Analysis of SAGE Results with Combined Learning Techniques

Hsuan-Tien Lin and Ling Li

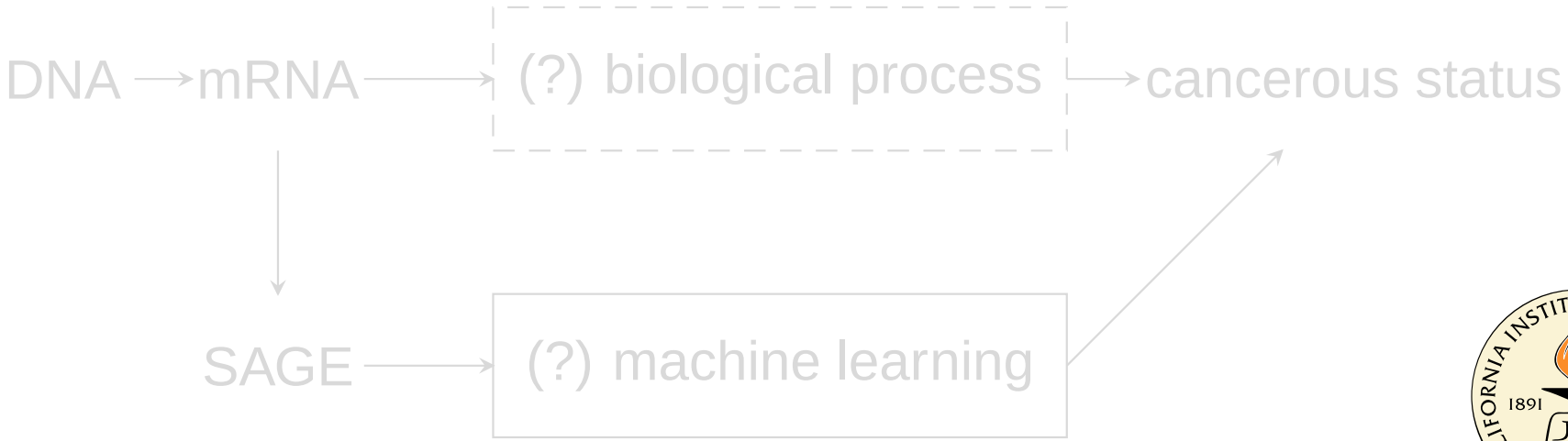
Learning Systems Group, Caltech

ECML/PKDD Discovery Challenge, 2005/10/07

Outline

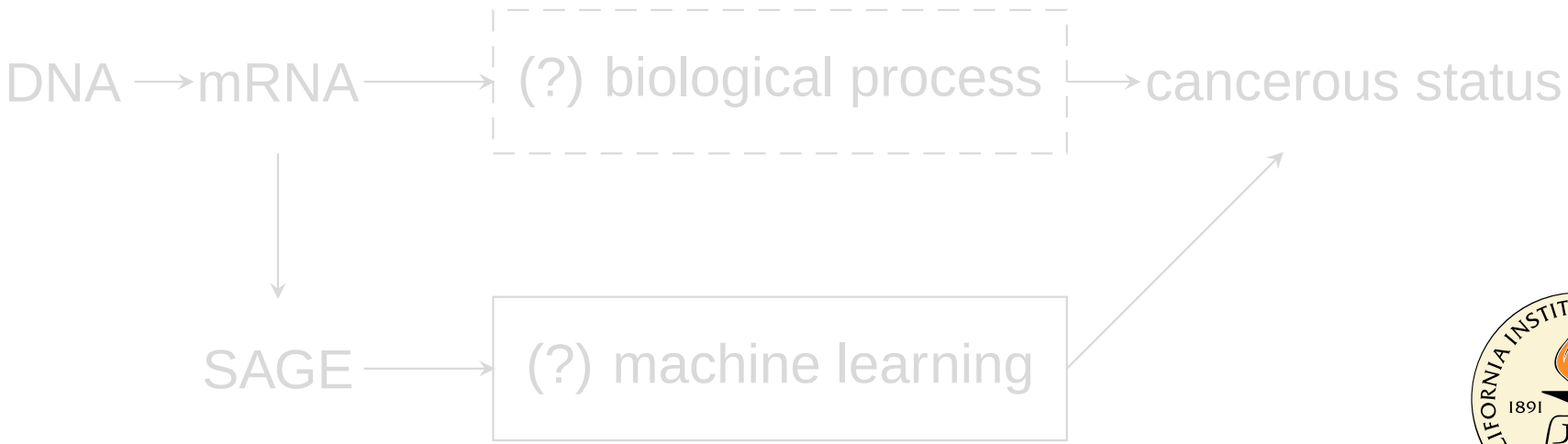
Problem Formulation

- SAGE: serial analysis of gene expressions
- the larger dataset: 90 samples (libraries) x_i, each with 27679 features (counts of SAGE tags) $(x_i)_d$
- labels y_i: 59 cancerous samples, and 31 normal ones
- **can we predict the cancerous status of the sample based on the features given?**



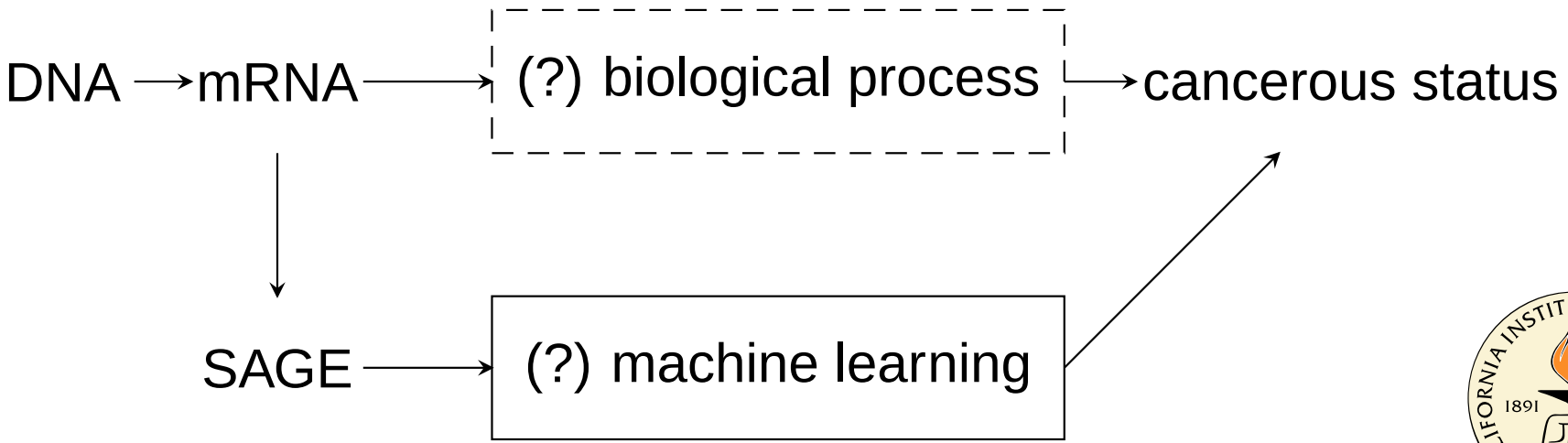
Problem Formulation

- SAGE: serial analysis of gene expressions
- the larger dataset: 90 samples (libraries) x_i, each with 27679 features (counts of SAGE tags) $(x_i)_d$
- labels y_i: 59 cancerous samples, and 31 normal ones
- **can we predict the cancerous status of the sample based on the features given?**



Problem Formulation

- SAGE: serial analysis of gene expressions
- the larger dataset: 90 samples (libraries) x_i, each with 27679 features (counts of SAGE tags) $(x_i)_d$
- labels y_i: 59 cancerous samples, and 31 normal ones
- **can we predict the cancerous status of the sample based on the features given?**



Difficulty of the Problem

- **how to build a classifier for the black box?**
  - many possibilities: linear models, decision trees, classifier ensembles, etc.
  - 27679 features with any models above can usually cover all possible labeling on 90 samples
    – fitting perfectly on 90 samples is as poor as fitting a random labeling
- **should all features be used in the black box?**
  - not all features are useful (Alves et al. 2005)
  - some features may even be misleading
- **how to compare different models?**
  - performance needs to be estimated with unseen samples
  - each sample is a precious one out of 90

Difficulty of the Problem

- **how to build a classifier for the black box?**
  - many possibilities: linear models, decision trees, classifier ensembles, etc.
  - 27679 features with any models above can usually cover all possible labeling on 90 samples
    – fitting perfectly on 90 samples is as poor as fitting a random labeling
- **should all features be used in the black box?**
  - not all features are useful (Alves et al. 2005)
  - some features may even be misleading
- **how to compare different models?**
  - performance needs to be estimated with unseen samples
  - each sample is a precious one out of 90

Difficulty of the Problem

- **how to build a classifier for the black box?**
  - many possibilities: linear models, decision trees, classifier ensembles, etc.
  - 27679 features with any models above can usually cover all possible labeling on 90 samples

    – fitting perfectly on 90 samples is as poor as fitting a random labeling
- **should all features be used in the black box?**
  - not all features are useful (Alves et al. 2005)
  - some features may even be misleading
- **how to compare different models?**
  - performance needs to be estimated with unseen samples
  - each sample is a precious one out of 90

Difficulty of the Problem

- **how to build a classifier for the black box?**
  - many possibilities: linear models, decision trees, classifier ensembles, etc.
  - 27679 features with any models above can usually cover all possible labeling on 90 samples
    – fitting perfectly on 90 samples is as poor as fitting a random labeling
- **should all features be used in the black box?**
  - not all features are useful (Alves et al. 2005)
  - some features may even be misleading
- **how to compare different models?**
  - performance needs to be estimated with unseen samples
  - each sample is a precious one out of 90

Difficulty of the Problem

- **how to build a classifier for the black box?**
  - many possibilities: linear models, decision trees, classifier ensembles, etc.
  - 27679 features with any models above can usually cover all possible labeling on 90 samples
    – fitting perfectly on 90 samples is as poor as fitting a random labeling
- **should all features be used in the black box?**
  - not all features are useful (Alves et al. 2005)
  - some features may even be misleading
- **how to compare different models?**
  - performance needs to be estimated with unseen samples
  - each sample is a precious one out of 90

Difficulty of the Problem

- **how to build a classifier for the black box?**
  - many possibilities: linear models, decision trees, classifier ensembles, etc.
  - 27679 features with any models above can usually cover all possible labeling on 90 samples
    – fitting perfectly on 90 samples is as poor as fitting a random labeling
- **should all features be used in the black box?**
  - not all features are useful (Alves et al. 2005)
  - some features may even be misleading
- **how to compare different models?**
  - performance needs to be estimated with unseen samples
  - each sample is a precious one out of 90

Difficulty of the Problem

- **how to build a classifier for the black box?**
  - many possibilities: linear models, decision trees, classifier ensembles, etc.
  - 27679 features with any models above can usually cover all possible labeling on 90 samples
    – fitting perfectly on 90 samples is as poor as fitting a random labeling
- **should all features be used in the black box?**
  - not all features are useful (Alves et al. 2005)
  - some features may even be misleading
- **how to compare different models?**
  - performance needs to be estimated with unseen samples
  - each sample is a precious one out of 90

Difficulty of the Problem

- **how to build a classifier for the black box?**
  - many possibilities: linear models, decision trees, classifier ensembles, etc.
  - 27679 features with any models above can usually cover all possible labeling on 90 samples
    – fitting perfectly on 90 samples is as poor as fitting a random labeling
- **should all features be used in the black box?**
  - not all features are useful (Alves et al. 2005)
  - some features may even be misleading
- **how to compare different models?**
  - performance needs to be estimated with unseen samples
  - each sample is a precious one out of 90

Difficulty of the Problem

- **how to build a classifier for the black box?**
  - many possibilities: linear models, decision trees, classifier ensembles, etc.
  - 27679 features with any models above can usually cover all possible labeling on 90 samples
    – fitting perfectly on 90 samples is as poor as fitting a random labeling
- **should all features be used in the black box?**
  - not all features are useful (Alves et al. 2005)
  - some features may even be misleading
- **how to compare different models?**
  - performance needs to be estimated with unseen samples
  - each sample is a precious one out of 90

Difficulty of the Problem

- **how to build a classifier for the black box?**
  - many possibilities: linear models, decision trees, classifier ensembles, etc.
  - 27679 features with any models above can usually cover all possible labeling on 90 samples
    – fitting perfectly on 90 samples is as poor as fitting a random labeling
- **should all features be used in the black box?**
  - not all features are useful (Alves et al. 2005)
  - some features may even be misleading
- **how to compare different models?**
  - performance needs to be estimated with unseen samples
  - each sample is a precious one out of 90

“Easiness” of the Problem

- 27679 features give each sample much information
- procedure: feature selection, then train with 89 samples, and test on the other
  - A: feature selection with 89 samples
  - B: feature selection with 90 samples
- B gets a test sample in data “preprocessing.”

how much does an extra sample in the “preprocessing” stage affect the prediction performance?

“Easiness” of the Problem

- 27679 features give each sample much information
- procedure: feature selection, then train with 89 samples, and test on the other
  - A: feature selection with 89 samples
  - B: feature selection with 90 samples
- B gets a test sample in data “preprocessing.”

how much does an extra sample in the “preprocessing” stage affect the prediction performance?

“Easiness” of the Problem

- 27679 features give each sample much information
- procedure: feature selection, then train with 89 samples, and test on the other
  - A: feature selection with 89 samples
  - B: feature selection with 90 samples
- B gets a test sample in data “preprocessing.”

how much does an extra sample in the “preprocessing” stage affect the prediction performance?

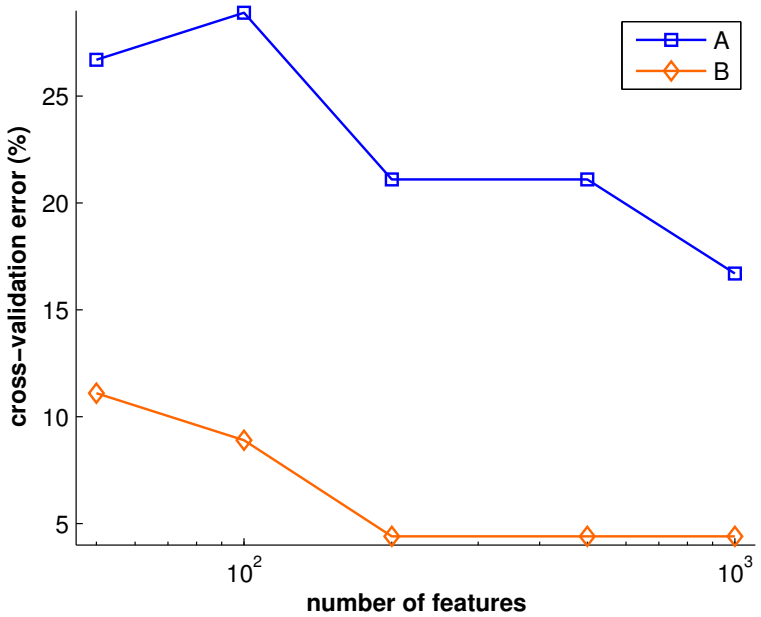
“Easiness” of the Problem

- 27679 features give each sample much information
- procedure: feature selection, then train with 89 samples, and test on the other
  - A: feature selection with 89 samples
  - B: feature selection with 90 samples
- B gets a test sample in data “preprocessing.”

how much does an extra sample in the “preprocessing” stage affect the prediction performance?

“Easiness” of the Problem

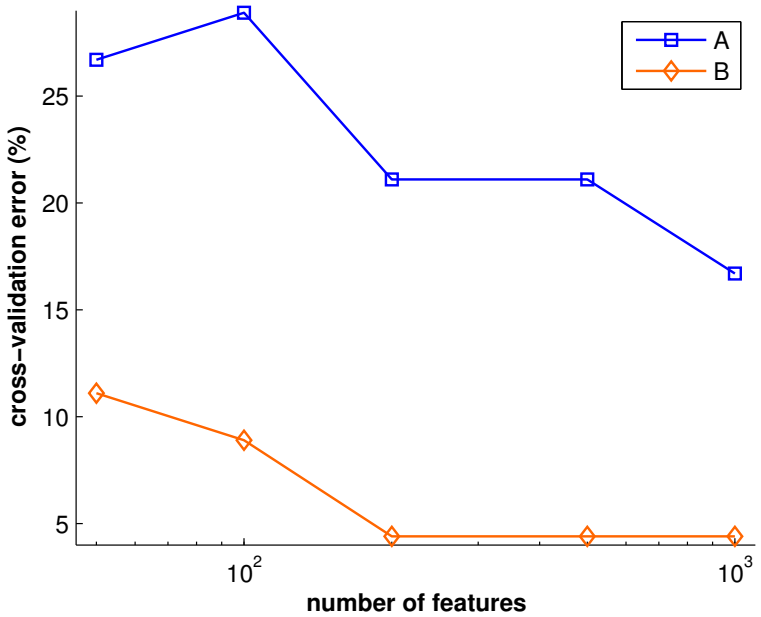
- procedure: feature selection, then train with 89, test on the other
  - A: feature selection with 89 samples
  - B: feature selection with 90 samples
- B is significantly biased towards the single sample



1. any piece of information can affect the result dramatically
2. careful NOT to look at any test information

“Easiness” of the Problem

- procedure: feature selection, then train with 89, test on the other
  - A: feature selection with 89 samples
  - B: feature selection with 90 samples
- B is significantly biased towards the single sample



1. any piece of information can affect the result dramatically
2. careful NOT to look at any test information

Our Approach of Analysis

- combination of classification, feature selection, and error estimation techniques
- use different combinations to show the relative usefulness of different techniques
- systematic and repeatable on similar datasets
- careful use of unseen samples
- robust conclusion with multiple combinations and error estimations

Our Approach of Analysis

- combination of classification, feature selection, and error estimation techniques
- use different combinations to show the relative usefulness of different techniques
- systematic and repeatable on similar datasets
- careful use of unseen samples
- robust conclusion with multiple combinations and error estimations

Our Approach of Analysis

- combination of classification, feature selection, and error estimation techniques
- use different combinations to show the relative usefulness of different techniques
- systematic and repeatable on similar datasets
- careful use of unseen samples
- robust conclusion with multiple combinations and error estimations

Our Approach of Analysis

- combination of classification, feature selection, and error estimation techniques
- use different combinations to show the relative usefulness of different techniques
- systematic and repeatable on similar datasets
- careful use of unseen samples
- robust conclusion with multiple combinations and error estimations

Our Approach of Analysis

- combination of classification, feature selection, and error estimation techniques
- use different combinations to show the relative usefulness of different techniques
- systematic and repeatable on similar datasets
- careful use of unseen samples
- robust conclusion with multiple combinations and error estimations

Classification Techniques

- techniques that avoid overfitting
- models that seem promising
- four classification algorithms
  - AdaBoost-Stump
  - SVM-Linear
  - SVM-Gaussian
  - SVM-Stump
    – a novel and promising paradigm through **infinite ensemble learning** (Lin and Li, ECML 2005)

Adaptive Boosting with Decision Stumps

- model:

$$\hat{g}(x) = \text{sign}\left(\sum_{t=1}^{T} w_t s_t(x)\right)$$

- a finite ensemble of weak rules
- each s_t is a decision stump (thresholding rule on a SAGE tag) – e.g. if the count of the tag 200 greater than 10, then cancerous
- each w_t: a nonnegative weight for s_t
- prediction: each s_t tells whether the sample is cancerous, and $\hat{g}$ reports the majority of weighted votes
- automatically selects $\leq T$ important tags and ignore others in prediction

Adaptive Boosting with Decision Stumps

- model:

$$\hat{g}(x) = \text{sign}\left(\sum_{t=1}^{T} w_t s_t(x)\right)$$

- a finite ensemble of weak rules
- each s_t is a decision stump (thresholding rule on a SAGE tag)
  – e.g. if the count of the tag 200 greater than 10, then cancerous
- each w_t: a nonnegative weight for s_t
- prediction: each s_t tells whether the sample is cancerous, and $\hat{g}$ reports the majority of weighted votes
- automatically selects $\leq T$ important tags and ignore others in prediction

Adaptive Boosting with Decision Stumps

- model:

$$\hat{g}(x) = \text{sign}\left(\sum_{t=1}^{T} w_t s_t(x)\right)$$

- a finite ensemble of weak rules
- each s_t is a decision stump (thresholding rule on a SAGE tag)
  – e.g. if the count of the tag 200 greater than 10, then cancerous
- each w_t: a nonnegative weight for s_t
- prediction: each s_t tells whether the sample is cancerous, and $\hat{g}$ reports the majority of weighted votes
- automatically selects $\leq T$ important tags and ignore others in prediction

Adaptive Boosting with Decision Stumps

- model:

$$\hat{g}(x) = \text{sign}\left(\sum_{t=1}^{T} w_t s_t(x)\right)$$

- a finite ensemble of weak rules
- each s_t is a decision stump (thresholding rule on a SAGE tag)
  – e.g. if the count of the tag 200 greater than 10, then cancerous
- each w_t: a nonnegative weight for s_t
- prediction: each s_t tells whether the sample is cancerous, and $\hat{g}$ reports the majority of weighted votes
- automatically selects $\leq T$ important tags and ignore others in prediction

Support Vector Machine with Linear Kernel

- model:

$$\hat{g}(x) = \text{sign}\left(\sum_{d=1}^{D} w_d (x)_d + b\right)$$

- a hyperplane in $\mathbb{R}^D$
  – e.g. if the weighted sum of all counts is greater than 10, then cancerous
- a large-margin hyperplane: clear separation between cancerous and normal samples
- each w_d: sensitivity for change of $(x)_d$
  – measure of the importance of tag d

Support Vector Machine with Linear Kernel

- model:

$$\hat{g}(x) = \text{sign}\left(\sum_{d=1}^{D} w_d (x)_d + b\right)$$

- a hyperplane in $\mathbb{R}^D$
  – e.g. if the weighted sum of all counts is greater than 10, then cancerous
- a large-margin hyperplane: clear separation between cancerous and normal samples
- each w_d: sensitivity for change of $(x)_d$
  – measure of the importance of tag d

Support Vector Machine with Linear Kernel

- model:

$$\hat{g}(x) = \text{sign}\left(\sum_{d=1}^{D} w_d (x)_d + b\right)$$

- a hyperplane in $\mathbb{R}^D$
  – e.g. if the weighted sum of all counts is greater than 10, then cancerous
- a large-margin hyperplane: clear separation between cancerous and normal samples
- each w_d: sensitivity for change of $(x)_d$
  – measure of the importance of tag d

Support Vector Machine with Gaussian Kernel

- model:

$$\hat{g}(x) = \text{sign}\left(\sum_{i=1}^{N} y_i \lambda_i \exp(-\gamma(x - x_i)^2)\right)$$

- a nonlinear classifier, similar to a radial basis function network
- large-margin hyperplane in an infinite dimensional space
- pros: powerful model, often good prediction performance
- cons: time-consuming to choose parameter γ, hard to interpret

Support Vector Machine with Gaussian Kernel

- model:

$$\hat{g}(x) = \text{sign}\left(\sum_{i=1}^{N} y_i \lambda_i \exp(-\gamma(x - x_i)^2)\right)$$

- a nonlinear classifier, similar to a radial basis function network
- large-margin hyperplane in an infinite dimensional space
- pros: powerful model, often good prediction performance
- cons: time-consuming to choose parameter γ, hard to interpret

Support Vector Machine with Gaussian Kernel

- model:

$$\hat{g}(x) = \text{sign}\left(\sum_{i=1}^{N} y_i \lambda_i \exp(-\gamma(x - x_i)^2)\right)$$

- a nonlinear classifier, similar to a radial basis function network
- large-margin hyperplane in an infinite dimensional space
- pros: powerful model, often good prediction performance
- cons: time-consuming to choose parameter γ, hard to interpret

Support Vector Machine with Stump Kernel

- model:

$$\hat{g}(x) = \text{sign}\left(\sum_{d=1}^{D}\sum_{q\in\pm 1}\left(\int w_{q,d}(\alpha)s_{q,d,\alpha}(x)\,d\alpha\right) + b\right)$$

- large-margin infinite ensemble of decision stumps: novel and promising
- pros: powerful model, often good performance
- superior power to AdaBoost-Stump due to infinity
- superior power to SVM-Linear due to nonlinearity
- faster parameter selection than SVM-Gauss
- model: partially interpreted
  – $w_{q,d}$ can estimate the importance of tag d

Support Vector Machine with Stump Kernel

- model:

$$\hat{g}(x) = \text{sign}\left(\sum_{d=1}^{D}\sum_{q\in\pm 1}\left(\int w_{q,d}(\alpha)s_{q,d,\alpha}(x)\,d\alpha\right) + b\right)$$

- large-margin infinite ensemble of decision stumps: novel and promising
- pros: powerful model, often good performance
- superior power to AdaBoost-Stump due to infinity
- superior power to SVM-Linear due to nonlinearity
- faster parameter selection than SVM-Gauss
- model: partially interpreted
  - $w_{q,d}$ can estimate the importance of tag d

Support Vector Machine with Stump Kernel

- model:

$$\hat{g}(x) = \text{sign}\left(\sum_{d=1}^{D}\sum_{q\in\pm 1}\left(\int w_{q,d}(\alpha)s_{q,d,\alpha}(x)\,d\alpha\right) + b\right)$$

- large-margin infinite ensemble of decision stumps: novel and promising
- pros: powerful model, often good performance
- superior power to AdaBoost-Stump due to infinity
- superior power to SVM-Linear due to nonlinearity
- faster parameter selection than SVM-Gauss
- model: partially interpreted
  - $w_{q,d}$ can estimate the importance of tag d

Support Vector Machine with Stump Kernel

- model:

$$\hat{g}(x) = \text{sign}\left(\sum_{d=1}^{D}\sum_{q\in\pm 1}\left(\int w_{q,d}(\alpha)s_{q,d,\alpha}(x)\,d\alpha\right) + b\right)$$

- large-margin infinite ensemble of decision stumps: novel and promising
- pros: powerful model, often good performance
- superior power to AdaBoost-Stump due to infinity
- superior power to SVM-Linear due to nonlinearity
- faster parameter selection than SVM-Gauss
- model: partially interpreted
  – $w_{q,d}$ can estimate the importance of tag d

Relative Comparison of Classification Techniques

- all four have some degree of regularization: avoid overfitting
- the first three were used in some gene/cancer related tasks
- SVM-Stump is closely related to AdaBoost-Stump
- pros and cons:

	AdaBoost -Stump	SVM -Linear	SVM -Gauss	SVM -Stump
model power(*)	–	–	↑	↑
interpretability	↑	↑	↓	–
speed	↑	–	↓	–

(*) it is hard to compare AdaBoost-Stump to SVM-Linear in power

Relative Comparison of Classification Techniques

- all four have some degree of regularization: avoid overfitting
- the first three were used in some gene/cancer related tasks
- SVM-Stump is closely related to AdaBoost-Stump
- pros and cons:

	AdaBoost -Stump	SVM -Linear	SVM -Gauss	SVM -Stump
model power(*)	–	–	↑	↑
interpretability	↑	↑	↓	–
speed	↑	–	↓	–

(*) it is hard to compare AdaBoost-Stump to SVM-Linear in power

Relative Comparison of Classification Techniques

- all four have some degree of regularization: avoid overfitting
- the first three were used in some gene/cancer related tasks
- SVM-Stump is closely related to AdaBoost-Stump
- pros and cons:

	AdaBoost -Stump	SVM -Linear	SVM -Gauss	SVM -Stump
model power(*)	–	–	↑	↑
interpretability	↑	↑	↓	–
speed	↑	–	↓	–

(*) it is hard to compare AdaBoost-Stump to SVM-Linear in power

Feature Selection with Ranking

Algorithm

1. rank (order) the features by their importance
2. select only the top M features

- a simple strategy
- relies on a good ranking algorithm
- three simple ranking algorithms:
  - Ranking with Fisher Score
  - Ranking with Linear Weight
  - Ranking with Stump Weight
- the first two have been used in similar tasks

Feature Selection with Ranking

Algorithm

1. rank (order) the features by their importance
2. select only the top M features

- a simple strategy
- relies on a good ranking algorithm
- three simple ranking algorithms:
  - Ranking with Fisher Score
  - Ranking with Linear Weight
  - Ranking with Stump Weight
- the first two have been used in similar tasks

Feature Ranking Techniques

- Rank with Fisher Score (RFS):
  how well can we use only $(x_i)_d$ to predict y_i?
- Rank with Linear Weight (RLW):
  what is the importance w_d of $(x)_d$ in the hyperplane

$$\sum w_d (x)_d + b$$

  found by SVM-Linear?
- Rank with Stump Weight (RSW):
  what is the amount of decision stumps $\sum_q \int w_{q,d}^2(\alpha)\, d\alpha$ needed for feature d in the ensemble

$$\sum_{d=1}^{D} \sum_{q \in \pm 1} \left(\int w_{q,d}(\alpha) s_{q,d,\alpha}(x)\, d\alpha \right) + b$$

  found by SVM-Stump?

Feature Ranking Techniques

- Rank with Fisher Score (RFS):
  how well can we use only $(x_i)_d$ to predict y_i?
- Rank with Linear Weight (RLW):
  what is the importance w_d of $(x)_d$ in the hyperplane

$$\sum w_d (x)_d + b$$

  found by SVM-Linear?
- Rank with Stump Weight (RSW):
  what is the amount of decision stumps $\sum_q \int w_{q,d}^2(\alpha)\, d\alpha$ needed for feature d in the ensemble

$$\sum_{d=1}^{D} \sum_{q \in \pm 1} \left(\int w_{q,d}(\alpha) s_{q,d,\alpha}(x)\, d\alpha \right) + b$$

  found by SVM-Stump?

Feature Ranking Techniques

- Rank with Fisher Score (RFS):
  how well can we use only $(x_i)_d$ to predict y_i?
- Rank with Linear Weight (RLW):
  what is the importance w_d of $(x)_d$ in the hyperplane

$$\sum w_d (x)_d + b$$

  found by SVM-Linear?
- Rank with Stump Weight (RSW):
  what is the amount of decision stumps $\sum_q \int w_{q,d}^2(\alpha)\, d\alpha$ needed for feature d in the ensemble

$$\sum_{d=1}^{D} \sum_{q \in \pm 1} \left(\int w_{q,d}(\alpha) s_{q,d,\alpha}(x)\, d\alpha \right) + b$$

  found by SVM-Stump?

Error Estimation Techniques

- v-fold cross-validation: economic use of samples
- training folds: $v - 1$ of the v folds
- test fold: the other folds is reserved unseen
- estimate: average error on the reduced test fold
- v-fold CV is a random process: can be repeated many times
- our setting: 10 fold $\times$10, 5 fold $\times$20, or 90 fold $\times$1
- 90 fold: also called leave-one-out

Error Estimation Techniques

- v-fold cross-validation: economic use of samples
- training folds: $v - 1$ of the v folds
- test fold: the other folds is reserved unseen
- estimate: average error on the reduced test fold
- v-fold CV is a random process: can be repeated many times
- our setting: 10 fold $\times$10, 5 fold $\times$20, or 90 fold $\times$1
- 90 fold: also called leave-one-out

Error Estimation Techniques

- v-fold cross-validation: economic use of samples
- training folds: $v - 1$ of the v folds
- test fold: the other folds is reserved unseen
- estimate: average error on the reduced test fold
- v-fold CV is a random process: can be repeated many times
- our setting: 10 fold $\times$10, 5 fold $\times$20, or 90 fold $\times$1
- 90 fold: also called leave-one-out

Error Estimation Techniques

- v-fold cross-validation: economic use of samples
- training folds: $v - 1$ of the v folds
- test fold: the other folds is reserved unseen
- estimate: average error on the reduced test fold
- v-fold CV is a random process: can be repeated many times
- our setting: 10 fold $\times$10, 5 fold $\times$20, or 90 fold $\times$1
- 90 fold: also called leave-one-out

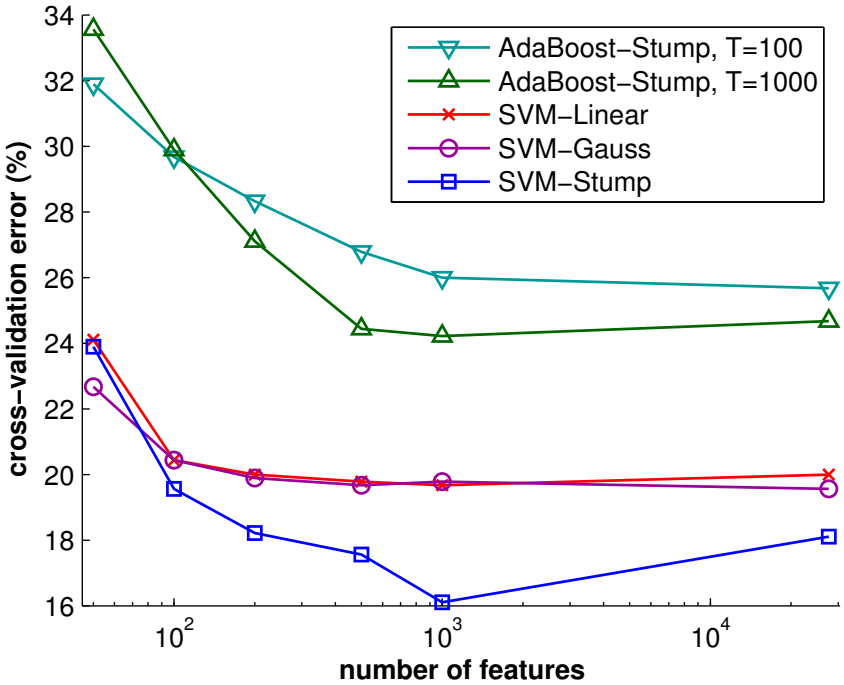
Experiment Settings

Experiment Setting

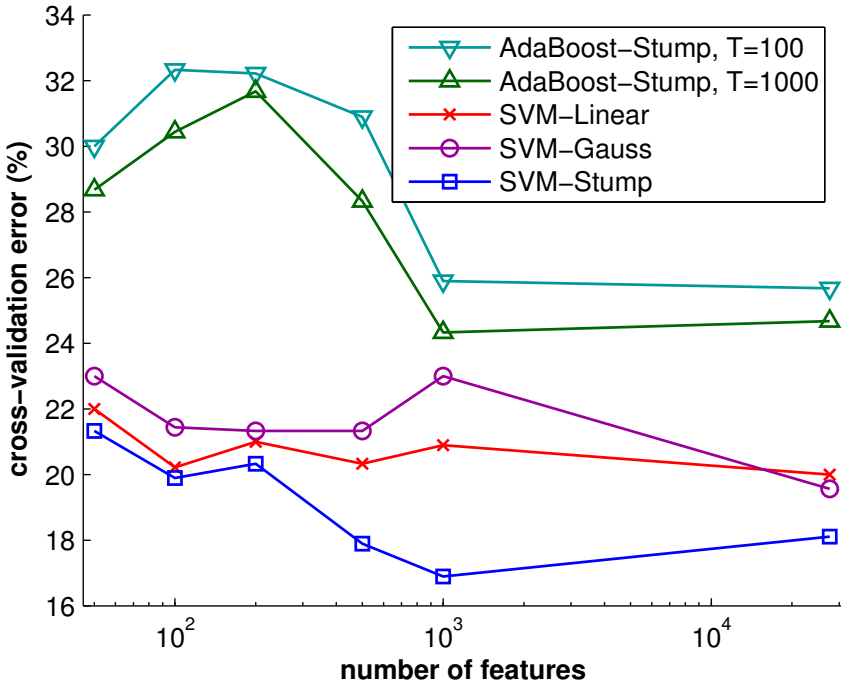
1. Cross-validation splitting to training folds/test fold
2. Feature ranking on training folds
3. Feature selection by ranking $(50, 100, 200, 500, 1000, 27679)$
4. Classification on the reduced training folds
5. Test on the reduced test fold

Comparison of Classification Techniques

Ranking with Linear Weight



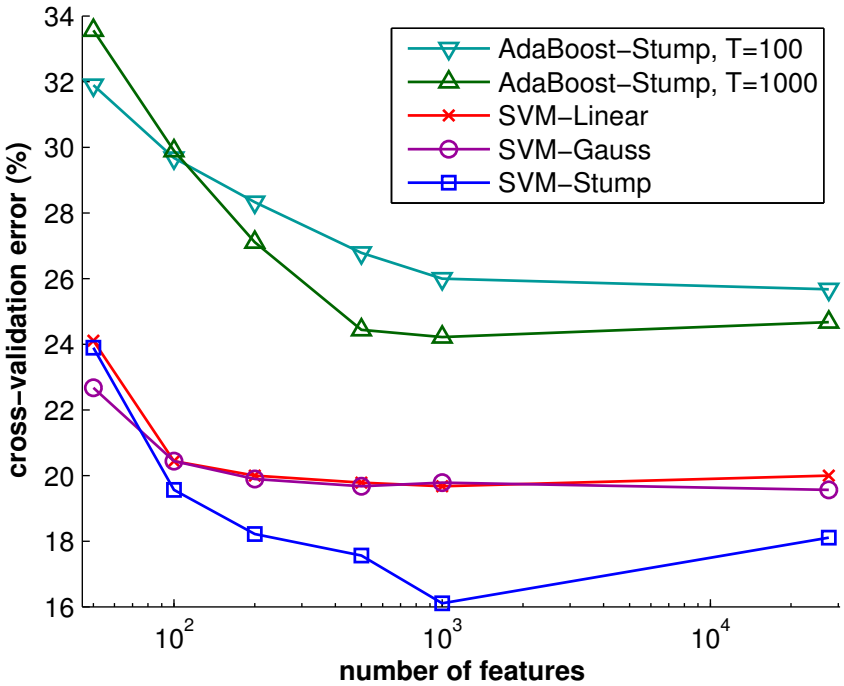
Ranking with Stump Weight



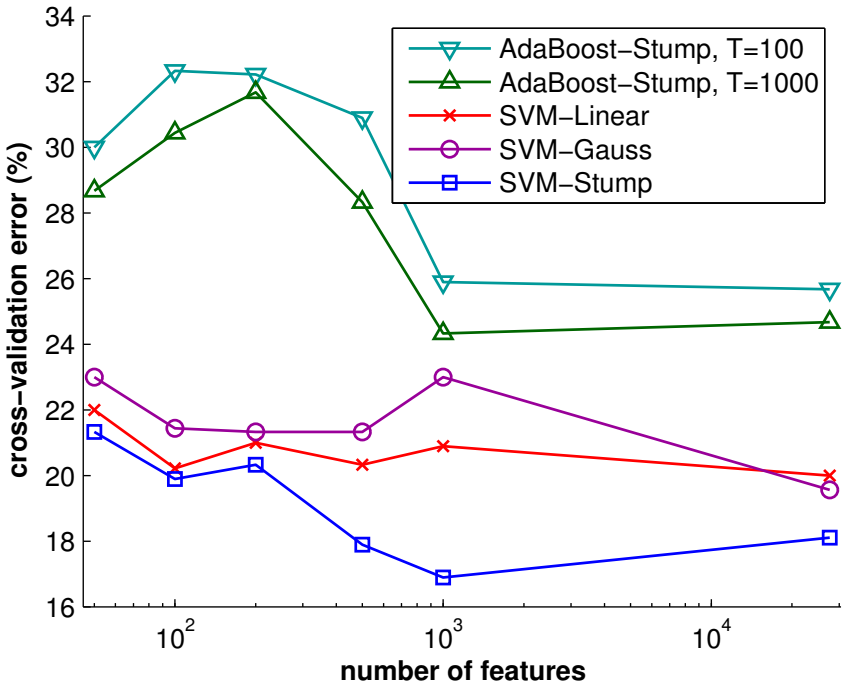
- results with 10 fold CV $\times 10$
- AdaBoost-Stump is not good
- SVM-Gauss is slightly worse than SVM-Linear
- SVM-Stump is slightly better than SVM-Linear

Comparison of Classification Techniques

Ranking with Linear Weight



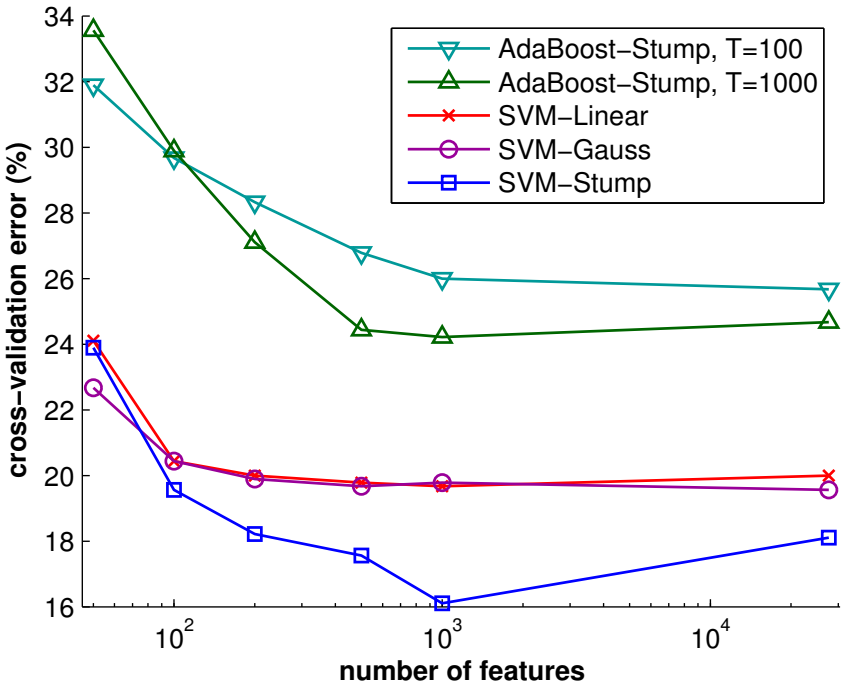
Ranking with Stump Weight



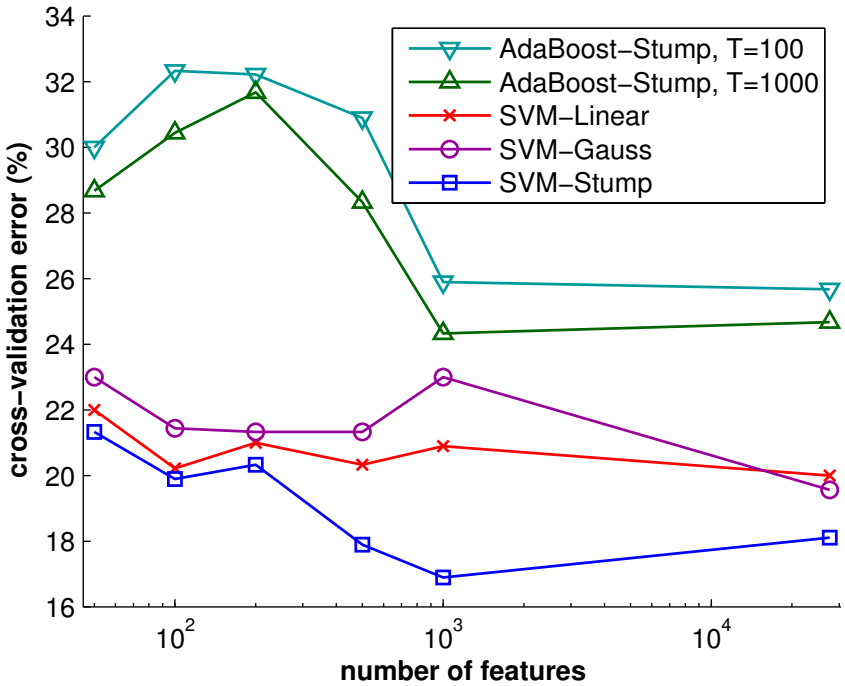
- results with 10 fold CV $\times 10$
- AdaBoost-Stump is not good
- SVM-Gauss is slightly worse than SVM-Linear
- SVM-Stump is slightly better than SVM-Linear

Comparison of Classification Techniques

Ranking with Linear Weight



Ranking with Stump Weight



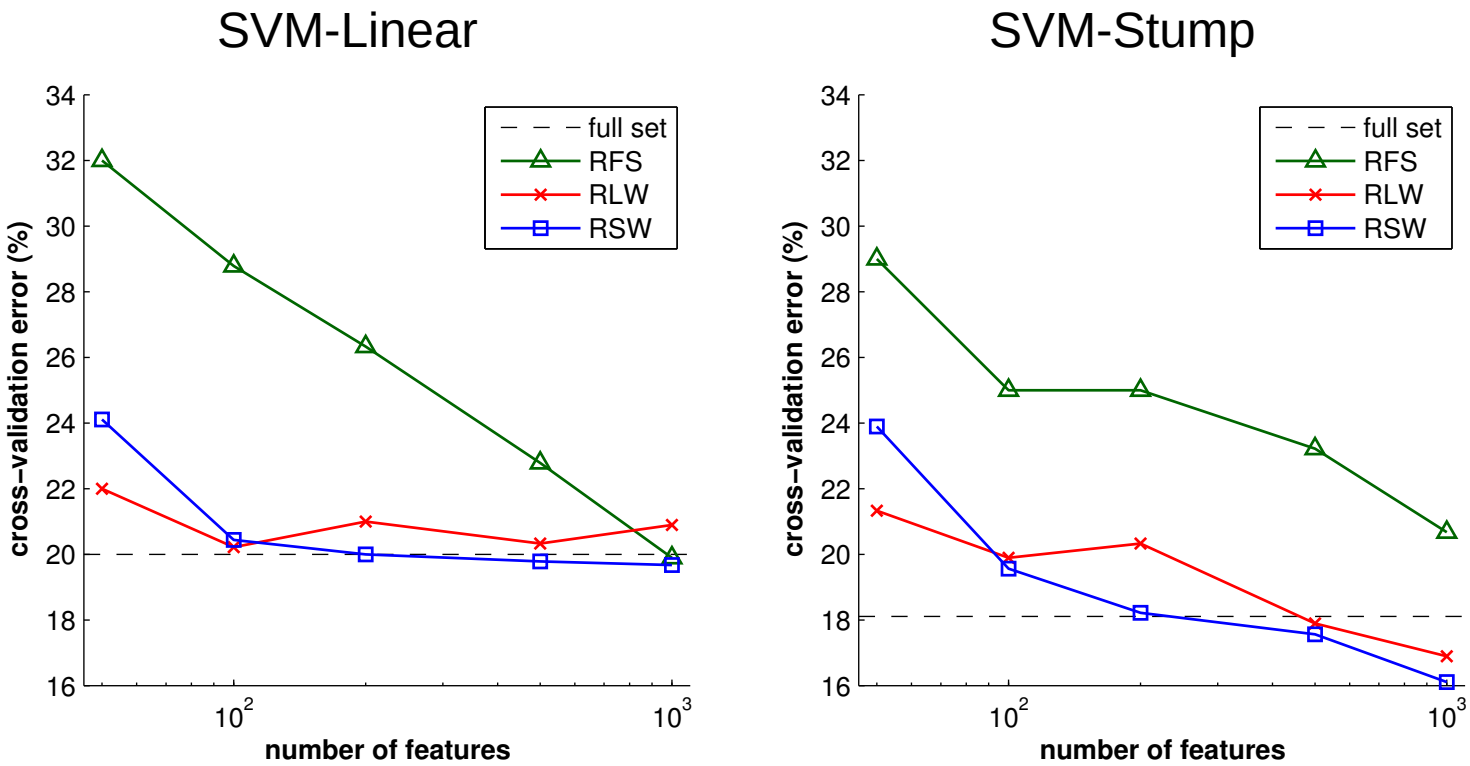
- results with 10 fold CV $\times 10$
- AdaBoost-Stump is not good
- SVM-Gauss is slightly worse than SVM-Linear
- SVM-Stump is slightly better than SVM-Linear

Comparison of Classification Techniques

SVM-Linear and SVM-Stump are the better choices

	AdaBoost -Stump	SVM -Linear	SVM -Gauss	SVM -Stump
model power	–	–	↑	↑
interpretability	↑	↑	↓	–
speed	↑	–	↓	–
performance	↓	↑	↑	↑

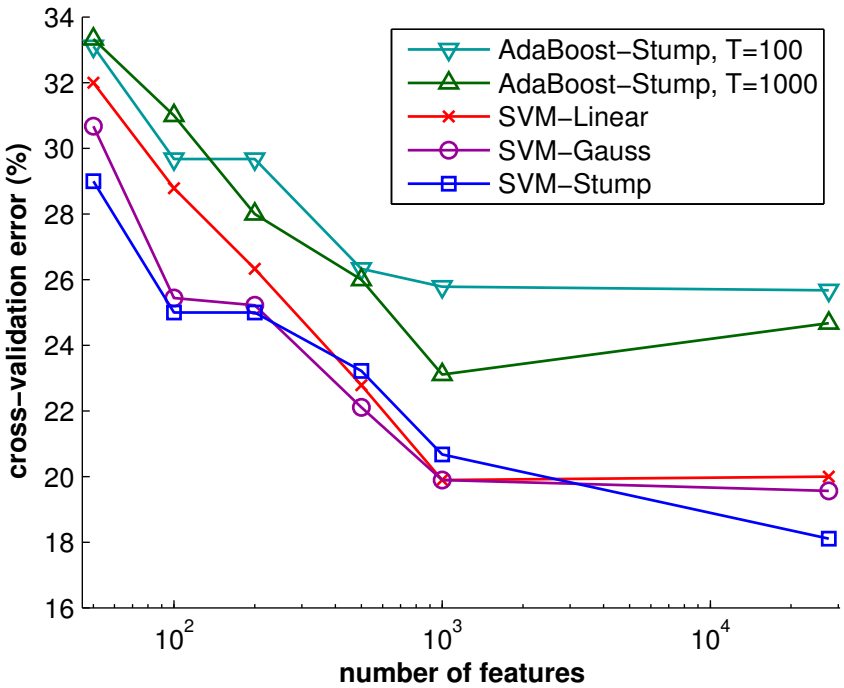
Comparison of Feature Selection Techniques



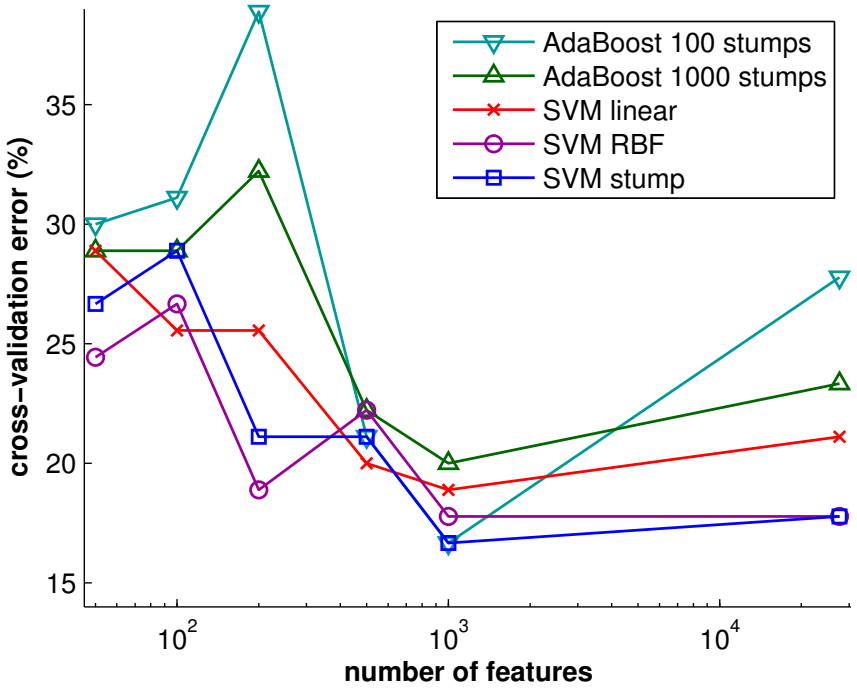
- results with 10 fold CV $\times 10$
- Ranking with F-Score is not good
- Ranking with Stump Weight is slightly better than with Linear Weight

Comparison of Error Estimation Techniques

Ranking with F-Score (10 fold $\times$ 10)



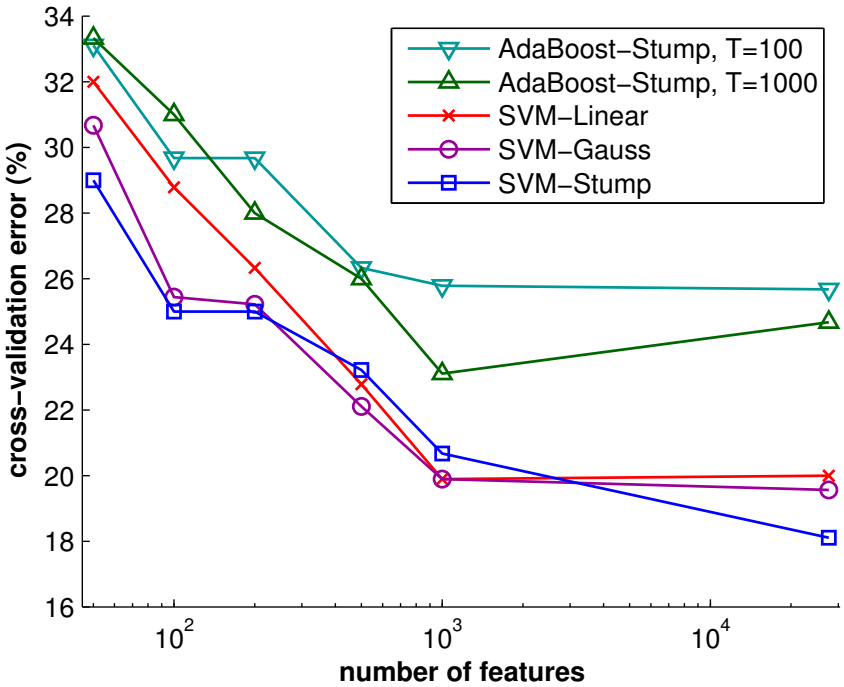
Ranking with F-Score (90 fold)



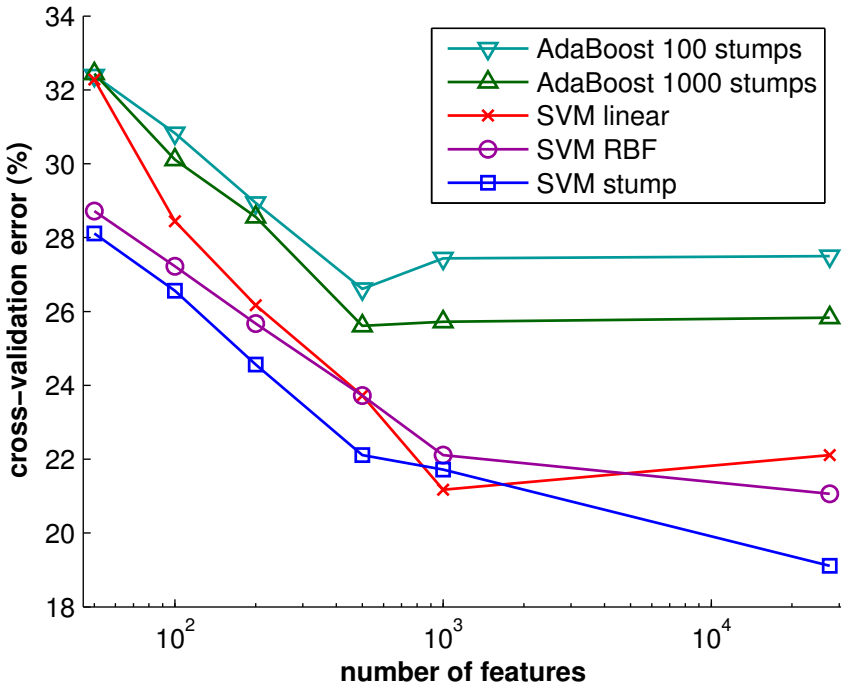
- leave-one-out does not give stable and explainable results

Comparison of Error Estimation Techniques

Ranking with F-Score (10 fold $\times$ 10)



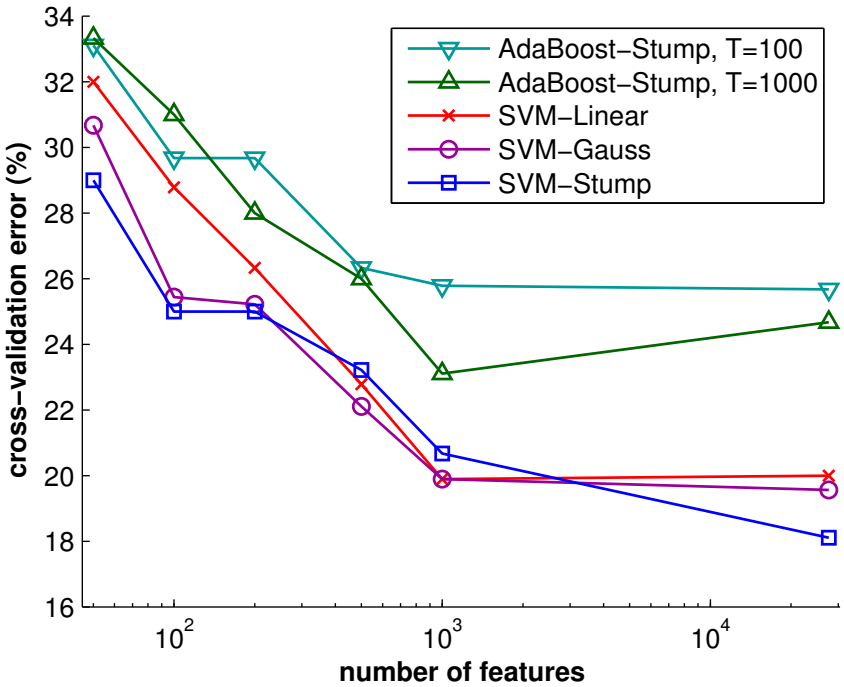
Ranking with F-Score (5 fold $\times$ 20)



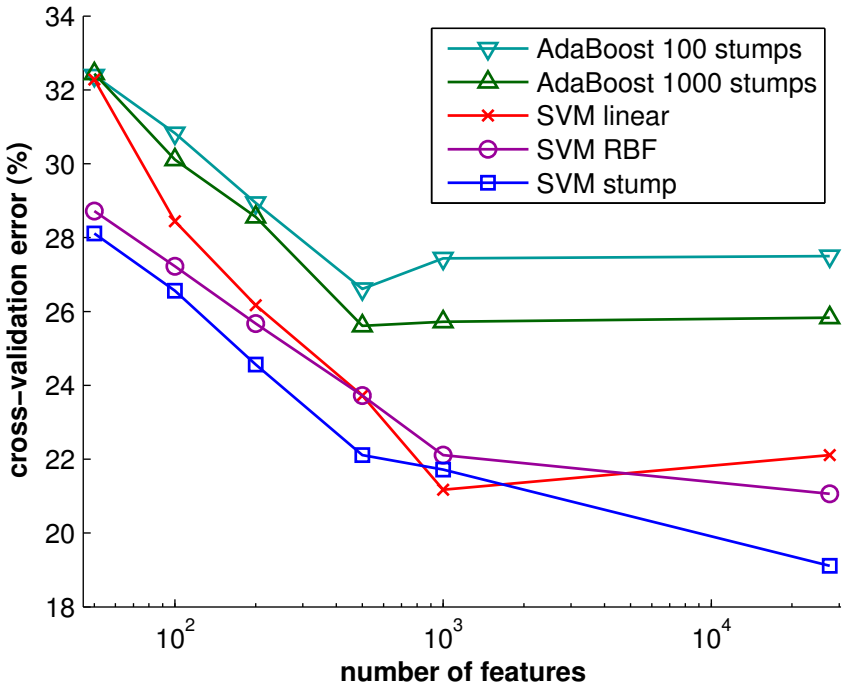
- similar conclusions from 5 fold and 10 fold CV
- 10-fold uses more samples for training
  – better choice considering the importance of samples

Comparison of Error Estimation Techniques

Ranking with F-Score (10 fold $\times$ 10)



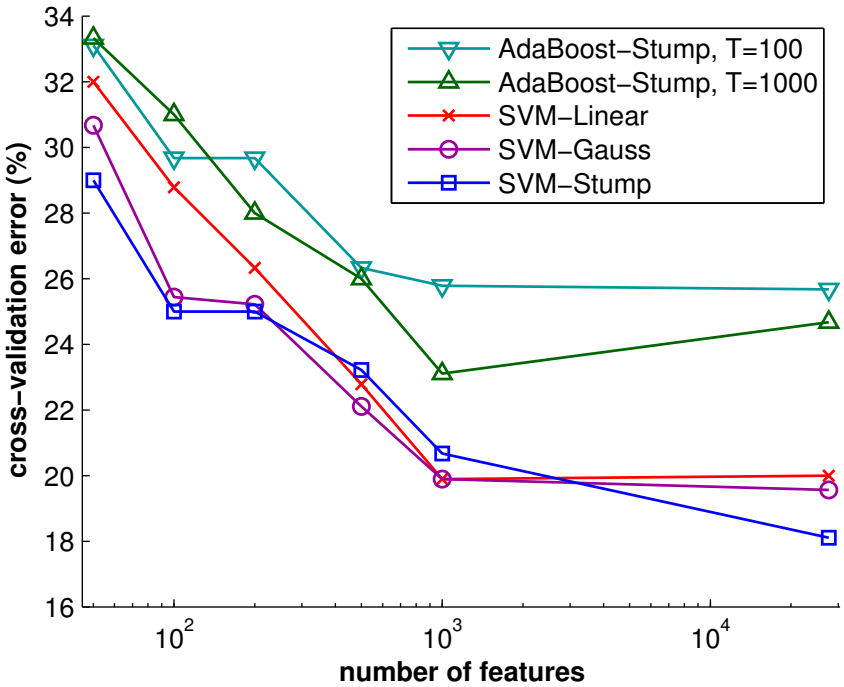
Ranking with F-Score (5 fold $\times$ 20)



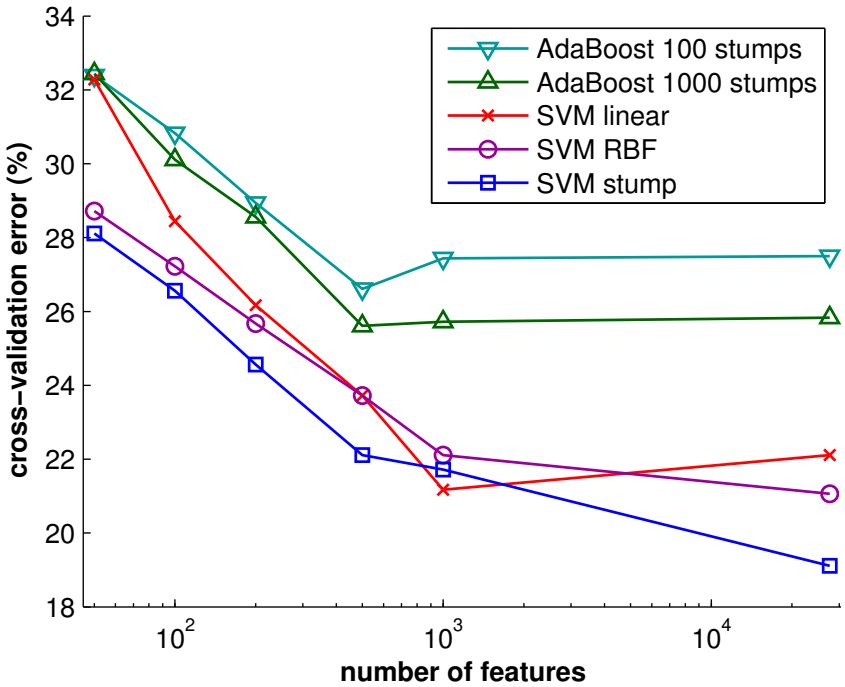
- similar conclusions from 5 fold and 10 fold CV
- 10-fold uses more samples for training
  – better choice considering the importance of samples

Comparison of Error Estimation Techniques

Ranking with F-Score (10 fold $\times$ 10)



Ranking with F-Score (5 fold $\times$ 20)



- similar conclusions from 5 fold and 10 fold CV
- 10-fold uses more samples for training
  – better choice considering the importance of samples

Conclusion

- **carefully analyzed the difficult SAGE dataset**
  - legitimate information only
  - robust conclusion through multiple testing
- classification: SVM-Linear and SVM-Stump are both promising
- feature selection: RLW and RSW are both good
  – possible to achieve better performance than full set
- error estimation: 10-fold CV seems to be a better choice and leave-one-out is bad
- how can we possibly distinguish between the linear model and the stump ensemble model?
  - are there more samples to verify the findings?
  - which model selects more biologically meaningful features?
  - which model is biologically more plausible?

Conclusion

- **carefully analyzed the difficult SAGE dataset**
  - legitimate information only
  - robust conclusion through multiple testing
- classification: SVM-Linear and SVM-Stump are both promising
- feature selection: RLW and RSW are both good
  – possible to achieve better performance than full set
- error estimation: 10-fold CV seems to be a better choice and leave-one-out is bad
- how can we possibly distinguish between the linear model and the stump ensemble model?
  - are there more samples to verify the findings?
  - which model selects more biologically meaningful features?
  - which model is biologically more plausible?

Conclusion

- **carefully analyzed the difficult SAGE dataset**
  - legitimate information only
  - robust conclusion through multiple testing
- classification: SVM-Linear and SVM-Stump are both promising
- feature selection: RLW and RSW are both good
  – possible to achieve better performance than full set
- error estimation: 10-fold CV seems to be a better choice and leave-one-out is bad
- how can we possibly distinguish between the linear model and the stump ensemble model?
  - are there more samples to verify the findings?
  - which model selects more biologically meaningful features?
  - which model is biologically more plausible?

Conclusion

- **carefully analyzed the difficult SAGE dataset**
  - legitimate information only
  - robust conclusion through multiple testing
- classification: SVM-Linear and SVM-Stump are both promising
- feature selection: RLW and RSW are both good
  – possible to achieve better performance than full set
- error estimation: 10-fold CV seems to be a better choice and leave-one-out is bad
- how can we possibly distinguish between the linear model and the stump ensemble model?
  - are there more samples to verify the findings?
  - which model selects more biologically meaningful features?
  - which model is biologically more plausible?

Conclusion

- **carefully analyzed the difficult SAGE dataset**
  - legitimate information only
  - robust conclusion through multiple testing
- classification: SVM-Linear and SVM-Stump are both promising
- feature selection: RLW and RSW are both good
  – possible to achieve better performance than full set
- error estimation: 10-fold CV seems to be a better choice and leave-one-out is bad
- how can we possibly distinguish between the linear model and the stump ensemble model?
  - are there more samples to verify the findings?
  - which model selects more biologically meaningful features?
  - which model is biologically more plausible?

Conclusion

- **carefully analyzed the difficult SAGE dataset**
  - legitimate information only
  - robust conclusion through multiple testing
- classification: SVM-Linear and SVM-Stump are both promising
- feature selection: RLW and RSW are both good
  – possible to achieve better performance than full set
- error estimation: 10-fold CV seems to be a better choice and leave-one-out is bad
- how can we possibly distinguish between the linear model and the stump ensemble model?
  - are there more samples to verify the findings?
  - which model selects more biologically meaningful features?
  - which model is biologically more plausible?

Conclusion

- **carefully analyzed the difficult SAGE dataset**
  - legitimate information only
  - robust conclusion through multiple testing
- classification: SVM-Linear and SVM-Stump are both promising
- feature selection: RLW and RSW are both good
  – possible to achieve better performance than full set
- error estimation: 10-fold CV seems to be a better choice and leave-one-out is bad
- how can we possibly distinguish between the linear model and the stump ensemble model?
  - are there more samples to verify the findings?
  - which model selects more biologically meaningful features?
  - which model is biologically more plausible?