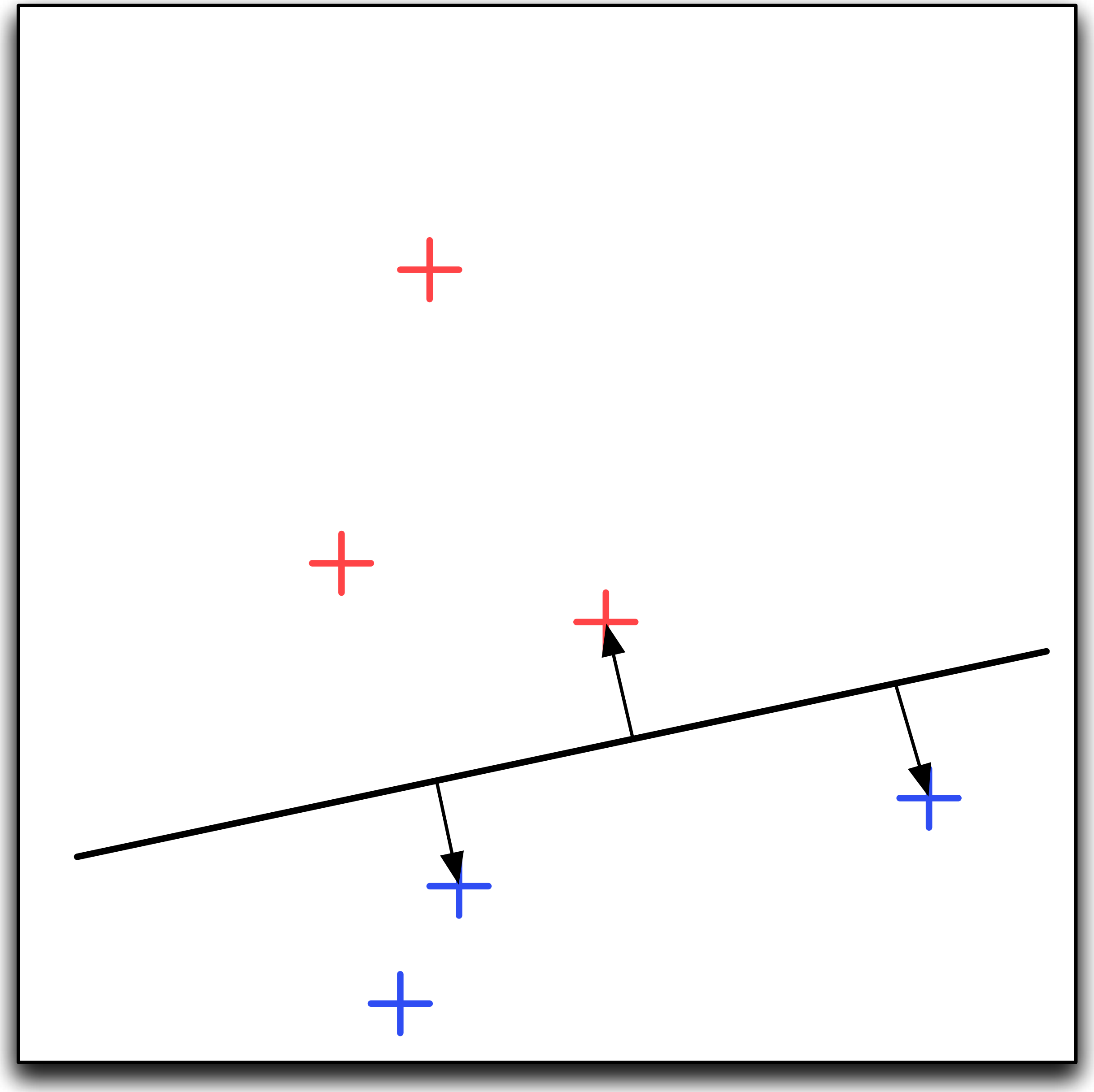
The Low Density Assumption

- Assumption
  - The area between the two classes has low density
  - Does not assume any specific form of cluster
- Support Vector Machine
  - Decision boundary is linear
  - Maximizes margin to closest instances
  - Can be learned in one Map-Reduce step (Stochastic Gradient Descent)



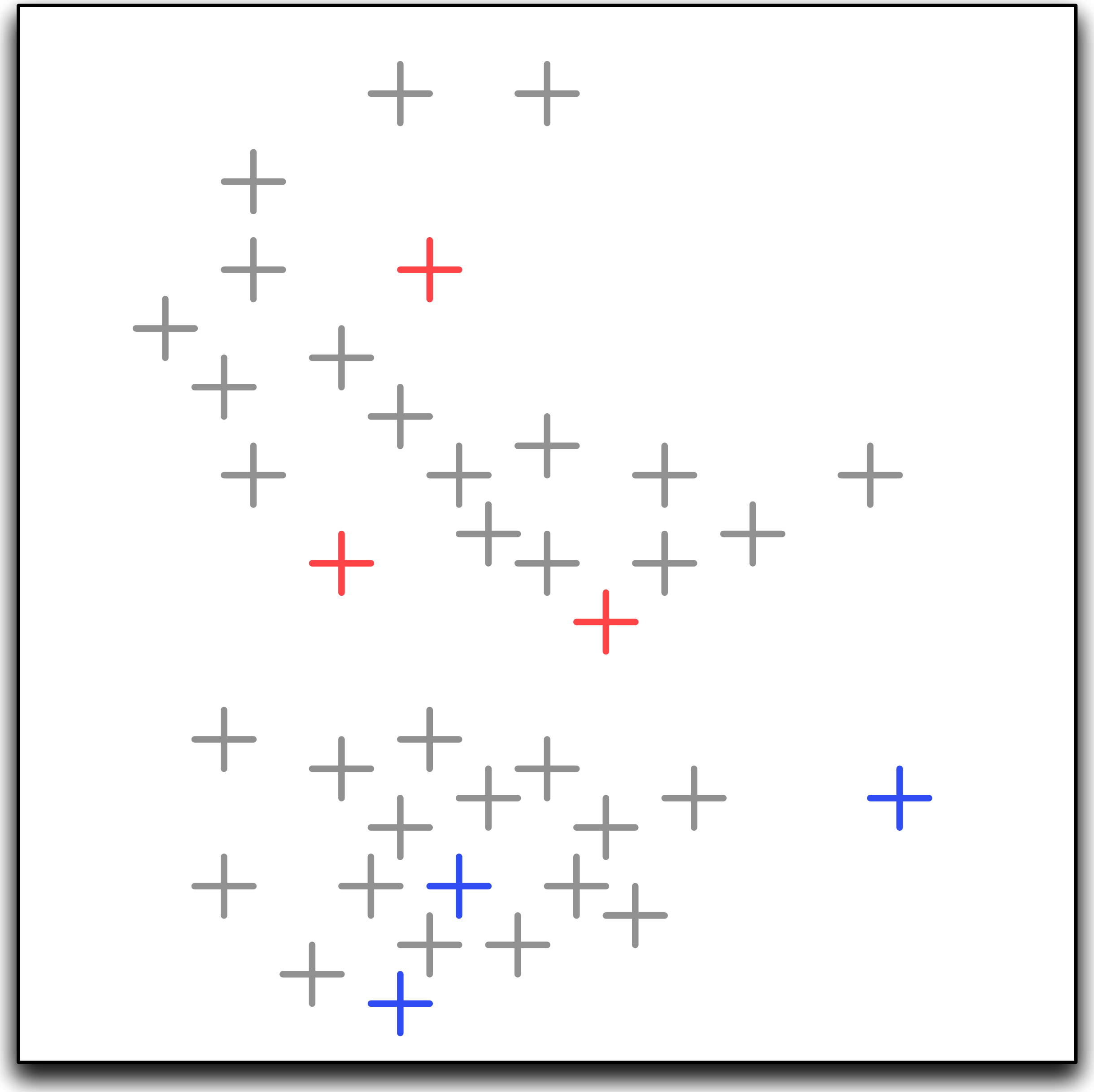
The Low Density Assumption

- Assumption
  - The area between the two classes has low density
  - Does not assume any specific form of cluster
- Support Vector Machine
  - Decision boundary is linear
  - Maximizes margin to closest instances
  - Can be learned in one Map-Reduce step (Stochastic Gradient Descent)



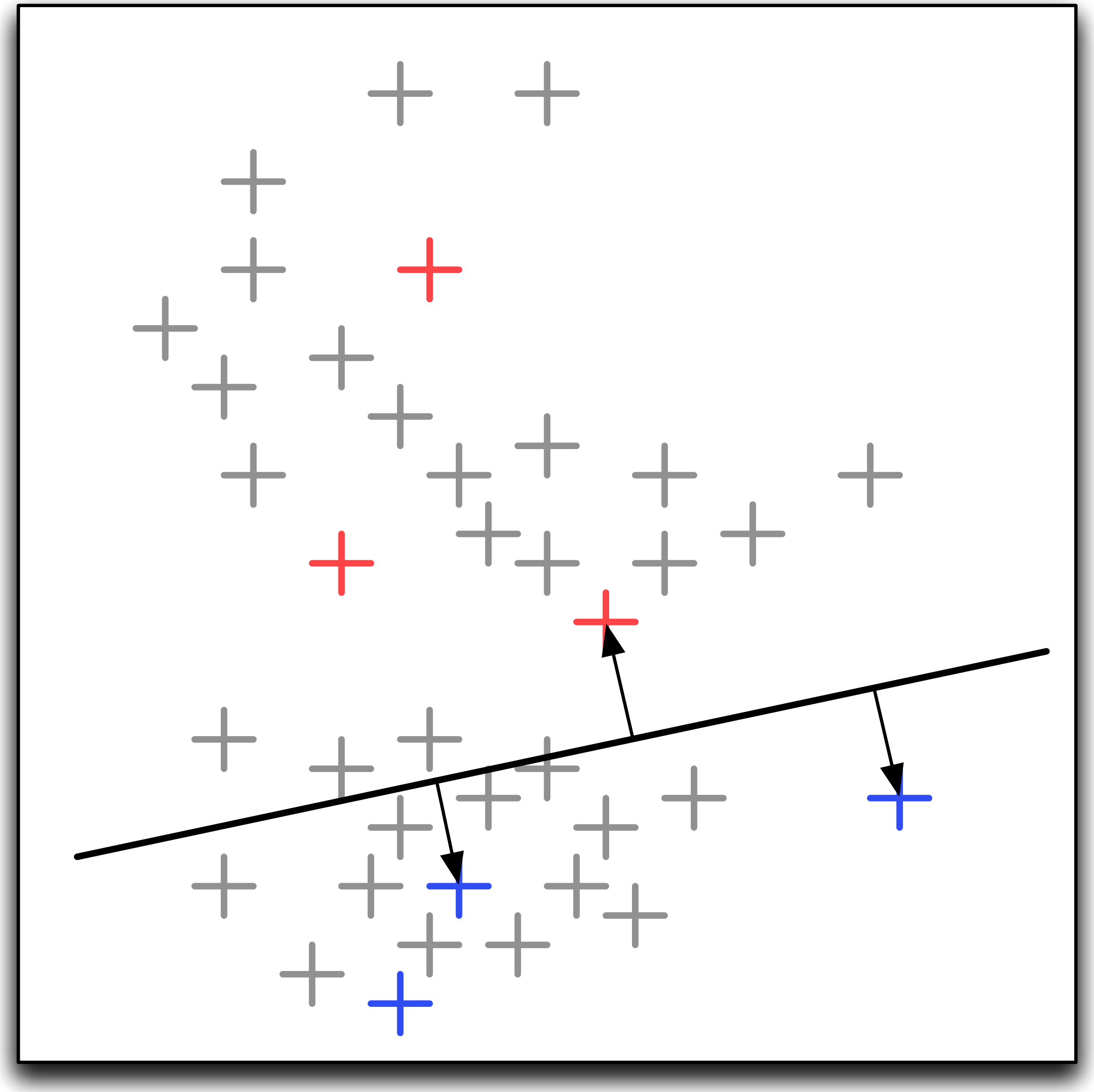
The Low Density Assumption

- Semi-Supervised Support Vector Machine
  - Minimize distance to labeled and unlabeled instances
  - Parameter to fine-tune influence of unlabeled instances
  - Additional constraint: keep class balance correct
- Implementation
  - Simple extension of SVM
  - But non-convex optimization problem



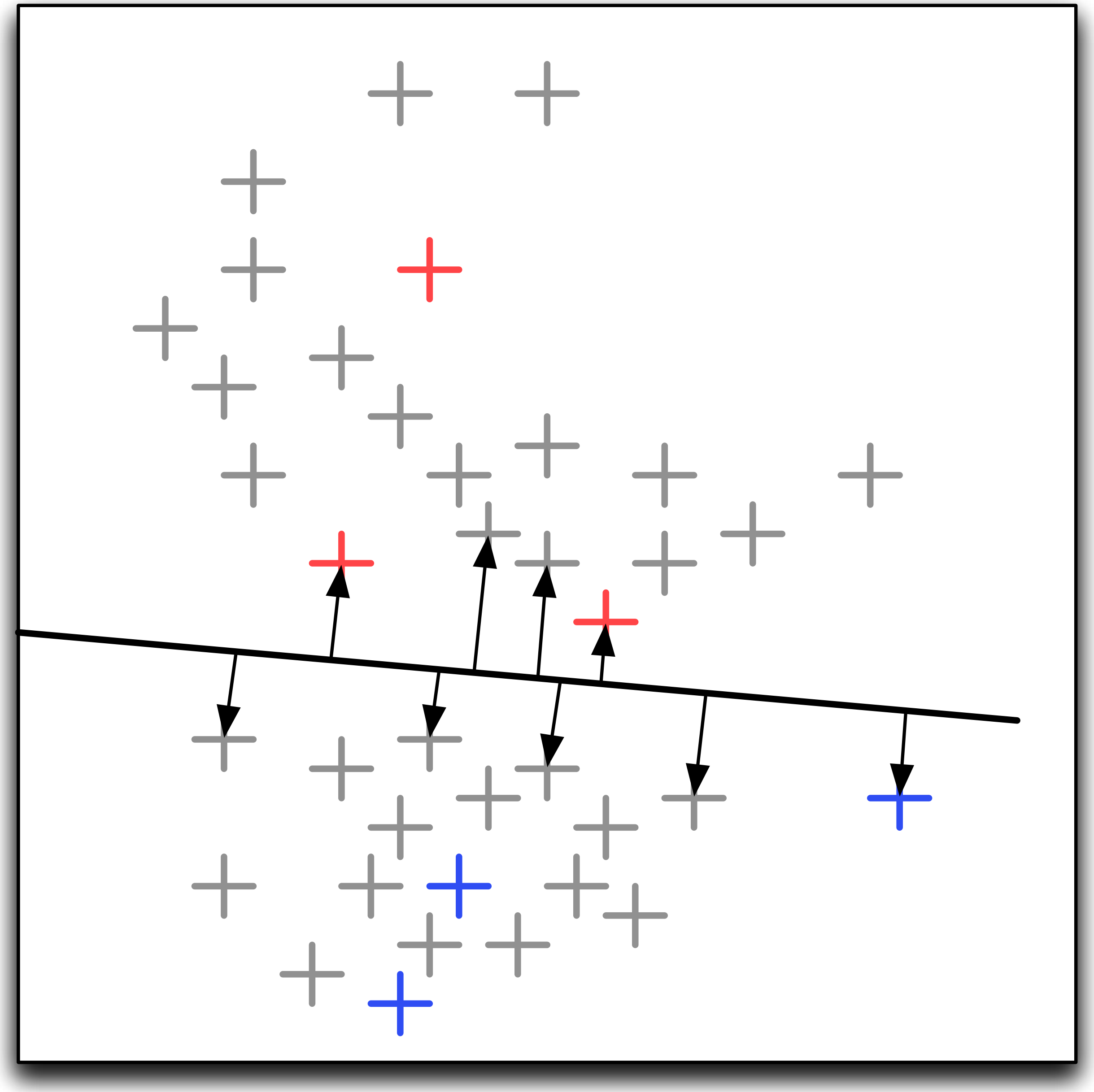
The Low Density Assumption

- Semi-Supervised Support Vector Machine
  - Minimize distance to labeled and unlabeled instances
  - Parameter to fine-tune influence of unlabeled instances
  - Additional constraint: keep class balance correct
- Implementation
  - Simple extension of SVM
  - But non-convex optimization problem



The Low Density Assumption

- Semi-Supervised Support Vector Machine
  - Minimize distance to labeled and unlabeled instances
  - Parameter to fine-tune influence of unlabeled instances
  - Additional constraint: keep class balance correct
- Implementation
  - Simple extension of SVM
  - But non-convex optimization problem



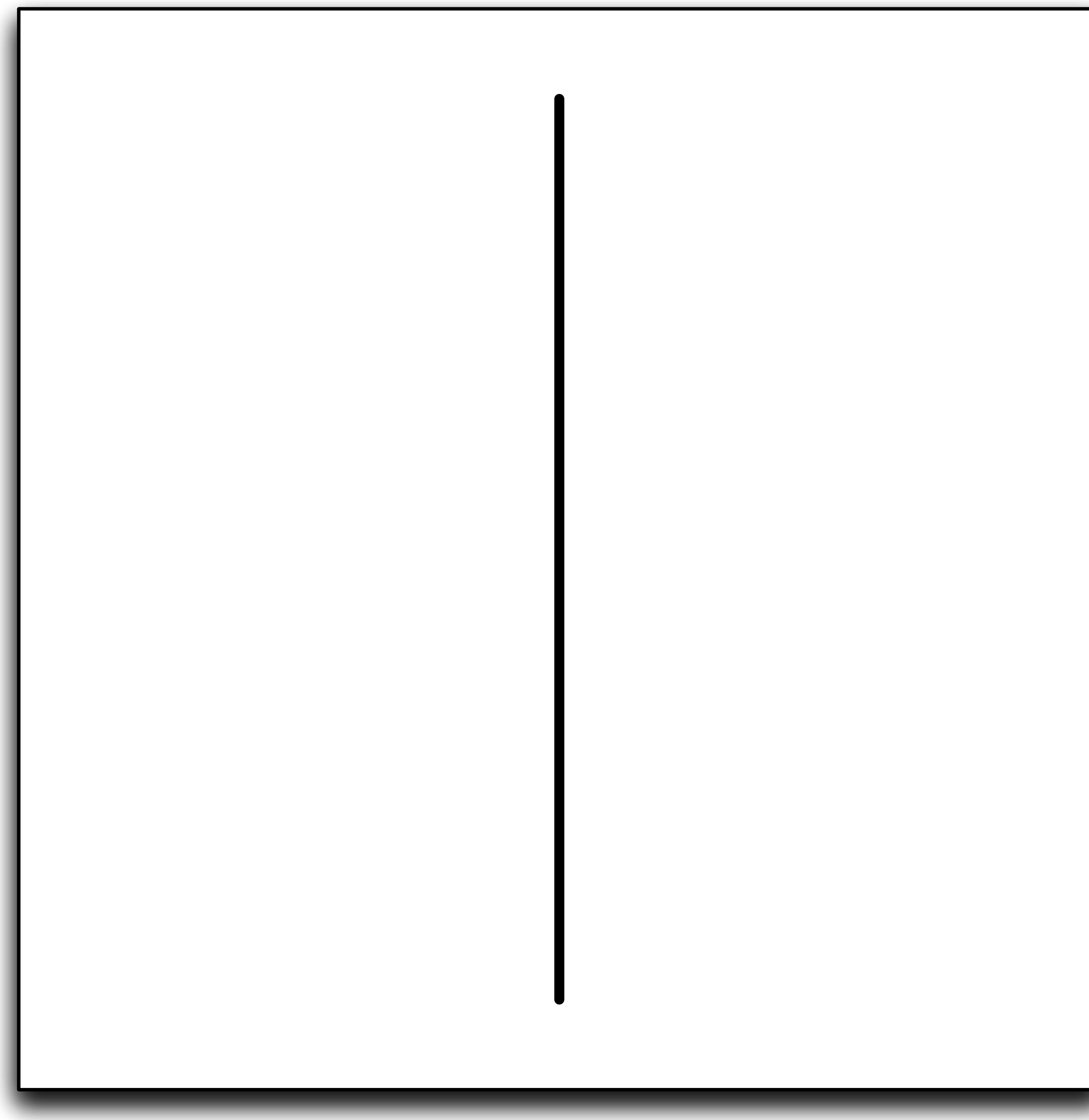
The Low Density Assumption

- Semi-Supervised Support Vector Machine
  - Minimize distance to labeled and unlabeled instances
  - Parameter to fine-tune influence of unlabeled instances
  - Additional constraint: keep class balance correct
- Implementation
  - Simple extension of SVM
  - But non-convex optimization problem

$$\min_w \overbrace{\boxed{\|w\|^2}}^{\text{Regularizer}} \overbrace{\boxed{+ C \sum_{i=1}^{n} \ell(y_i(w^T x_i + b))}}^{\text{Margins of labeled instances}} \underbrace{\boxed{+ C^\star \sum_{i=n+1}^{n+m} \ell(|w^T x_i + b|)}}_{\text{Margins of unlabeled instances}}$$

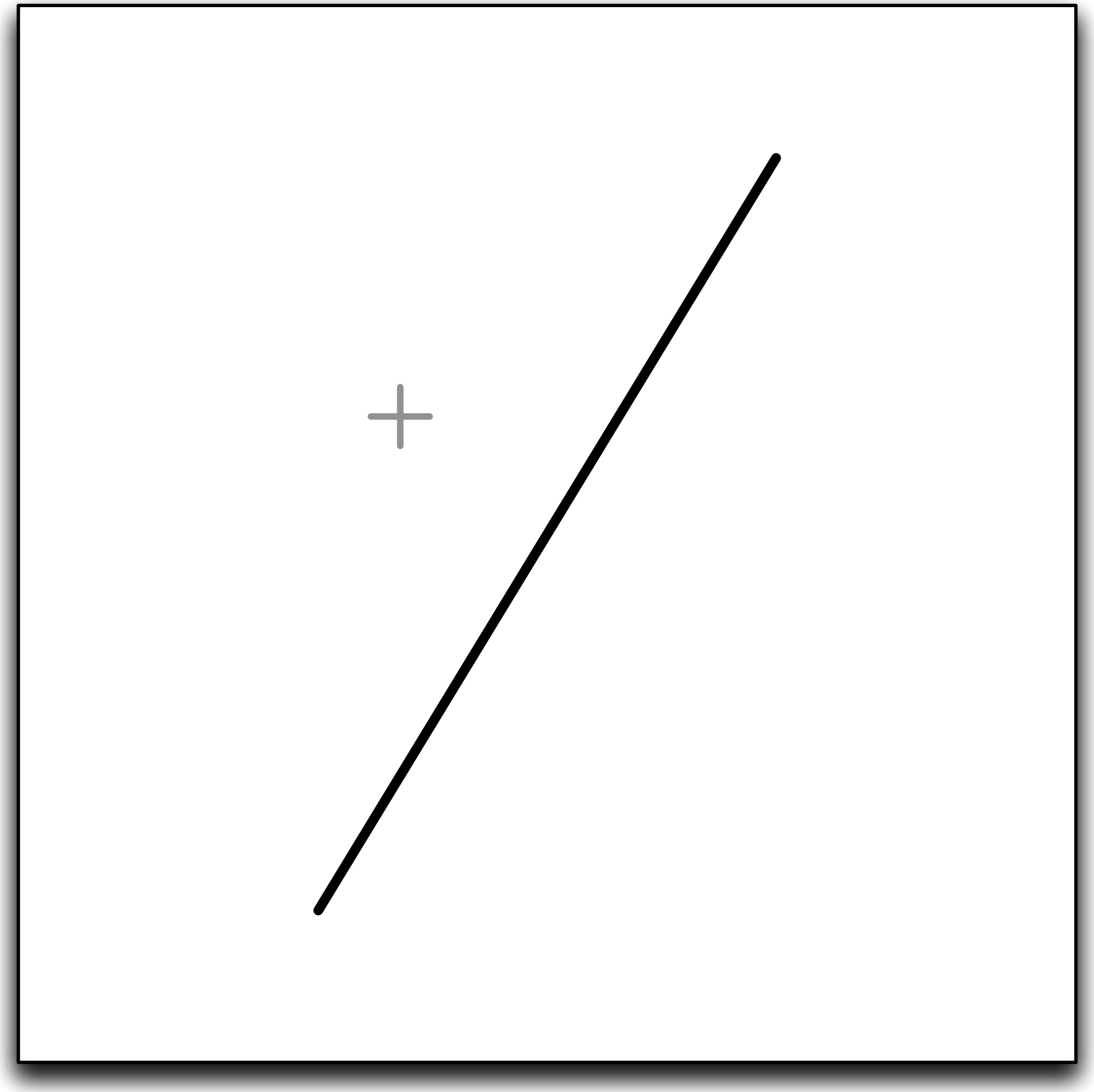
Semi-Supervised SVM

- Stochastic Gradient Descent
  - One run over the data in random order
  - Each misclassified or unlabeled instance moves classifier a bit
  - Steps get smaller over time
- Implementation on Hadoop
  - Mapper: send data to reducer in random order
  - Reducer: update linear classifier for unlabeled or misclassified instances
  - Many random runs to find best one



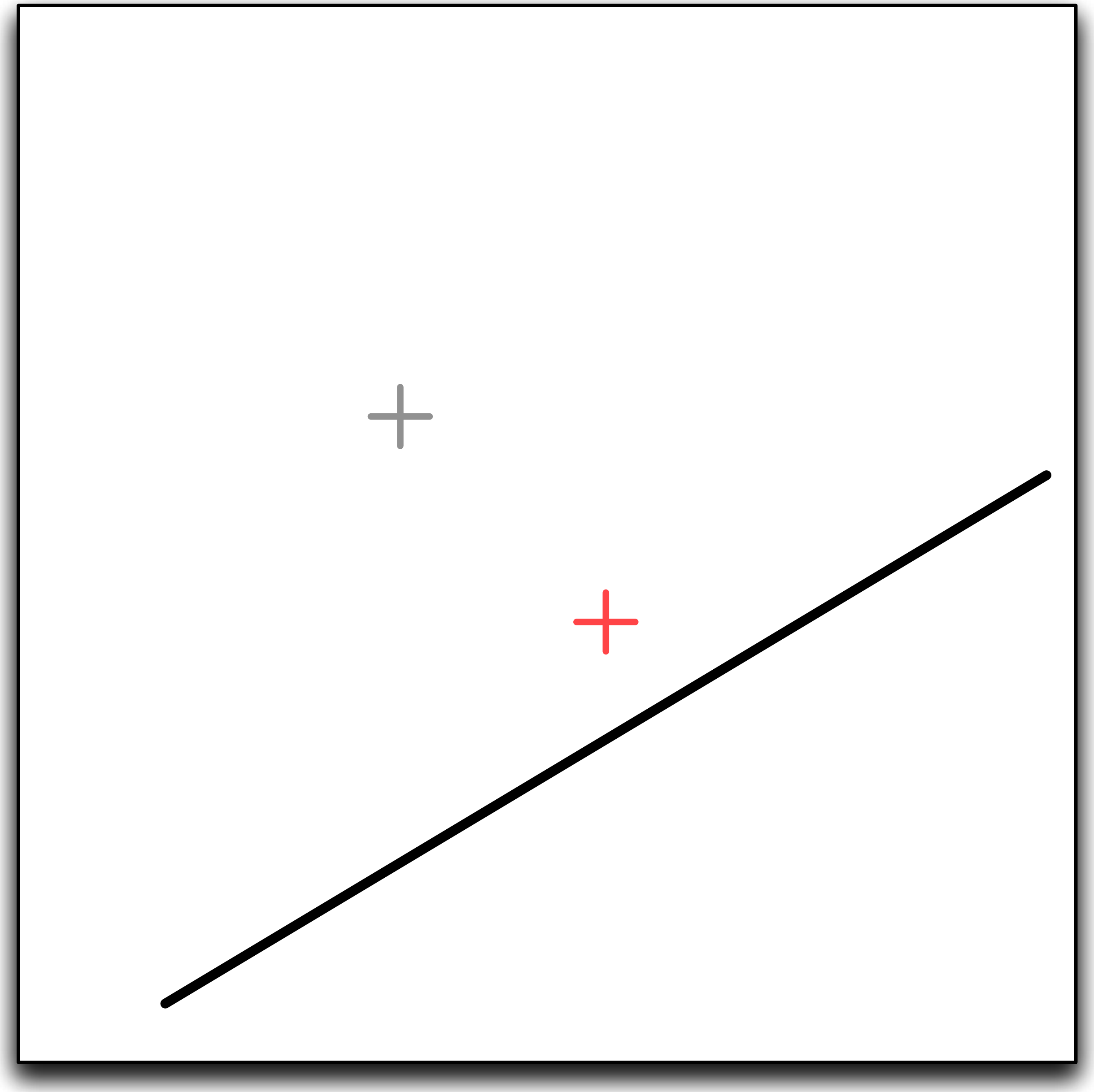
Semi-Supervised SVM

- Stochastic Gradient Descent
  - One run over the data in random order
  - Each misclassified or unlabeled instance moves classifier a bit
  - Steps get smaller over time
- Implementation on Hadoop
  - Mapper: send data to reducer in random order
  - Reducer: update linear classifier for unlabeled or misclassified instances
  - Many random runs to find best one



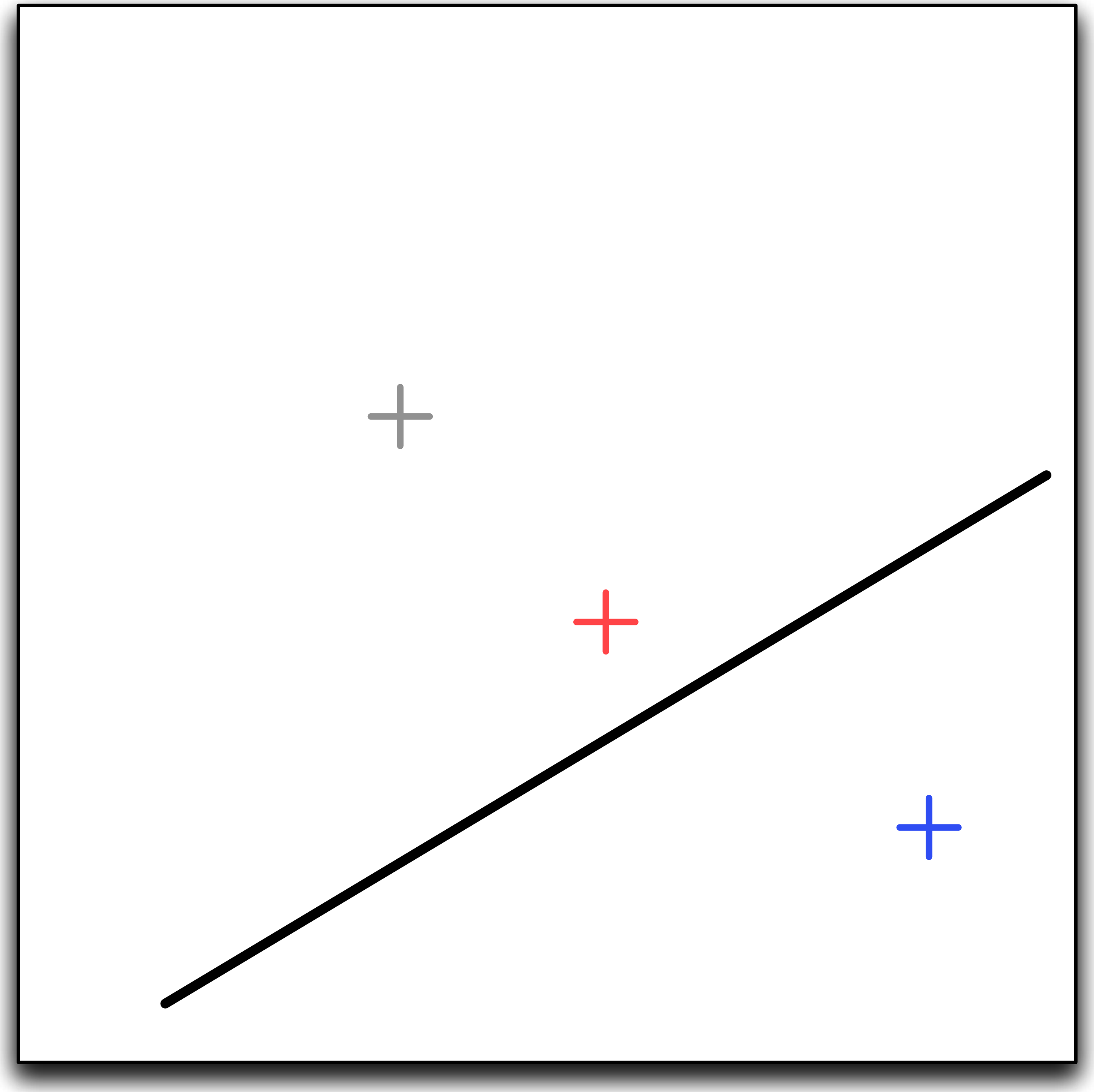
Semi-Supervised SVM

- Stochastic Gradient Descent
  - One run over the data in random order
  - Each misclassified or unlabeled instance moves classifier a bit
  - Steps get smaller over time
- Implementation on Hadoop
  - Mapper: send data to reducer in random order
  - Reducer: update linear classifier for unlabeled or misclassified instances
  - Many random runs to find best one



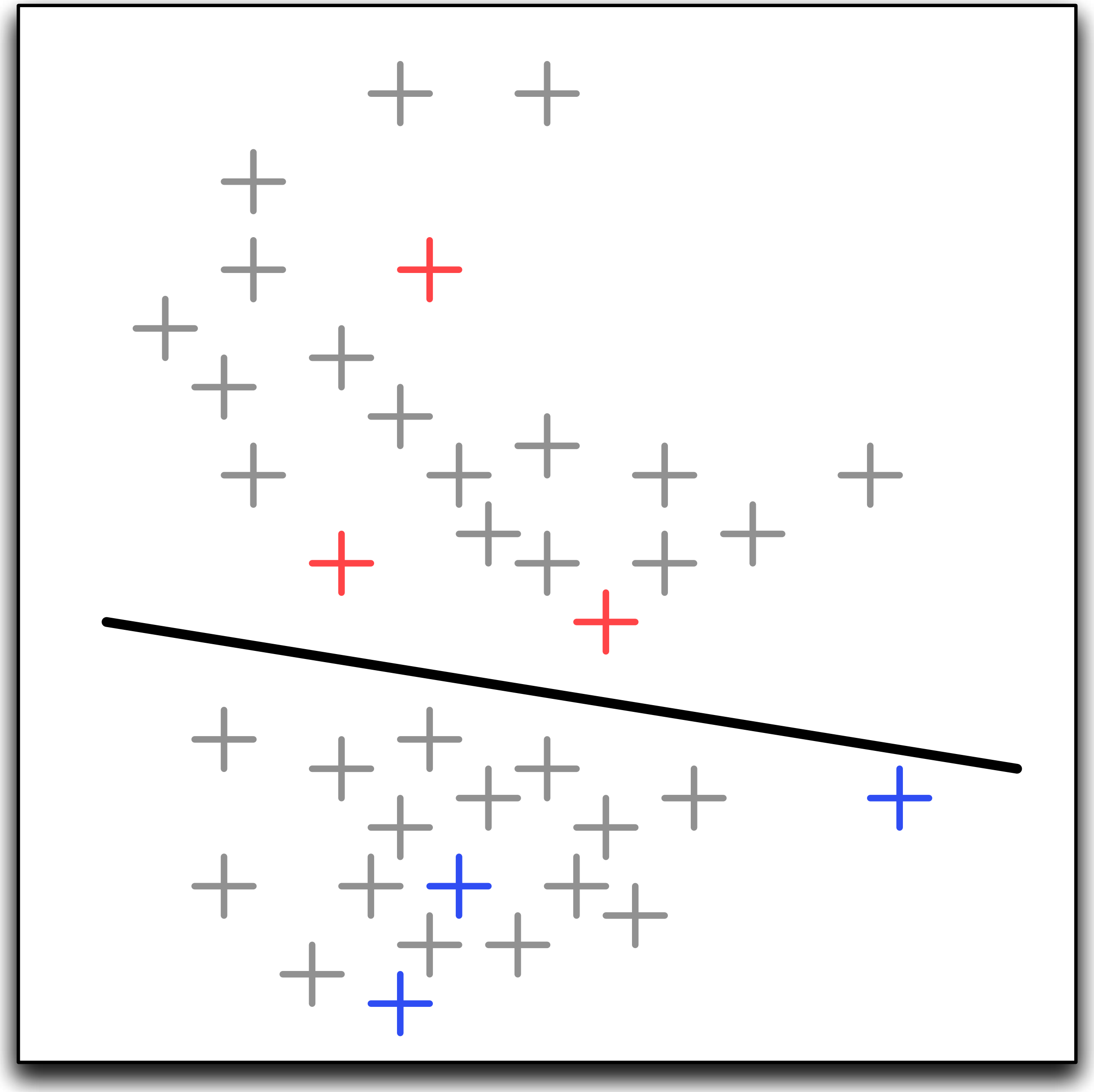
Semi-Supervised SVM

- Stochastic Gradient Descent
  - One run over the data in random order
  - Each misclassified or unlabeled instance moves classifier a bit
  - Steps get smaller over time
- Implementation on Hadoop
  - Mapper: send data to reducer in random order
  - Reducer: update linear classifier for unlabeled or misclassified instances
  - Many random runs to find best one



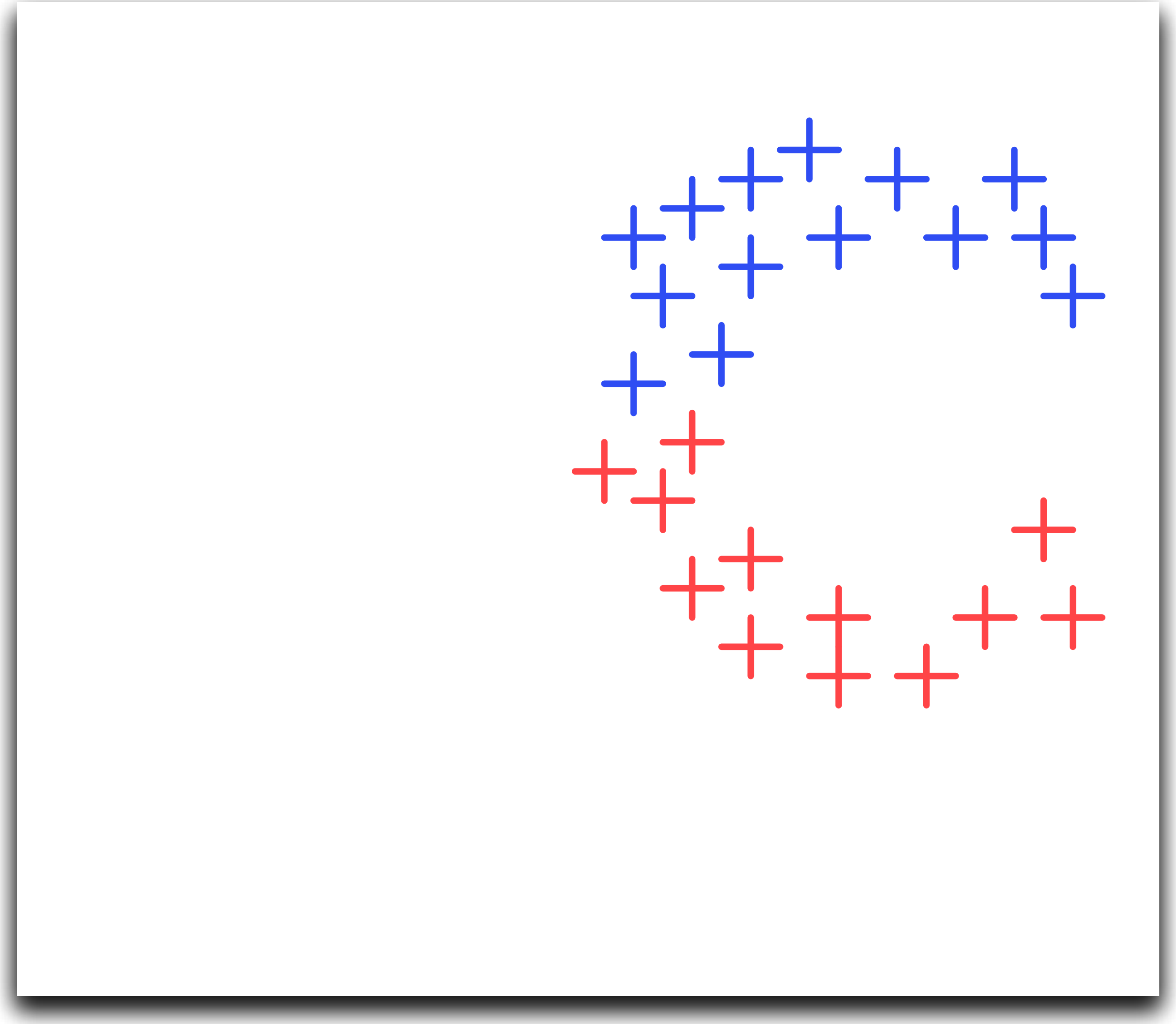
Semi-Supervised SVM

- Stochastic Gradient Descent
  - One run over the data in random order
  - Each misclassified or unlabeled instance moves classifier a bit
  - Steps get smaller over time
- Implementation on Hadoop
  - Mapper: send data to reducer in random order
  - Reducer: update linear classifier for unlabeled or misclassified instances
  - Many random runs to find best one



The Manifold Assumption

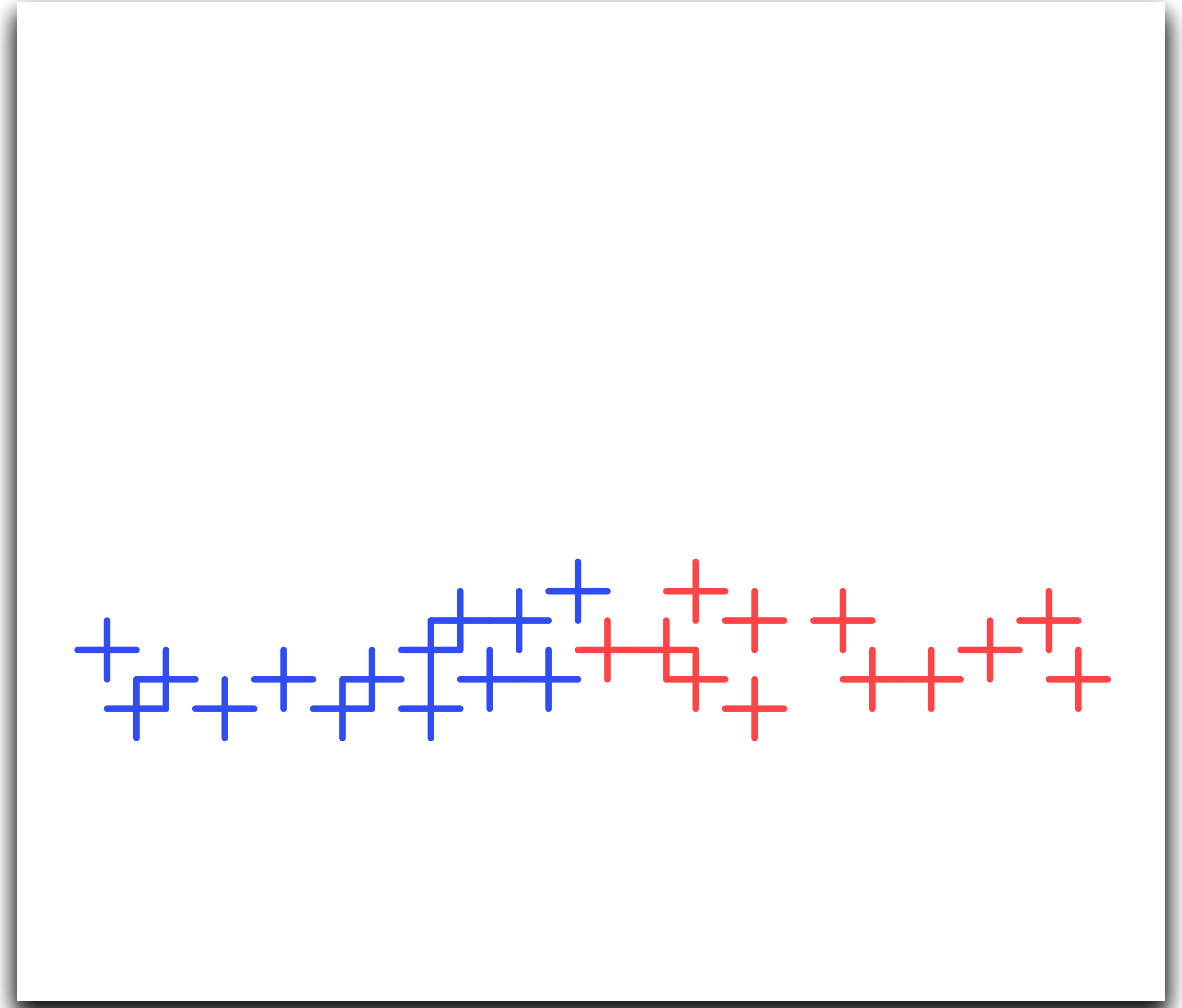
- The Assumption
  - Training data is (roughly) contained in a low dimensional manifold
  - One can perform learning in a more meaningful low-dimensional space
  - Avoids curse of dimensionality



- Similarity Graphs
  - Idea: compute similarity scores between instances
  - Create network where the nearest neighbors are connected

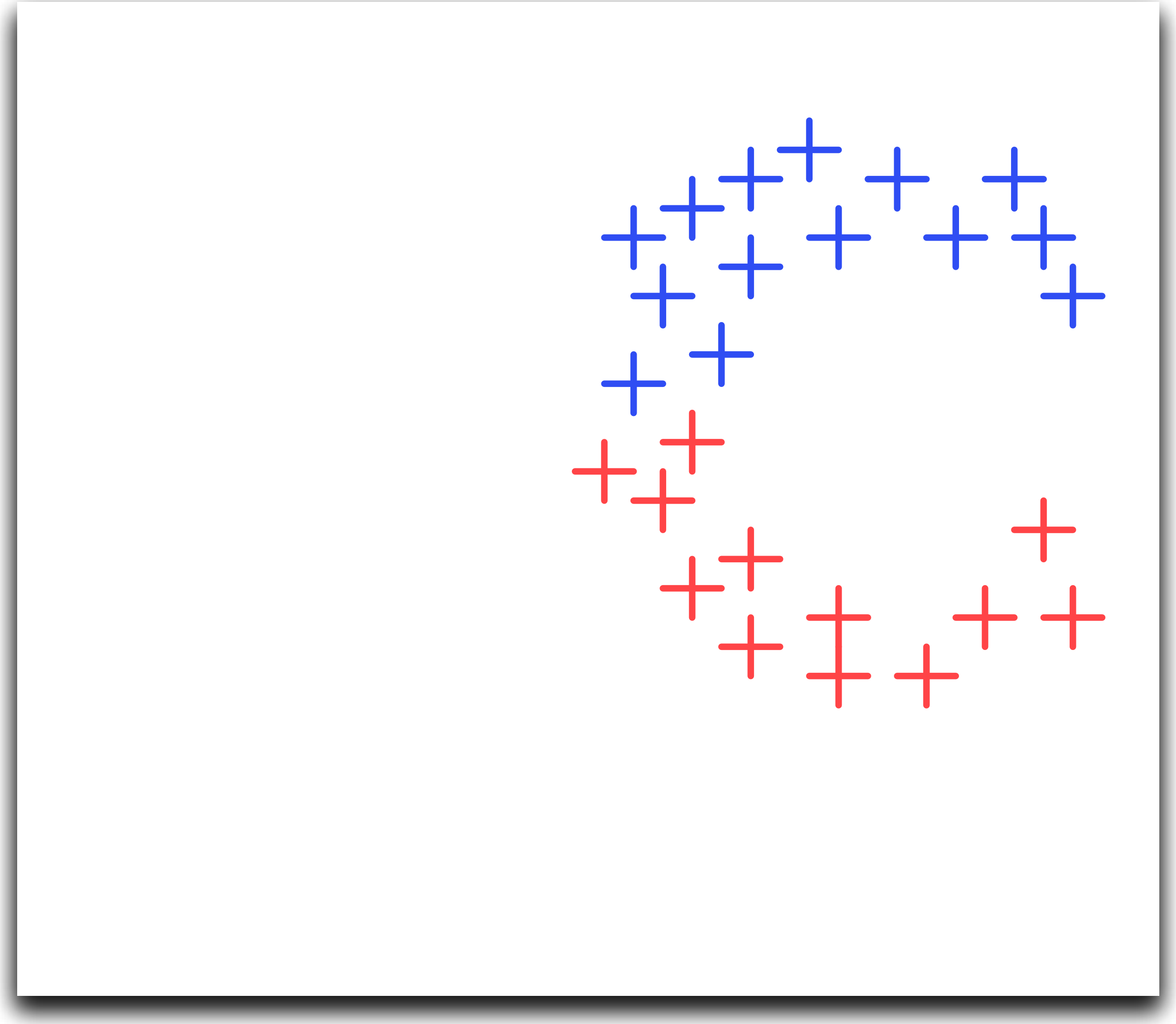
The Manifold Assumption

- The Assumption
  - Training data is (roughly) contained in a low dimensional manifold
  - One can perform learning in a more meaningful low-dimensional space
  - Avoids curse of dimensionality
- Similarity Graphs
  - Idea: compute similarity scores between instances
  - Create network where the nearest neighbors are connected



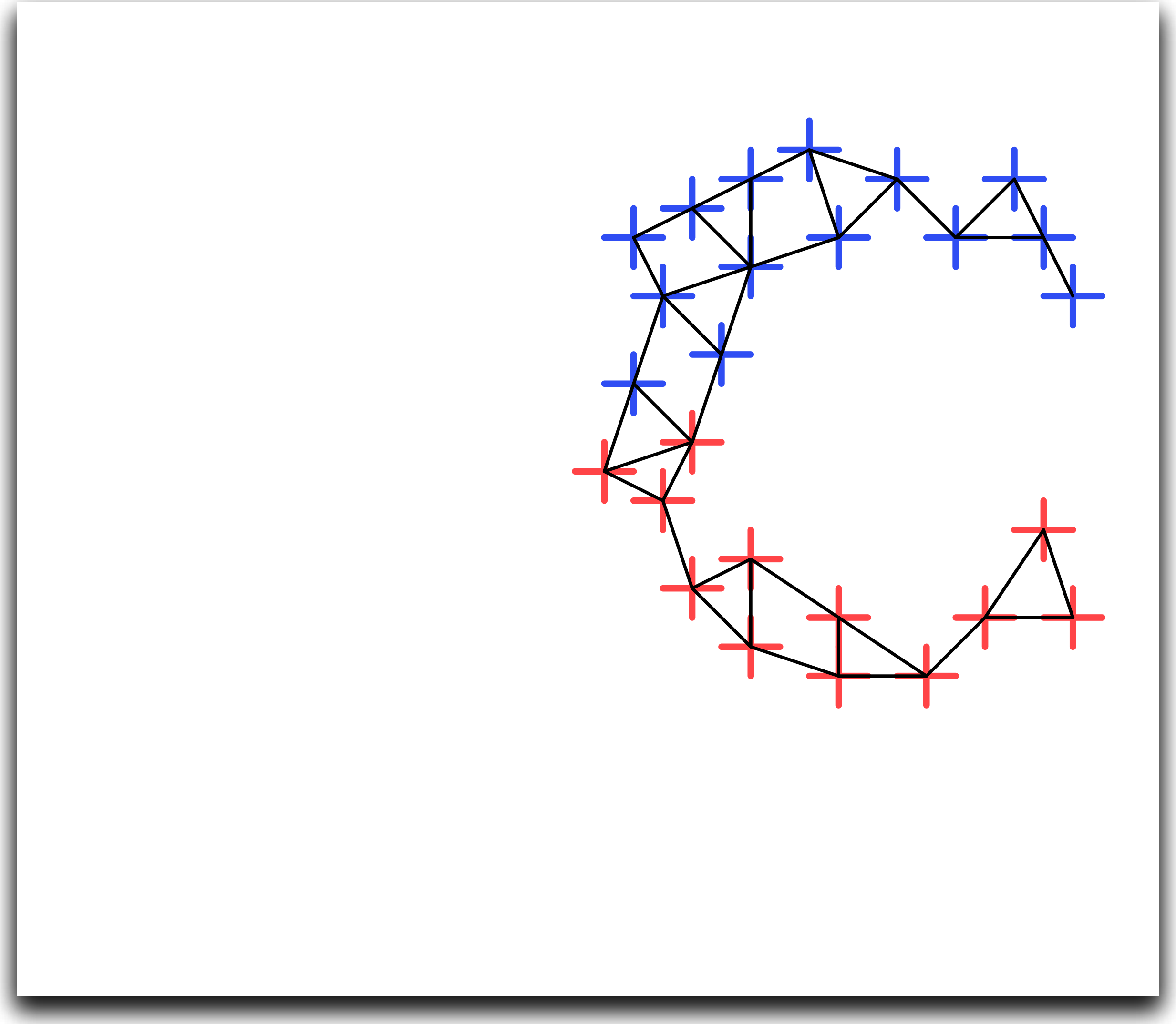
The Manifold Assumption

- The Assumption
  - Training data is (roughly) contained in a low dimensional manifold
  - One can perform learning in a more meaningful low-dimensional space
  - Avoids curse of dimensionality
- Similarity Graphs
  - Idea: compute similarity scores between instances
  - Create network where the nearest neighbors are connected



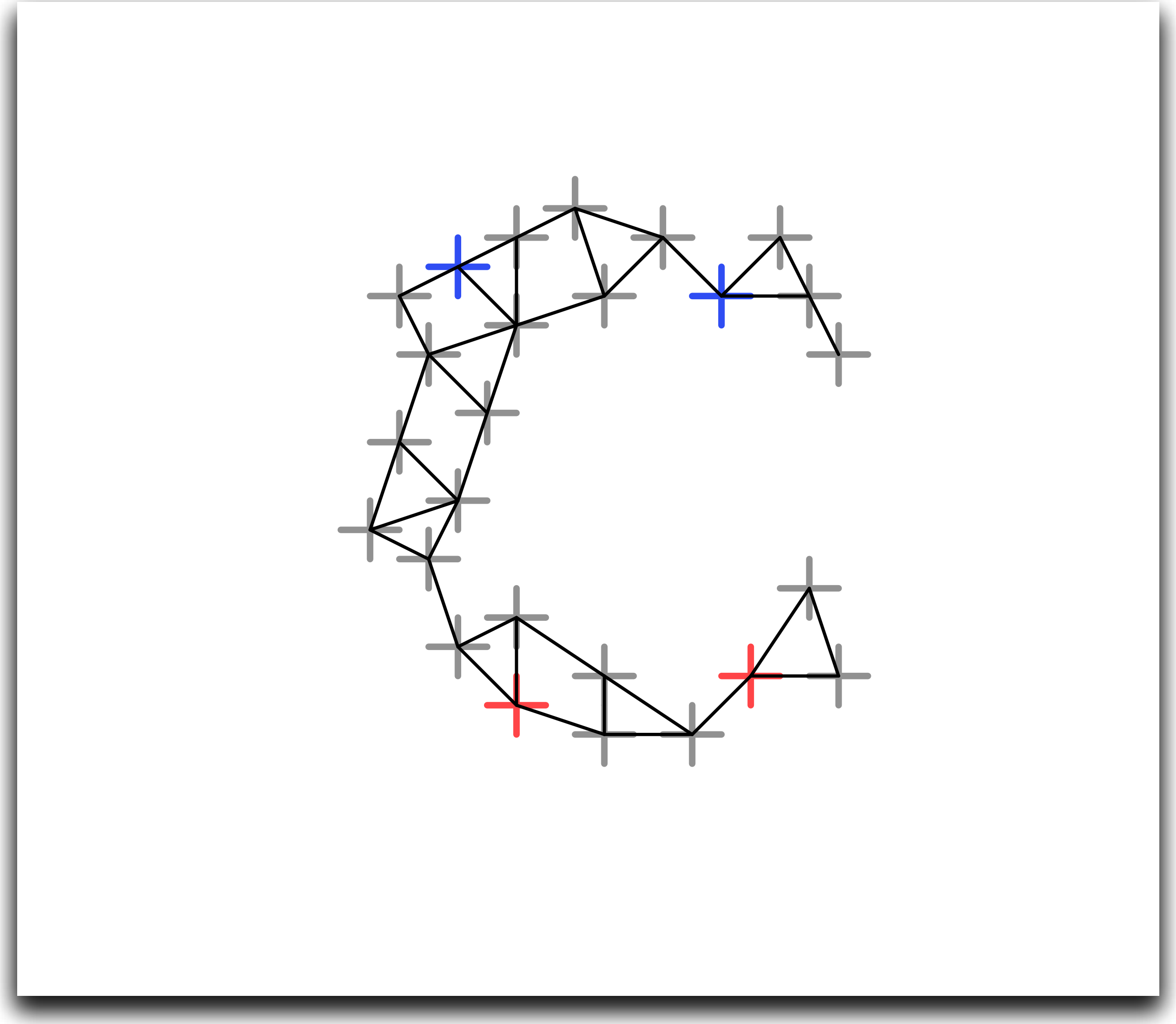
The Manifold Assumption

- The Assumption
  - Training data is (roughly) contained in a low dimensional manifold
  - One can perform learning in a more meaningful low-dimensional space
  - Avoids curse of dimensionality
- Similarity Graphs
  - Idea: compute similarity scores between instances
  - Create network where the nearest neighbors are connected



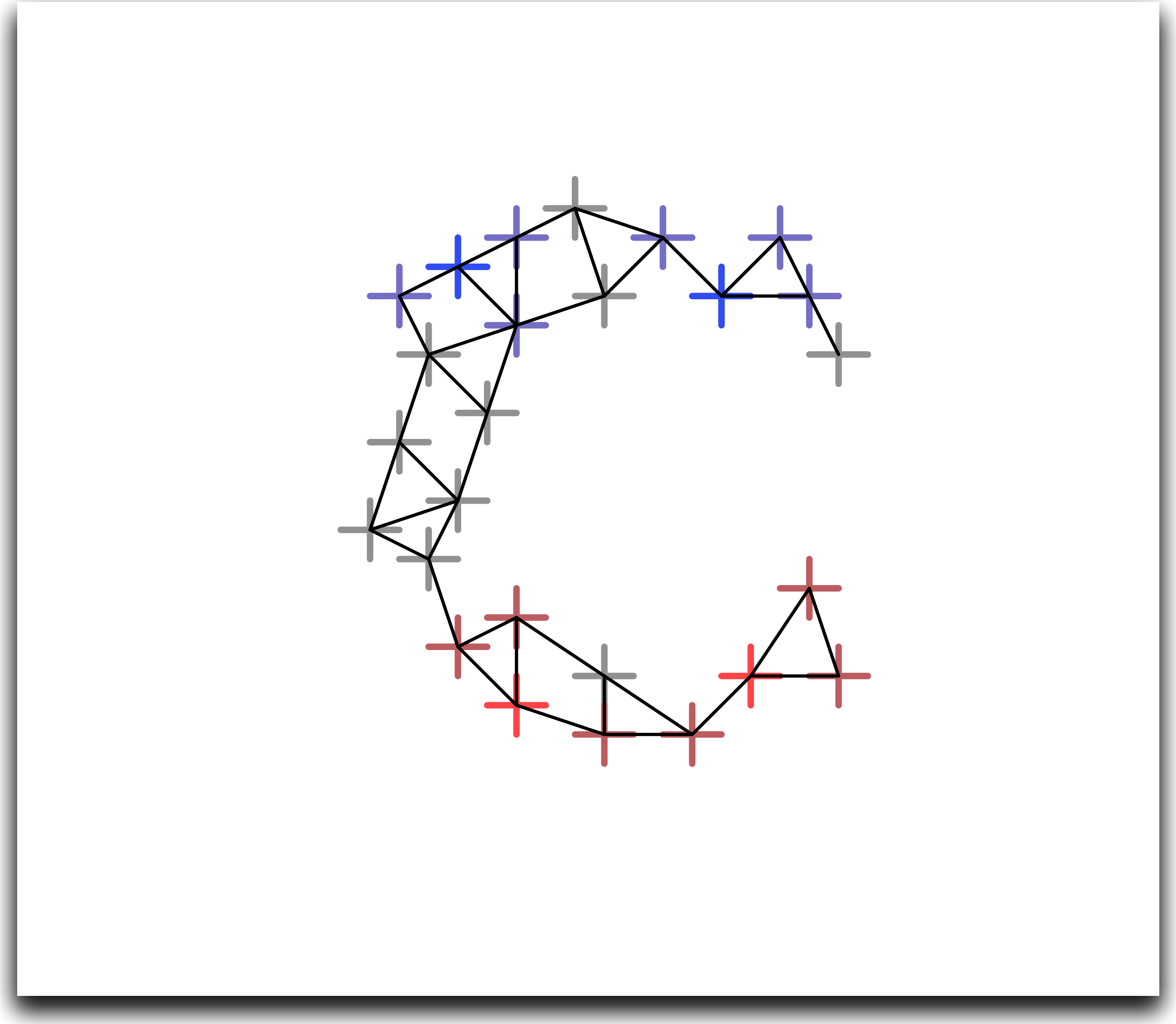
Label Propagation

- Main Idea
  - Propagate label information to neighboring instances
  - Then repeat until convergence
  - Similar to PageRank
- Theory
  - Known to converge under weak conditions
  - Equivalent to matrix inversion



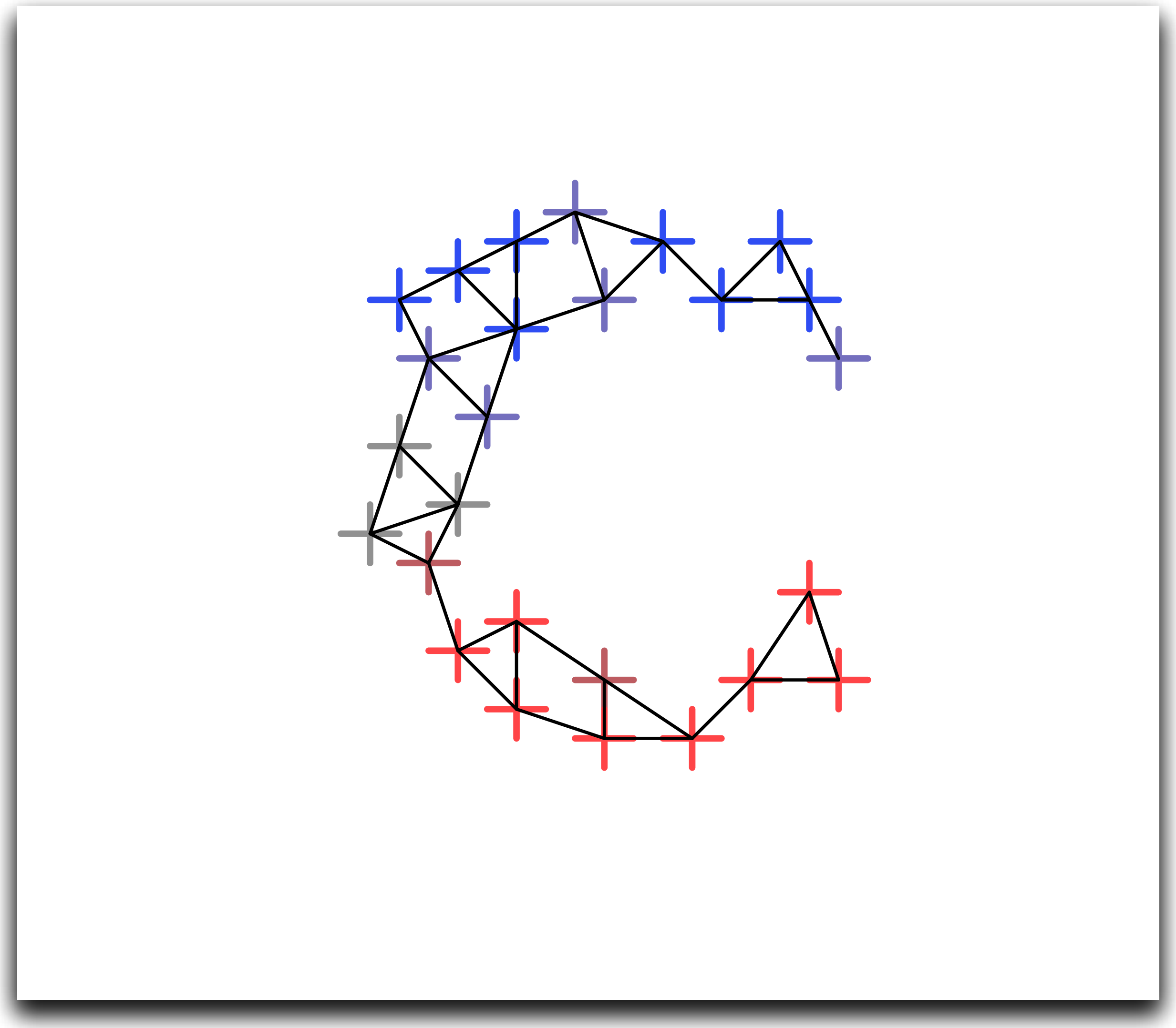
Label Propagation

- Main Idea
  - Propagate label information to neighboring instances
  - Then repeat until convergence
  - Similar to PageRank
- Theory
  - Known to converge under weak conditions
  - Equivalent to matrix inversion



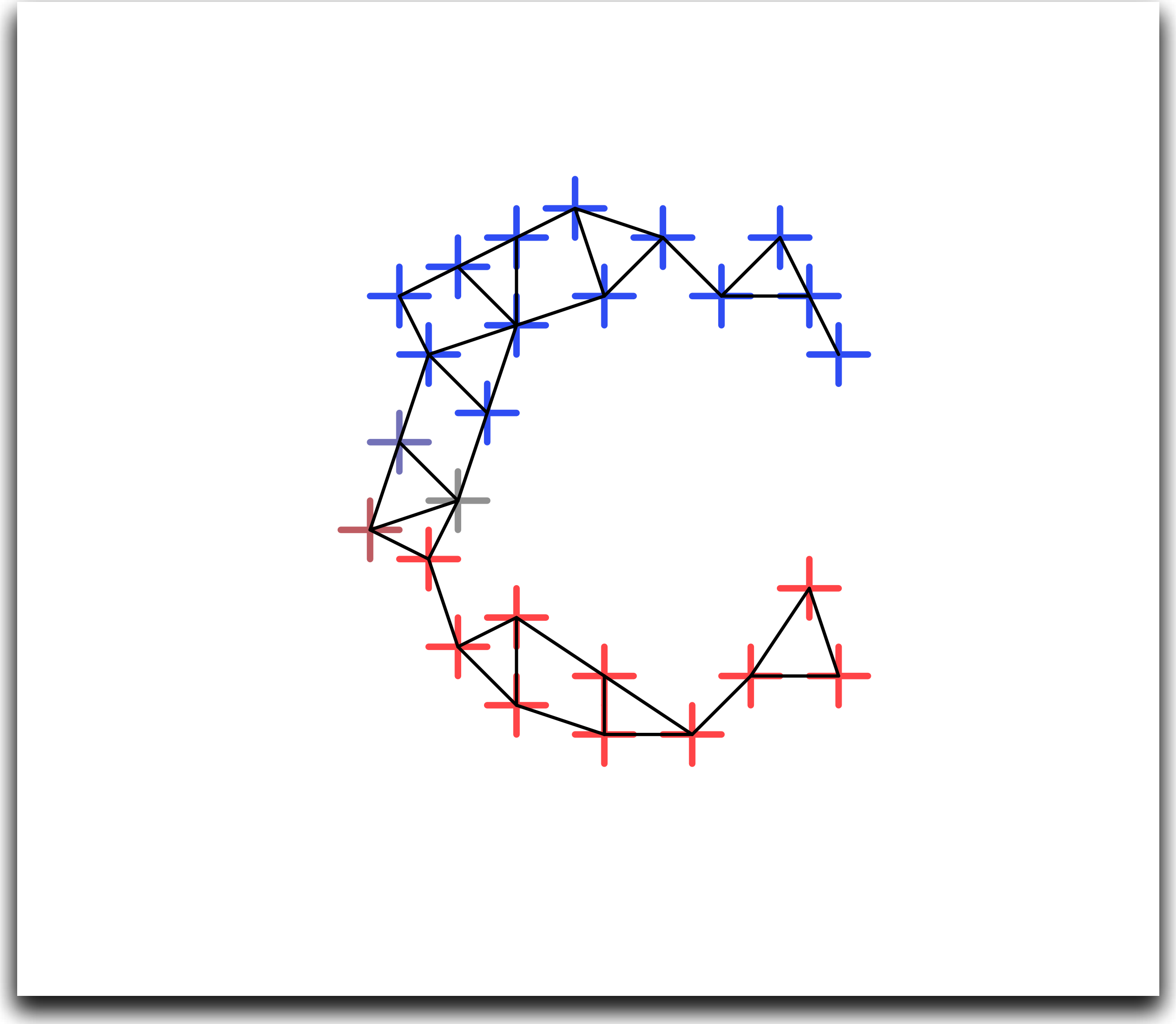
Label Propagation

- Main Idea
  - Propagate label information to neighboring instances
  - Then repeat until convergence
  - Similar to PageRank
- Theory
  - Known to converge under weak conditions
  - Equivalent to matrix inversion



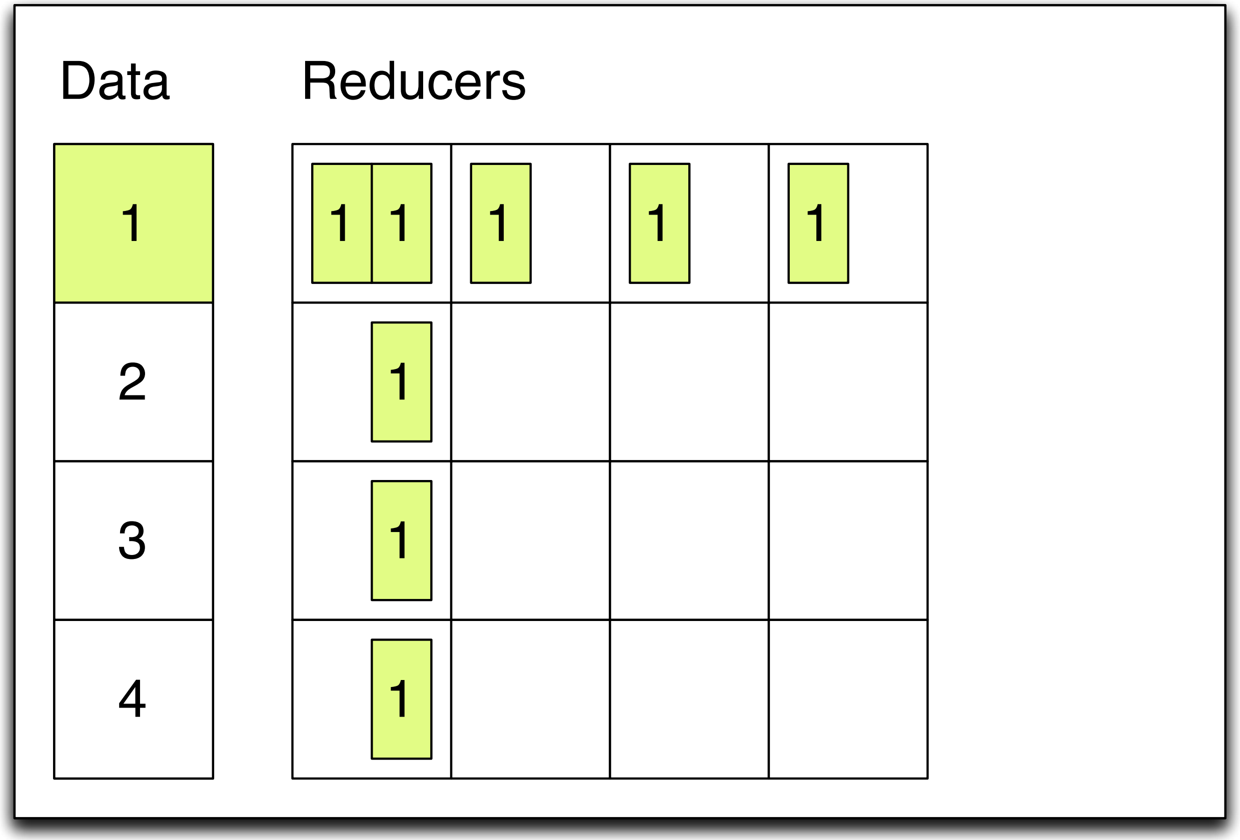
Label Propagation

- Main Idea
  - Propagate label information to neighboring instances
  - Then repeat until convergence
  - Similar to PageRank
- Theory
  - Known to converge under weak conditions
  - Equivalent to matrix inversion



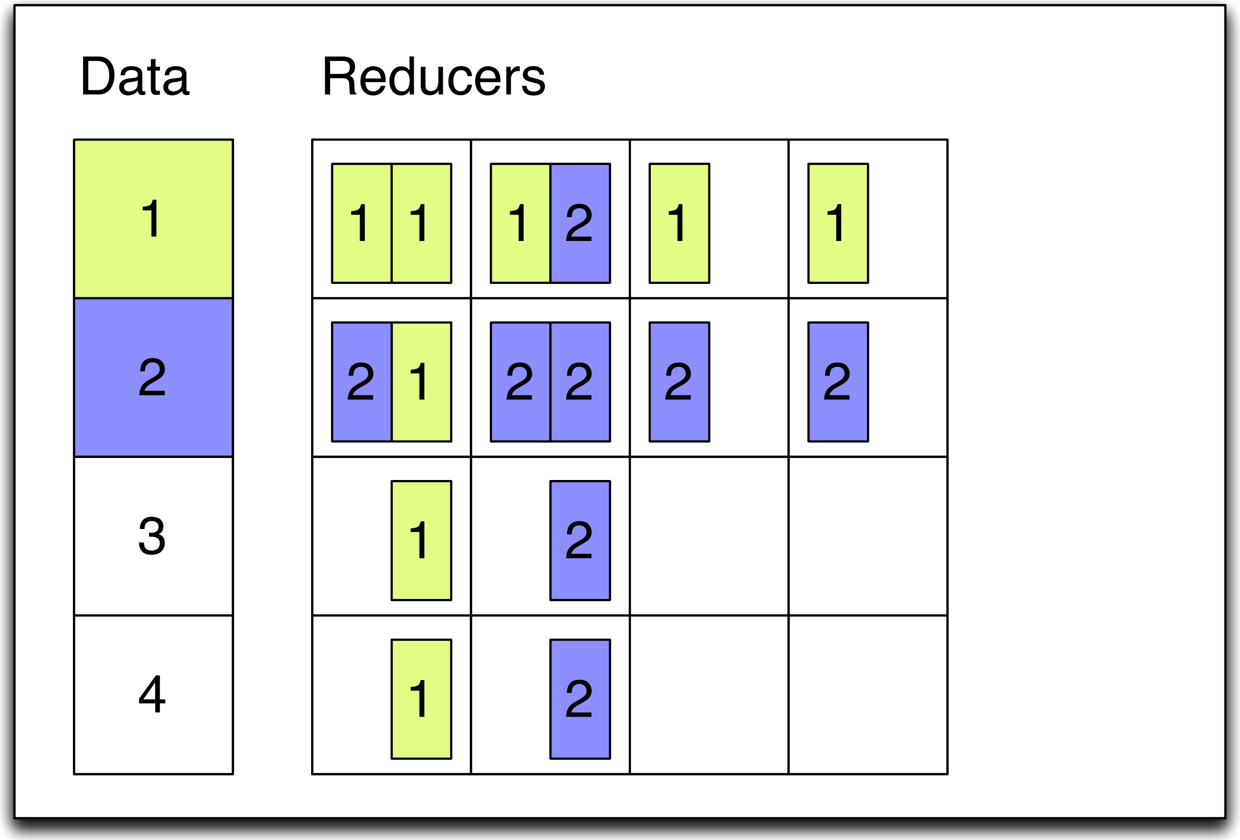
Nearest Neighbor Join

- Block Nested Loop Join
  - 1st MR job: partition data into blocks, compute nearest neighbors between blocks
  - 2nd MR job: filter out the overall nearest neighbors
- Smarter Approaches
  - Use spatial information to avoid unnecessary comparisons
  - R trees, space filling curves, locality sensitive hashing
  - Example implementations available online



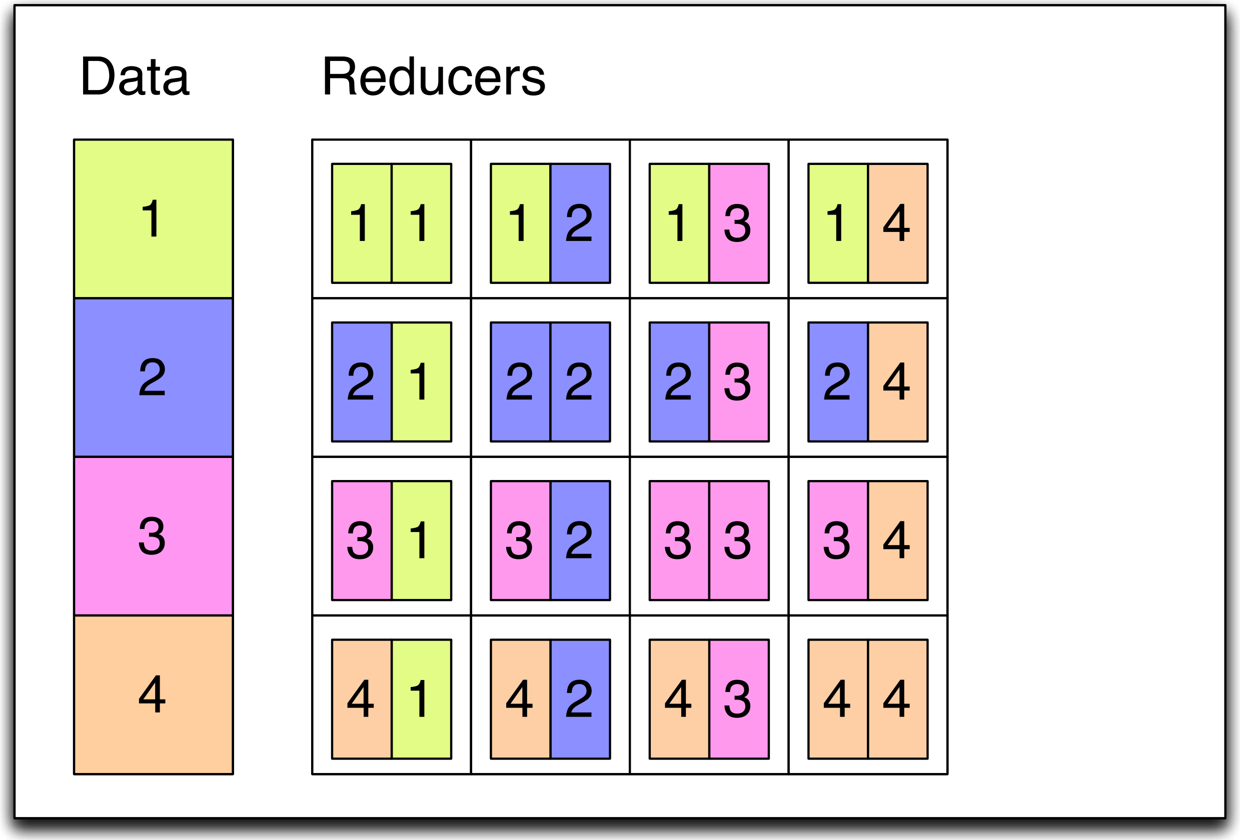
Nearest Neighbor Join

- Block Nested Loop Join
  - 1st MR job: partition data into blocks, compute nearest neighbors between blocks
  - 2nd MR job: filter out the overall nearest neighbors
- Smarter Approaches
  - Use spatial information to avoid unnecessary comparisons
  - R trees, space filling curves, locality sensitive hashing
  - Example implementations available online



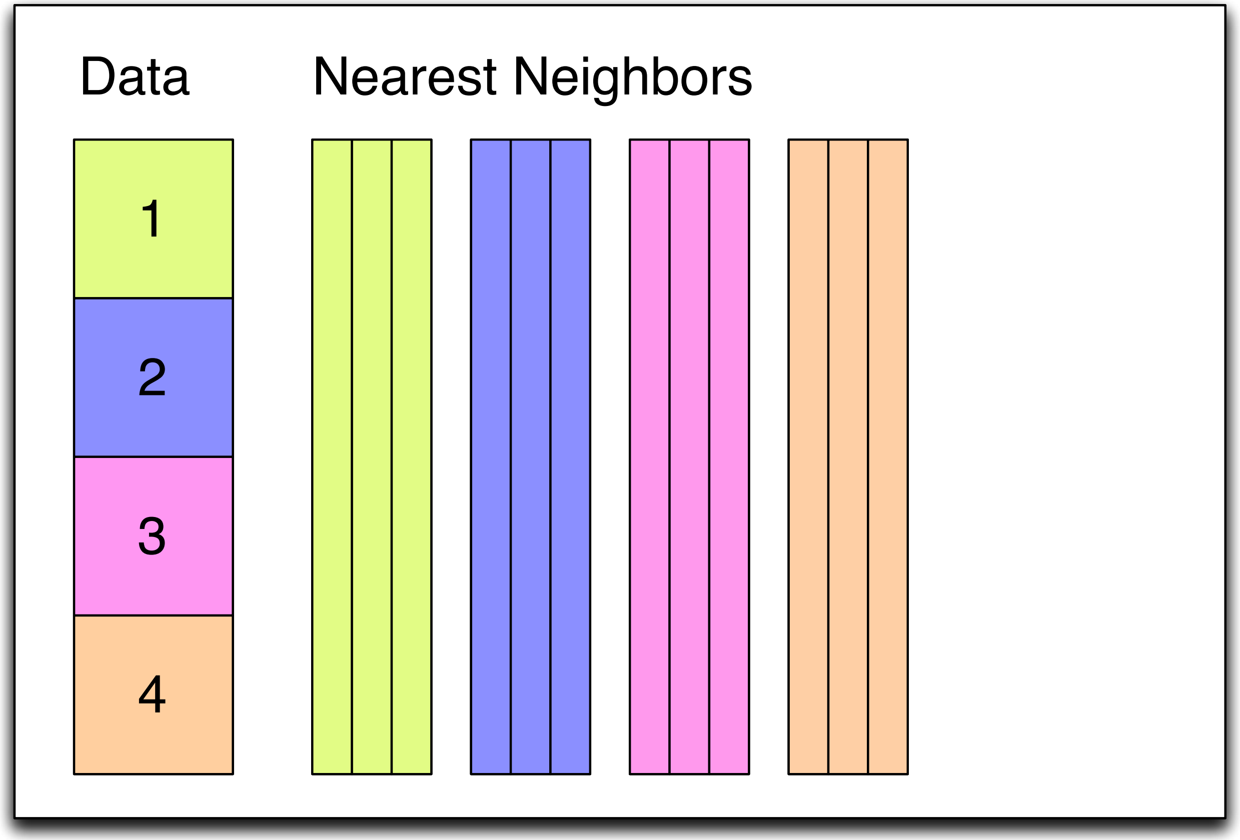
Nearest Neighbor Join

- Block Nested Loop Join
  - 1st MR job: partition data into blocks, compute nearest neighbors between blocks
  - 2nd MR job: filter out the overall nearest neighbors
- Smarter Approaches
  - Use spatial information to avoid unnecessary comparisons
  - R trees, space filling curves, locality sensitive hashing
  - Example implementations available online



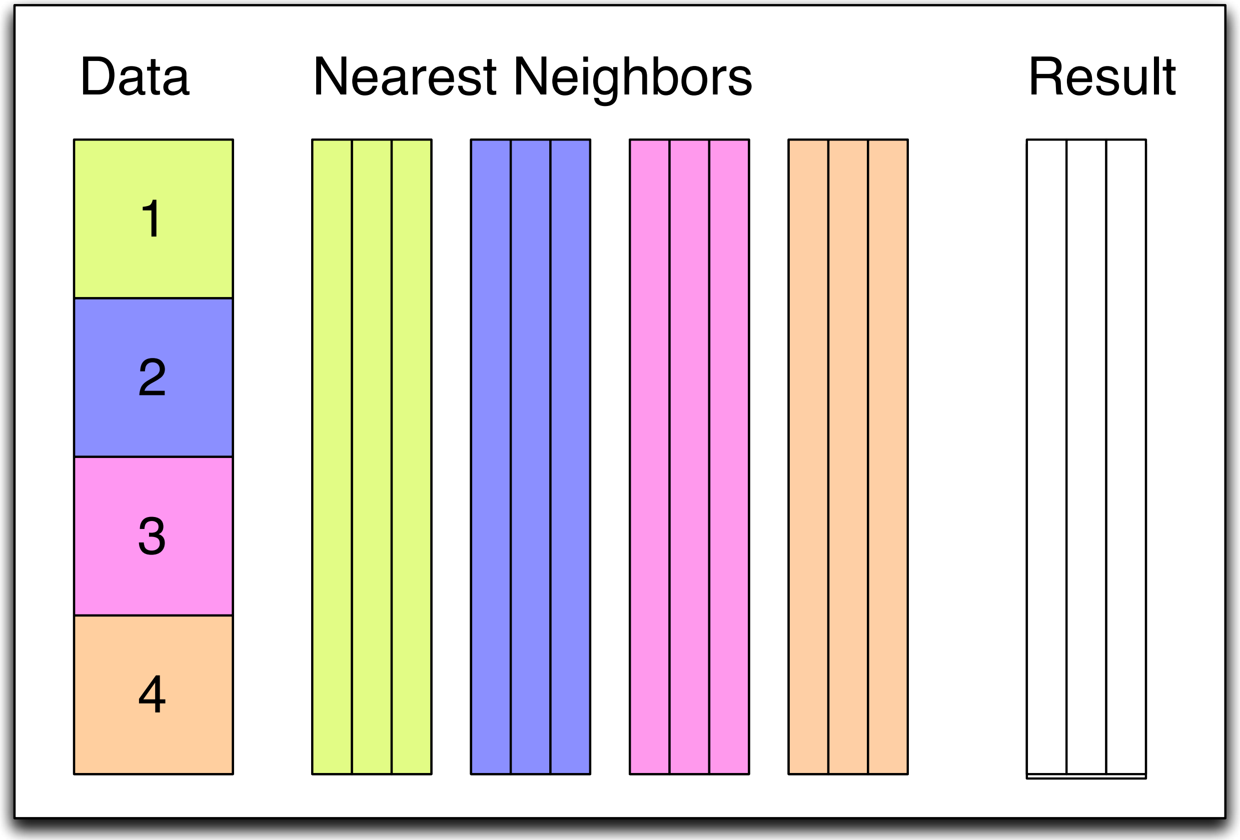
Nearest Neighbor Join

- Block Nested Loop Join
  - 1st MR job: partition data into blocks, compute nearest neighbors between blocks
  - 2nd MR job: filter out the overall nearest neighbors
- Smarter Approaches
  - Use spatial information to avoid unnecessary comparisons
  - R trees, space filling curves, locality sensitive hashing
  - Example implementations available online



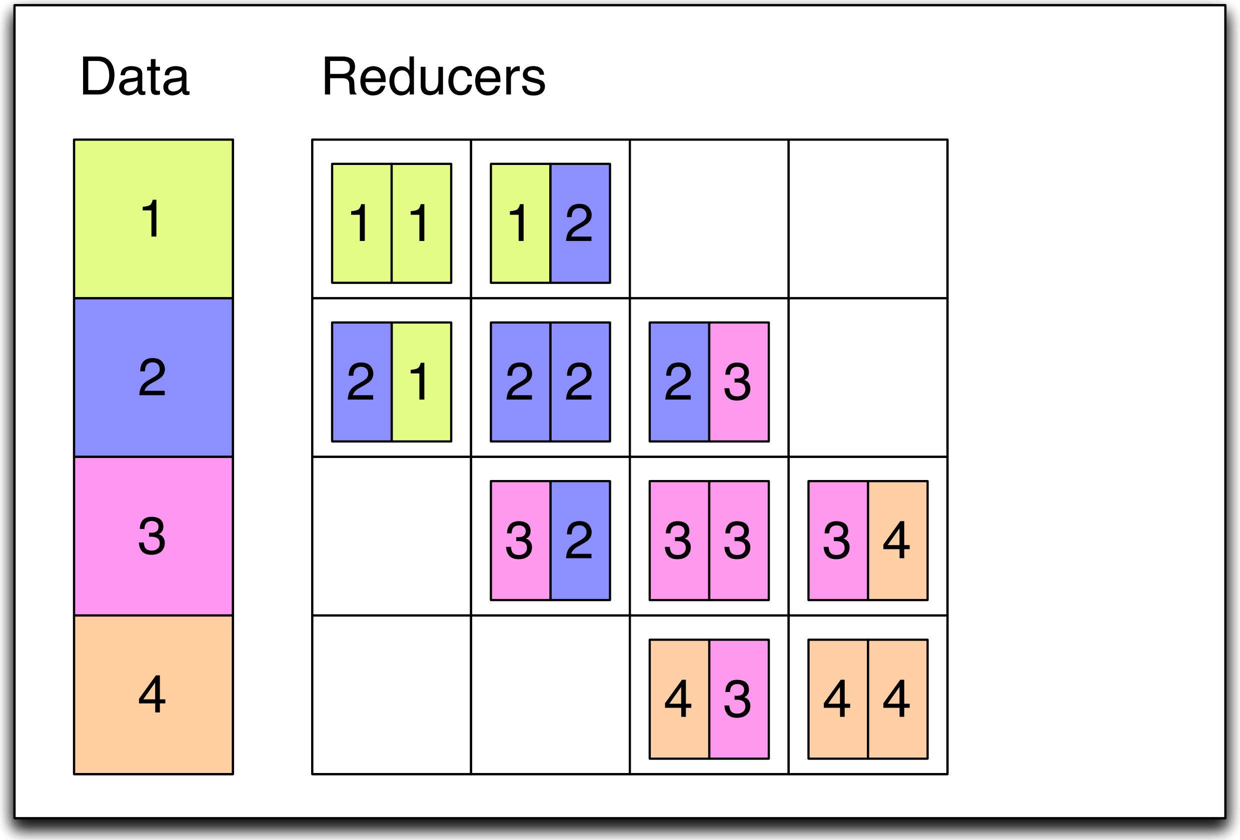
Nearest Neighbor Join

- Block Nested Loop Join
  - 1st MR job: partition data into blocks, compute nearest neighbors between blocks
  - 2nd MR job: filter out the overall nearest neighbors
- Smarter Approaches
  - Use spatial information to avoid unnecessary comparisons
  - R trees, space filling curves, locality sensitive hashing
  - Example implementations available online



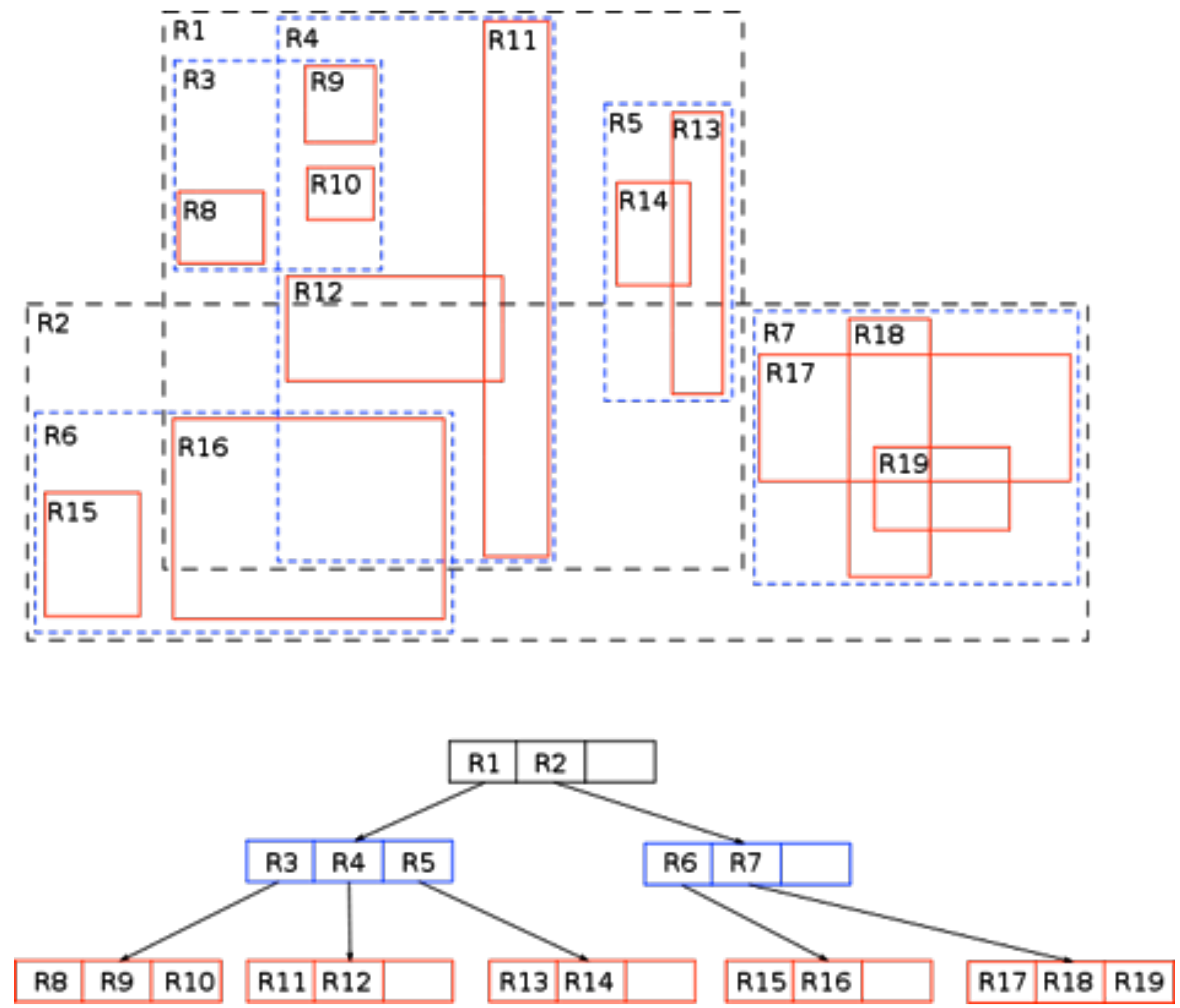
Nearest Neighbor Join

- Block Nested Loop Join
  - 1st MR job: partition data into blocks, compute nearest neighbors between blocks
  - 2nd MR job: filter out the overall nearest neighbors
- Smarter Approaches
  - Use spatial information to avoid unnecessary comparisons
  - R trees, space filling curves, locality sensitive hashing
  - Example implementations available online



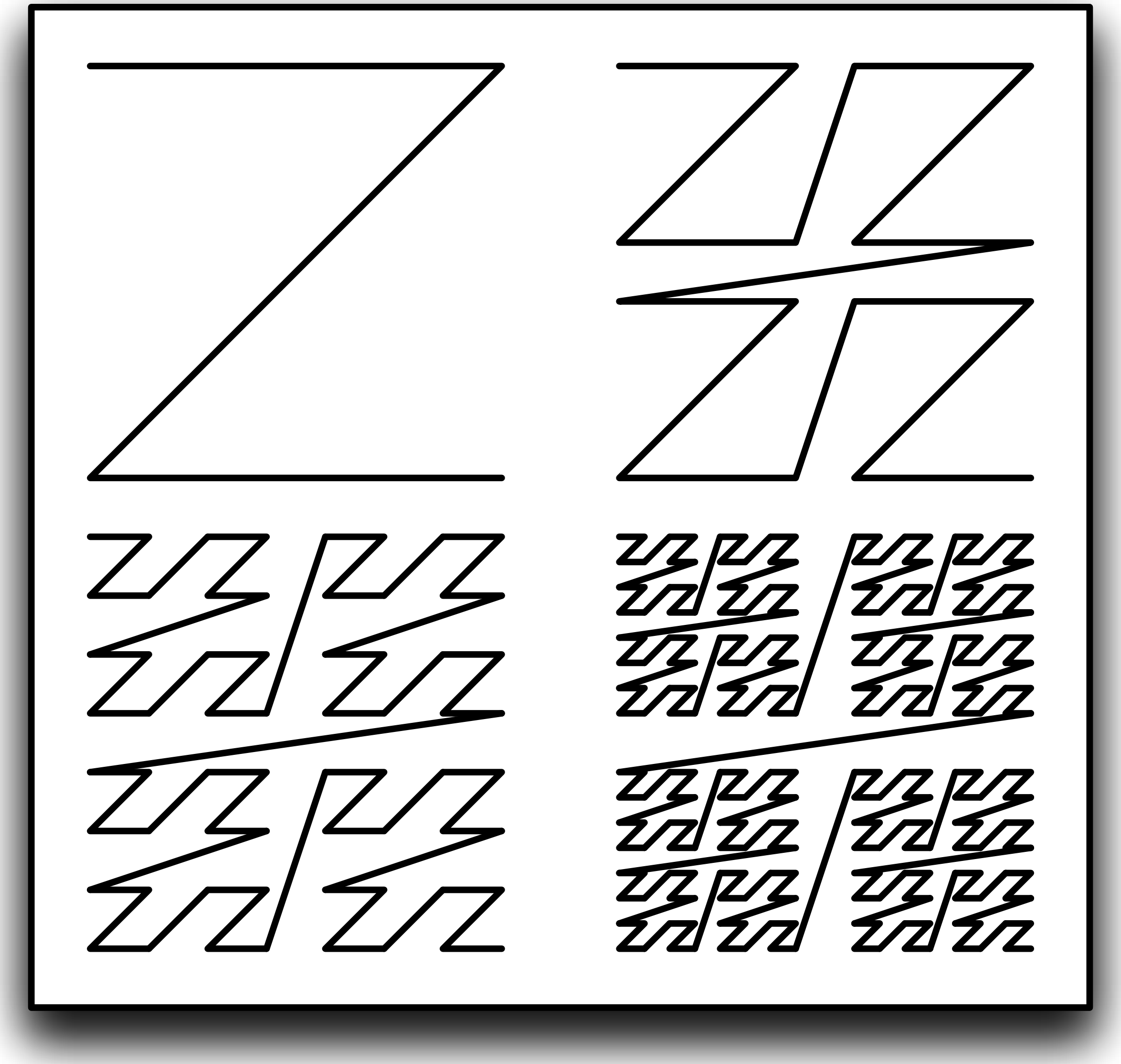
Nearest Neighbor Join

- Block Nested Loop Join
  - 1st MR job: partition data into blocks, compute nearest neighbors between blocks
  - 2nd MR job: filter out the overall nearest neighbors
- Smarter Approaches
  - Use spatial information to avoid unnecessary comparisons
  - R trees, space filling curves, locality sensitive hashing
  - Example implementations available online



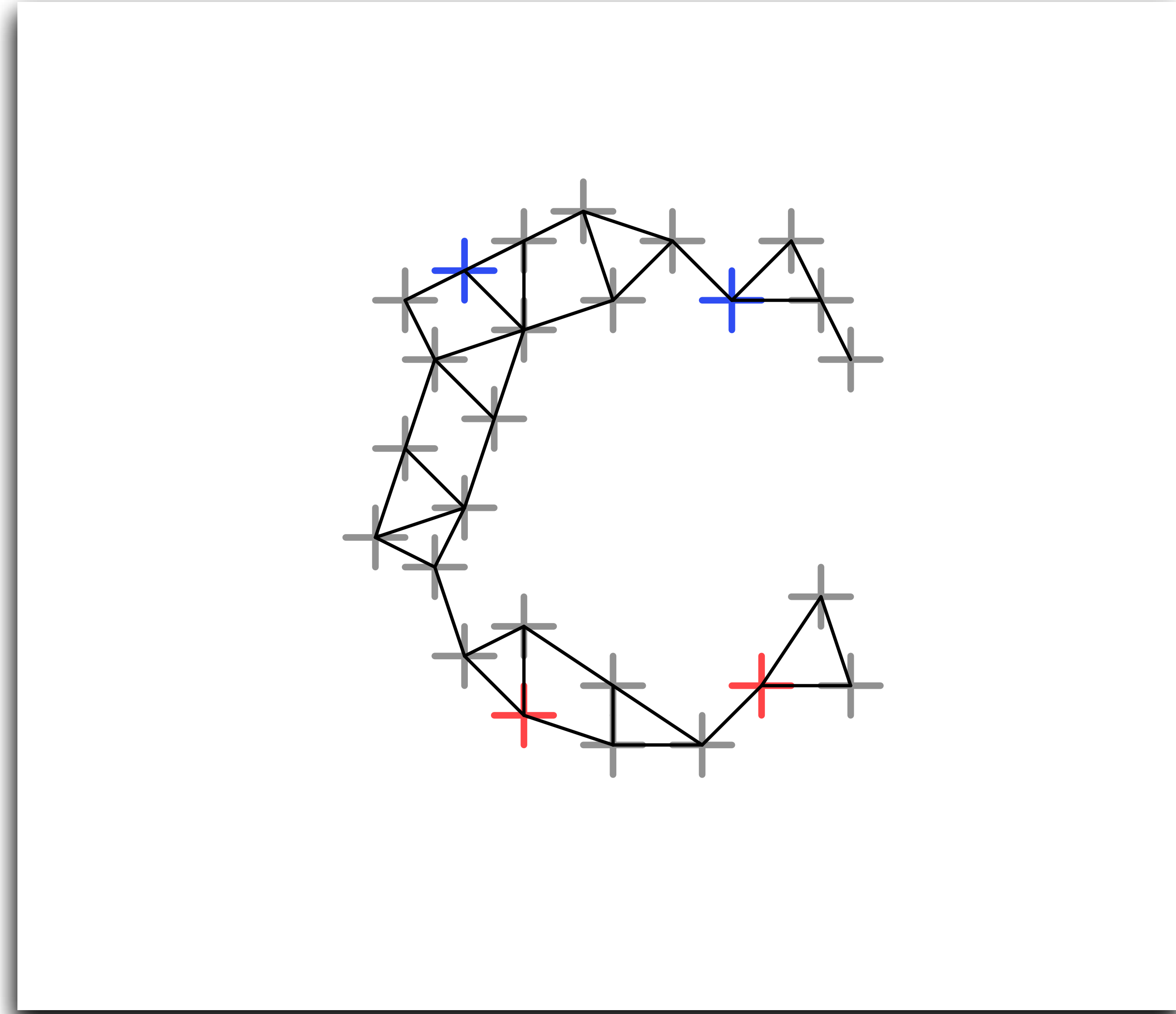
Nearest Neighbor Join

- Block Nested Loop Join
  - 1st MR job: partition data into blocks, compute nearest neighbors between blocks
  - 2nd MR job: filter out the overall nearest neighbors
- Smarter Approaches
  - Use spatial information to avoid unnecessary comparisons
  - R trees, space filling curves, locality sensitive hashing
  - Example implementations available online



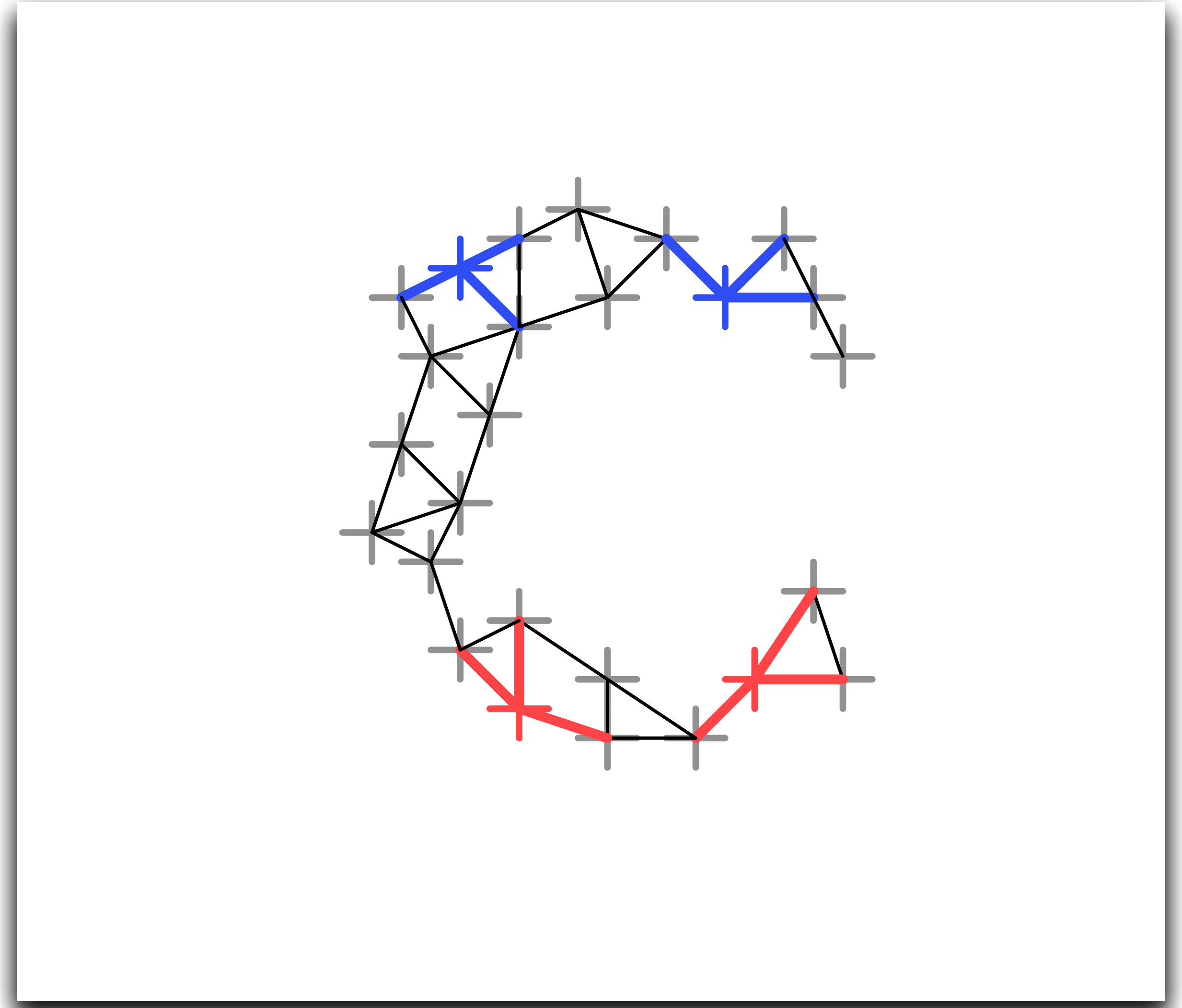
Label Propagation on Hadoop

- Iterative Procedure
  - Mapper: Emit neighbor-label pairs
  - Reducer: Collect incoming labels per node and combine them into new label
  - Repeat until convergence
- Improvement
  - Cluster data and perform within-cluster propagation locally
  - Sometimes it's more efficient to perform matrix inversion instead



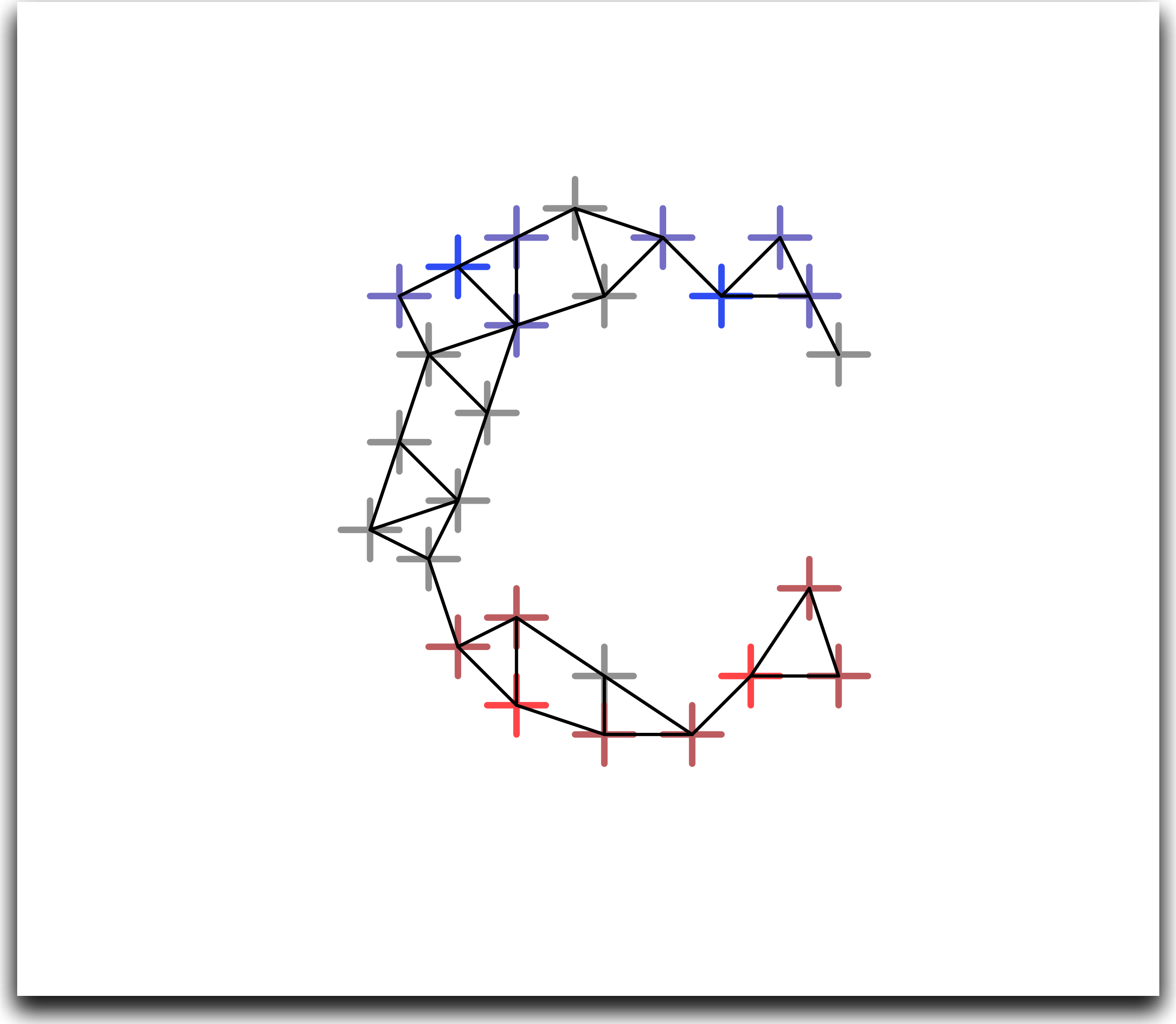
Label Propagation on Hadoop

- Iterative Procedure
  - Mapper: Emit neighbor-label pairs
  - Reducer: Collect incoming labels per node and combine them into new label
  - Repeat until convergence
- Improvement
  - Cluster data and perform within-cluster propagation locally
  - Sometimes it's more efficient to perform matrix inversion instead



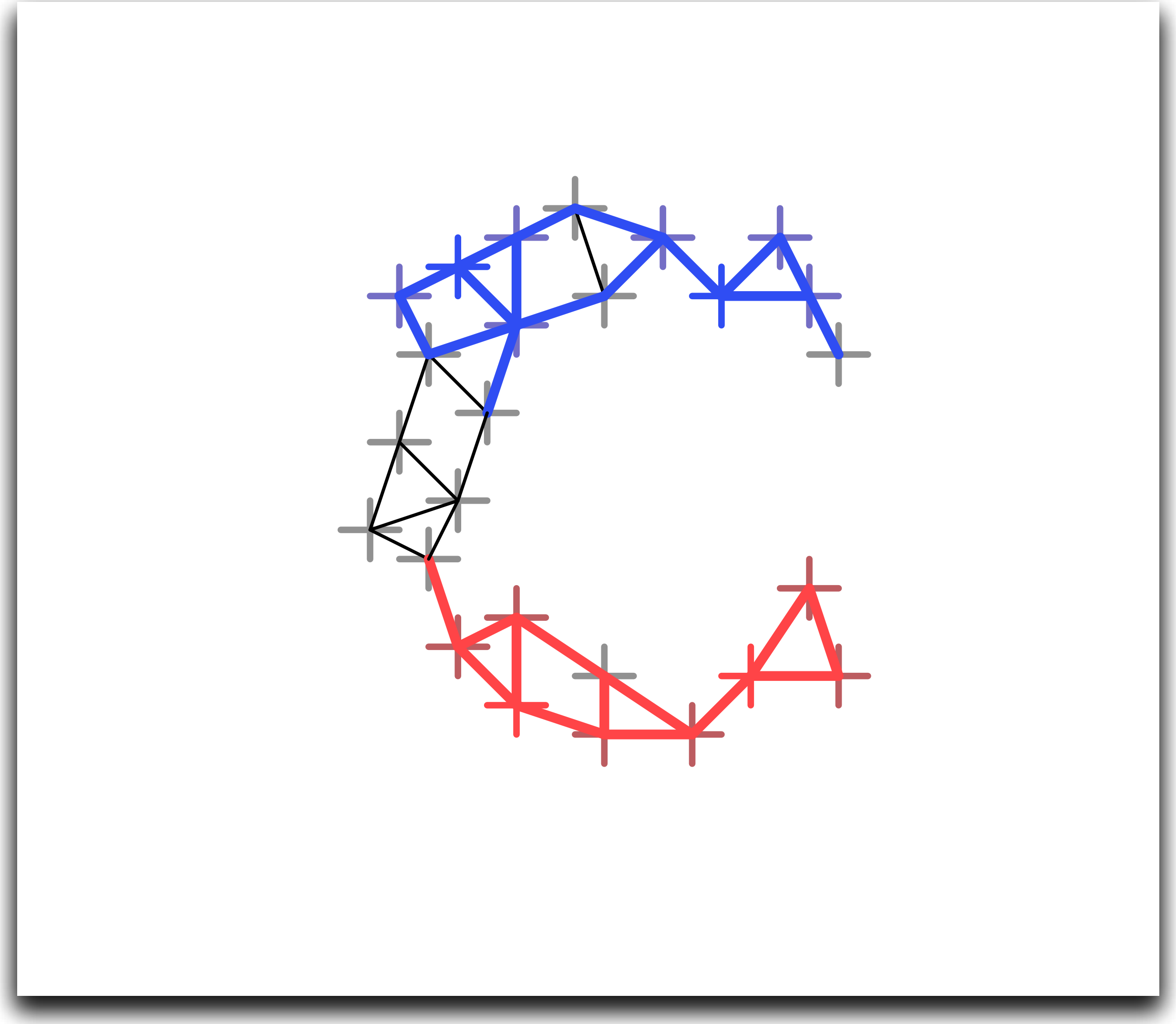
Label Propagation on Hadoop

- Iterative Procedure
  - Mapper: Emit neighbor-label pairs
  - Reducer: Collect incoming labels per node and combine them into new label
  - Repeat until convergence
- Improvement
  - Cluster data and perform within-cluster propagation locally
  - Sometimes it's more efficient to perform matrix inversion instead



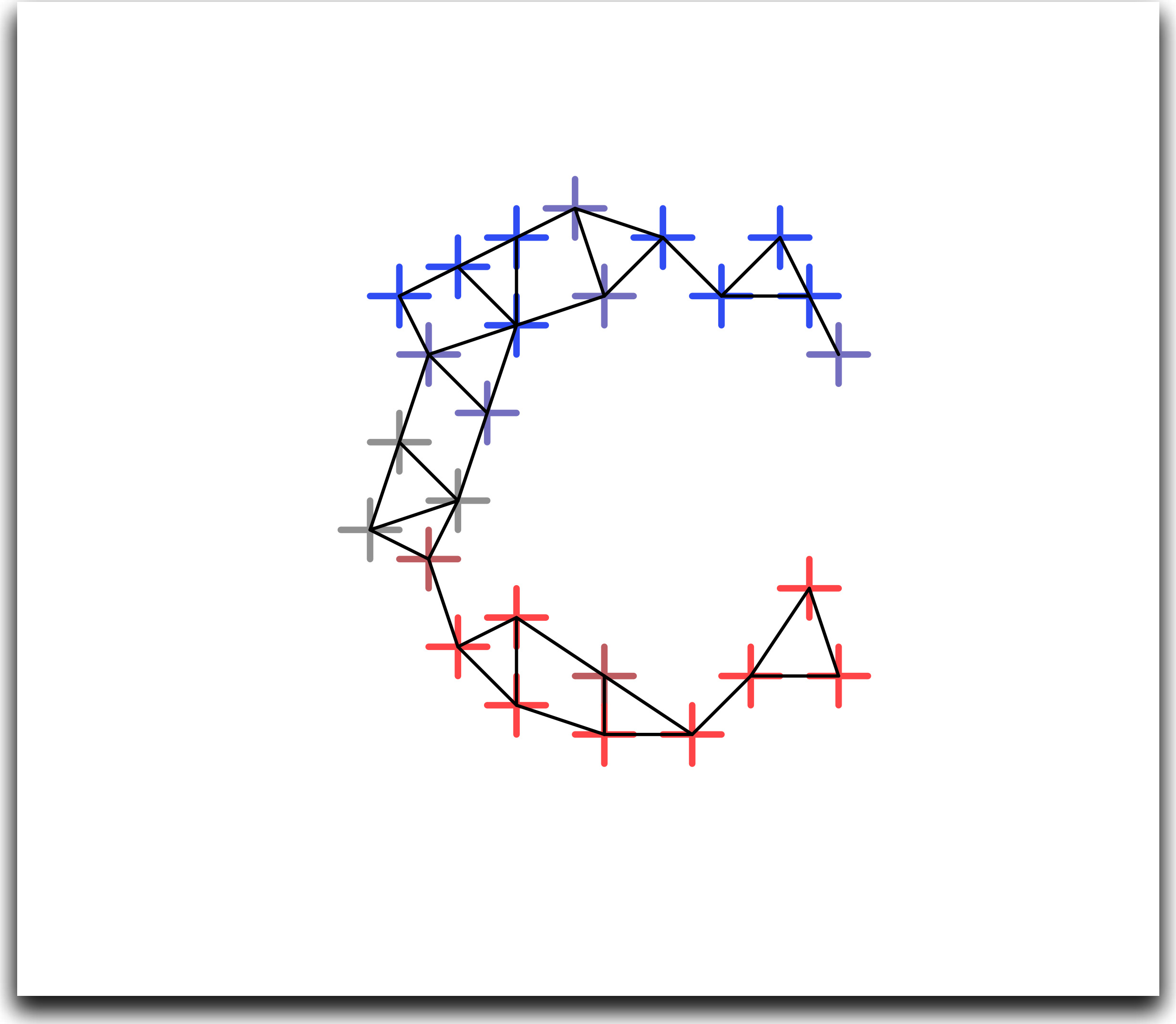
Label Propagation on Hadoop

- Iterative Procedure
  - Mapper: Emit neighbor-label pairs
  - Reducer: Collect incoming labels per node and combine them into new label
  - Repeat until convergence
- Improvement
  - Cluster data and perform within-cluster propagation locally
  - Sometimes it's more efficient to perform matrix inversion instead



Label Propagation on Hadoop

- Iterative Procedure
  - Mapper: Emit neighbor-label pairs
  - Reducer: Collect incoming labels per node and combine them into new label
  - Repeat until convergence
- Improvement
  - Cluster data and perform within-cluster propagation locally
  - Sometimes it's more efficient to perform matrix inversion instead



Conclusion

- Semi-Supervised Learning
  - Only few training instances have labels
  - Unlabeled instances can still provide valuable signal
- Different assumptions lead to different approaches
  - Cluster assumption: generative models
  - Low density assumption: semi-supervised support vector machines
  - Manifold assumption: label propagation
- Code available: https://github.com/Datameer-Inc/lesel

@Datameer