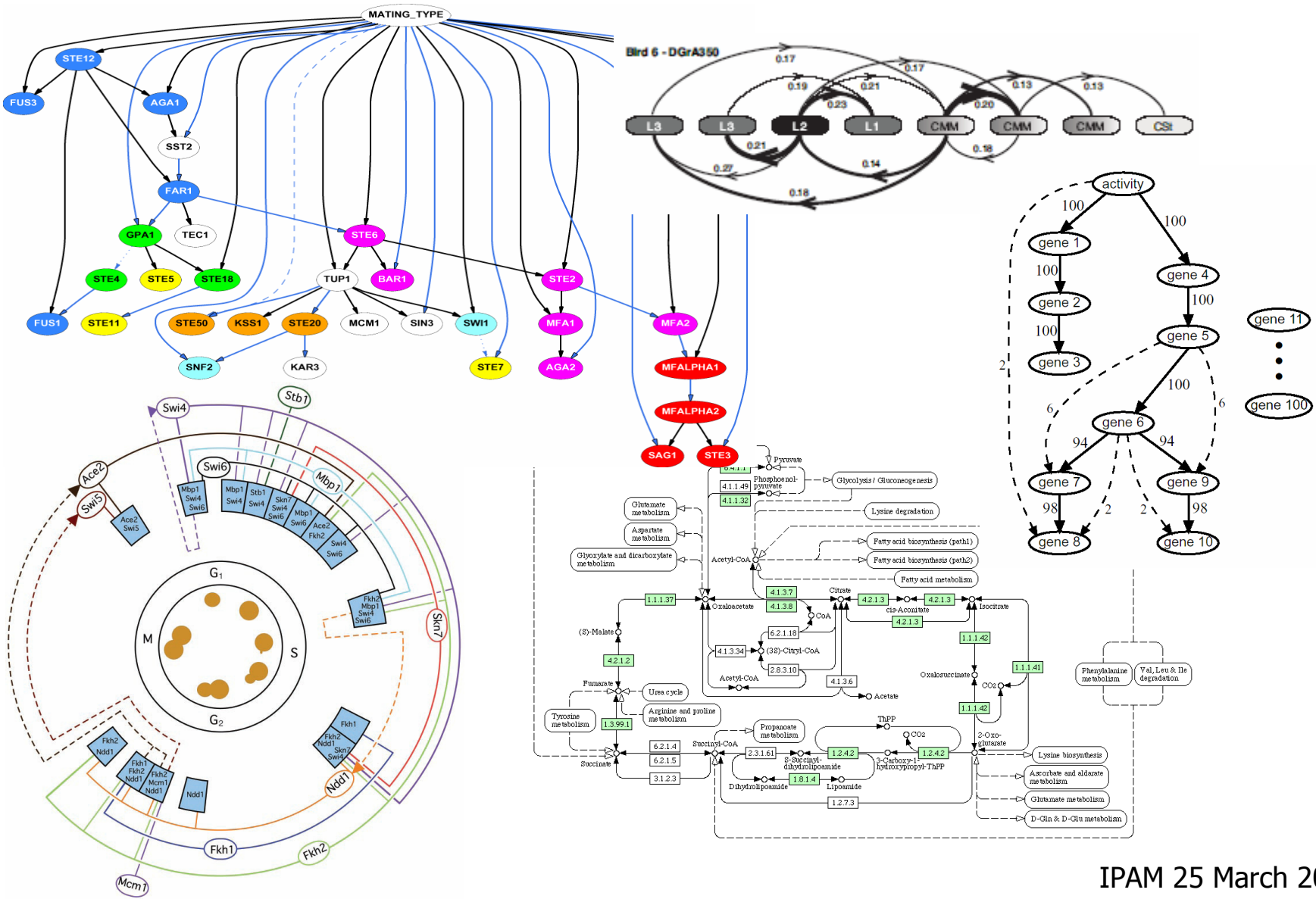
Classification in high-dimensional spaces: joint classification and feature selection

Alexander Hartemink

Department of Computer Science
Center for Bioinformatics and Computational Biology
Duke University

IPAM Workshop, 25 March 2004

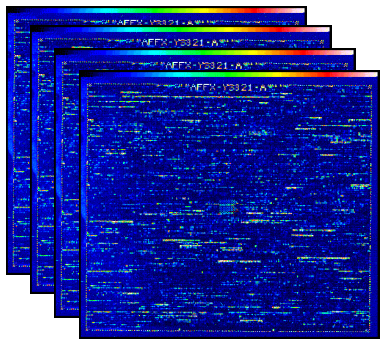
Network reconstruction

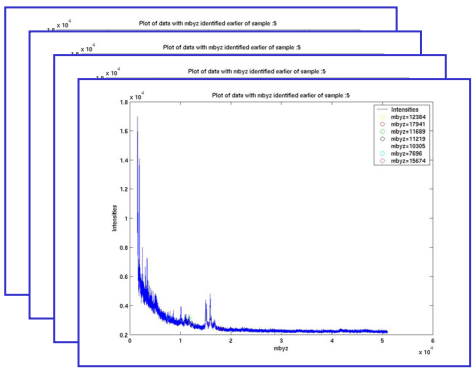


Outline

- Problem motivation
- Feature selection
- Classification
- Joint classifier and feature optimization (JCFO)
- Results

Problem setup and motivation



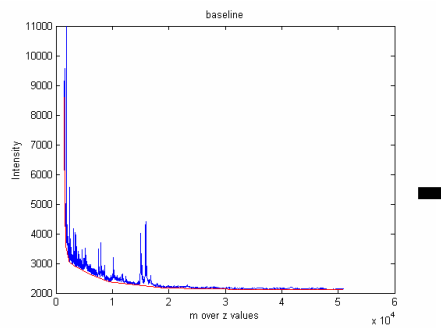


- Let's say we have n tissue samples from which we can collect gene or protein expression data
- Let us denote the p expression levels for each sample by a feature vector

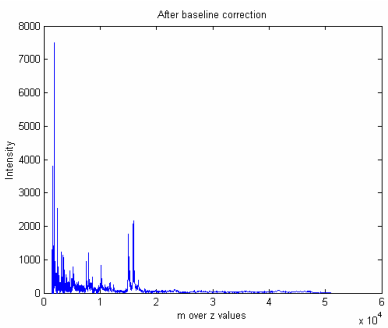
$$\boldsymbol{x} = [x_1, x_2, \ldots, x_p]^T \in \mathbb{R}^p$$

- What can we learn from a database of n such feature vectors?
- This is a task for machine learning (or statistical learning, or pattern recognition, or data mining)

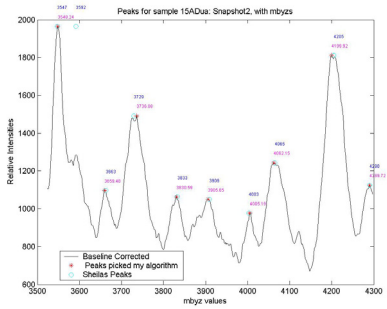
Caveat: feature extraction is not quite so simple in a mass-spectrometry context



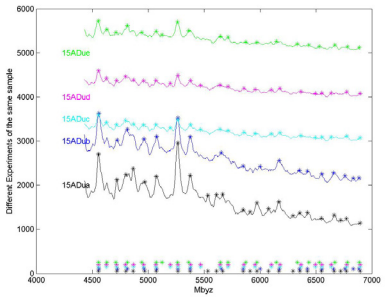
Baseline correction



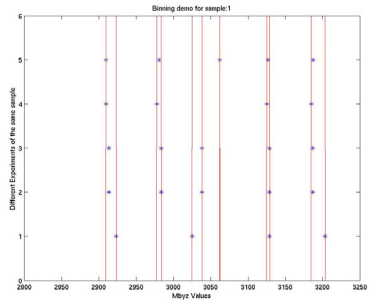
Peak identification



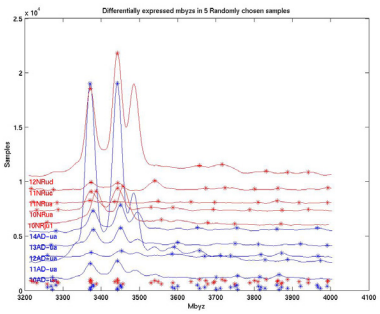
Peak alignment



Peak registration 1



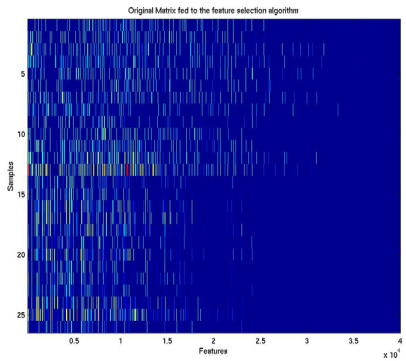
Peak registration 2



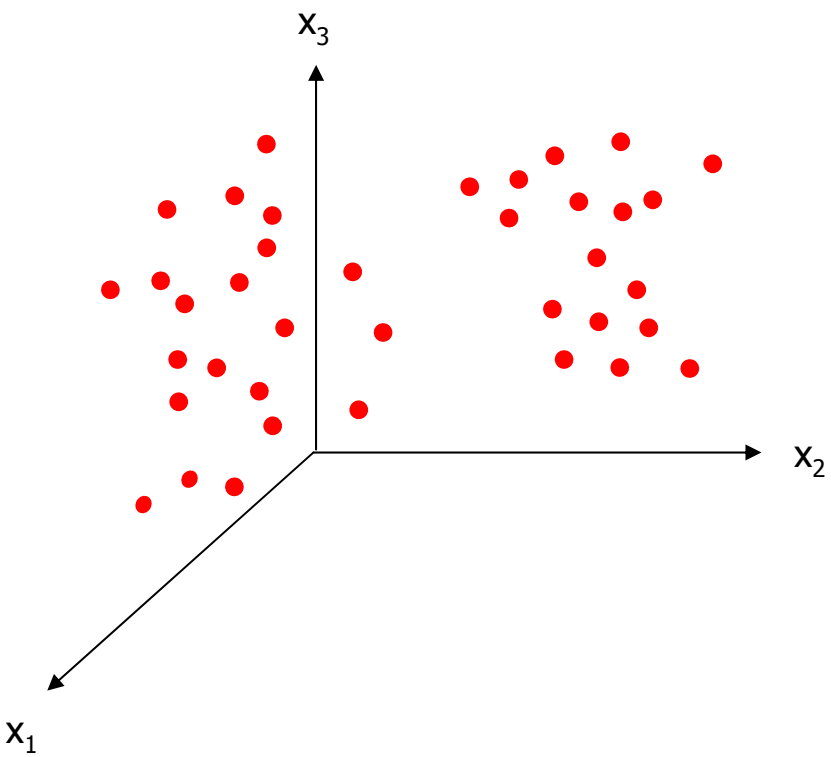
Build feature vectors

p features

n samples



Visual representation of the problem



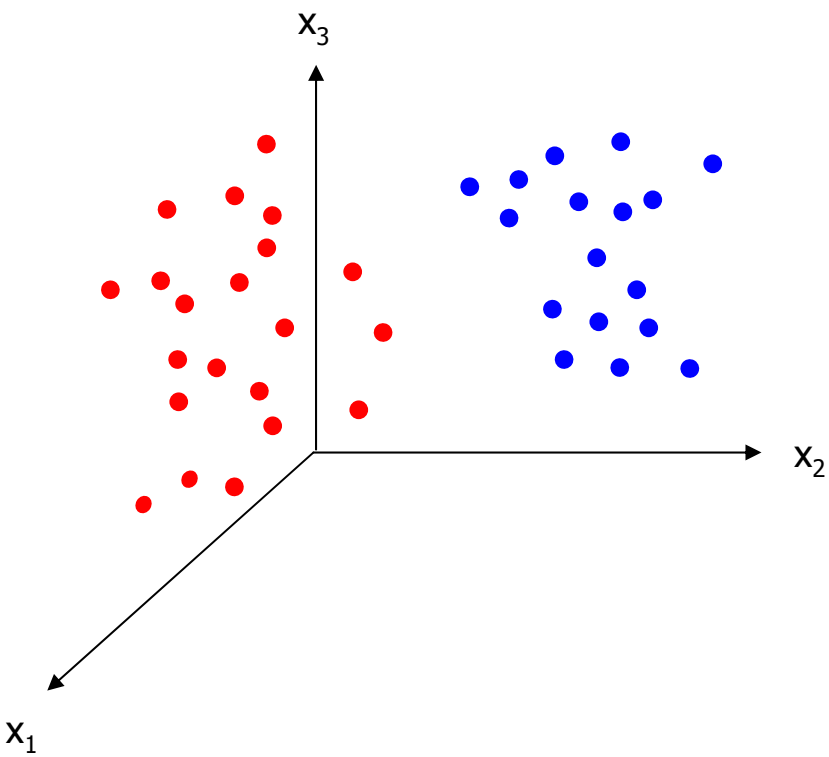
n points in $\mathbb{R}^p$

"unsupervised" learning context: discover patterns based on how the points are organized spatially (clustering methods, density estimation, etc.)

"Supervised" learning context

- Now suppose that samples fall into a discrete number of groups or subsets or classes
- Further suppose each of our samples $\boldsymbol{x}^{(i)}$ has been given a label $y^{(i)}$ to indicate the class to which the sample belongs
- We now have a classification problem
- Simplest case is dichotomous classification, in which $y^{(i)} \in \{0, 1\}$

Visual representation of the problem



n points in $\mathbb{R}^p$
each of which now has a label (color)

Problem definition

- We have a database of n samples, each with a feature vector and a label:

$$D = \left\{ \left(\boldsymbol{x}^{(i)}, y^{(i)} \right) : \boldsymbol{x}^{(i)} \in \mathbb{R}^p, y^{(i)} \in \{0, 1\} \right\}_{i=1}^{n}$$

- Can we learn a function $f(\boldsymbol{x})$ for predicting the label $y^{(i)}$ of a sample based on its corresponding feature vector $\boldsymbol{x}^{(i)}$, or in a probabilistic setting, predicting $P(y = 1 | \boldsymbol{x})$?

Many classification options already exist

- Gaussian mixture modeling
- class-conditional density estimation
- naïve Bayes
- *k* nearest neighbor (*k*-NN)
- decision trees or stumps
- classification and regression trees (CART)
- random forests
- bagging
- boosting (Adaboost, e.g.)
- linear or quadratic discriminant analysis
- weighted voting schemes
- artificial neural networks (ANN)
- Gaussian processes
- support vector machines (SVM)
- relevance vector machines (RVM)
- logistic regression or sparse logistic regression
- probit regression or sparse probit regression (SPR)

Why another classifier?

- p is typically on the order of 2,000-20,000
- n is typically on the order of 20-200
- This is what statisticians refer to as a " large p , small n " problem
- Danger of overfitting, leading to poor generalization performance
- Curse of dimensionality suggests the importance of feature selection

Outline

- Problem motivation
- Feature selection
- Classification
- Joint classifier and feature optimization (JCFO)
- Results

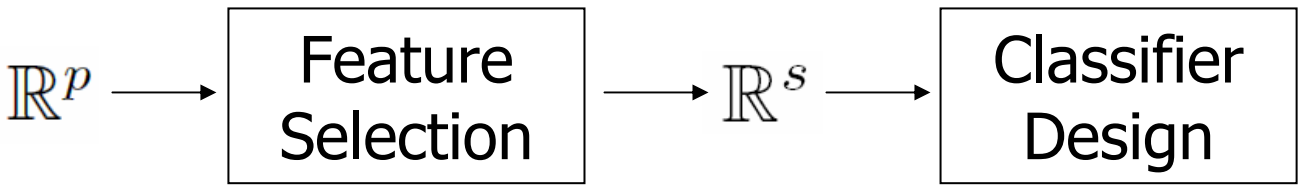
Why feature selection?

- Leads to better classification performance (classifier should fit decision surface based only on features that matter, not noise)
- Can provide insights into mechanisms behind class distinctions (or identify features for subsequent network analysis)
- Leads to simpler and cheaper diagnostic tests if deployed clinically

Approaches to feature selection

- Methods fall into three basic categories:
  - » Filter methods
  - » Wrapper methods
  - » Embedded methods
- Almost all existing methods are in one of the first two categories

Filter methods



$$s < p$$

- Serial approach: features are scored independently and the top s make it through the filter before reaching the classifier
- Score: correlation, mutual information, *t*-statistic, decision stump error, signal-to-noise ratio, FDR, etc.

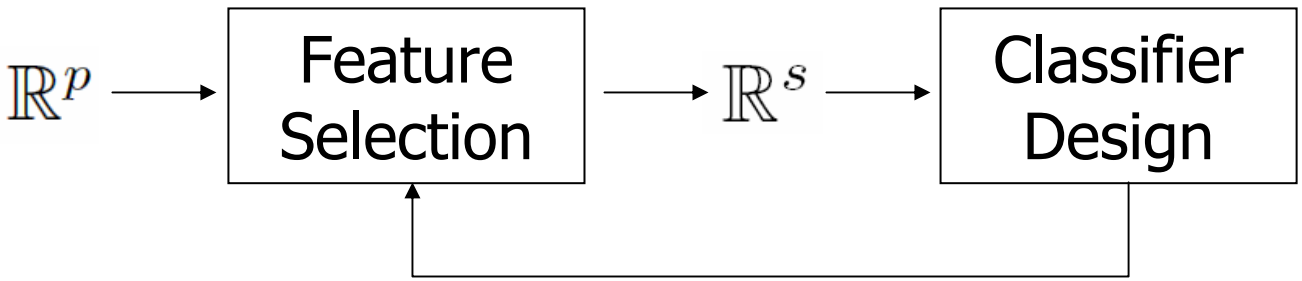
Problems with filter methods

- Redundancy in selected features: features are considered independently and not measured on the basis of whether they contribute any new information
- Inability to detect higher-order relationships between combinations of features and class labels
- No feedback from the classifier in terms of how useful the features are

Dimension reduction: a filter method with a twist

- Rather than retain a subset of s features, perform dimension reduction by projecting features onto s principal components of variation (PCA/SVD, etc.)
- Problem is that the s new basis vectors are combinations of all of the features and are often quite uninterpretable: what exactly is a supergene or eigengene anyway?

Wrapper methods



- Iterative approach: many feature subsets are scored based on classification performance and best is used
- Selection of subsets: forward selection, backward selection, forward-backward selection, recursive feature elimination (RFE), shaving, etc.

Problems with wrapper methods

- Computationally expensive: for each feature subset to be considered, a classifier must be learned, and also evaluated
- There are 2^p feature subsets to consider (and recall that p here is order 2,000-200,000) so we need heuristics to select feature subsets
- Multiple-hypothesis testing issues

Embedded methods

- Attempt to jointly or simultaneously learn both a classifier and a feature subset
- Often optimize an objective function that jointly rewards accuracy of classification and penalizes use of more features
- Connections between penalized/regularized regression and Bayesian regression

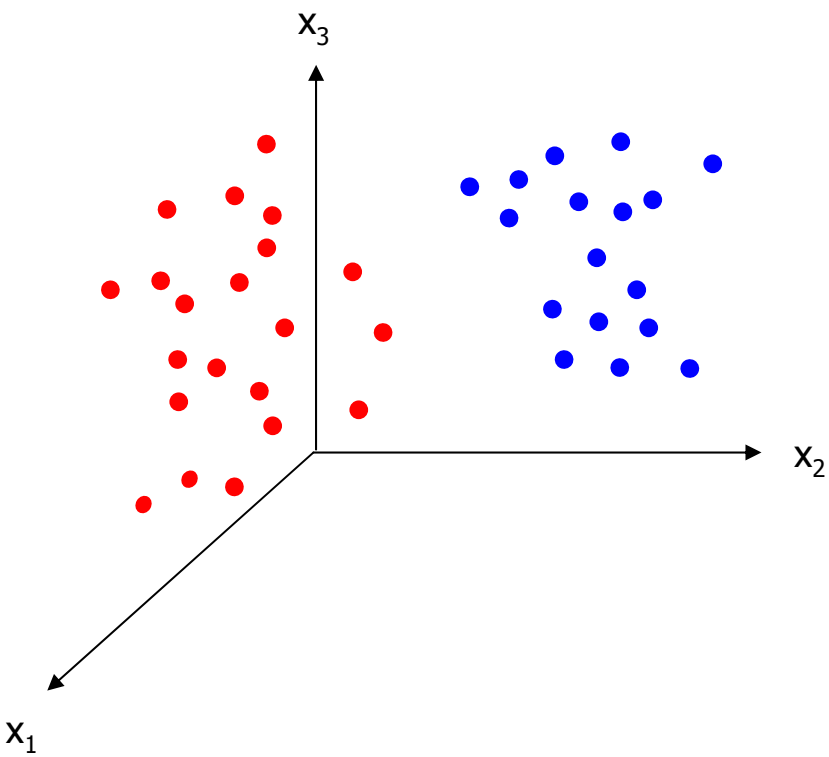
Outline

- Problem motivation
- Feature selection
- Classification
- Joint classifier and feature optimization (JCFO)
- Results

Linear family of possible decision surfaces

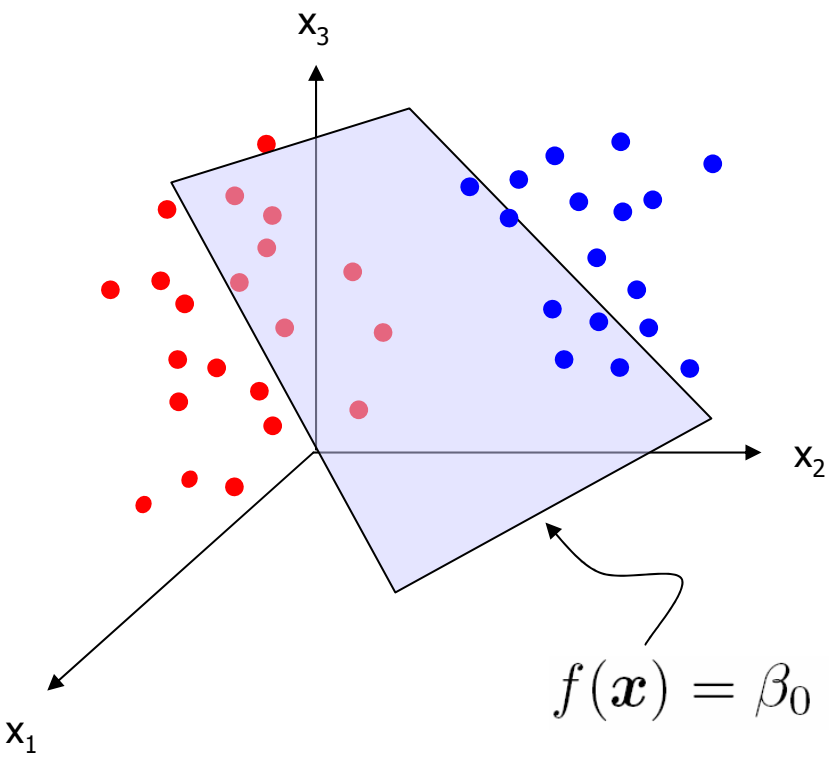
- We are good with linear functions so let us assume that the function $f(x)$ will be linear
- In a discriminative setting, this corresponds to learning a hyperplane decision surface
- What if a linear boundary is not suitable or appropriate? We've got a trick for that in a minute…

Visual representation of the problem



n points in $\mathbb{R}^p$
each of which now has a label (color)

Visual representation of the problem



n points in $\mathbb{R}^p$
each of which now
has a label (color)

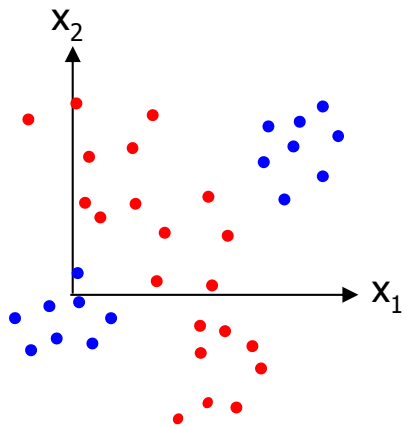
linear hyperplane
as decision surface

$$f(\boldsymbol{x}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

What if we need a nonlinear decision surface to separate the classes?

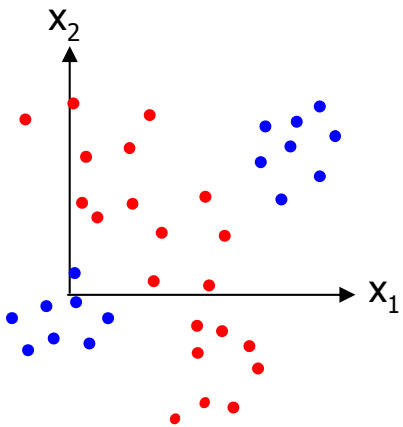
- Map original feature space $\mathbb{R}^p$ into a new space $\mathbb{R}^q$ (typically $p \ll q$; q may be ∞)
- The n points in $\mathbb{R}^p$ are now n points in $\mathbb{R}^q$
- If we learn a linear decision surface in new space, its pre-image in the original feature space will typically be a nonlinear decision surface

Illustration of feature space mapping

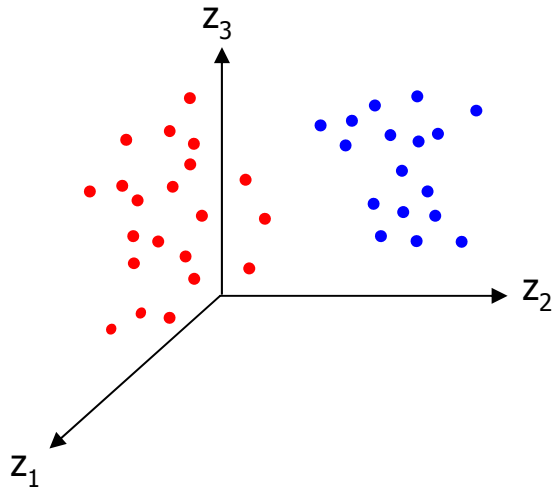


n points in $\mathbb{R}^p$

Illustration of feature space mapping



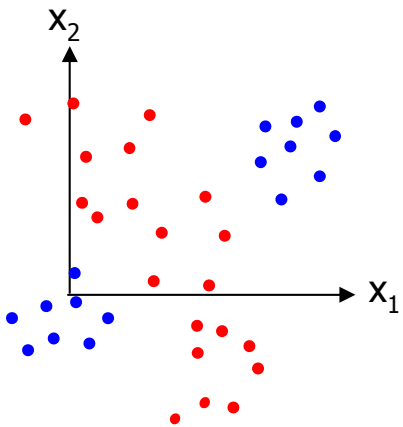
$\phi : \mathbb{R}^p \longrightarrow \mathbb{R}^q$



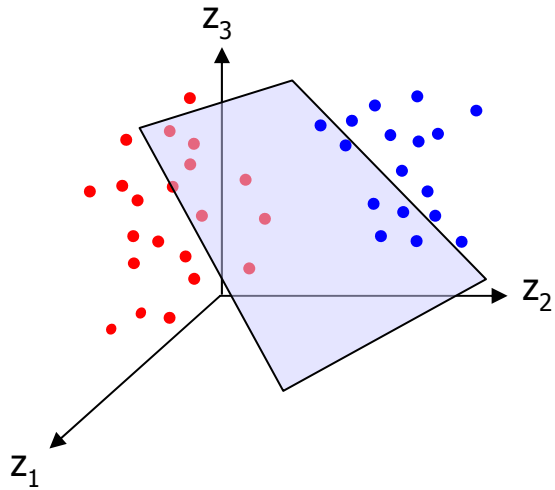
n points in $\mathbb{R}^p$

n points in $\mathbb{R}^q$

Illustration of feature space mapping



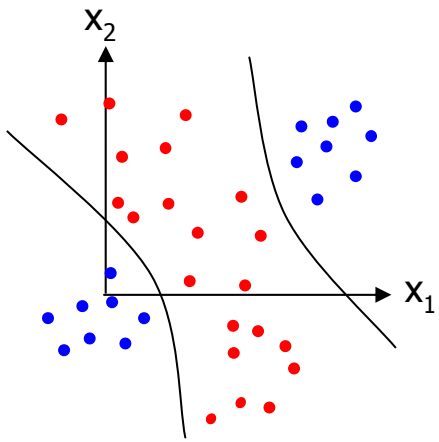
$$\phi : \mathbb{R}^p \longrightarrow \mathbb{R}^q$$



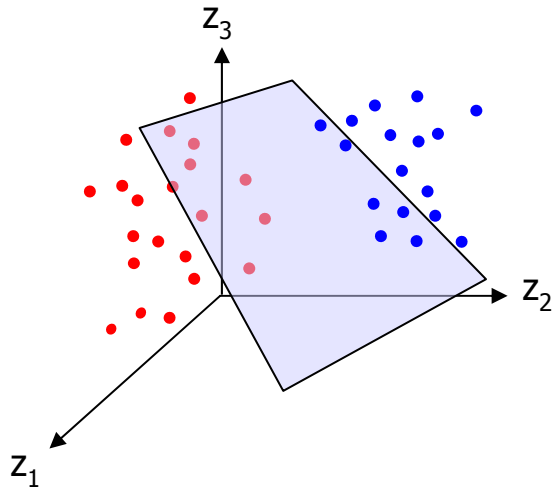
n points in $\mathbb{R}^p$

n points in $\mathbb{R}^q$

Illustration of feature space mapping



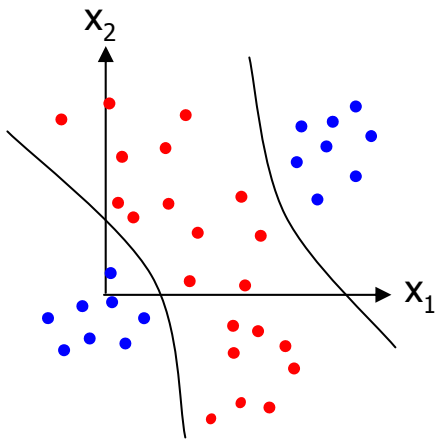
$$\phi : \mathbb{R}^p \longrightarrow \mathbb{R}^q$$



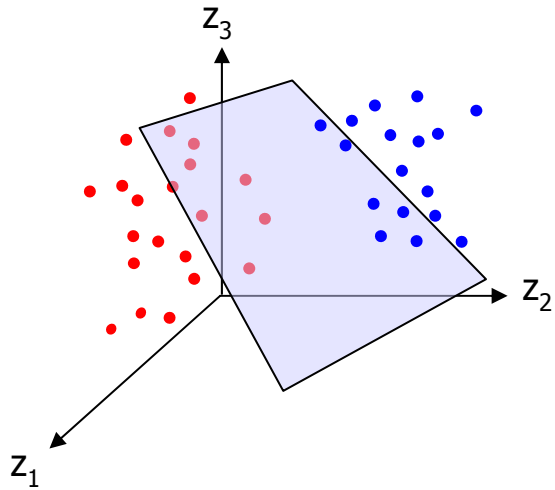
n points in $\mathbb{R}^p$

n points in $\mathbb{R}^q$

Illustration of feature space mapping



$$\phi : \mathbb{R}^p \longrightarrow \mathbb{R}^q$$

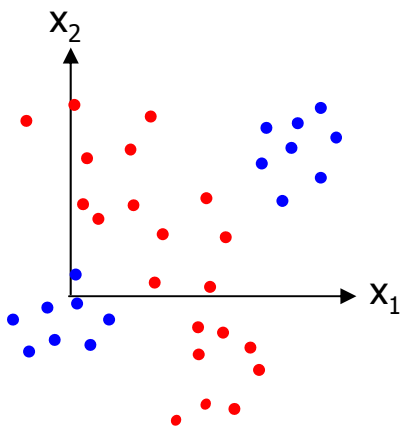


n points in $\mathbb{R}^p$

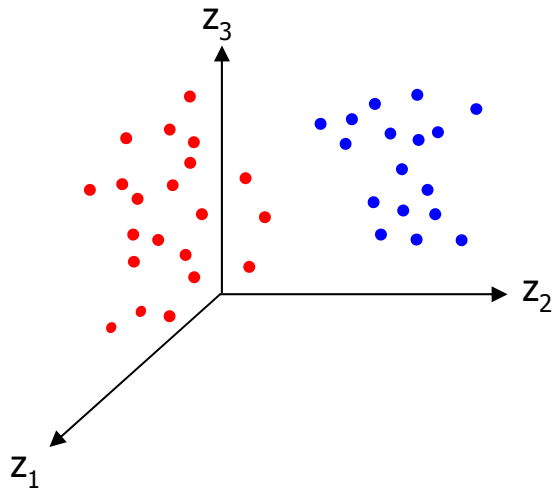
n points in $\mathbb{R}^q$

- But haven't we just made the dimensionality problem worse?
- And how do we know what mapping to use?

Mapping into a kernel space

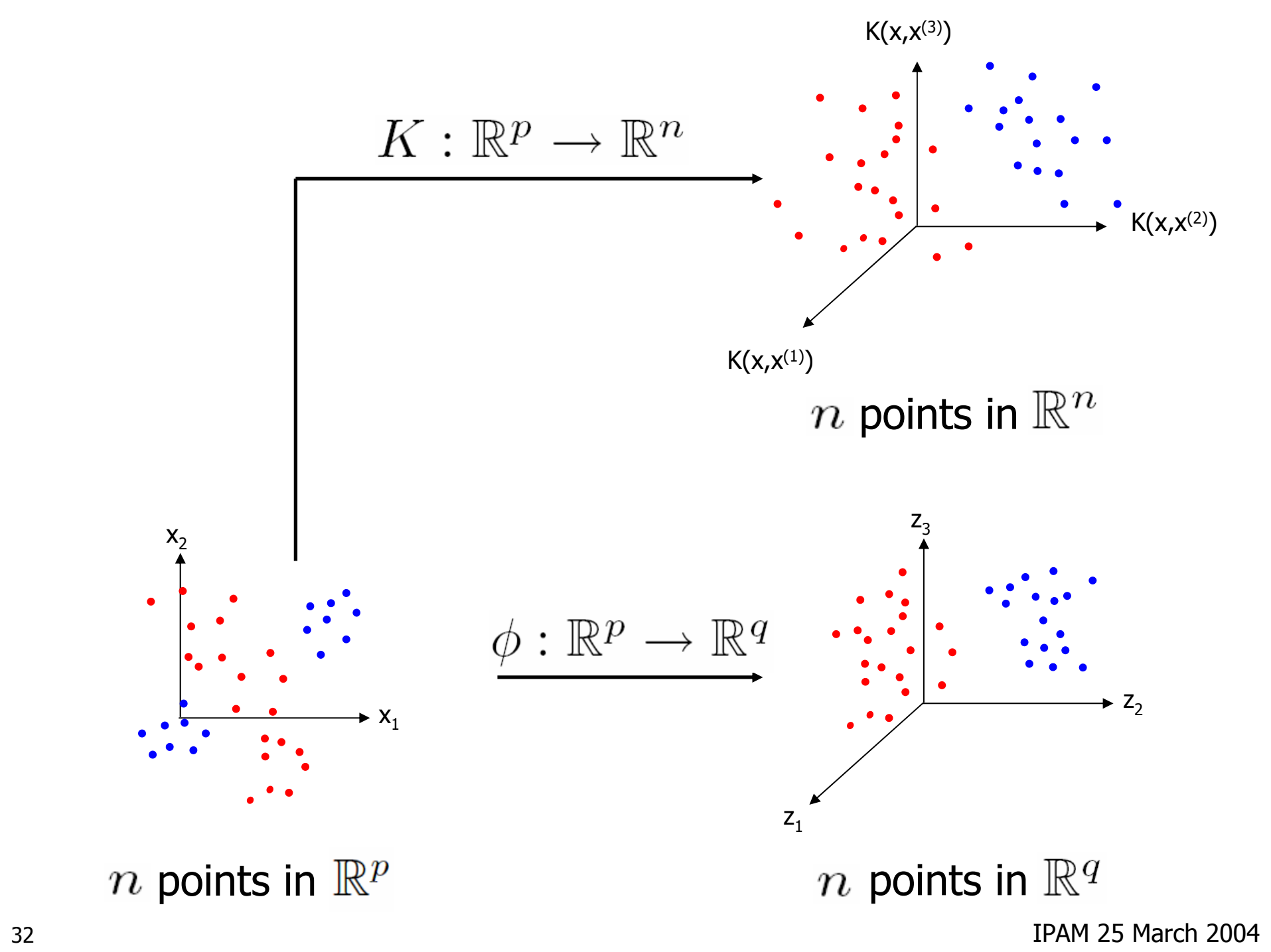


$$\phi : \mathbb{R}^p \longrightarrow \mathbb{R}^q$$

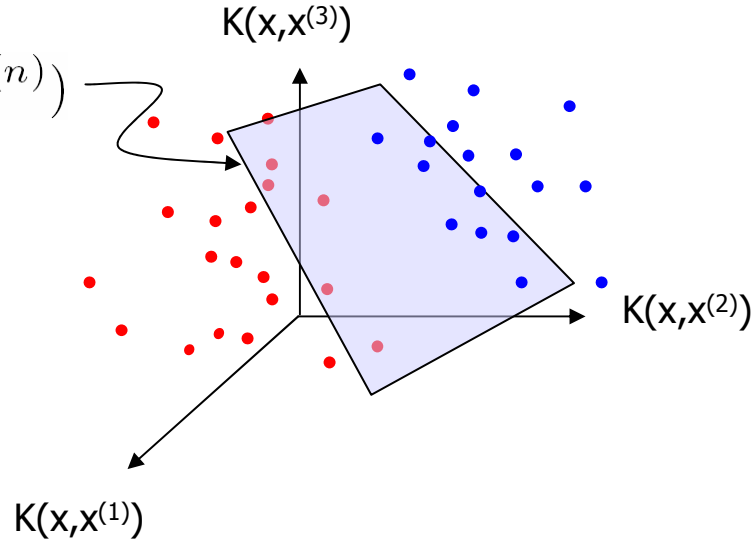


n points in $\mathbb{R}^p$

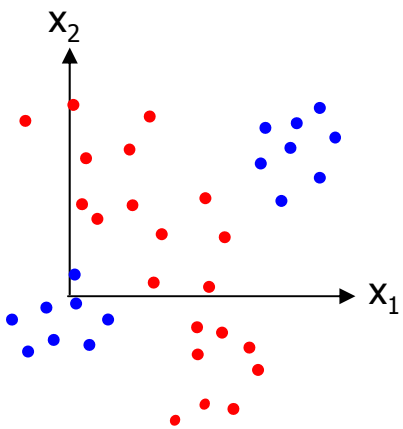
n points in $\mathbb{R}^q$


$K : \mathbb{R}^p \longrightarrow \mathbb{R}^n$
K(x,x(3))
K(x,x(2))
K(x,x(1))
n points in $\mathbb{R}^n$
x2
x1
$\phi : \mathbb{R}^p \longrightarrow \mathbb{R}^q$
z3
z2
z1
n points in $\mathbb{R}^p$
n points in $\mathbb{R}^q$

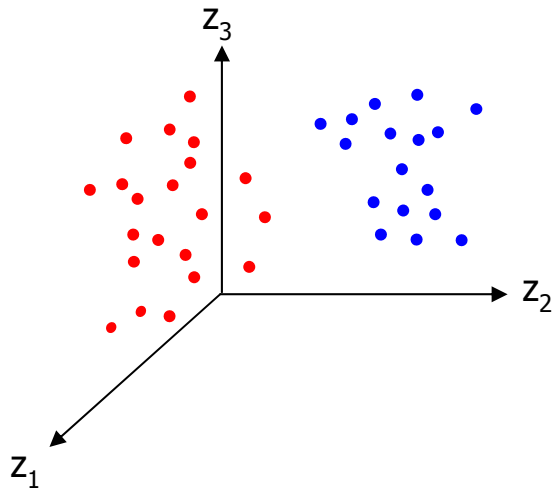
$$f(\boldsymbol{x}) = \beta_0 + \beta_1 K(\boldsymbol{x}, \boldsymbol{x}^{(1)}) + \cdots + \beta_n K(\boldsymbol{x}, \boldsymbol{x}^{(n)})$$



n points in $\mathbb{R}^n$



n points in $\mathbb{R}^p$



n points in $\mathbb{R}^q$

Linear family of possible decision surfaces

- At the end of the day, we are still going to learn a linear hyperplane
- The linear hyperplane decision surface we learn will typically correspond to a nonlinear decision surface in the original feature space
- We can bypass the high-dimensional space and go directly to the kernel space
- Choice of kernel depends on the problem's geometry

There are many different criteria we could employ for learning this plane

- A learning theory bound leads to a criterion that maximizes the margin (SVM, e.g.)
- Simple probability theory leads to a criterion that maximizes the likelihood under a link function (logistic or probit regression, e.g.)
- Bayesian statistical theory leads to a criterion that maximizes the evidence (RVM, e.g.) or the posterior probability (SPR, JCFO, e.g.)

Outline

- Problem motivation
- Feature selection
- Classification
- Joint classifier and feature optimization (JCFO)
- Results

Joint classification and feature optimization (JCFO)

- JCFO is an extension of SPR which is a refinement of the RVM which is a Bayesian formulation of the SVM
- SVM: sparse in use of kernel basis functions (leads to small set of "support vectors")
- SPR and RVM: sparse in use of kernel basis functions when in kernel mode; or sparse in use of features when in non-kernel mode; but not both
- JCFO: both sparse in use of kernel basis functions and sparse in use of features when constructing those kernel basis functions

JCFO formulation

- Probit regression formalism:

$$P(y=1|\boldsymbol{x}) = \Phi\left(\beta_0 + \sum_{i=1}^{n} \beta_i K_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}^{(i)})\right)$$

- Kernel functions depend on scaling factors
  - » Polynomial: $K_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}^{(i)}) = \left(1 + \sum_{k=1}^{p} \theta_k x_k x_k^{(i)}\right)^r$
  - » Gaussian RBF: $K_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}^{(i)}) = \exp\left(-\sum_{k=1}^{p} \theta_k \left(x_k - x_k^{(i)}\right)^2\right)$

JCFO objective function

- Learning via MAP criterion means selecting parameters to maximize posterior probability:

$$(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}}) \;=\; \arg\max_{(\boldsymbol{\beta},\boldsymbol{\theta})} \left(p(D|\boldsymbol{\beta},\boldsymbol{\theta})\, p(\boldsymbol{\beta}|\gamma_1)\, p(\boldsymbol{\theta}|\gamma_2)\right)$$

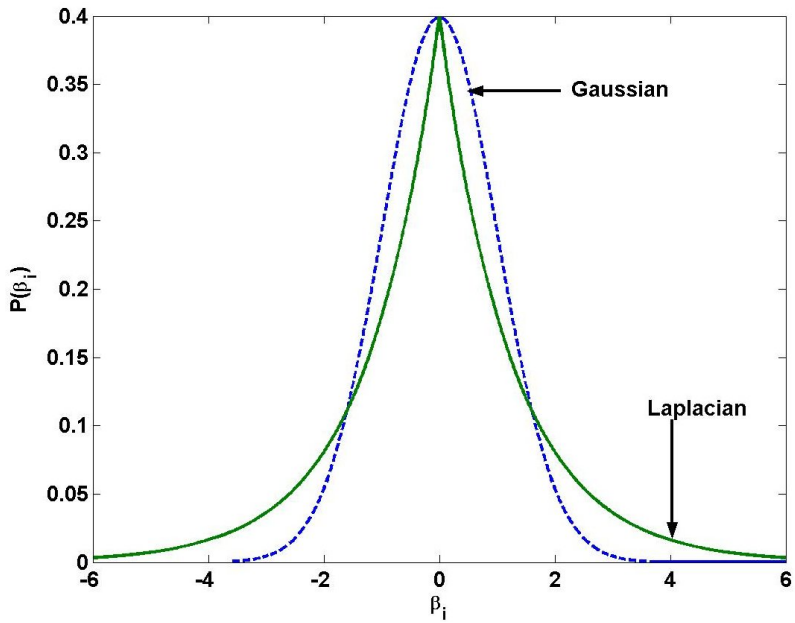
- Likelihood function is a binomial:

$$p(D|\boldsymbol{\beta},\boldsymbol{\theta}) = \prod_{i=1}^{n} \left[\Phi\left(\beta_0 + \sum_{j=1}^{n} \beta_i K_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})\right)\right]^{y^{(i)}} \left[\Phi\left(-\beta_0 - \sum_{j=1}^{n} \beta_i K_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})\right)\right]^{(1-y^{(i)})}$$

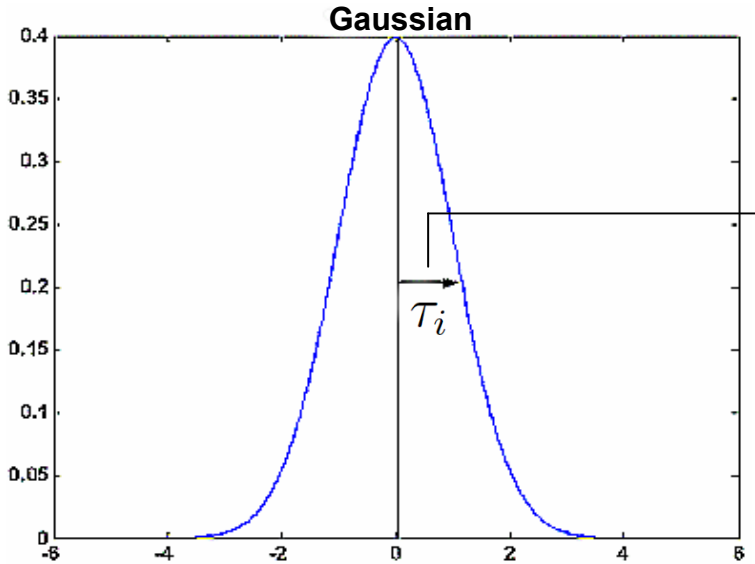
- What about the priors?

Sparsity-promoting priors

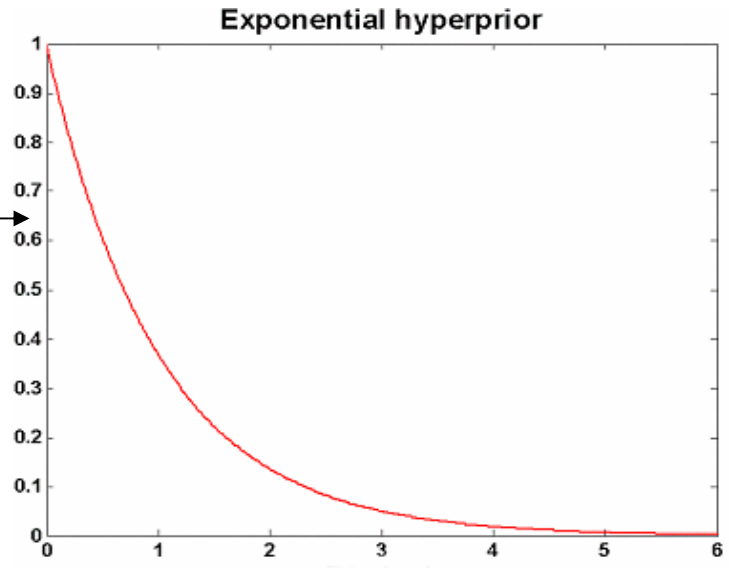
- We use Laplacian priors
- Equivalent to an ℓ_1 penalty; related to LASSO
- Mathematically challenging, so we exploit equivalent hierarchical priors



Hierarchical representation of Laplacian



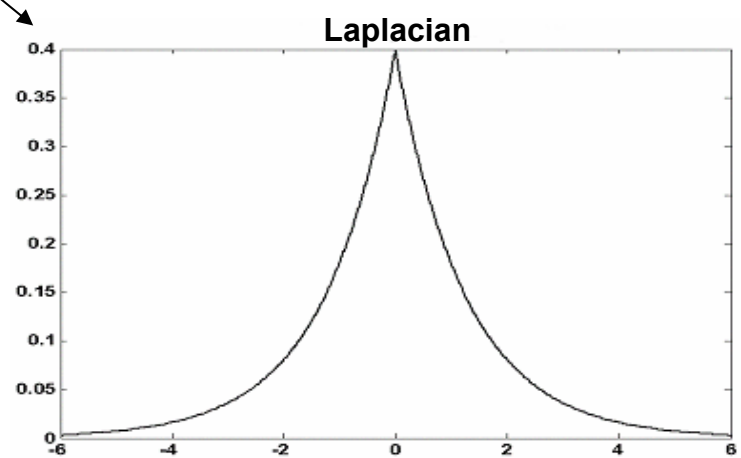
$$p(\beta_i|\tau_i) = \mathcal{N}(\beta_i|0,\tau_i)$$



$$p(\tau_i|\gamma_1) = (\gamma_1/2)\exp(-\gamma_1\,\tau_i/2)$$

$$p(\beta_i|\gamma_1) = \int_0^\infty p(\beta_i|\tau_i)\,p(\tau_i|\gamma_1)\,d\tau_i$$

$$= \frac{\sqrt{\gamma_1}}{2}\exp(-\sqrt{\gamma_1}\,|\beta_i|)$$



EM algorithm used to maximize log posterior probability

- E step is completely analytical, very fast
- M step involves this maximization:

$$\hat{\boldsymbol{\beta}}^{(t+1)}, \hat{\boldsymbol{\theta}}^{(t+1)} = \arg\max_{\boldsymbol{\beta},\boldsymbol{\theta}} Q\left(\boldsymbol{\beta}, \boldsymbol{\theta} \,\middle|\, \hat{\boldsymbol{\beta}}^{(t)}, \hat{\boldsymbol{\theta}}^{(t)}\right)$$

$$= \arg\max_{\boldsymbol{\beta},\boldsymbol{\theta}} \; -\boldsymbol{\beta}^T \boldsymbol{H}^T \boldsymbol{H} \boldsymbol{\beta} + 2\boldsymbol{\beta}^T \boldsymbol{H}^T \boldsymbol{v} - \boldsymbol{\beta}^T \boldsymbol{\Omega} \boldsymbol{\beta} - \boldsymbol{\theta}^T \boldsymbol{\Delta} \boldsymbol{\theta}$$

- Done in two parts: matrix inversion to find $\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}^{(t+1)}$ and numerical optimization to find $\hat{\boldsymbol{\theta}}^{(t+1)}$

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}^{(t+1)} = (\boldsymbol{\Omega} + \boldsymbol{H}^T \boldsymbol{H})^{-1} \boldsymbol{H}^T \boldsymbol{v}$$

$$= \boldsymbol{\kappa}(\boldsymbol{I} + \boldsymbol{\kappa} \boldsymbol{H}^T \boldsymbol{H} \boldsymbol{\kappa})^{-1} \boldsymbol{\kappa} \boldsymbol{H}^T \boldsymbol{v}$$

Outline

- Problem motivation
- Feature selection
- Classification
- Joint classifier and feature optimization
- Results

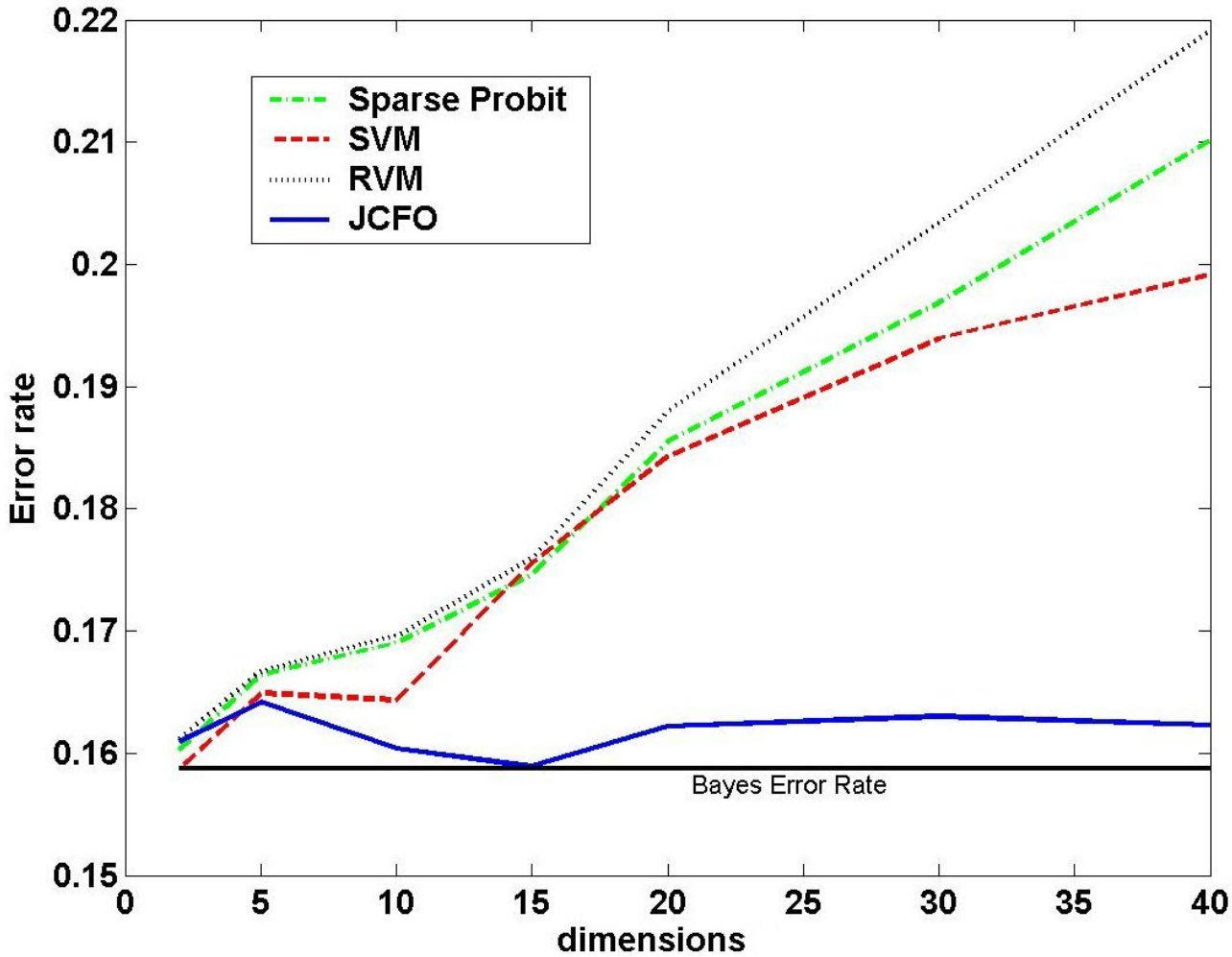
Evaluating effect of irrelevant features

- Sample 100 points from each of two spherical Gaussian distributions in $\mathbb{R}^p$ as training set
- Gaussian means only differ along first two dimensions:

$$\boldsymbol{\mu}_1 = -\boldsymbol{\mu}_2 = [\,1/\sqrt{2}, 1/\sqrt{2}, \underbrace{0, 0, \ldots, 0}_{(p-2)\ \text{zeros}}\,]^T$$

- Bayesian optimal classifier only cares about first two features and has error rate of 0.159

JCFO error rate stable as number of irrelevant features grows



Application to benchmark machine learning datasets

Classifier	Ripley	Pima	Crabs	WBC
Linear discriminant [14]	N/A	67	3	19
Neural network [20]	N/A	75	3	N/A
Gaussian process [20]	92	67	3	8
SVM [14]	106	64	4	9
Logistic regression [20]	N/A	66	4	N/A
RVM [18]	93	65	0	9
Sparse probit regression	95	62	0	9
JCFO	92	64	0	8

Application to high dimensional gene expression data

Classifier	AML/ALL	Colon
Adaboosting (Decision stumps) [2]	95.8	72.6
SVM (Linear kernel) [2]	94.4	77.4
SVM (Quadratic kernel) [2]	95.8	74.2
Logistic regression (No kernel: on feature space) [9]	97.2	71.0
RVM (No kernel: on feature space) [9]	97.2	88.7
Sparse probit regression (No kernel: on feature space)	97.2	88.7
Sparse probit regression (Linear kernel)	97.2	91.9
Sparse probit regression (Quadratic kernel)	95.8	84.6
JCFO (Linear kernel)	100.0	96.8
JCFO (Quadratic kernel)	98.6	88.7

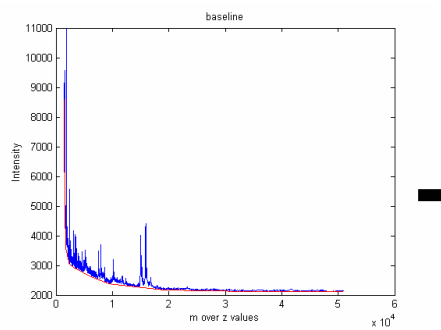
Application to high dimensional gene expression data

Classifier	**Duke ER status**	**Duke LN status**	**Lund**
SVM (Linear kernel)	97.4	78.9	87.9
RVM (Linear kernel)	94.7	92.1	88.5
RVM (No kernel)	89.5	81.6	96.5
Sparse probit regression (Linear kernel)	97.4	89.5	86.2
Sparse probit regression (No kernel)	84.2	89.5	96.5
JCFO (Linear kernel)	97.4	94.7	98.3

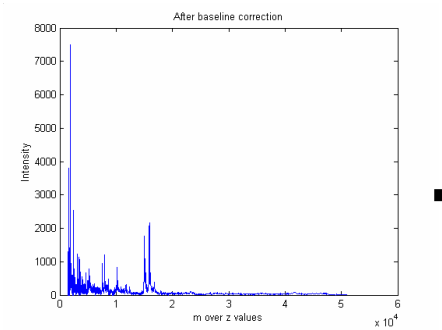
JCFO returns the relative weights of features in the kernel similarity function

θ_i	Index	Accession	Gene Name	Gene Description
2.10	1357	T84051	● CDC42	cell division cycle 42 (GTP binding protein, 25kDa)
1.76	974	U00968	SREBF1	sterol regulatory element binding transcription factor 1
1.47	1924	H64807		placental folate transporter (H. sapiens)
1.44	1873	L07648	● MXI1	MAX interacting protein 1
1.41	350	D26129	● RNASE1	ribonuclease, RNase A family, 1 (pancreatic)
1.38	377	Z50753	● GUCA2B	guanylate cyclase activator 2B (uroguanylin)
1.21	1757	H16096	PMPCB	peptidase (mitochondrial processing) beta
1.01	765	M76378	CSRP1	cysteine and glycine-rich protein 1
0.86	1346	T62947	RPL24	ribosomal protein L24
0.84	1976	K03474	AMH	anti-Mullerian hormone
0.75	792	R88740	ATP5J	ATP synthase, H+ transporting, mitochondrial F0 complex, subunit F6
0.74	70	T61661	PFN1	profilin 1
0.74	554	H24401		MAP kinase phosphatase-1 (H. sapiens)
0.74	698	T51261	SERPINE2	serine (or cysteine) proteinase inhibitor, clade E (nexin), member 2
0.72	1546	T51493	PPP2R5C	protein phosphatase 2, regulatory subunit B (B56), gamma isoform
0.64	1740	M81651	SEMG2	semenogelin II
0.50	1641	K02268	PDYN	prodynorphin
0.42	1024	R65697	● REL	v-rel reticuloendotheliosis viral oncogene homolog (avian)
0.37	1644	R80427		C4-dicarboxylate transport sensor protein DCTB (R. leguminosarum)
0.32	1623	T94993	● FGFR2	fibroblast growth factor receptor 2 (keratinocyte growth factor receptor)
0.14	1909	U10886	● PTPRJ	protein tyrosine phosphatase, receptor type, J
0.12	1482	T64012	CHRND	cholinergic receptor, nicotinic, delta polypeptide
0.10	1094	R33481	ATF7	activating transcription factor 7
0.06	187	T51023	HSPCB	heat shock 90kDa protein 1, beta
0.06	1504	H78386	IL1R2	interleukin 1 receptor, type II
0.03	1241	T64885		general negative regulator of transcription subunit 1 (S. cerevisiae)

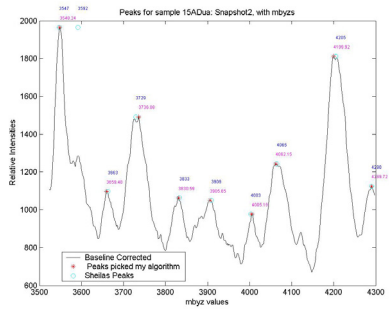
But what about proteomic data?



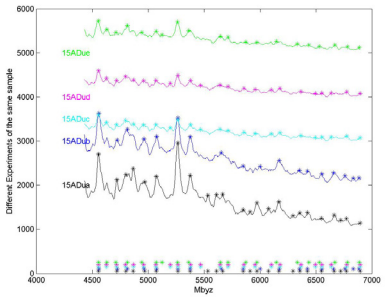
Baseline correction



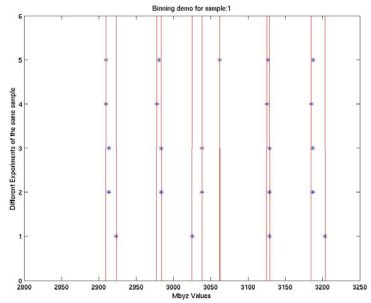
Peak identification



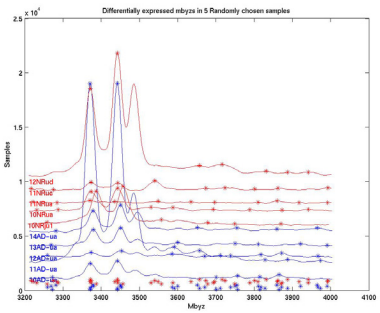
Peak alignment



Peak registration 1



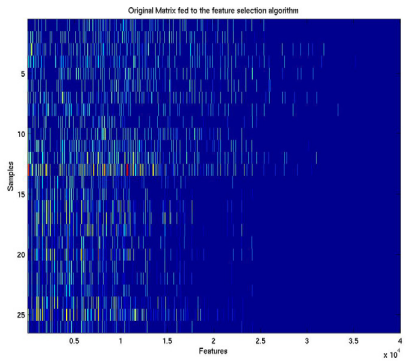
Peak registration 2



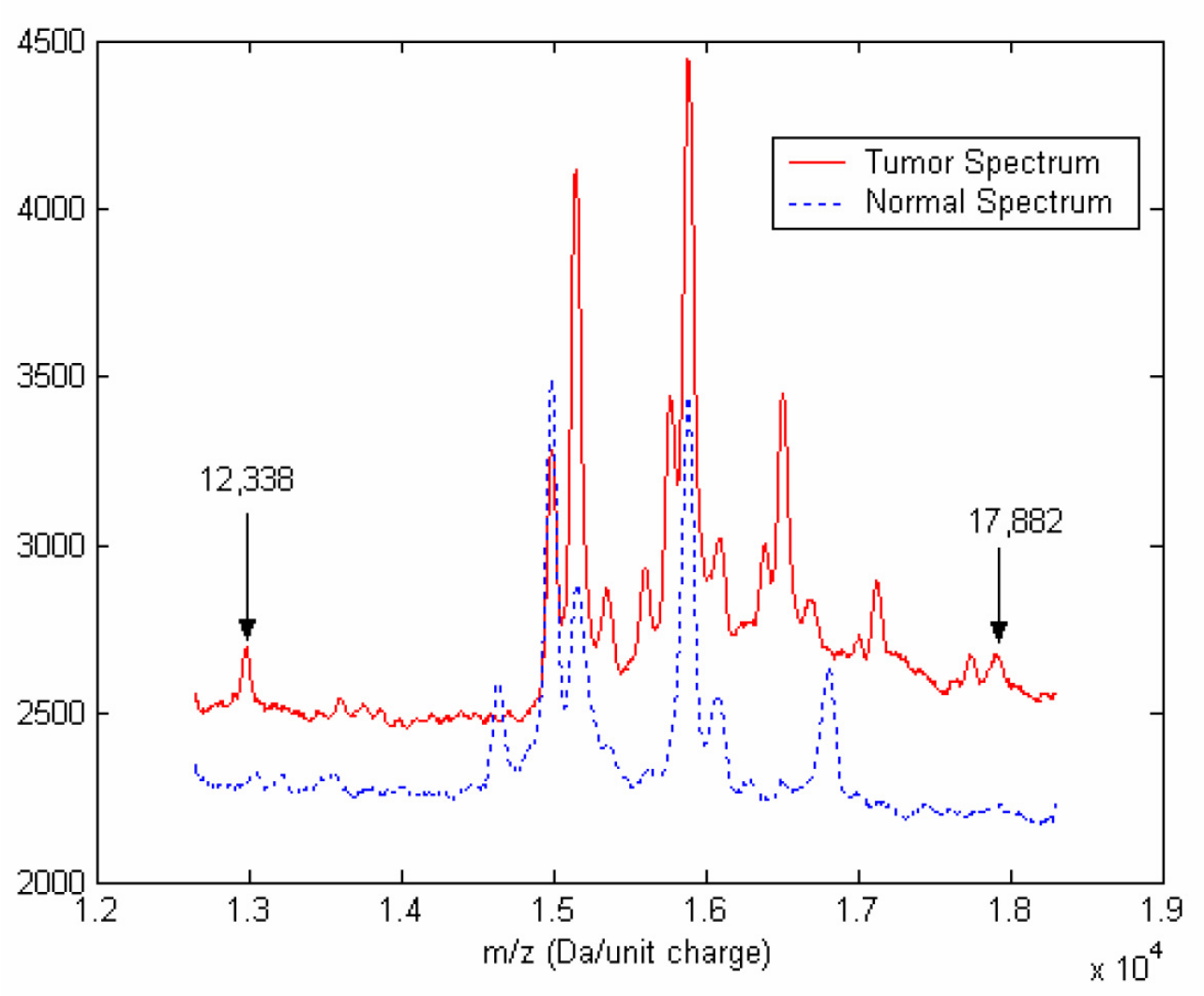
Build feature vectors

p features

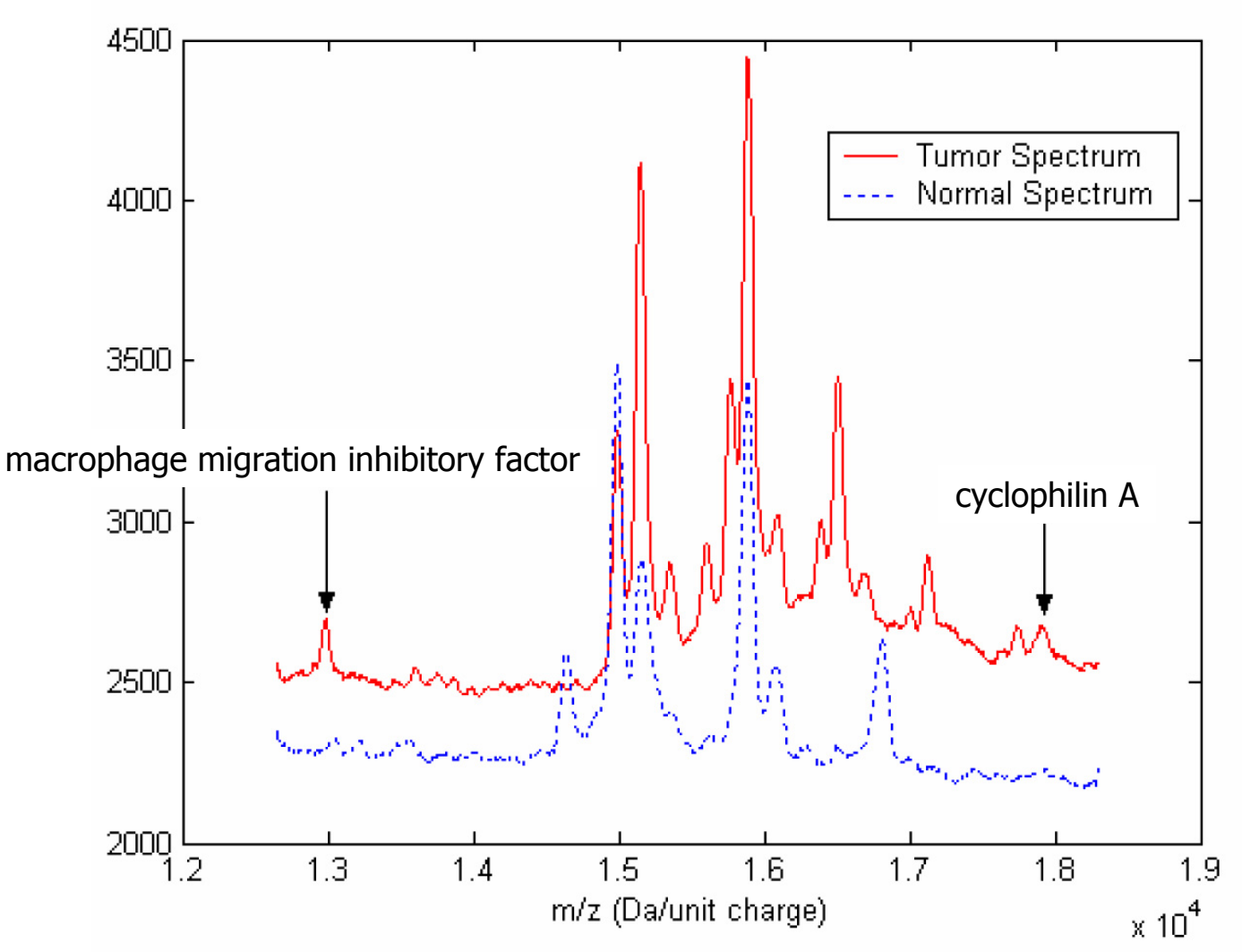
n samples



Caveat: relevant features not necessarily biomarkers, still need to be identified



Caveat: relevant features not necessarily biomarkers, still need to be identified



Conclusions

- When working in high-dimensional feature spaces with many irrelevant features, feature selection is critical to prevent overfitting
- Existing feature selection methods are rarely embedded in the classifier design problem
- JCFO jointly learns a classifier and selects the features most relevant to the classification decision
- Performance is state-of-the-art and derives mostly from better feature selection
- In a proteomic context, proper feature extraction is a necessary precondition for effective feature selection

Acknowledgements

- Balaji Krishnapuram, Duke ECE
- Larry Carin, Duke ECE
- Mario Figueiredo, Instituto Superior Tecnico, Portugal
- Pallavi Pratapa, Duke CS
- Qiuhua Liu, Duke ECE
- Ned Patz, Duke Radiology
- Michael Campa, Duke Radiology

Contact information

Alexander J. Hartemink

Department of Computer Science
Center for Bioinformatics and Computational Biology
Duke University

amink@cs.duke.edu