Combining Kernels for Classification

Doctoral Thesis Seminar

Darrin P. Lewis

dplewis@cs.columbia.edu

Outline

- **Summary of Contribution**
- Stationary kernel combination
- Nonstationary kernel combination
- Sequential minimal optimization
- Results
- Conclusion

Summary of Contribution

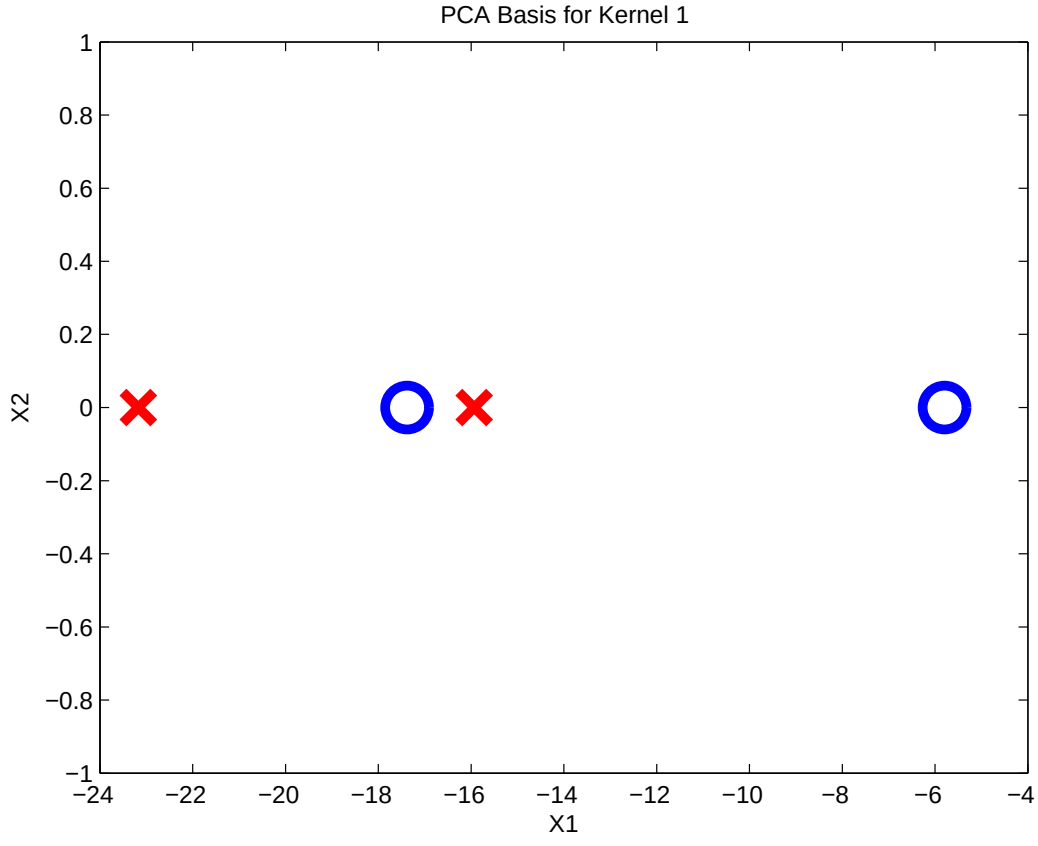
- Empirical study of kernel averaging versus SDP weighted kernel combination
- Nonstationary kernel combination
- Double Jensen bound for latent MED
- Efficient iterative optimization
- Implementation

Outline

- Summary of Contribution
- **Stationary kernel combination**
- Nonstationary kernel combination
- Sequential minimal optimization
- Results
- Conclusion

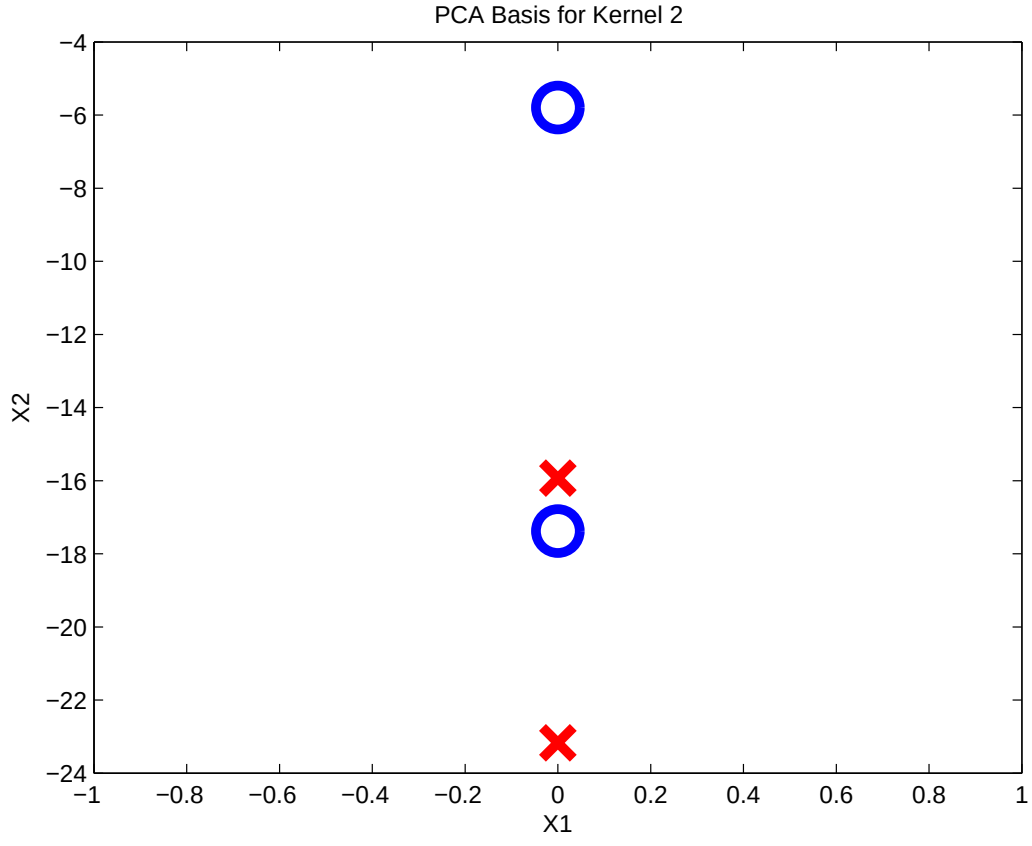
Example Kernel One

$$\begin{bmatrix} 1 & 4 & 2.75 & 3 \\ 4 & 16 & 11 & 12 \\ 2.75 & 11 & 7.5625 & 8.25 \\ 3 & 12 & 8.25 & 9 \end{bmatrix}$$



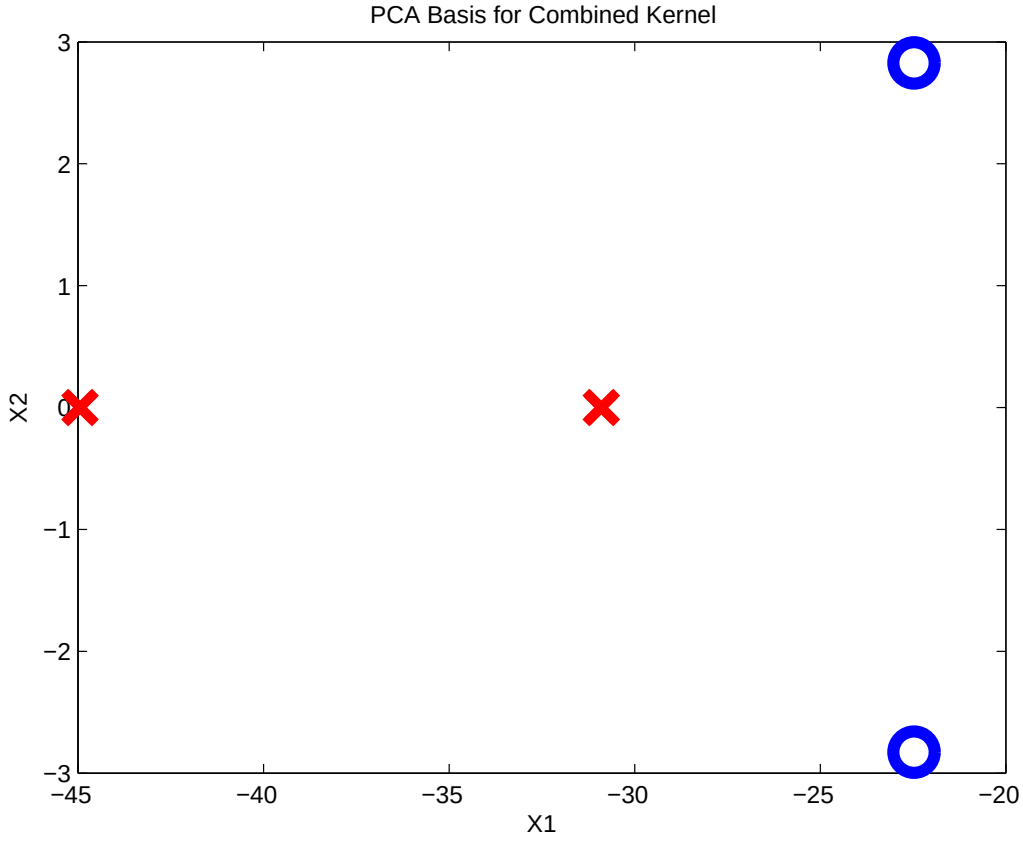
Example Kernel Two

$$\begin{bmatrix} 9 & 12 & 8.25 & 3 \\ 12 & 16 & 11 & 4 \\ 8.25 & 11 & 7.5625 & 2.75 \\ 3 & 4 & 2.75 & 1 \end{bmatrix}$$



Example Kernel Combination

$$\begin{bmatrix} 10 & 16 & 11 & 6 \\ 16 & 32 & 22 & 16 \\ 11 & 22 & 15.125 & 11 \\ 6 & 16 & 11 & 10 \end{bmatrix}$$



Effect of Combination

$$
\begin{aligned}
K_C(x,z) &= K_1(x,z) + K_2(x,z) \\
&= \langle \phi_1(x), \phi_1(z) \rangle + \langle \phi_2(x), \phi_2(z) \rangle \\
&= \langle \phi_1(x) : \phi_2(x), \phi_1(z) : \phi_2(z) \rangle
\end{aligned}
$$

- The implicit feature space of the combined kernel is a concatenation of the feature spaces of the individual kernels.
- A basis in the combined feature space may be lower dimensional than the sum of the dimensions of the individual feature spaces.

Combination Weights

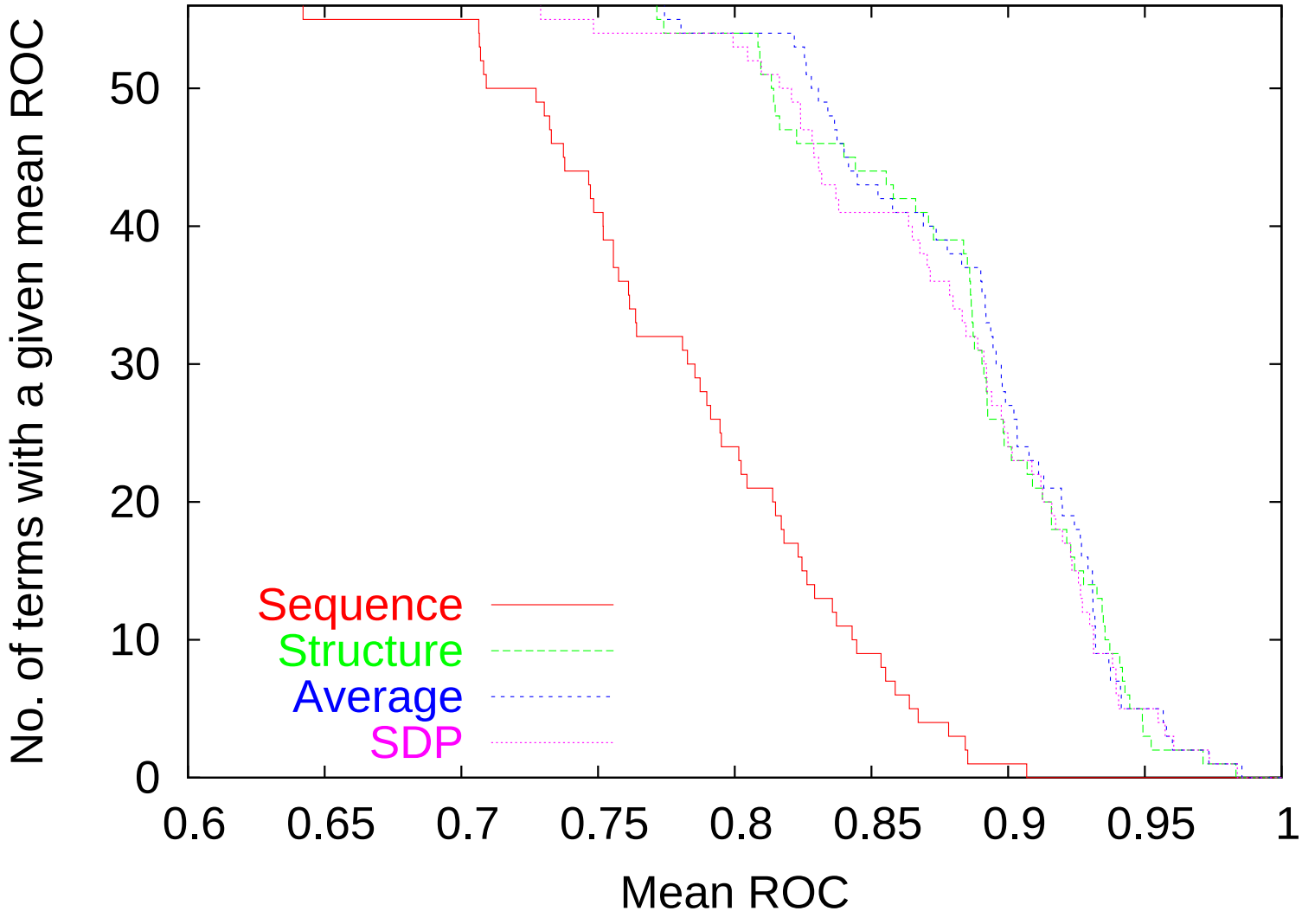
There are several ways in which the combination weights can be determined:

- *equal weight*: or unweighted combination. This is also essentially kernel averaging[14].
- *optimized weight*: SDP weighted combination[6]. Weights and SVM Lagrange multipliers are determined in a single optimization. To regularize the kernel weights, a constraint is enforced to keep the trace of the combined kernel constant.

Sequence/Structure

- We compare[10] the state-of-the-art SDP and simple averaging for conic combinations of kernels
- Drawbacks of SDP include optimization time and lack of a free implementation
- We determined the cases in which averaging is preferable and those in which SDP is required
- Our experiments predict Gene Ontology[2] (GO) terms using a combination of amino acid sequence and protein structural information
- We use the 4,1-Mismatch sequence kernel[8] and MAMMOTH (sequence-independent) structure kernel[13]

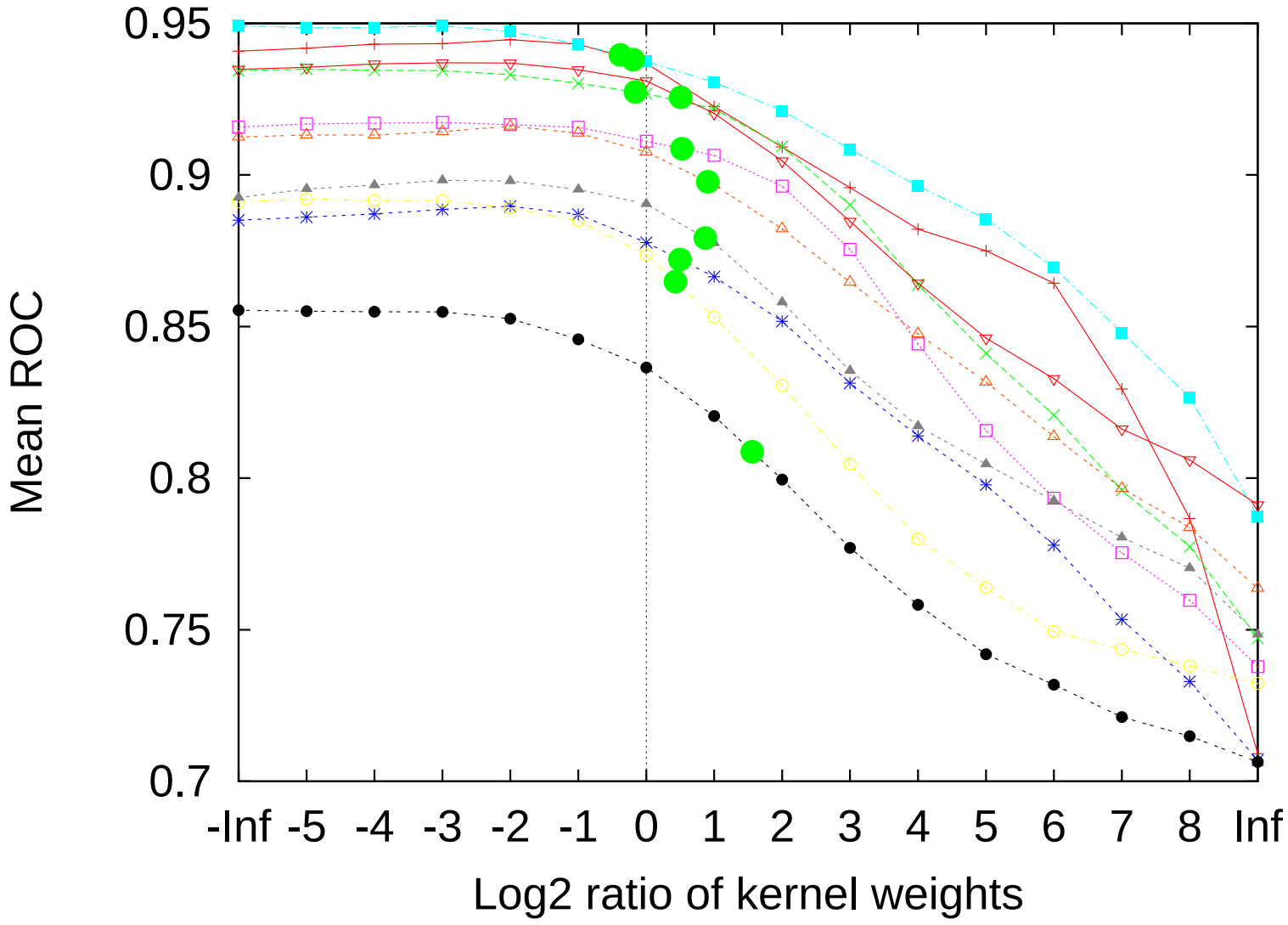
Cumulative ROC AUC



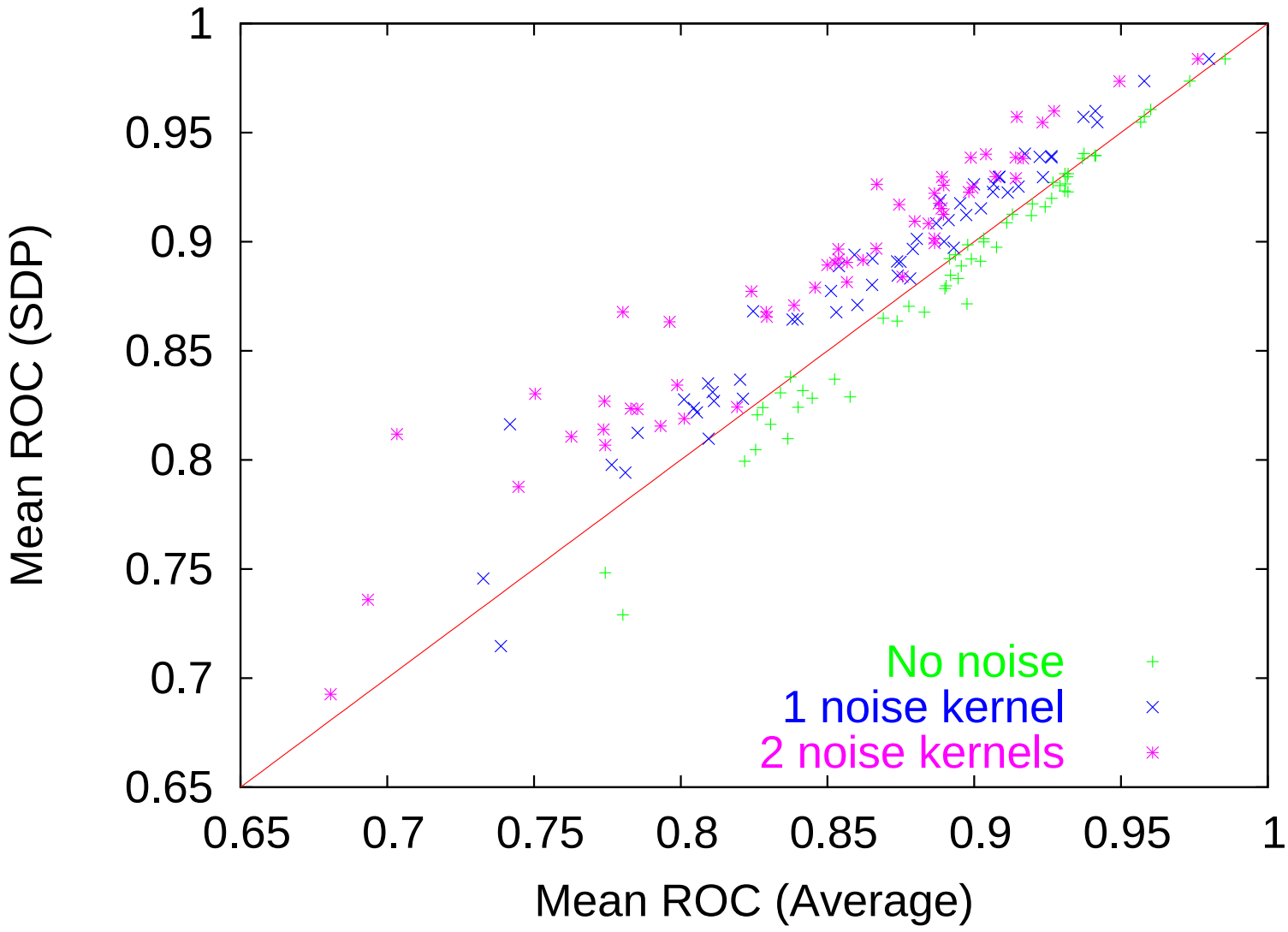
Mean ROC AUC Top 10 GO Terms

GO term	Structure	Sequence	Average	SDP
GO:0008168	0.941 ± 0.014	0.709 ± 0.020	0.937 ± 0.016	0.938 ± 0.015
GO:0005506	0.934 ± 0.008	0.747 ± 0.015	0.927 ± 0.012	0.927 ± 0.012
GO:0006260	0.885 ± 0.014	0.707 ± 0.020	0.878 ± 0.016	0.870 ± 0.015
GO:0048037	0.916 ± 0.015	0.738 ± 0.025	0.911 ± 0.016	0.909 ± 0.016
GO:0046483	0.949 ± 0.007	0.787 ± 0.011	0.937 ± 0.008	0.940 ± 0.008
GO:0044255	0.891 ± 0.012	0.732 ± 0.012	0.874 ± 0.015	0.864 ± 0.013
GO:0016853	0.855 ± 0.014	0.706 ± 0.029	0.837 ± 0.017	0.810 ± 0.019
GO:0044262	0.912 ± 0.007	0.764 ± 0.018	0.908 ± 0.006	0.897 ± 0.006
GO:0009117	0.892 ± 0.015	0.748 ± 0.016	0.890 ± 0.012	0.880 ± 0.012
GO:0016829	0.935 ± 0.006	0.791 ± 0.013	0.931 ± 0.008	0.926 ± 0.007
GO:0006732	0.823 ± 0.011	0.781 ± 0.013	0.845 ± 0.011	0.828 ± 0.013
GO:0007242	0.898 ± 0.011	0.859 ± 0.014	0.903 ± 0.010	0.900 ± 0.011
GO:0005525	0.923 ± 0.008	0.884 ± 0.015	0.931 ± 0.009	0.931 ± 0.009
GO:0004252	0.937 ± 0.011	0.907 ± 0.012	0.932 ± 0.012	0.931 ± 0.012
GO:0005198	0.809 ± 0.010	0.795 ± 0.014	0.828 ± 0.010	0.824 ± 0.011

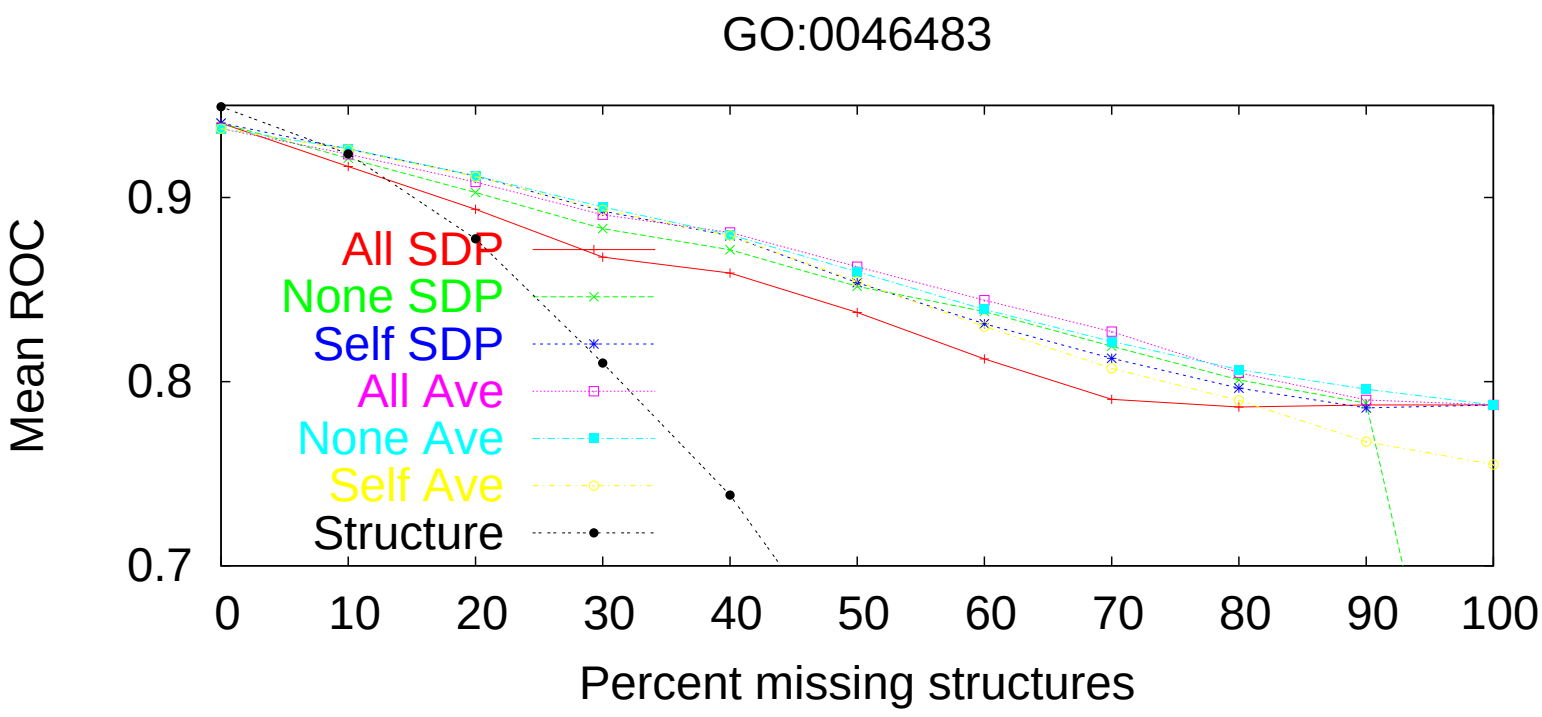
Varying Ratio Top 10 GO Terms



Noisy Kernels 56 GO Terms



Missing Data Typical GO Term



Outline

- Summary of Contribution
- Stationary kernel combination
- **Nonstationary kernel combination**
- Sequential minimal optimization
- Results
- Conclusion

Kernelized Discriminants

Single:

$$f(x) = \sum_t y_t \lambda_t k(x_t, x) + b$$

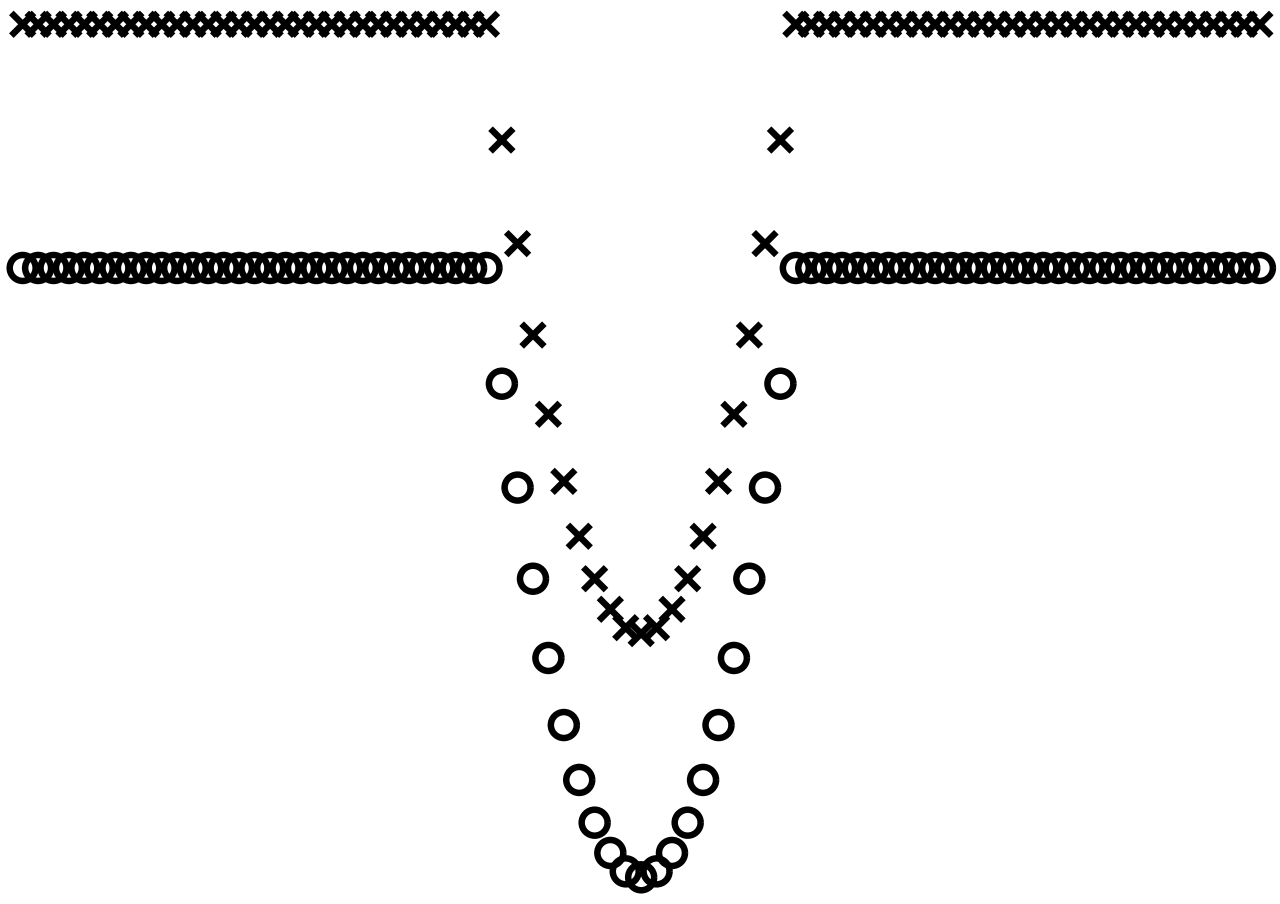
Linear combination:

$$f(x) = \sum_t y_t \lambda_t \sum_m \nu_m k_m(x_t, x) + b$$

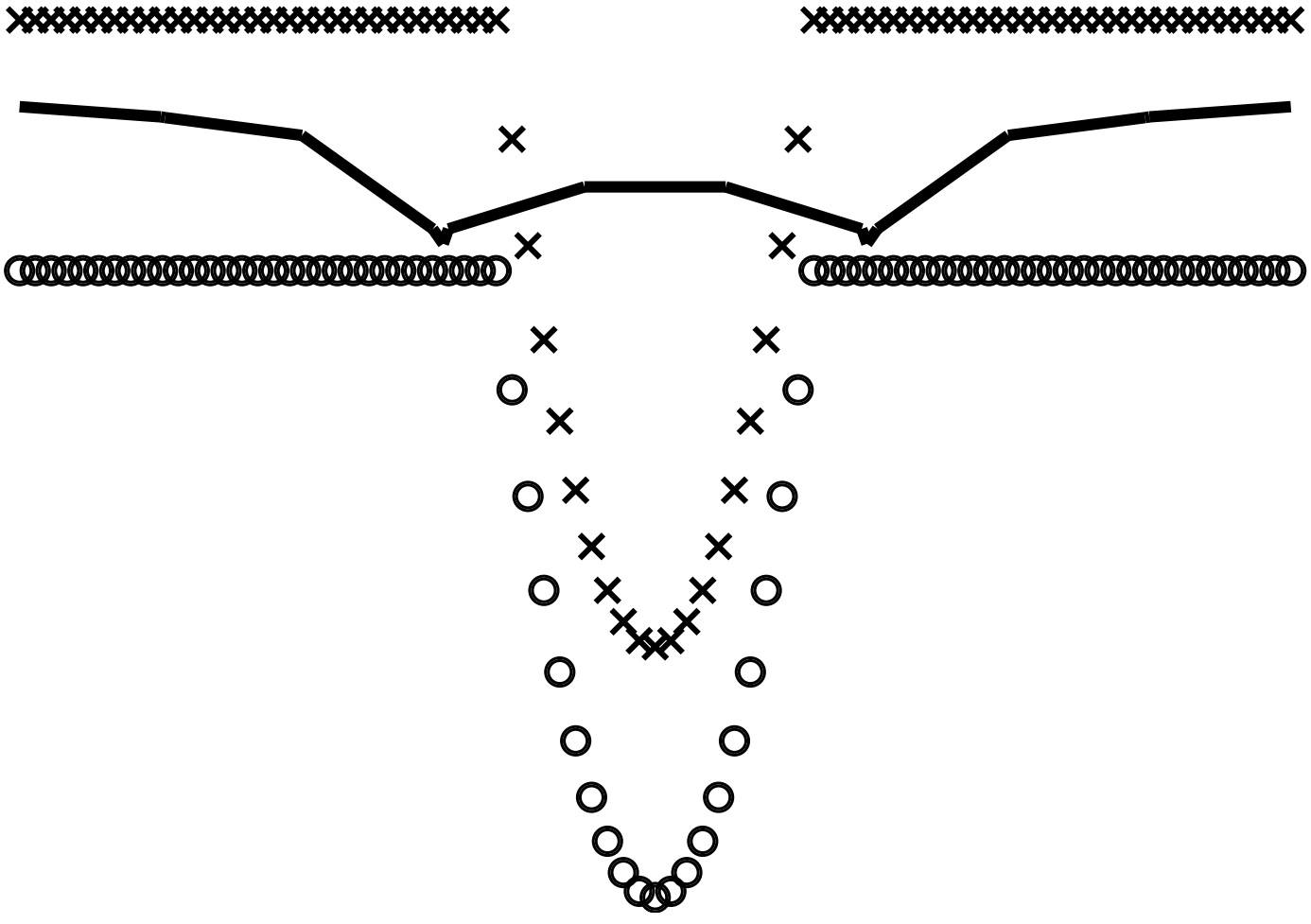
Nonstationary combination[9]:

$$f(x) = \sum_t y_t \lambda_t \sum_m \nu_{m,t}(x) k_m(x_t, x) + b$$

Parabola-Line Data



Parabola-Line SDP

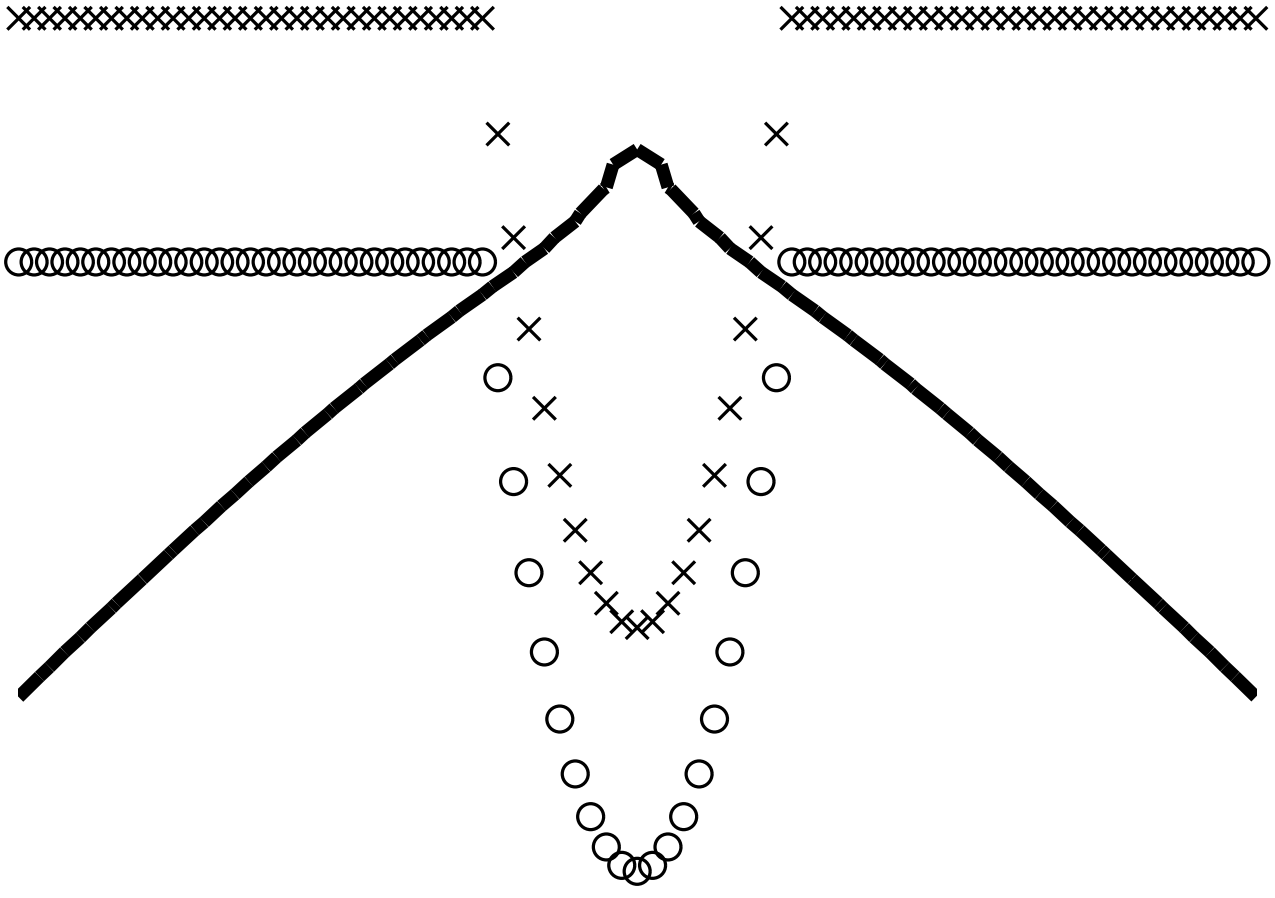


Ratio of Gaussian Mixtures

$$\mathcal{L}(X_t;\Theta) = \ln \frac{\sum_{m=1}^{M} \alpha_m \mathcal{N}(\phi_m^+(X_t)|\mu_m^+, I)}{\sum_{n=1}^{N} \beta_n \mathcal{N}(\phi_n^-(X_t)|\mu_n^-, I)} + b$$

- μ_m^+, μ_n^- Gaussian means
- α, β mixing proportions
- b scalar bias
- For now, maximum likelihood parameters are estimated independently for each model.
- Note explicit feature maps, ϕ^+, ϕ^-.

Parabola-Line ML



Ratio of Generative Models

$$\mathcal{L}(X_t;\Theta) = \ln \frac{\sum_{m=1}^{M} P(m, \phi_m^+(X_t)|\theta_m^+)}{\sum_{n=1}^{N} P(n, \phi_n^-(X_t)|\theta_n^-)} + b$$

- Find distribution $P(\Theta)$ rather than specific Θ^*
- Classify using $\hat{y} = \text{sign}\left(\int_\Theta P(\Theta)\mathcal{L}(X_t;\Theta)d\Theta\right)$

Max Ent Parameter Estimation

- Find $P(\Theta)$ to satisfy "moment" constraints:
  $\int_\Theta P(\Theta) y_t \mathcal{L}(X_t;\Theta) d\Theta \geq \gamma_t \quad \forall t \in \mathcal{T}$
  while assuming nothing additional.
- Minimize Shannon relative entropy:
  $D(P\|P^{(0)}) = \int_\Theta P(\Theta) \ln \frac{P(\Theta)}{P^{(0)}(\Theta)} d\Theta$
  to allow the use of a prior $P^{(0)}(\Theta)$.
- Classic ME solution[3] is:
  $P(\Theta) = \frac{1}{Z(\lambda)} P^{(0)}(\Theta) e^{\sum_{t\in\mathcal{T}} \lambda_t [y_t \mathcal{L}(X_t|\Theta) - \gamma_t]}$
- λ fully specifies $P(\Theta)$.
- Maximize log-concave objective $J(\lambda) = -\log Z(\lambda)$.

Tractable Partition

$$\ddot{Z}(\lambda, Q|q) = \int_{\Theta} P^{(0)}(\Theta)$$
$$\exp\Big(\sum_{t\in\mathcal{T}^+} \lambda_t \Big(\sum_m q_t(m) \ln P(m, \phi_m^+(X_t)|\theta_m^+) + H(q_t)$$
$$- \sum_n Q_t(n) \ln P(n, \phi_n^-(X_t)|\theta_n^-) - H(Q_t) + b - \gamma_t \Big) \Big)$$
$$\exp\Big(\sum_{t\in\mathcal{T}^-} \lambda_t \Big(\sum_n q_t(n) \ln P(n, \phi_n^-(X_t)|\theta_n^-) + H(q_t)$$
$$- \sum_m Q_t(m) \ln P(m, \phi_m^+(X_t)|\theta_m^+) - H(Q_t) - b - \gamma_t \Big) \Big) d\Theta$$

- Introduce variational distributions q_t over the correct class log-sums and Q_t over the incorrect class log-sums to replace them with upper and lower bounds, respectively.
- $\operatorname{argmin}_Q \operatorname{argmax}_q \ddot{Z}(\lambda, Q|q) = Z(\lambda)$
- Iterative optimization is required.

MED Gaussian Mixtures

$$\mathcal{L}(X_t;\Theta) = \ln \frac{\sum_{m=1}^{M} \alpha_m \mathcal{N}(\phi_m^+(X_t)|\mu_m^+, I)}{\sum_{n=1}^{N} \beta_n \mathcal{N}(\phi_n^-(X_t)|\mu_n^-, I)} + b$$

- Gaussian priors $\mathcal{N}(0, I)$ on μ_m^+, μ_n^-
- Non-informative Dirichlet priors on α, β
- Non-informative Gaussian $\mathcal{N}(0, \infty)$ prior on b.

These assumptions simplify the objective and result in a set of linear equality constraints on the convex optimization.

Convex Objective

$$
\begin{aligned}
\ddot{J}(\lambda, Q|q) \quad = \quad & \sum_{t\in\mathcal{T}} \lambda_t (H(Q_t) - H(q_t)) + \sum_{t\in\mathcal{T}} \lambda_t \gamma_t \\
& - \frac{1}{2} \sum_{t,t'\in\mathcal{T}^+} \lambda_t \lambda_{t'} \Big(\sum_m q_t(m) q_{t'}(m) k_m^+(t,t') \\
& \qquad + \sum_n Q_t(n) Q_{t'}(n) k_n^-(t,t') \Big) \\
& - \frac{1}{2} \sum_{t,t'\in\mathcal{T}^-} \lambda_t \lambda_{t'} \Big(\sum_m Q_t(m) Q_{t'}(m) k_m^+(t,t') \\
& \qquad + \sum_n q_t(n) q_{t'}(n) k_n^-(t,t') \Big) \\
& + \sum_{\substack{t\in\mathcal{T}^+ \\ t'\in\mathcal{T}^-}} \lambda_t \lambda_{t'} \Big(\sum_m q_t(m) Q_{t'}(m) k_m^+(t,t') \\
& \qquad + \sum_n Q_t(n) q_{t'}(n) k_n^-(t,t') \Big)
\end{aligned}
$$

Optimization

- For now, we discard the $H(Q_t)$ entropy terms.
- We redefine $\lambda \leftarrow Q\lambda$ and optimize with a quadratic program.
- Subsumes SVM (M=N=1)

The following constraints must be satisfied:

$$\sum_{t \in \mathcal{T}^-} \lambda_t Q_t(m) = \sum_{t \in \mathcal{T}^+} \lambda_t q_t(m) \;\; \forall m = 1 \ldots M$$

$$\sum_{t \in \mathcal{T}^+} \lambda_t Q_t(n) = \sum_{t \in \mathcal{T}^-} \lambda_t q_t(n) \;\; \forall n = 1 \ldots N$$

$$0 \leq \lambda_t \leq c \;\; \forall t = 1 \ldots T$$

Expected Gaussian LL

$$
\begin{aligned}
E\{\ln \mathcal{N}(\phi_m^+(X_t)|\mu_m^+)\} = & \\
& -\frac{D}{2}\ln(2\pi) - \frac{1}{2} - \frac{1}{2}k_m^+(X_t, X_t) \\
& + \sum_{\tau \in \mathcal{T}^+} \lambda_\tau q_\tau(m) k_m^+(X_\tau, X_t) \\
& - \sum_{\tau \in \mathcal{T}^-} \lambda_\tau Q_\tau(m) k_m^+(X_\tau, X_t) \\
& - \frac{1}{2} \sum_{\tau,\tau' \in \mathcal{T}^+} \lambda_\tau \lambda_{\tau'} q_\tau(m) q_{\tau'}(m) k_m^+(X_\tau, X_{\tau'}) \\
& - \frac{1}{2} \sum_{\tau,\tau' \in \mathcal{T}^-} \lambda_\tau \lambda_{\tau'} Q_\tau(m) Q_{\tau'}(m) k_m^+(X_\tau, X_{\tau'}) \\
& + \sum_{\substack{\tau \in \mathcal{T}^+ \\ \tau' \in \mathcal{T}^-}} \lambda_\tau \lambda_{\tau'} q_\tau(m) Q_{\tau'}(m) k_m^+(X_\tau, X_{\tau'})
\end{aligned}
$$

Expected Mixing/Bias LL

$$a_m = E\{\ln \alpha_m\} + \tfrac{1}{2}E\{b\} \quad \forall m = 1..M$$

$$b_n = E\{\ln \beta_n\} - \tfrac{1}{2}E\{b\} \quad \forall n = 1..N$$

When $\lambda_t \in (0, c)$ we must achieve the following with equality:

$$\sum_m q_t(m)(a_m + E\{\ln \mathcal{N}(\phi_m^+(X_t)|\mu_m^+)\}) + H(q_t) =$$

$$\sum_n Q_t(n)(b_n + E\{\ln \mathcal{N}(\phi_n^-(X_t)|\mu_n^-)\}) + H(Q_t) + \gamma_t \quad \forall t \in \mathcal{T}^+$$

$$\sum_n q_t(n)(b_n + E\{\ln \mathcal{N}(\phi_n^-(X_t)|\mu_n^-)\}) + H(q_t) =$$

$$\sum_m Q_t(m)(a_m + E\{\ln \mathcal{N}(\phi_m^+(X_t)|\mu_m^+)\}) + H(Q_t) + \gamma_t \quad \forall t \in \mathcal{T}^-$$

We solve for a_m for $m = 1..M$ and b_n for $n = 1..N$ in this (over-constrained) linear system, obtaining the expected bias and mixing proportions.

Tractable Prediction

$$\hat{y} = \ln \frac{\sum_m \exp\left(E\{\ln \mathcal{N}(\phi_m^+(X)|\mu_m^+)\} + a_m\right)}{\sum_n \exp\left(E\{\ln \mathcal{N}(\phi_n^-(X)|\mu_n^-)\} + b_n\right)}$$

Nonstationary Weights

Recall the nonstationary kernelized discriminant:

$$f(x) = \sum_t y_t \lambda_t \sum_m \nu_{m,t}(x) k_m(x_t, x) + b.$$

To view a MED Gaussian mixture as nonstationary kernel combination, we choose weight functions of the form:

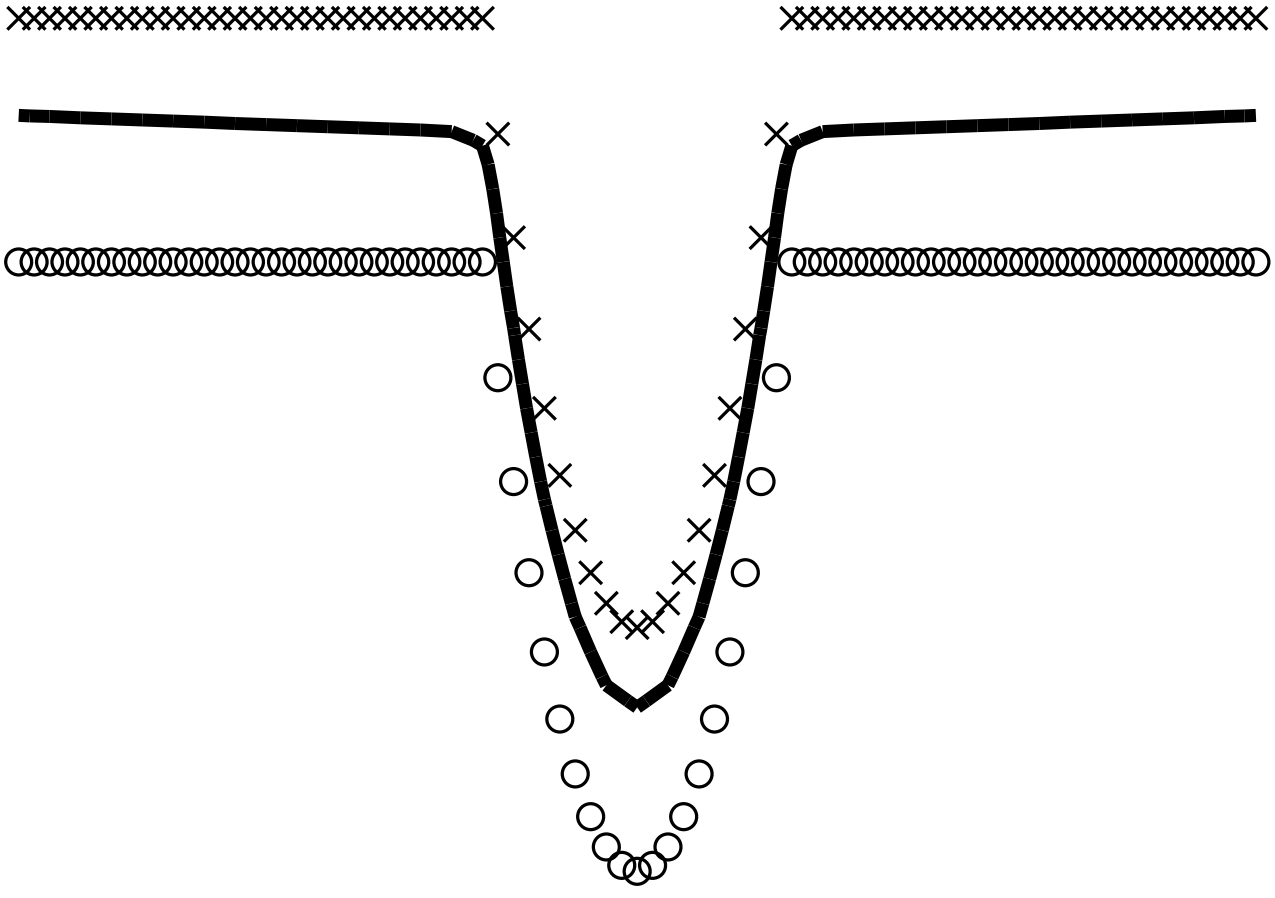
$$\nu^+_{m,t}(X) = \frac{\exp(E\{\ln \mathcal{N}(\phi^+_m(X)|\mu^+_m)\} + a_m)}{\sum_m \exp(E\{\ln \mathcal{N}(\phi^+_m(X)|\mu^+_m)\} + a_m)}.$$

Note how the kernel weight depends on the Gaussian components.

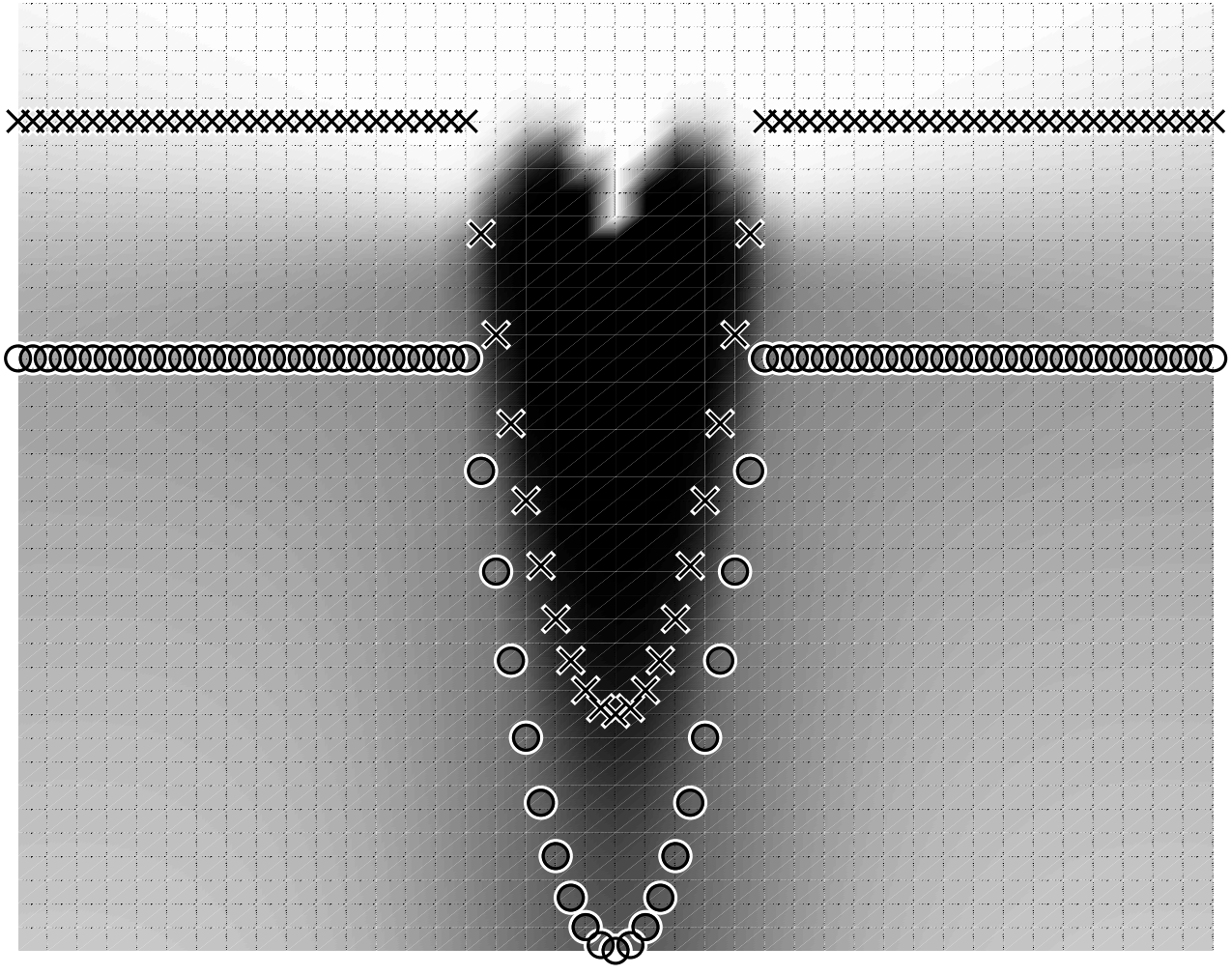
NSKC Prediction

$$\hat{y} = \sum_{\tau \in \mathcal{T}^+} \sum_m \lambda_\tau Q_\tau(m) \nu_m^+(X) k_m^+(X_\tau, X)$$
$$- \sum_{\tau \in \mathcal{T}^-} \sum_m \lambda_\tau Q_\tau(m) \nu_m^+(X) k_m^+(X_\tau, X)$$
$$- \sum_{\tau \in \mathcal{T}^-} \sum_n \lambda_\tau Q_\tau(n) \nu_n^-(X) k_n^-(X_\tau, X)$$
$$+ \sum_{\tau \in \mathcal{T}^+} \sum_n \lambda_\tau Q_\tau(n) \nu_n^-(X) k_n^-(X_\tau, X)$$
$$+ \sum_m \nu_m^+(X) k_m^+(X, X) - \sum_n \nu_n^-(X) k_n^-(X, X)$$
$$+ constant.$$

Parabola-Line NSKC



Parabola-Line NSKC Weight



Outline

- Summary of Contribution
- Stationary kernel combination
- Nonstationary kernel combination
- **Sequential minimal optimization**
- Results
- Conclusion

SMO

$\text{argmin}_\lambda J(\lambda) = c^T\lambda + \frac{1}{2}\lambda^T\mathbf{H}\lambda$ subject to:

$$\begin{bmatrix} \dots & q_{u_1} & q_{u_1} & \dots & -1 & 0 & \dots & q_{w_1} & q_{w_1} & \dots \\ \dots & q_{u_2} & q_{u_2} & \dots & 0 & -1 & \dots & q_{w_2} & q_{w_2} & \dots \\ \dots & 1 & 0 & \dots & -q_{v_1} & -q_{v_1} & \dots & 1 & 0 & \dots \\ \dots & 0 & 1 & \dots & -q_{v_2} & -q_{v_2} & \dots & 0 & 1 & \dots \end{bmatrix} \begin{bmatrix} \vdots \\ \lambda_{u_1} \\ \lambda_{u_2} \\ \vdots \\ \lambda_{v_1} \\ \lambda_{v_2} \\ \vdots \\ \lambda_{w_1} \\ \lambda_{w_2} \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Inter-class

We can maintain the constraints using the following equalities in vector form:

$$q_u\left(\hat{\lambda_u}^T 1\right) - \hat{\lambda_v} = q_u\left(\lambda_u^T 1\right) - \lambda_v$$

$$\hat{\lambda_u} - q_v\left(\hat{\lambda_v}^T 1\right) = \lambda_u - q_v\left(\lambda_v^T 1\right).$$

Then, we can write

$$\Delta\lambda_v = (\Delta\lambda_u^T 1)q_u$$

$$\Delta\lambda_u = (\Delta\lambda_v^T 1)q_v = (((\Delta\lambda_u^T 1)q_u)^T 1)q_v = (\Delta\lambda_u^T 1)q_v.$$

Analytic Update

$(\Delta\lambda_u^T 1) = (\Delta\lambda_v^T 1) = \Delta s$. We have $\Delta\lambda_v = \Delta s q_u$ and $\Delta\lambda_u = \Delta s q_v$. The change in the quadratic objective function for the axes u and v is

$$\begin{aligned}\Delta J_{uv}(\Delta\lambda) = & c_u^T \Delta\lambda_u + c_v^T \Delta\lambda_v \\ & + \tfrac{1}{2}\Delta\lambda_u^T \mathbf{H}_{uu}\Delta\lambda_u + \Delta\lambda_u^T \mathbf{H}_{uv}\Delta\lambda_v + \tfrac{1}{2}\Delta\lambda_v^T \mathbf{H}_{vv}\Delta\lambda_v \\ & + \sum\nolimits_{t\neq u,v} (\Delta\lambda_t^T \mathbf{H}_{tu}\Delta\lambda_u + \Delta\lambda_t^T \mathbf{H}_{tv}\Delta\lambda_v).\end{aligned}$$

We must express the change in the objective, $\Delta J_{uv}(\Delta\lambda)$ as a function of Δs. The resulting one-dimensional quadratic objective function, $\Delta J_{uv}(\Delta s)$, can be analytically optimized by finding the root of the derivative under the box constraint.

Other Cases

Intra-class:

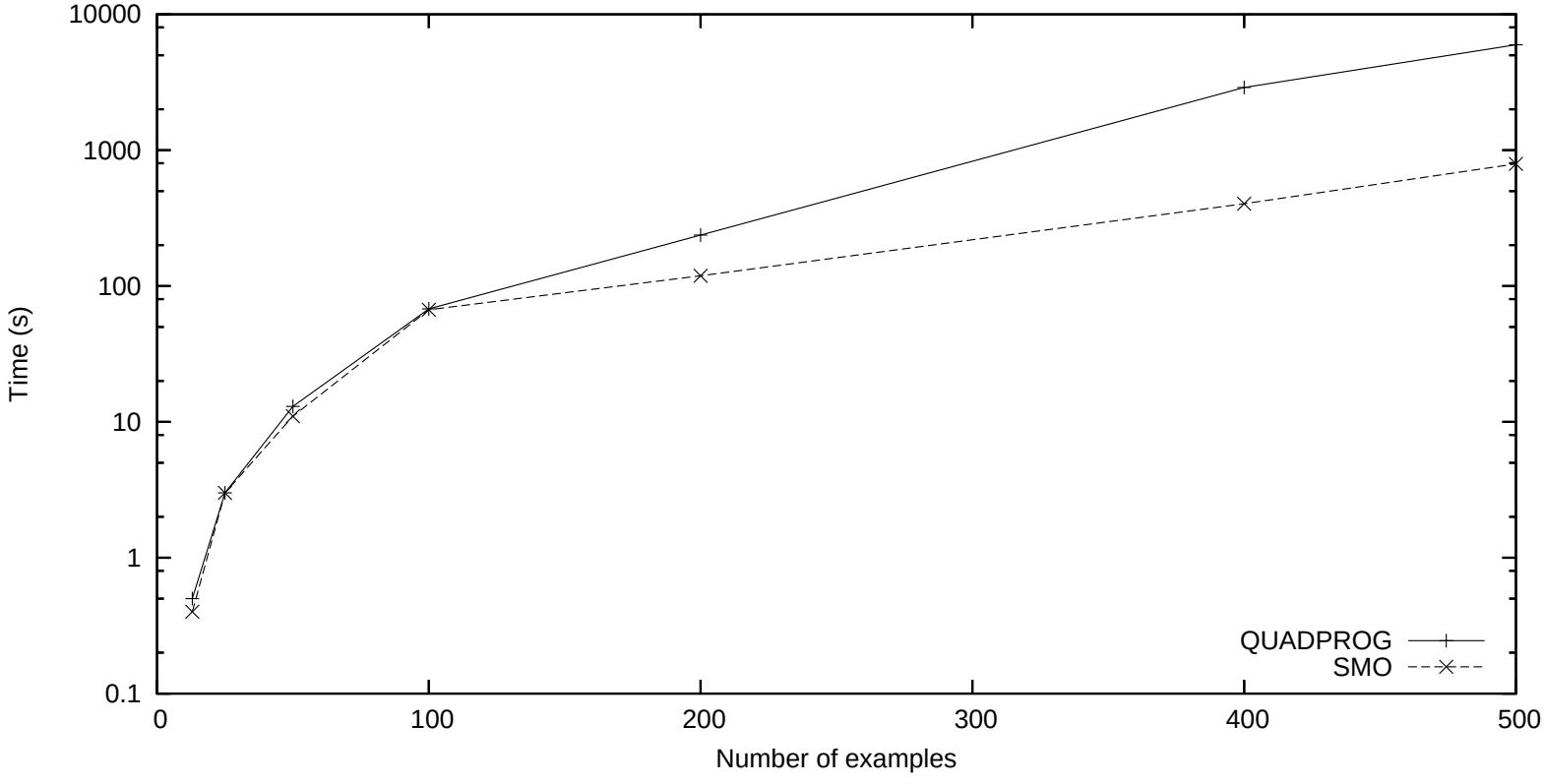
$$q_u\left(\hat{\lambda_u}^T 1\right) + q_w\left(\hat{\lambda_w}^T 1\right) = q_u\left(\lambda_u^T 1\right) + q_w\left(\lambda_w^T 1\right)$$

$$\hat{\lambda_u} + \hat{\lambda_w} = \lambda_u + \lambda_w$$

Newton Step:

- Occasionally interleave a second-order step[1] over a larger set of axes.
- We discovered that SMO can get trapped in a local plateau in the objective function.
- Though the objective and constraints are convex, choosing a minimal set of axes to update results in slow convergence.

SMO Timing


10000
1000
100
10
1
0.1
Time (s)
0
100
200
300
400
500
Number of examples
QUADPROG
SMO

Outline

- Summary of Contribution
- Stationary kernel combination
- Nonstationary kernel combination
- Sequential minimal optimization
- **Results**
- Conclusion

Benchmark data sets

We validate NSKC on UCI[11] Breast Cancer, Sonar, and Heart data sets. We use a quadratic kernel $k_1(x_1, x_2) = (1 + x_1^T x_2)^2$, an RBF kernel $k_2(x_1, x_2) = \exp(-0.5(x_1 - x_2)^T(x_1 - x_2)/\sigma)$, and a linear kernel $k_3(x_1, x_2) = x_1^T x_2$.

- All three kernels are normalized so that their features lie on the surface of a unit hypersphere.
- As in Lanckriet et al.[6], we use a hard margin ($c = 10,000$)
- RBF width parameter σ is set to 0.5 (Cancer), 0.1 (Sonar) and 0.5 (Heart).

Breast Cancer

Algorithm	Mean ROC
quadratic	0.5486 ± 0.091
RBF	0.6275 ± 0.019
linear	0.5433 ± 0.087
SDP	0.8155 ± 0.015
ML	0.5573 ± 0.03
NSKC	$\mathbf{0.8313 \pm 0.014}$

Sonar

Algorithm	Mean ROC
quadratic	0.8145 ± 0.01
RBF	0.8595 ± 0.009
linear	0.7297 ± 0.01
SDP	0.8595 ± 0.009
ML	0.6817 ± 0.022
NSKC	$\mathbf{0.8634 \pm 0.008}$

Heart

Algorithm	Mean ROC
quadratic	**0.6141** $\pm$ **0.032**
RBF	0.5556 $\pm$ 0.01
linear	0.5237 $\pm$ 0.02
SDP	0.5556 $\pm$ 0.01
ML	0.5361 $\pm$ 0.024
NSKC	0.6052 $\pm$ 0.016

Yeast Experiment

We compare NSKC against three single-kernel SVMs and against an SDP combination of the three kernels. This is the data set used for the original SDP experiments[7;5].

- Gene expression kernel
- Protein domain kernel
- Sequence kernel
- MIPS MYGD labels
- 500 randomly sampled genes in a 5x3cv experiment

Protein Function Annotation

Class	Exp	Dom	Seq	SDP	NSKC
1	0.630	0.717	**0.750**	0.745	0.747
2	0.657	0.664	0.718	0.751	**0.755**
3	0.668	0.706	0.729	0.768	**0.774**
4	0.596	0.756	0.752	0.766	**0.778**
5	0.810	0.773	0.789	0.834	**0.836**
6	0.617	0.690	0.668	0.698	**0.717**
7	0.554	0.715	**0.740**	0.720	0.738
8	0.594	0.636	0.680	0.697	**0.699**
9	0.535	0.564	**0.603**	0.582	0.576
10	0.554	0.616	**0.706**	0.697	0.687
11	0.506	0.470	0.480	0.524	**0.526**
12	0.682	0.896	0.883	0.916	**0.918**

Sequence/Structure Revisited

GO term	Average	SDP	NSKC
GO:0008168	0.937 ± 0.016	0.938 ± 0.015	0.944 ± 0.014
GO:0005506	0.927 ± 0.012	0.927 ± 0.012	0.926 ± 0.013
GO:0006260	0.878 ± 0.016	0.870 ± 0.015	0.880 ± 0.015
GO:0048037	0.911 ± 0.016	0.909 ± 0.016	0.918 ± 0.015
GO:0046483	0.937 ± 0.008	0.940 ± 0.008	0.941 ± 0.008
GO:0044255	0.874 ± 0.015	0.864 ± 0.013	0.874 ± 0.012
GO:0016853	0.837 ± 0.017	0.810 ± 0.019	0.823 ± 0.018
GO:0044262	0.908 ± 0.006	0.897 ± 0.006	0.906 ± 0.007
GO:0009117	0.890 ± 0.012	0.880 ± 0.012	0.887 ± 0.012
GO:0016829	0.931 ± 0.008	0.926 ± 0.007	0.928 ± 0.008

- NSKC and averaging are in a statistical tie
- NSKC is significantly better than SDP

Outline

- Summary of Contribution
- Stationary kernel combination
- Nonstationary kernel combination
- Sequential minimal optimization
- Results
- **Conclusion**

Conclusion

- Prior work
- Contributions
- Future directions

Prior work

To complete this research we built upon an impressive foundation of prior work in:

- Kernel methods[16]
- Support vector machines[18]
- Multi-kernel learning[14;6;12;17]
- Maximum entropy discrimination[3]
- Protein function annotation from heterogeneous data sets[7;17]
- Optimization[15;1]

In particular, this thesis extends the work of Jebara[4]. William Noble and Tony Jebara are my advisors and co-authors and greatly influenced the work.

Contributions

- Empirical study of averaging versus SDP
- Nonstationary kernel combination
- Double Jensen bound for latent MED
- Efficient optimization
- Implementation

Averaging vs. SDP

- We present a comparison of SDP and averaging for combining protein sequence and structure kernels for the prediction of function.
- We analyze the outcomes and suggest when each approach is appropriate.
- We conclude that in all practical cases, averaging is worthwhile.
- This result is significant to practitioners because it indicates that a simple, fast, free technique is also very effective.

Nonstationary kernel combination

- We propose a novel way to combine kernels that generalizes upon the state-of-the-art.
- NSKC allows kernel combination weight to depend on the input space.
- We demonstrate our technique with a synthetic problem that existing techniques cannot solve.
- We validate NSKC with several common benchmark data sets and two real-world problems.
- NSKC usually outperforms existing techniques.

Double Jensen, SMO, Implementation

- The new double Jensen variational bound is tight and assures that latent MED optimization will converge to a local optimum.
- Sequential minimal optimization for MED Gaussian mixtures improves optimization speed and helps to make the technique practical.
- SMO is faster than the `quadprog` standard QP solver and matches the speed of the highly optimized commercial Mosek optimization software.
- Our C++ SMO implementation and our Matlab classes for kernels, learning algorithms, and cross validation experiments will be freely available for academic use.

Future directions

- Saddle-point optimization of indefinite objective
- Entropy terms for Q
- Transduction
- Other latent variable models

References

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Prentice-Hall, 2003. To appear. Available at http://www.stanford.edu/~boyd/cvxbook.html.

[2] Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nat Genet*, 25(1):25–9, 2000.

[3] T. Jaakkola, M. Meila, and T. Jebara. Maximum entropy discrimination. In *Advances in Neural Information Processing Systems*, volume 12, December 1999.

[4] T. Jebara. *Machine Learning: Discriminative and Generative*. Kluwer Academic, Boston, MA, 2004.

[5] G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.

[6] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semi-definite programming. In C. Sammut and A. Hoffman, editors, *Proceedings of the 19th International Conference on Machine Learning*, Sydney, Australia, 2002. Morgan Kauffman.

[7] G. R. G. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble. Kernel-based data fusion and its application to protein function prediction in yeast. In R. B. Altman, A. K. Dunker, L. Hunter, T. A. Jung, and T. E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing*, pages 300–311. World Scientific, 2004.

[8] C. Leslie, E. Eskin, J. Weston, and W. S. Noble. Mismatch string kernels for SVM protein classification. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems*, pages 1441–1448, Cambridge, MA, 2003. MIT Press.

[9] D. Lewis, T. Jebara, and W. S. Noble. Nonstationary kernel combination. In *23rd International Conference on Machine Learning (ICML)*, 2006.

[10] D. Lewis, T. Jebara, and W. S. Noble. Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure. Submitted, April 2006.

[11] P. M. Murphy and D. W. Aha. UCI repository of machine learning databases. Dept. of Information and Computer Science, UC Irvine, 1995.

[12] C. S. Ong, A. J. Smola, and R. C. Williamson. Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, 6:1043–1071, 2005.

[13] A. R. Ortiz, C. E. M. Strauss, and O. Olmea. MAMMOTH (Matching molecular models obtained from theory): An automated method for model comparison. *Protein Science*, 11:2606–2621, 2002.

[14] P. Pavlidis, J. Weston, J. Cai, and W. S. Noble. Learning gene functional classifications from multiple data types. *Journal of Computational Biology*, 9(2):401–411, 2002.

[15] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods*. MIT Press, 1999.

[16] B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA, 1999.

[17] K. Tsuda, H.J. Shin, and B. Schölkopf. Fast protein classification with multiple networks. In *ECCB*, 2005.

[18] V. N. Vapnik. *Statistical Learning Theory*. Adaptive and learning systems for signal processing, communications, and control. Wiley, New York, 1998.