Advanced Methods for Sequence Analysis

Petra Philips

Friedrich Miescher Laboratory, Tübingen

Vorlesung WS 2006/2007
Eberhard Karls Universität Tübingen

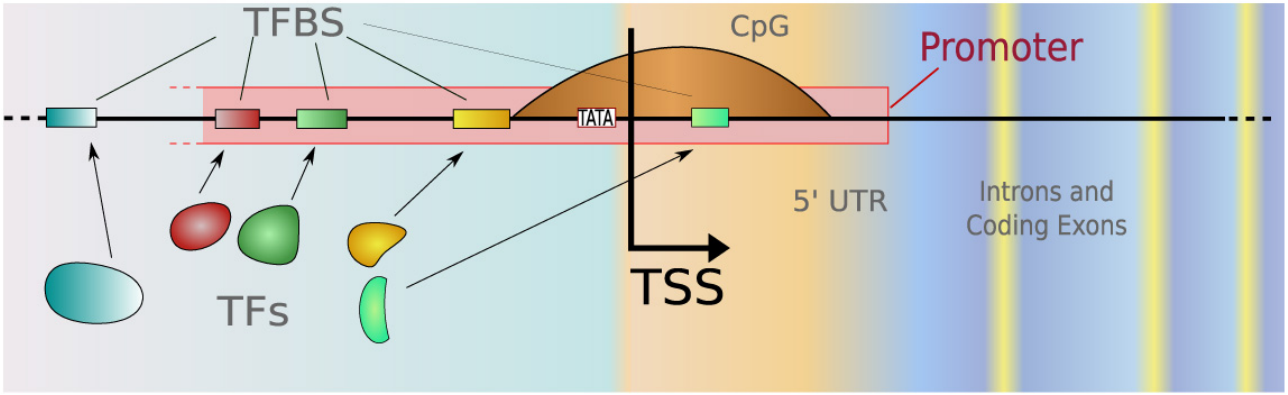
31 January 2007

http://www.fml.mpg.de/raetsch/lectures/amsa

Topics

- Multiple Kernel Learning (MKL)
- Interpretability

Example: Promoter Detection



- POL II binds to a rather vague region of $\approx [-20, +20]$ bp
- Upstream of TSS: promoter containing transcription factor binding sites
- Downstream of TSS: 5' UTR, and further downstream coding regions and introns (different statistics)
- 3D structure of the promoter must allow the transcription factors to bind

Features to Describe the TSS

- TFBS in Promotor region
- Condition: DNA should not be too twisted
- CpG islands (often over TSS/first exon; in most, but not all promoters)
- TSS with TATA box (≈ -30 bp upstream)
- Exon content in UTR 5" region
- Distance to first donor splice site

Idea: Combine weak features to build strong promoter predictor.

Approach: SVM

SVM classifier

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} y_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b\right)$$

Key ingredient is kernel $k(\mathbf{x}, \mathbf{x}')$ – similarity of two sequences.

Natural idea

Use 5 kernels to model each of the heterogeneous features.

The 5 Kernels

1. TSS signal (including parts of core promoter with TATA box)
   – use **Weighted Degree Shift kernel**
2. CpG Islands, distant enhancers and TFBS upstream of TSS
   – use **Spectrum kernel** (large window upstream of TSS)
3. model coding sequence TFBS downstream of TSS
   – use **Spectrum kernel** (small window downstream of TSS)
4. stacking energy of DNA
   – use *btwist* energy of dinucleotides with **Linear kernel**
5. twistedness of DNA
   – use btwist angle of dinucleotides with **Linear kernel**

Combining Classifiers

- Train individual classifiers and combine outputs by voting methods, SVMs, ...
- Aggregate feature spaces and define ONE kernel.

Problems

- Data scaling problem.
- Combination of kernels.

Recall

If k_1 and k_2 are kernels so are

- αk_1, if $\alpha \geq 0$
- $k_1 + k_2$
- $k_1 k_2$
- ... [Schölkopf and Smola, 2002]

Multiple Kernel Learning (MKL)

Possible solution

Linear combination of kernels,

$$k(\mathbf{x}, \mathbf{x}') = \sum_{p=1}^{M} \beta_p k_p(\mathbf{x}, \mathbf{x}'), \;\; \beta_p \geq 0.$$

Discussion

- Choice of weights is a way of feature selection.
- Optimal weights teach us something about problem.

Learning weights from data

- Automatic feature selection.
- Interpretability.

SVMs with kernels (Primal)

Minimize

$$\frac{1}{2}\sum_{i,j}^{N}\alpha_i\alpha_j k(\mathbf{x}_i,\mathbf{x}_j)+C\sum_{i=1}^{N}\xi_i$$

Subject to

$$y_i\left(\sum_{j=1}^{N}\alpha_j k(\mathbf{x}_i,\mathbf{x}_j)+b\right)\geqslant 1-\xi_i$$
$$\xi_i\geqslant 0$$
for all $i=1,\ldots,N$.

Minimize

$$\frac{1}{2}\|\mathbf{w}\|_2^2+C\sum_{i=1}^{N}\xi_i \quad (1)$$

Subject to

$$y_i\left(\mathbf{w}^\top\Phi(\mathbf{x}_i)+b\right)\geqslant 1-\xi_i \quad (2)$$
$$\xi_i\geqslant 0$$
for all $i=1,\ldots,N$.

(1) Maximize the margin (regularization).

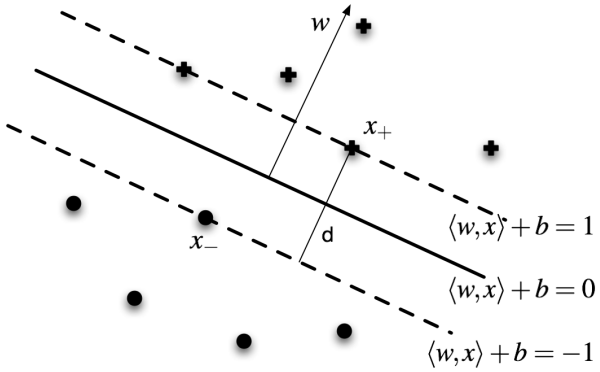
(2) Correctly classify the training data.

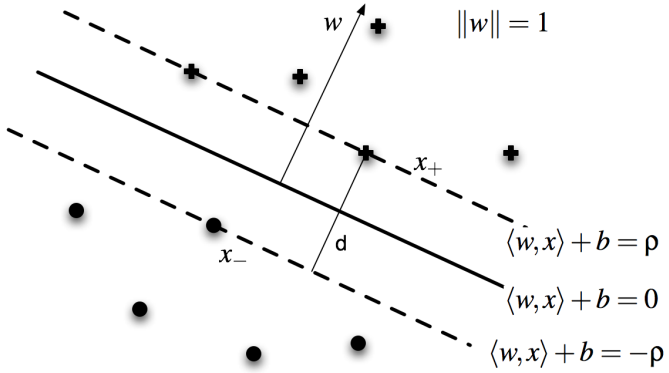
Representer Theorem: $\mathbf{w}^*=\sum_{n=1}^{N}\alpha_n\Phi(\mathbf{x}_n)$

because of l_2 norm $\|\mathbf{w}\|_2$.

Margin Maximization

Two equivalent formulations.





Minimize $\|\mathbf{w}\|_2$

Subject to $y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geqslant 1$
for all $i = 1, \ldots, N$.

Maximize ρ

Subject to $y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geqslant \rho$,
for all $i = 1, \ldots, N$,
$\|\mathbf{w}\|_2 = 1$.

Norms and Sparsity

l_2-Norm

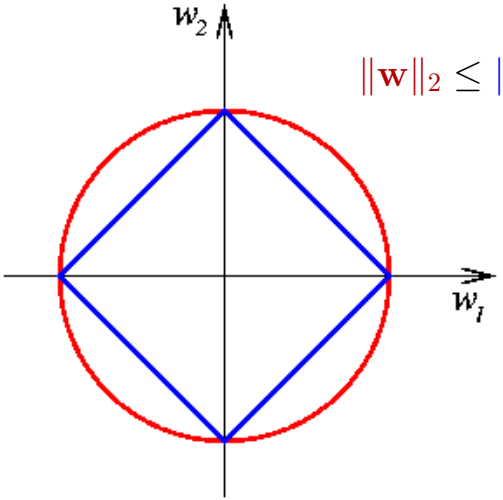
Maximize ρ

Subject to $y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geqslant \rho,$
for all $i = 1, \ldots, N,$
$\|\mathbf{w}\|_2 = 1.$

l_1-Norm

Maximize ρ

Subject to $y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geqslant \rho,$
for all $i = 1, \ldots, N,$
$\|\mathbf{w}\|_1 = 1.$



$\|\mathbf{w}\|_2 \leq \|\mathbf{w}\|_1$

Using l_1 norm leads to sparser solution!

SVMs with kernels (Primal)

Minimize

$$\frac{1}{2}\sum_{i,j}^{N}\alpha_i\alpha_j k(\mathbf{x}_i,\mathbf{x}_j) + C\sum_{i=1}^{N}\xi_i$$

Subject to

$$y_i\left(\sum_{j=1}^{N}\alpha_j k(\mathbf{x}_i,\mathbf{x}_j) + b\right) \geqslant 1-\xi_i$$
$$\xi_i \geqslant 0$$
for all $i = 1,\ldots,N$.

Minimize

$$\frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i$$

Subject to

$$y_i\left(\mathbf{w}^\top\Phi(\mathbf{x}_i) + b\right) \geqslant 1-\xi_i$$
$$\xi_i \geqslant 0$$
for all $i = 1,\ldots,N$.

MKL: $$k(\mathbf{x}_i,\mathbf{x}_j) = \sum_{p=1}^{M}\beta_p k_p(\mathbf{x}_i,\mathbf{x}_j),\ \beta_p \geq 0.$$

MKL Optimization Problem

$$\min \quad \frac{1}{2}\left(\sum_{p=1}^{M}\beta_p \left\|\mathbf{w}_p\right\|_2\right)^2 + C\sum_{i=1}^{N}\xi_n$$

$$\text{w.r.t.} \quad \mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_M), \mathbf{w}_p \in \mathbb{R}^{k_p}, \boldsymbol{\xi} \in \mathbb{R}_+^N, \boldsymbol{\beta} \in \mathbb{R}_+^M, b \in \mathbb{R}$$

$$\text{s.t.} \quad y_i\left(\sum_{p=1}^{M}\beta_p \mathbf{w}_p^\top \Phi(\mathbf{x}_{i,p}) + b\right) \geq 1 - \xi_i, \forall i = 1, \ldots, N$$

Problem

Minimization over β_p shrinks them to zero!

$\Rightarrow$ Constraining β_p is necessary.

Constraining the Weights

l_2-Norm

Dense $\|\boldsymbol{\beta}\|_2 = 1$ (Lin and Zhang 2004)

l_1-Norm

Sparse $\|\boldsymbol{\beta}\|_1 = 1$ (Bach, Lanckriet and Jordan 2004)

- Convex combination of kernels.
- Sparse solution in terms of kernels.
- Allows for interpretation of result.

Constraints on β_p

$$\sum_{p=1}^{M} \beta_p = 1, \ \beta_p \geq 0$$

MKL Optimization Problem I

Primal Formulation (Bach, Lanckriet, Jordan 2004)

$$\min \quad \frac{1}{2}\left(\sum_{p=1}^{M}\beta_p \left\|\mathbf{w}_p\right\|_2\right)^2 + C\sum_{i=1}^{N}\xi_n$$

$$\text{s.t.} \quad y_i\left(\sum_{p=1}^{M}\beta_p \mathbf{w}_p^\top \Phi(\mathbf{x}_{i,p}) + b\right) \geq 1-\xi_i, \forall i = 1,\ldots,N$$

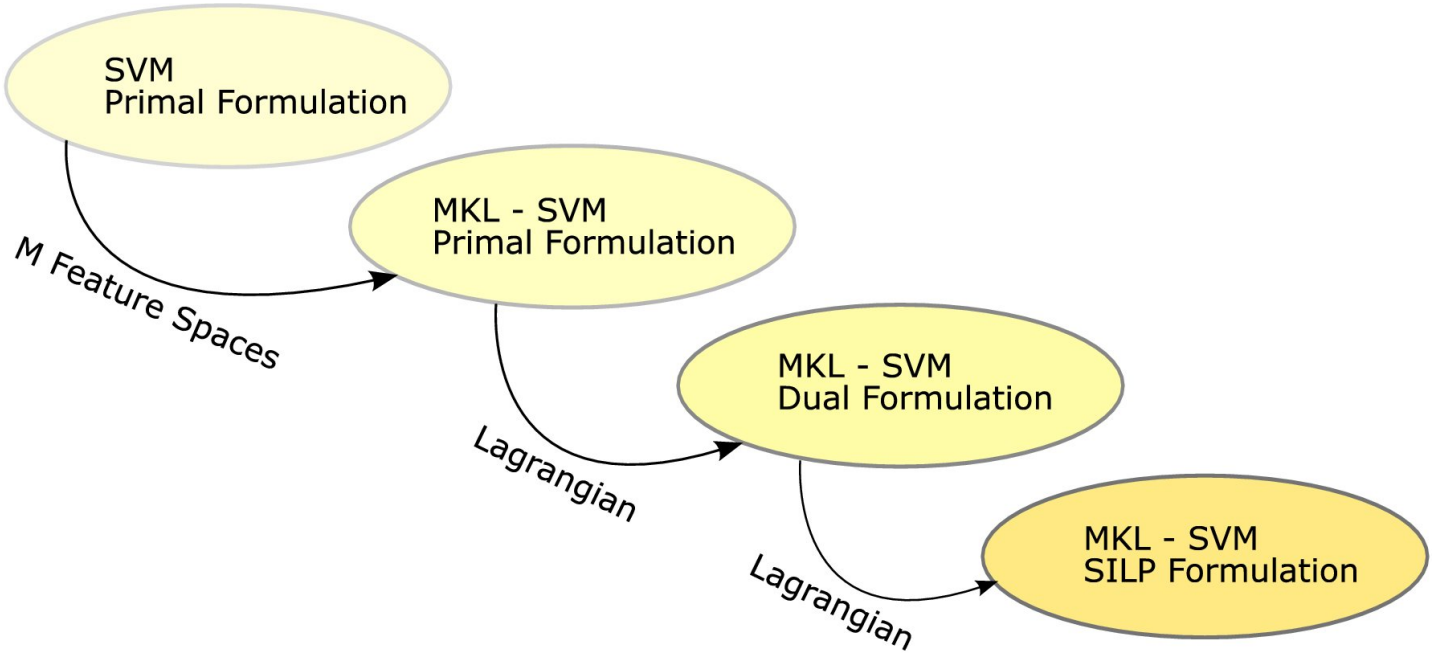
$$\sum_{p=1}^{M}\beta_p = 1, \;\; \beta_p \geq 0$$

Properties

- Equivalent to linear SVM for $M = 1$.
- Solution sparse in "blocks".
- Each block p corresponds to one kernel.

Rewriting the MKL formulation

From the Standard SVM Primal to the Semi-Infinite Linear Program



MKL Optimization Problem II

Dual Formulation (Bach, Lanckriet, Jordan 2004)

$$\begin{aligned}
\min \quad & \frac{1}{2}\gamma^2 - \sum_{i=1}^{N} \alpha_i \\
\text{w.r.t.} \quad & \gamma \in \mathbb{R}, \boldsymbol{\alpha} \in \mathbb{R}^N \\
\text{s.t.} \quad & 0 \leq \boldsymbol{\alpha} \leq C, \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& \underbrace{\sum_{r=1}^{N}\sum_{s=1}^{N} \alpha_r \alpha_s y_r y_s K_p(\mathbf{x}_r, \mathbf{x}_s)}_{=:S_p(\boldsymbol{\alpha})} - \gamma^2 \leq 0, \ \forall p = 1, \ldots, M
\end{aligned}$$

"partial Lagrangian:"

$$L := \frac{1}{2}\gamma^2 - \sum_{i=1}^{N} \alpha_i + \sum_{p=1}^{M} \beta_p (S_p(\boldsymbol{\alpha}) - \gamma^2)$$

MKL Optimization Problem III

Reformulation as Semi-Infinite Linear Program

$$\max_{\boldsymbol{\beta}} \min_{\boldsymbol{\alpha}} \sum_{p=1}^{M} \beta_p \left(\frac{1}{2} S_p(\boldsymbol{\alpha}) - \sum_{i=1}^{N} \alpha_i \right)$$

$$\text{s.t. } 0 \leq \boldsymbol{\alpha} \leq C, \sum_{i=1}^{N} \alpha_i y_i = 0, \sum_{p=1}^{M} \beta_p = 1$$

$$\begin{aligned} \max \quad & \theta \\ \text{w.r.t.} \quad & \theta \in \mathbb{R}, \boldsymbol{\beta} \in \mathbb{R}_+^M \text{ with } \sum_{p=1}^{M} \beta_p = 1 \\ \text{s.t.} \quad & \sum_{p=1}^{M} \beta_p \left(\frac{1}{2} S_p(\boldsymbol{\alpha}) - \sum_{i=1}^{N} \alpha_i \right) \geq \theta \\ & \text{for all } \boldsymbol{\alpha} \text{ with } 0 \leq \boldsymbol{\alpha} \leq C \text{ and } \sum_{i=1}^{N} y_i \alpha_i = 0 \end{aligned}$$

$$\sum_{p=1}^{M} \beta_p \left(\frac{1}{2} S_p(\boldsymbol{\alpha}) - \sum_{i=1}^{N} \alpha_i \right) = \frac{1}{2} \sum_{r=1}^{N} \sum_{s=1}^{N} \alpha_r \alpha_s y_r y_s \sum_{p=1}^{M} \beta_p k_p(\mathbf{x}_r, \mathbf{x}_s) - \sum_{i=1}^{N} \alpha_i,$$

Semi-Infinite Linear Program

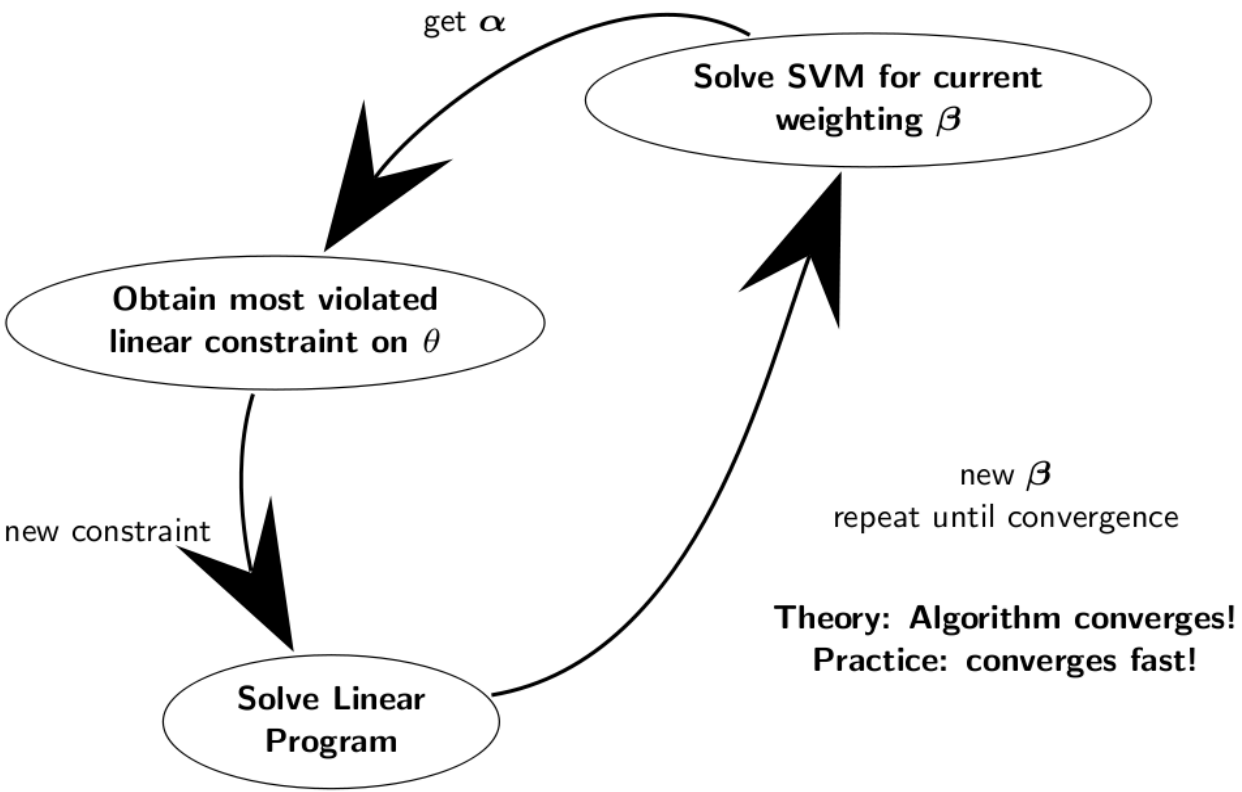
$$
\begin{aligned}
\max \quad & \theta \\
\text{w.r.t.} \quad & \theta \in \mathbb{R}, \boldsymbol{\beta} \in \mathbb{R}_+^M \text{ with } \sum_{p=1}^{M} \beta_p = 1 \\
\text{s.t.} \quad & \sum_{p=1}^{M} \beta_p \left(\frac{1}{2} \sum_{r=1}^{N} \sum_{s=1}^{N} \alpha_r \alpha_s y_r y_s K_p(\mathbf{x}_r, \mathbf{x}_s) - \sum_{i=1}^{N} \alpha_i \right) \geq \theta \\
& \text{for all } \boldsymbol{\alpha} \text{ with } 0 \leq \boldsymbol{\alpha} \leq C \text{ and } \sum_{i=1}^{N} y_i \alpha_i = 0
\end{aligned}
$$

$\Rightarrow$ **Linear Problem, but infinitely many constraints**

- One constraint for each $\boldsymbol{\alpha}$.
- Can be solved using Column Generation.
- Most violated constraints can be identified by solving SVM.

Column Generation

Sonnenburg et al. [2006b]



Summary

MKL Algorithm

- Automatically computes best convex combination of kernels

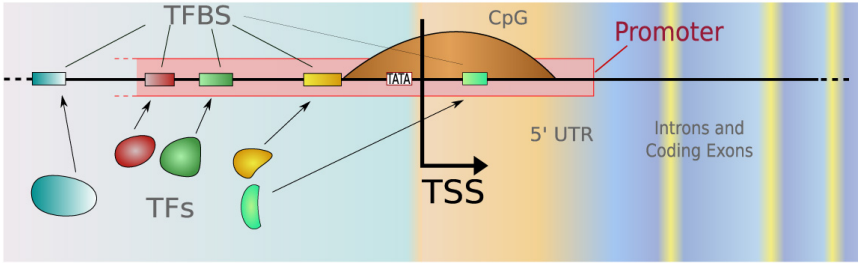
$$k(\mathbf{x}, \mathbf{x}') = \sum_{p=1}^{M} \beta_p k_p(\mathbf{x}, \mathbf{x}'), \quad \sum_{p=1}^{M} \beta_p = 1, \ \beta_p \geq 0.$$

- SILP formulation makes large scale training and evaluation possible.

Possible Applications

- Heterogeneous data.
- Improving interpretability.

Promoter Detection with MKL



1. TSS signal – **Weighted Degree Shift kernel**
2. Features upstream of TSS – **Spectrum kernel**
3. Features downstream of TSS – **Spectrum kernel**
4. Stacking energy of DNA – **Linear kernel**
5. Twistedness of DNA – **Linear kernel**

MKL to combine different kernels.
Large scale training and evaluation possible.

Data Generation

True TSS:

- From dbTSSv4 (based on hg16) extract putative TSS windows of size $[-1000, +1000]$.

Decoy TSS:

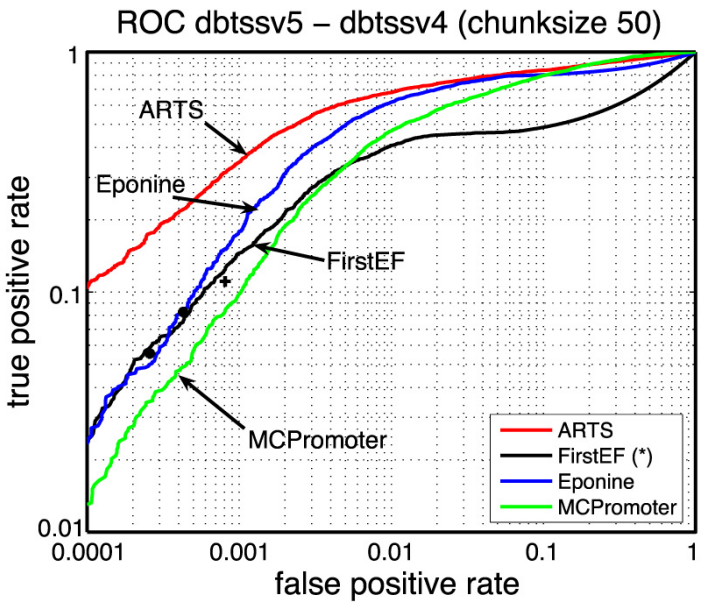
- Annotate dbTSSv4 with transcription-stop (via $BLAT$ alignment of mRNAs).
- From the interior of the gene ($+100bp$ to gene end) sample negatives for training (10 per positive), again windows $[-1000, +1000]$.

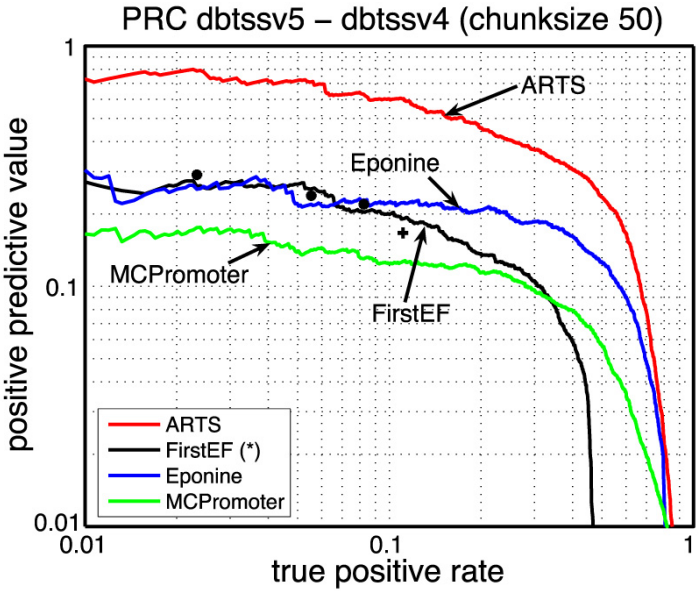
Processing:

- 8508 positive, 85042 negative examples.
- Split into disjoint training and validation set ($50\% : 50\%$).

Results

Receiver Operator Characteristic Curve *and* Precision Recall Curve

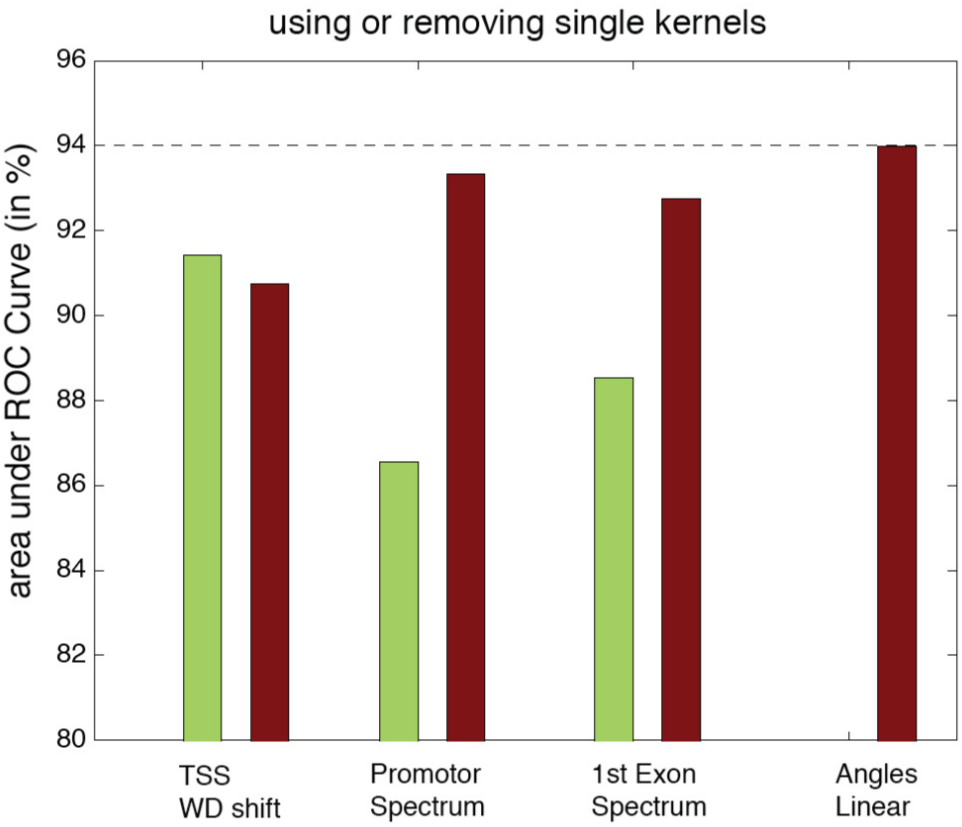




$\Rightarrow 35\%$ **true positives at a false positive rate of** $1/1000$
(best other method find about a half (18%**))**

See Sonnenburg et al. [2006c] for more details.

Which kernel is most important?



$\Rightarrow$ **Weighted Degree Shift kernel modeling TSS signal**

MKL and Interpretability

Different scaling of heterogeneous feature spaces leads to different weightings!

MKL for Promoter Detection

- State-of-the-art recognition accuracy.
- Elegant and flexible formulation.
- Easy and principled combination of heterogeneous features.
- NO interpretation!

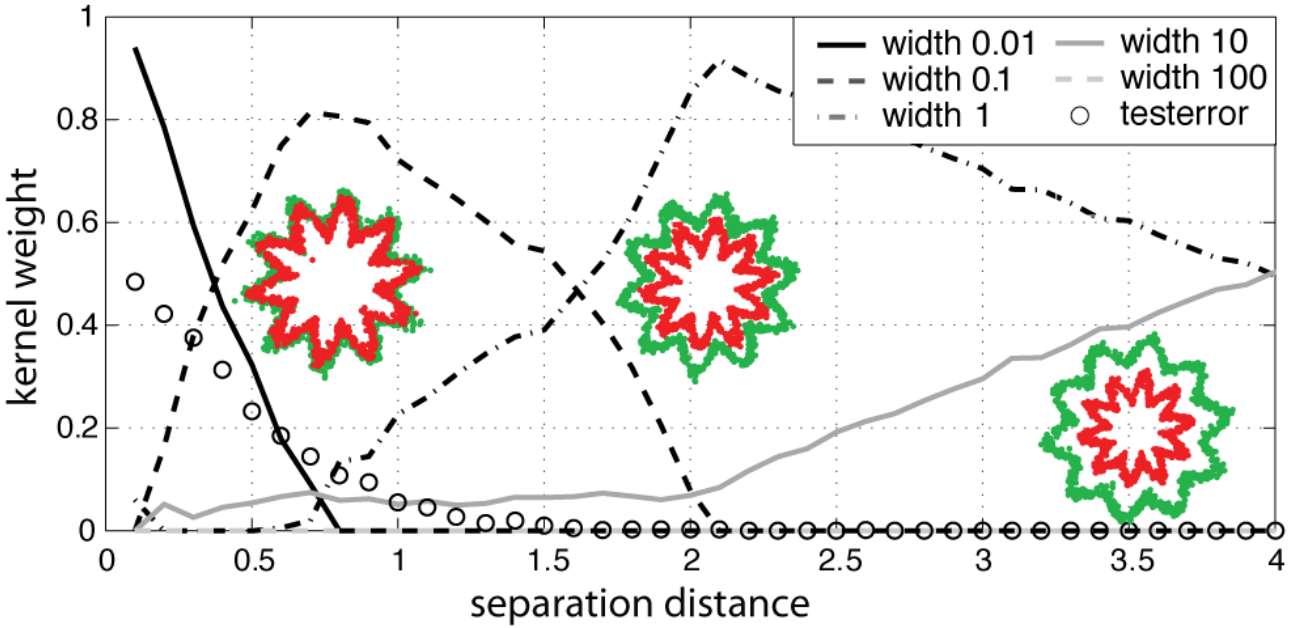
A Closer Look

How about homogeneous feature spaces?

Toy Application I

MKL for Classification

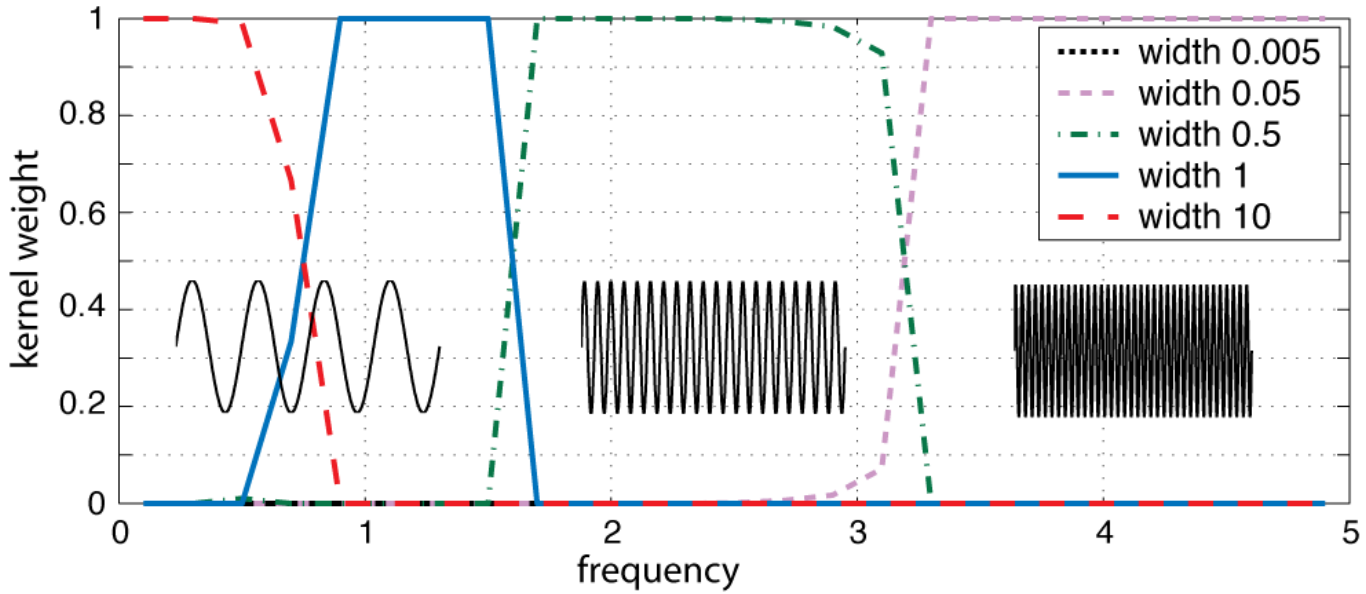
- Separation of two star-shaped classes with different class distances.
- Combination of 5 RBF-kernels with different widths.



Toy Application II

MKL for Regression

- Learning of a sinus wave.
- Combination of 5 RBF-kernels with different widths.



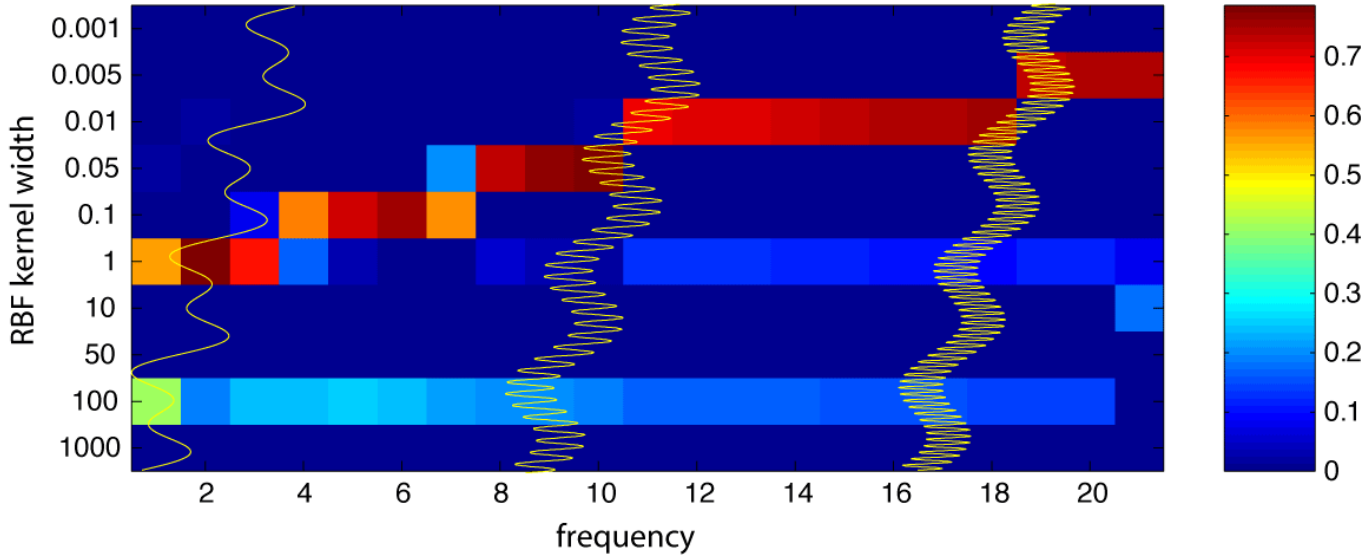
Toy Application III

MKL for Regression

- Learning of a complex function with varying frequency a

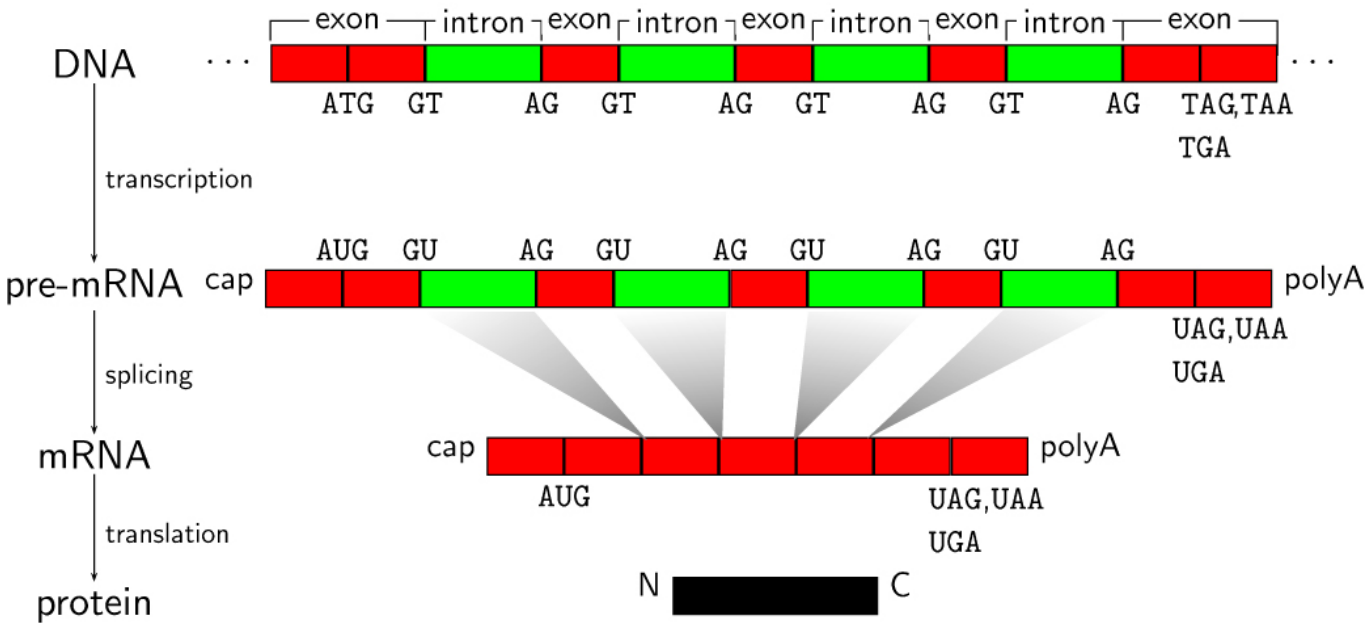
$$f(x) = \sin(ax) + \sin(bx) + cx$$

- Combination of 10 RBF-kernels with different widths.



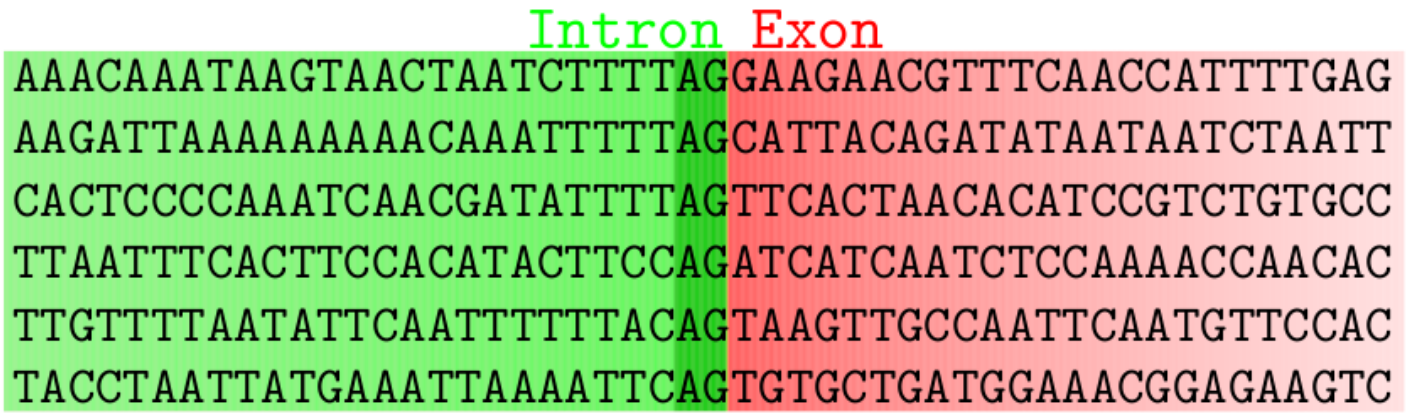
Splice Site Detection

Splice sites are locations on DNA at boundaries of
- exons (which code for proteins)
- introns (which do not)



Detection of Splice Sites

Focus on Acceptor Splice Site Prediction



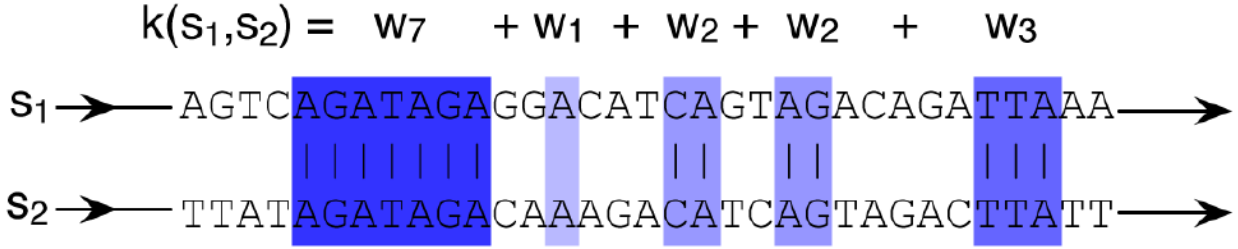
- Aligned sequences of fixed length (AG always at same position).
- Task: distinguish *C. elegans* splice sites from fake splice sites.

$\Rightarrow$ 2-class classification problem by using substring comparisons.

SVM for Splice Site Detection

Weighted Degree kernel

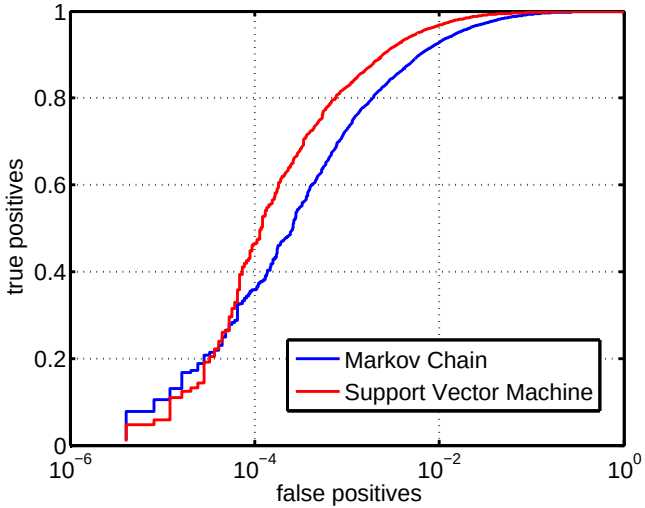
- Count matching substrings of length upto D at each position.



HMM vs. SVM

Standard Approach: Use two higher order Markov Chains

- one model for fake splice sites and one for true splice sites trained on $\approx 600,000$ examples, of them only $\approx 36,000$ true splice sites



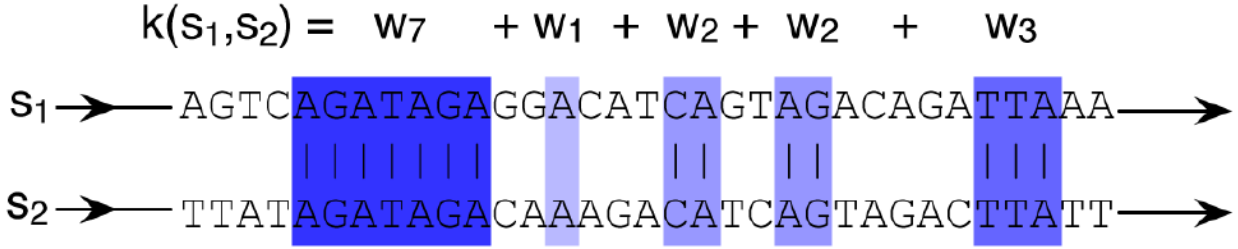
SVM classification results are better!

Drawback: We loose interpretability of the result!

MKL for Splice Site Detection

Weighted Degree kernel

- Count matching substrings of length upto D at each position.



- Linear combination of LD kernels

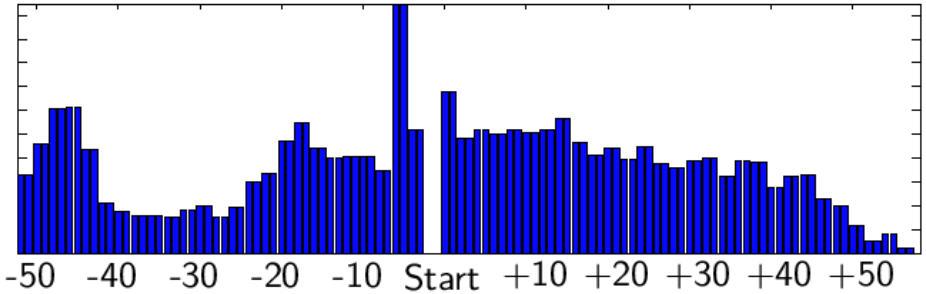
$$k(\mathbf{x}, \mathbf{x}') = \sum_{d=1}^{D} \sum_{l=1}^{L-d+1} \gamma_{l,d} \mathbf{I}(\mathbf{u}_{l,d}(\mathbf{x}) = \mathbf{u}_{l,d}(\mathbf{x}')).$$

- Learn weights $\gamma_{l,d}$ with MKL.
- Use weights $\gamma_{l,d}$ to interprete SVM decision.

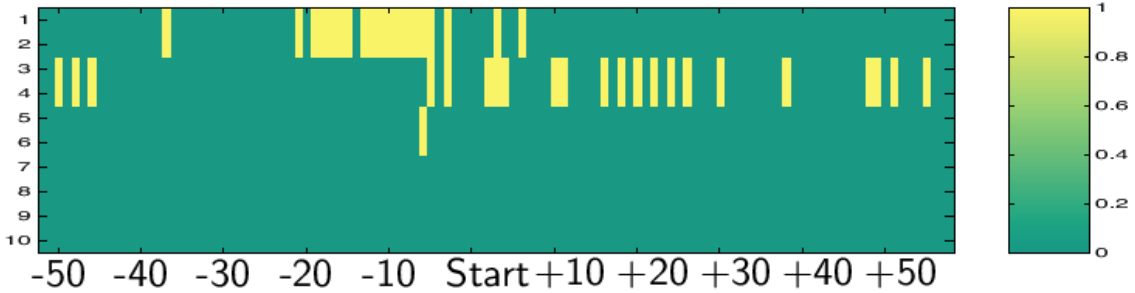
Understanding the SVM Decision

Splice Sites [Rätsch et al., 2006]

1. Which positions in the sequence are important for discrimination?



2. What characterizes those positions?



3. Which motifs at which position are important?

Understanding the SVM Decision

Positional Oligomer Importance Matrix (POIM)
[Sonnenburg et al., 2006a]

For each k-mer at each position

- Weight contribution of the k-mer observed at this certain position.
- Weight contribution for all higher order overlapping k-mers.
- Efficiently computed for WD kernels using suffix tries representation.

Toy Application

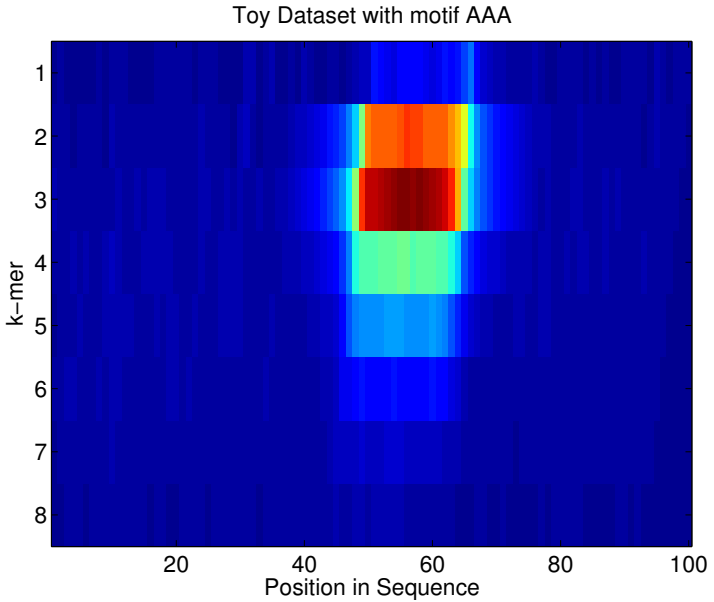
- Finding a motif AAA at varying positions.

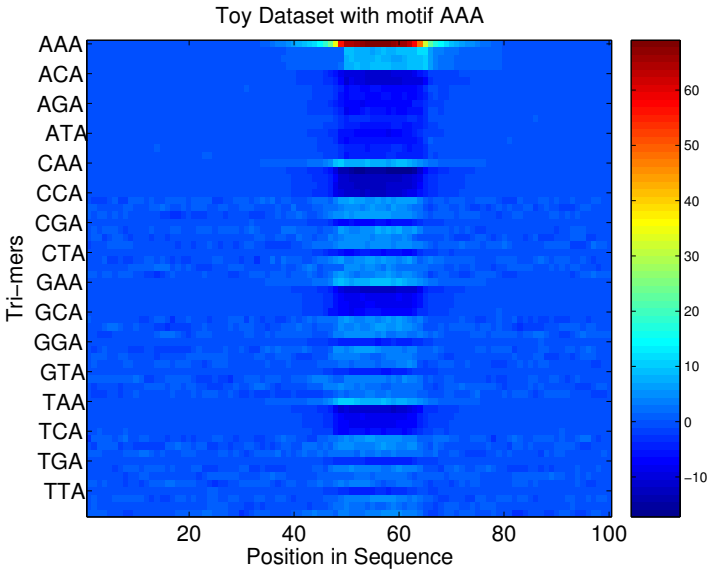
```
AAACAAATAAGTAACTAATCTTTTAAAGAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAAACAAATTTCATTAAATACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTAAATTTCACTAACACATCCGTCTGTGC
```

- WD kernel with shifts.



POIM for Motif Finding

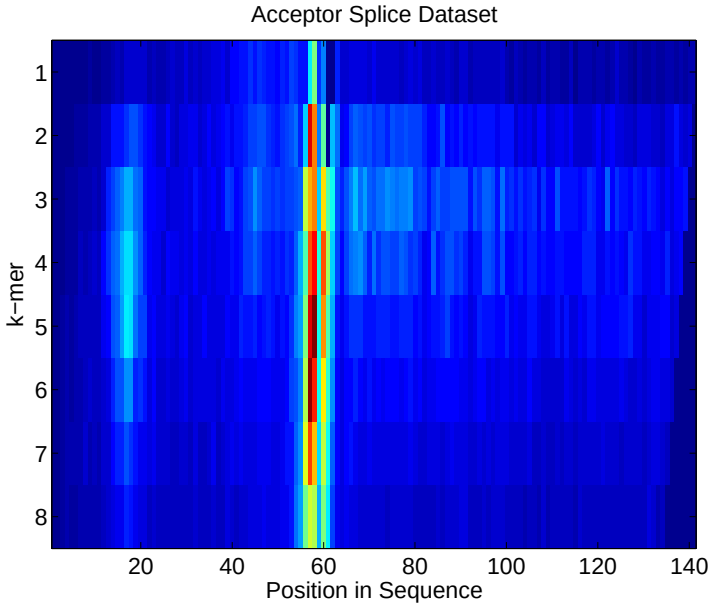


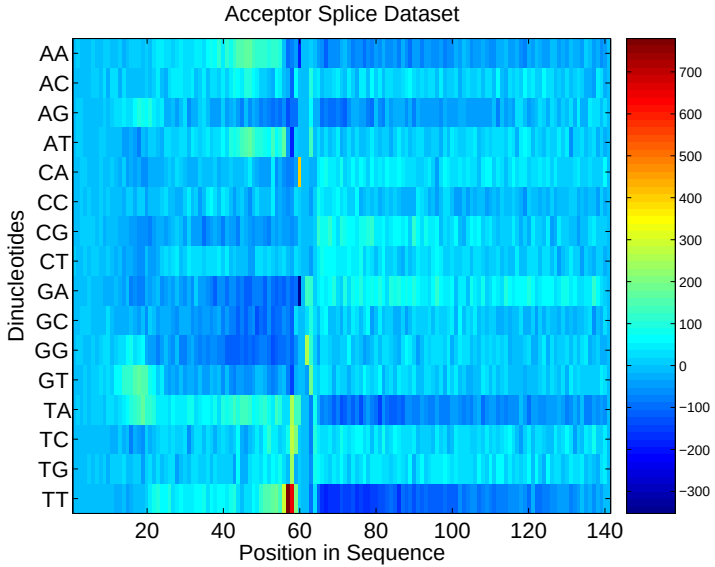


Splice Site Detection

POIM for Interpretation

- WD kernel with shifts.





Summary

Multiple Kernel Learning (MKL)

- Allows automatic training of convex combinations of kernels.
- Useful for combining heterogeneous features.
- SILP formulation makes large scale applications possible.
- Weightings allow interpretation in case of homogeneous features.

Positional Oligomer Importance Matrix (POIM)

- Efficient computation.
- Allow interpretation of SVM decision.

References

G. Rätsch, S. Sonnenburg, and C. Schäfer. Learning interpretable svms for biological sequence classification. *BMC Bioinformatics*, 7 (Suppl 1):S9, February 2006.

B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

Sören Sonnenburg, Gunnar Rätsch, and Konrad Rieck. Large scale learning with string kernels. In O. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, *Large Scale Kernel Machines*. MIT Press, 2006a. In press.

Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research*, 7:1531–1565, July 2006b.

Sören Sonnenburg, Alexander Zien, and Gunnar Rätsch. ARTS: Accurate Recognition of Transcription Starts in Human. *Bioinformatics*, 22(14):e472–480, 2006c.