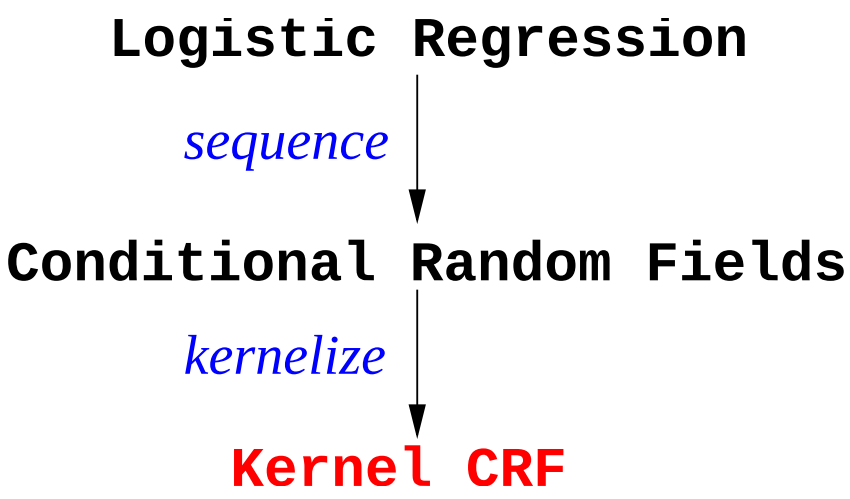
Kernel Conditional Random Fields

Jerry Xiaojin Zhu

Joint work with:
John Lafferty
Yan Liu

February 20, 2004


Logistic Regression
sequence
Conditional Random Fields
kernelize
Kernel CRF

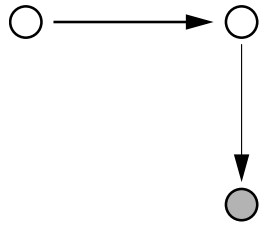
Non-independent Data

- Independent data:
  - logistic regression, SVM, etc.
- Non-independent data: sequences, trees, graphs
  - e.g. text, speech, proteins
  - HMMs, CRFs, etc.

From HMMs to CRFs: special case of sequence

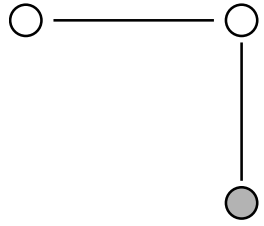
HMM:

$$p(\mathbf{y} \mid \mathbf{x}) \quad \propto \quad \prod_{t=1}^{T} p(y_t \mid y_{t-1}) p(x_t \mid y_t)$$



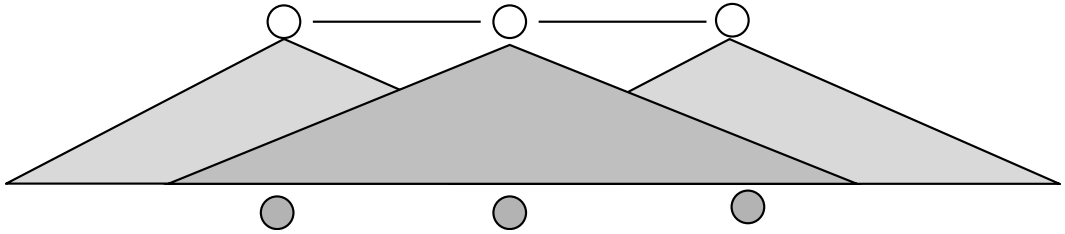
CRF:

$$p(\mathbf{y} \mid \mathbf{x}) \quad = \quad \frac{1}{Z_{\mathbf{x}}} \prod_{t=1}^{T} \exp \left[\begin{array}{l} \sum_j \lambda_j \phi_j(y_t, y_{t-1}) \\ + \sum_k \lambda_k \phi_k(y_t, \mathbf{x}) \end{array} \right]$$



(Lafferty, McCallum and Pereira, 2001), also (Johnson et al., 2001)

Conditional Random Fields



$$p(\mathbf{y}\,|\,\mathbf{x}) \;=\; \frac{1}{Z(\mathbf{x})}\exp\Big(\textstyle\sum_{c\in\text{cliques}}\sum_{i\in\text{features}}\lambda_{ci}\Phi_{ci}(\mathbf{x},\mathbf{y}_c)\Big)$$

- Global normalization. Undirected graphical models.
- Model $p(\text{label sequence } \mathbf{y}\,|\,\text{observation sequence } \mathbf{x})$ rather than joint probability $p(\mathbf{y},\mathbf{x})$
- Allow arbitrary (e.g. long range) dependencies on the observation sequence

Conditional Random Fields

- Still efficient (Viterbi, forward-backward) if dependencies within the state sequence $\mathbf{y}$ are local (sequences, trees).

- Promising results in
  - tagging, parsing, information extraction (Collins, 2001), (Sha and Pereira, 2003), (Pinto et al., 2003)
  - image processing (Kumar and Hebert, 2003)

Explicit Features vs. Kernels

CRFs were based on explicit feature representations.

$$p(\mathbf{y}\,|\,\mathbf{x}) \;=\; \frac{1}{Z(\mathbf{x})}\exp\left(\sum_{c\in\text{cliques}}\sum_{i\in\text{features}} \lambda_{ci}\Phi_{ci}(\mathbf{x},\mathbf{y}_c)\right)$$

λ_{ci} : weight parameters

$\Phi_{ci}(\mathbf{x},\mathbf{y}_c)$: features

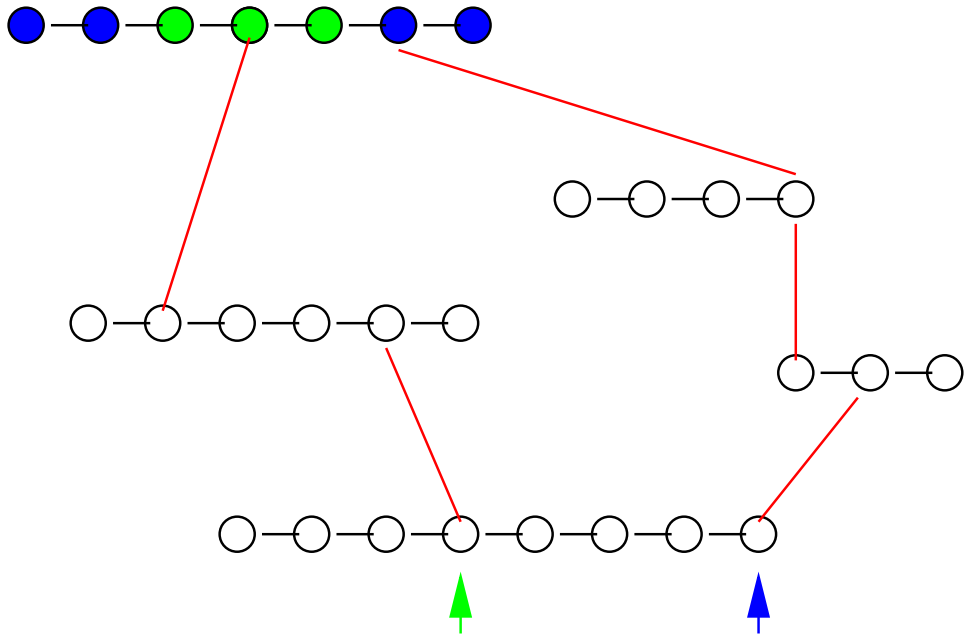
Kernels: implicit feature representations.

$$K_c(\mathbf{x},\mathbf{y}_c;\,\mathbf{x}',\mathbf{y}'_c) = \sum_i \Phi_{ci}(\mathbf{x},\mathbf{y}_c)\Phi_{ci}(\mathbf{x}',\mathbf{y}'_c)$$

Why Kernels

- Kernel machines are very successful;

- Semi-supervised learning on sequences with graph kernels.



The Negative Log Loss

$$p(\mathbf{y}\,|\,\mathbf{x}) \;=\; \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{c \in \text{cliques}} \sum_{i \in \text{features}} \lambda_{ci} \Phi_{ci}(\mathbf{x}, \mathbf{y}_c) \right)$$

Introduce functions f_c

$$p(\mathbf{y}\,|\,\mathbf{x}) \;=\; \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{c \in \text{cliques}} f_c(\mathbf{x}, \mathbf{y}_c) \right)$$

The negative log loss (logistic loss)

$$\begin{aligned} \psi(\mathbf{x}, \mathbf{y}, f) \;&=\; -\log p(\mathbf{y}\,|\,\mathbf{x}) \\ &=\; -\sum_{c \in \text{cliques}} f_c(\mathbf{x}, \mathbf{y}_c) + \log \sum_{\mathbf{y}'} \exp\left(\sum_{c \in \text{cliques}} f_c(\mathbf{x}, \mathbf{y}'_c) \right) \end{aligned}$$

Regularized Risk

$$R_\psi f = \sum_{i=1}^{n} \psi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, f) + \Omega(\|f\|)$$

Regularizer Ω is a monotonically increasing function.

The risk minimizer f^* is the MAP estimate of the CRF.

A different ψ (hinge loss?) seems to correspond to Max-Margin Markov (M^3) Networks (Taskar et al., 2003).

Representer Theorem for CRFs

The minimizer f^* of

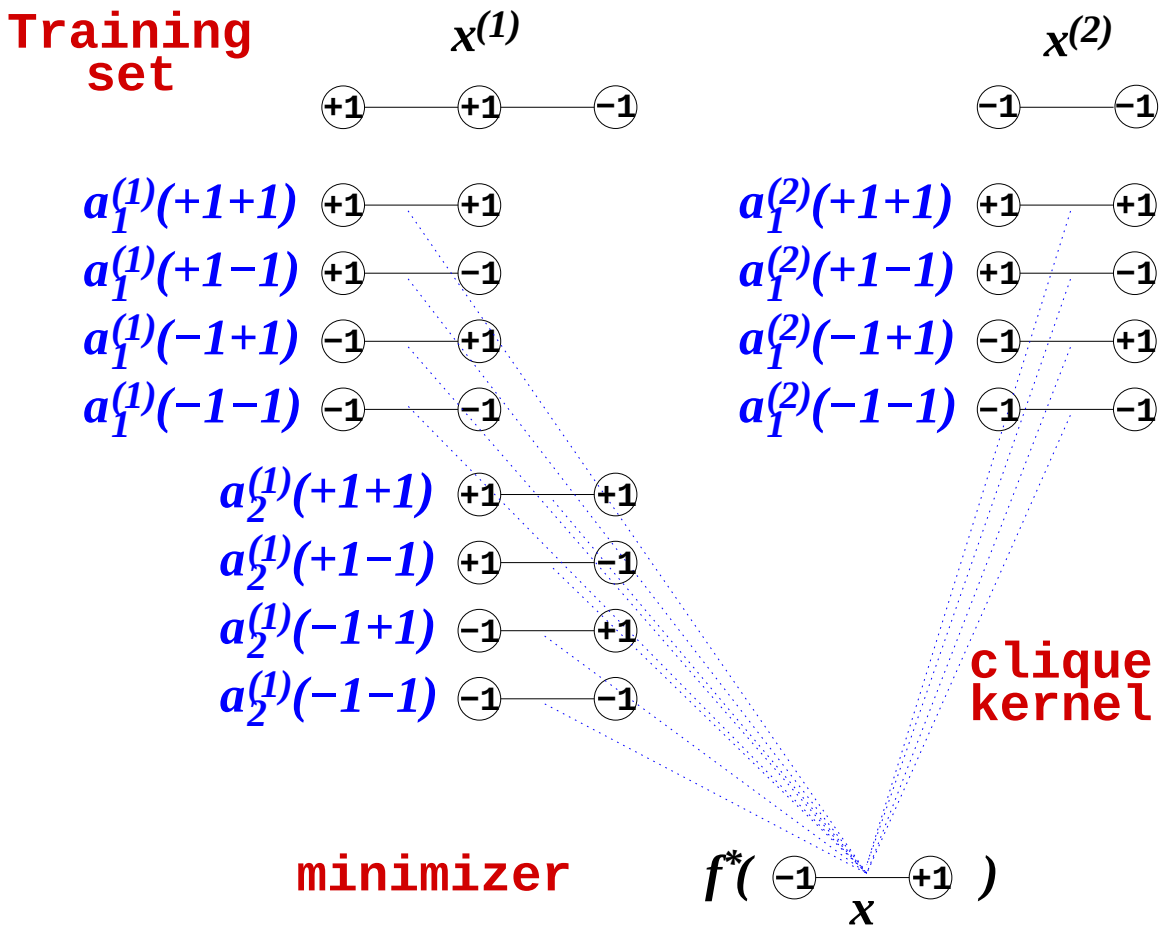
$$R_\psi f = \sum_{i=1}^{n} \psi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, f) + \Omega(\|f\|)$$

has the form

$$f_c^*(\cdot) = \sum_{i=1}^{n} \sum_{c \in \text{cliques}} \sum_{\text{all } \mathbf{y}_c} \alpha_c^{(i)}(\mathbf{y}_c) K_c(\mathbf{x}^{(i)}, \mathbf{y}_c; \cdot)$$

Dual parameters $\alpha_c^{(i)}(\mathbf{y}_c)$: all labeling, not only those in training.

Example 1: Edge cliques



Example 2: Kernel Logistic Regression

With only vertex cliques, and

$$K(\mathbf{x}, y_v; \mathbf{x}', y_{v'}) = K(x_v, x_{v'})\delta(y_v, y_{v'})$$

The minimizer has the form

$$f^*(x_0, y_0) = \sum_{v \in \text{vertices}} \alpha_v(y_0) K(x_v, x_0)$$

This is simply kernel logistic regression.

The KCRF training problem

Given training set $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}$, clique kernels K_c, find the dual parameters

$$\alpha_c^{(i)}(\mathbf{y}_c)$$

for

$$f_c(\cdot) = \sum_{i=1}^{n} \sum_{c \in \text{cliques}} \sum_{\text{all } \mathbf{y}_c} \alpha_c^{(i)}(\mathbf{y}_c) K_c(\mathbf{x}^{(i)}, \mathbf{y}_c; \cdot)$$

that minimizes the regularized risk

$$R_\psi f = \sum_{i=1}^{n} \psi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, f) + \frac{1}{2}\|f\|^2$$

unconstrained convex optimization problem, global solution.

The Derivatives

Each parameter $\alpha_c^{(i)}(\mathbf{y}_c)$ is associated with a basis function

$$h(\cdot) = K_c(\mathbf{x}^{(i)}, \mathbf{y}_c; \cdot)$$

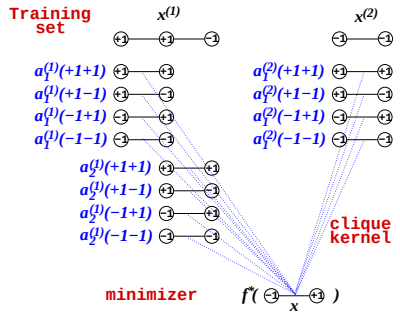
$$\frac{\partial R_\psi f}{\partial \alpha_c^{(i)}(\mathbf{y}_c)} = E_f[h] - \tilde{E}[h] + < f, h >_K$$

$E_f[h]$: expectation of h under the current KCRF parameters (need clique marginals, forward-backward)

$\tilde{E}[h]$: training set empirical expectation of h

Newton's algorithm, BFGS.

Too Many Parameters



Every labeling of every training clique has a parameter.

$$(\text{num of training cliques}) * (\text{clique size})^{\text{num of labels}}$$

Negative log loss (KCRF) $\rightarrow$ everything is a 'support clique'

Sparse training: select a good subset.

Sparse Training: Greedy Clique Selection

Initially Active Set $\mathcal{A}$ empty, all parameters zero. Repeat:

1. candidates $\leftarrow$ training cliques not in $\mathcal{A}$

2. select the candidate clique $K_c(\mathbf{x}^{(i)}, \mathbf{y}_c; \cdot)$ with the highest gain. $\mathcal{A} \leftarrow \mathcal{A} \cup \{K_c(\mathbf{x}^{(i)}, \mathbf{y}_c; \cdot)\}$

3. train KCRF with the active cliques in $\mathcal{A}$ by minimizing $R_\psi f$.

Gain

Gain: $R_\psi(f) - R_\psi(f, K_c(\mathbf{x}^{(i)}, \mathbf{y}_c; \cdot))$; Problem: has to estimate $\alpha_c^{(i)}(\mathbf{y}_c)$.

Linear approximation (functional derivative):

$$R_\psi(f, \epsilon K_c(\mathbf{x}^{(i)}, \mathbf{y}_c; \cdot)) \approx R_\psi(f) + \epsilon \frac{\partial R_\psi f}{\partial \alpha_c^{(i)}(\mathbf{y}_c)}$$

Select the candidate with the largest gradient magnitude:

$$\left| \frac{\partial R_\psi f}{\partial \alpha_c^{(i)}(\mathbf{y}_c)} \right| = \left| E_f[h] - \tilde{E}[h] + < f, h >_K \right|$$

The clique whose model and empirical expectations mismatch the most.

Detour: Semi-Supervised Learning

Classification needs labeled training data.

Often, labeled data scarce, unlabeled data abundant.

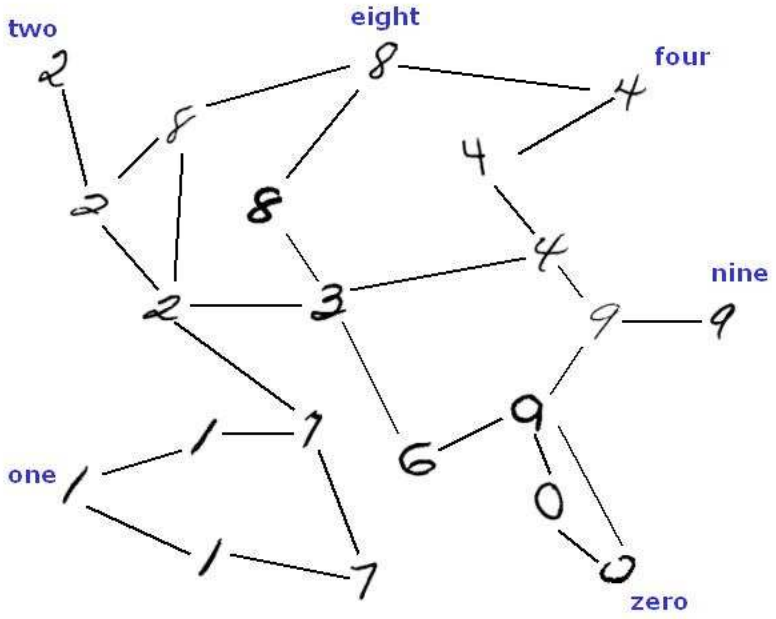
Semi-supervised learning: use unlabeled data to help classification.

Graph kernels: emerging theme in semi-supervised learning.

Semi-Supervised Learning: the Graph

(Note: not the KCRF graph)

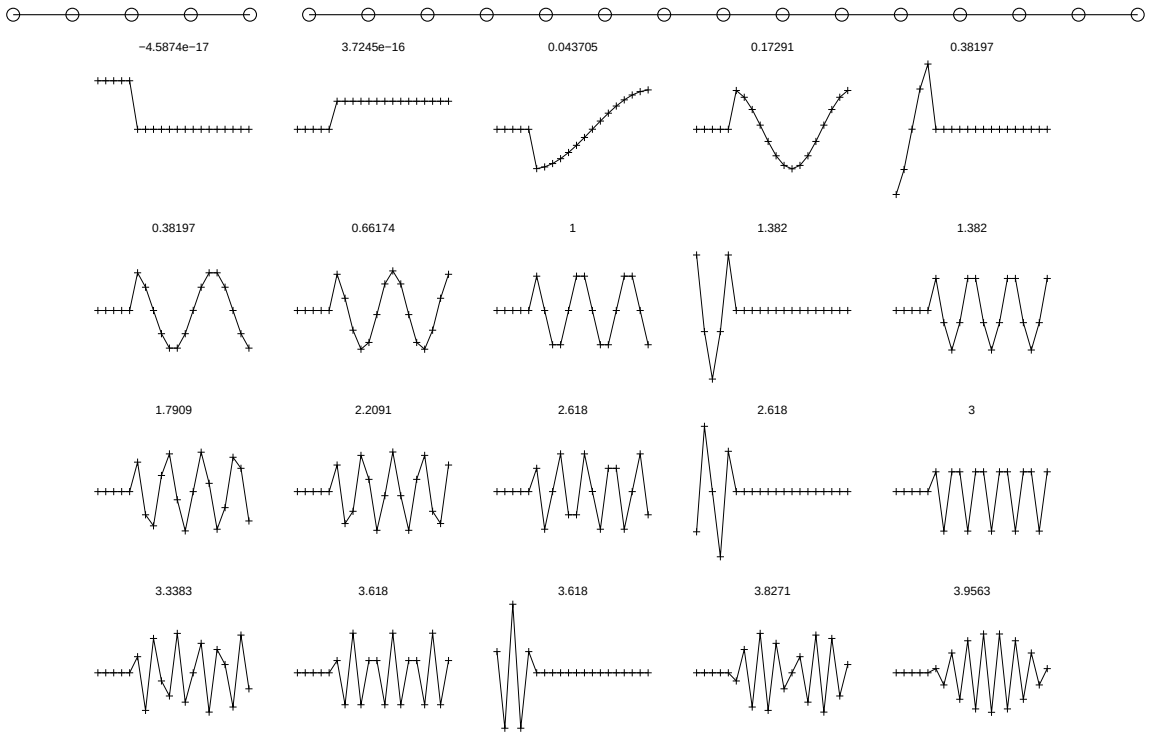
nodes: labeled and unlabeled data; edges: local similarity



Labels propagate, smooth on graph

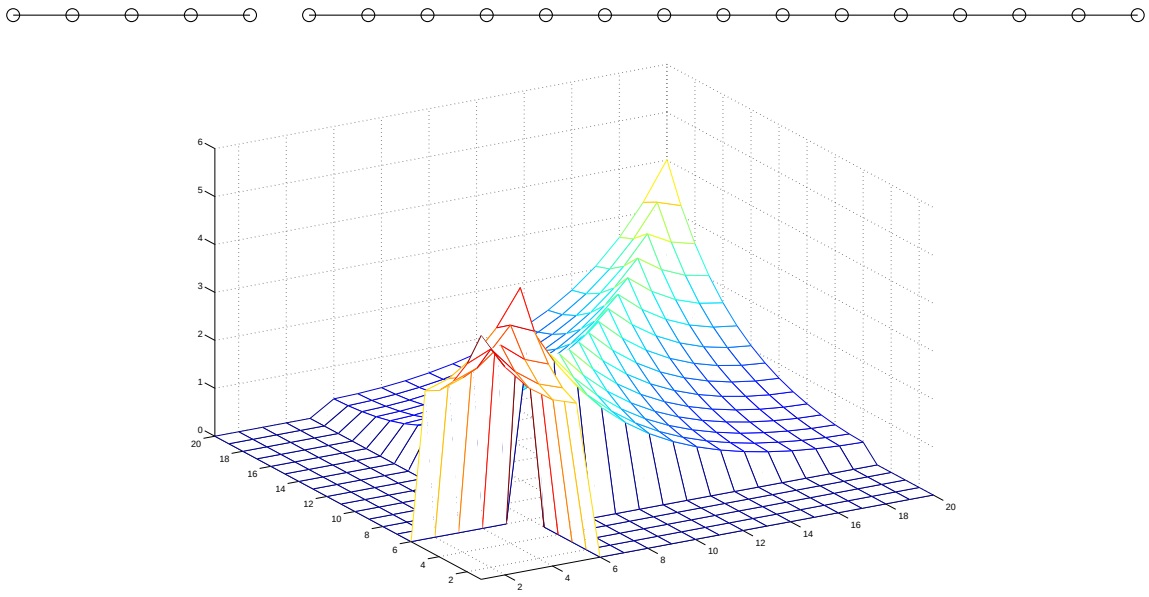
Semi-Supervised Learning: the Laplacian

weights W; degrees (diagonal) D; Laplacian $D - W$
Spectrum $D - W = \sum_i \lambda_i \phi_i \phi_i^\top$.
λ_i: frequencies; ϕ_i: vibration modes. Low frequencies smoother.



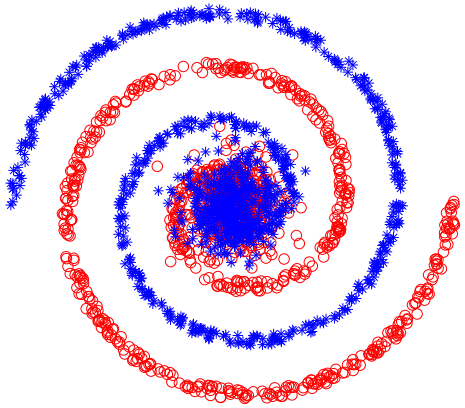
Semi-Supervised Kernels

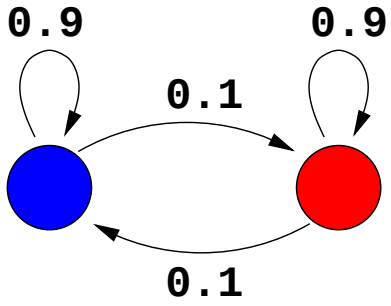
Emphasize low frequency components, e.g. $K = \sum_i \frac{1}{\lambda_i + 0.05} \phi_i \phi_i^\top$



(Smola and Kondor, 2003), (Zhu, Gharahmani and Lafferty, 2003), etc.

Back to KCRF: a Synthetic Example





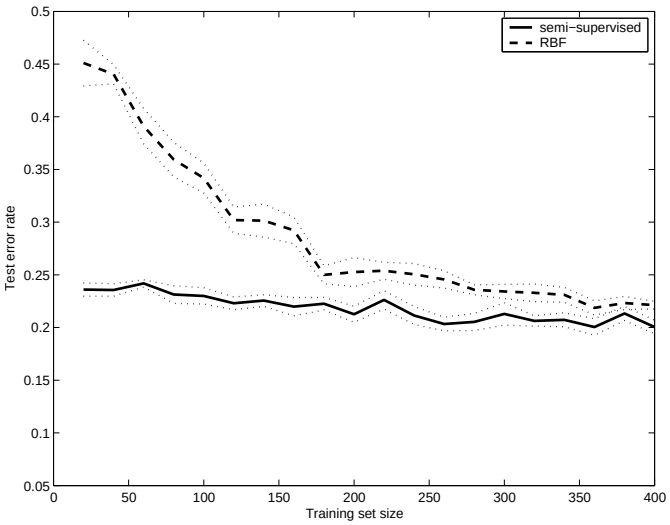
sequences: generated from an HMM

Semi-supervised kernel: 10 nearest neighbor unweighted graph

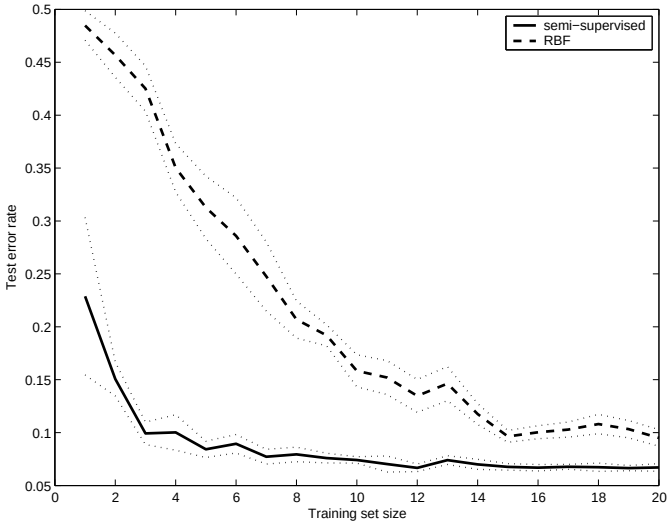
RBF kernel

Synthetic Example

test error vs. training set size



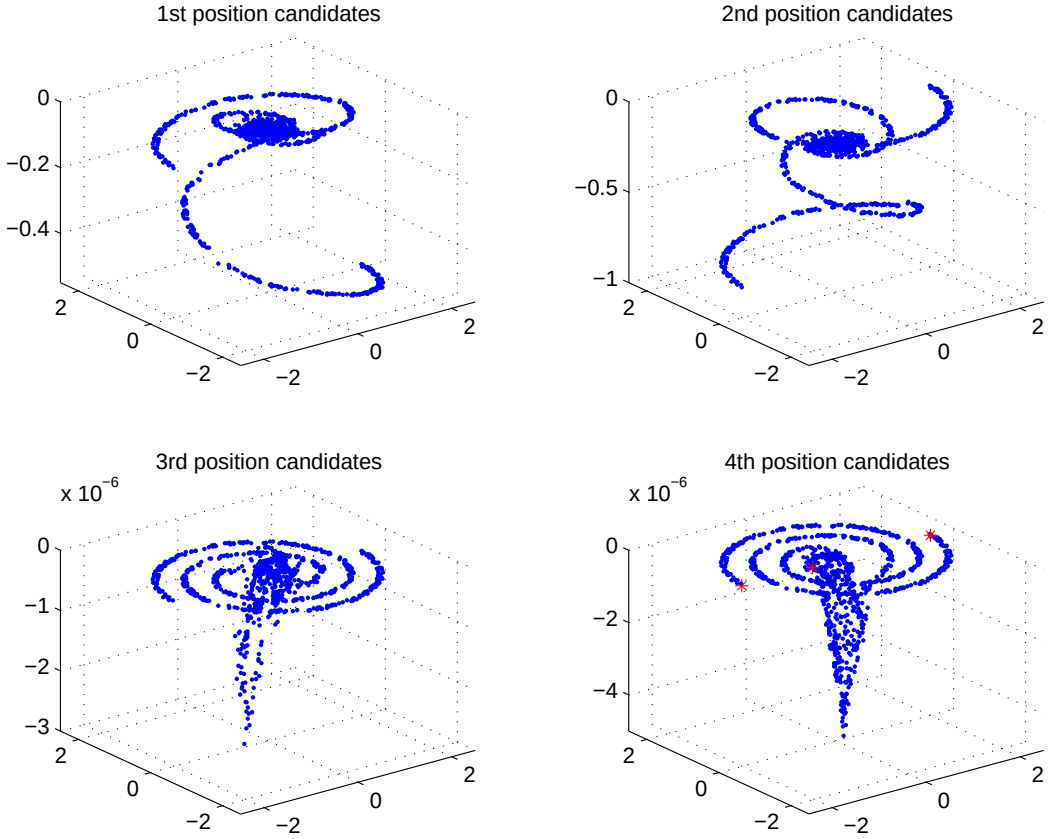
kernel logistic regression



KCRF

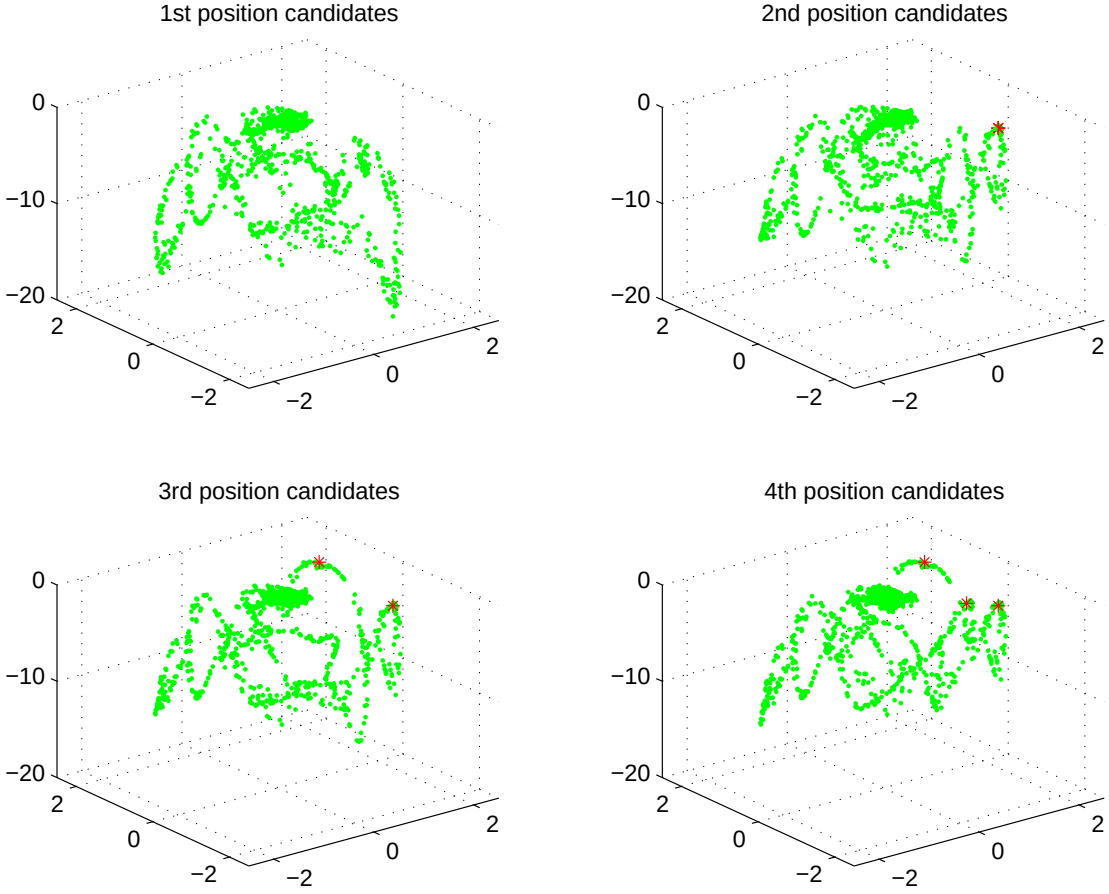
Synthetic Example

KCRF candidate gain, graph kernel



Synthetic Example

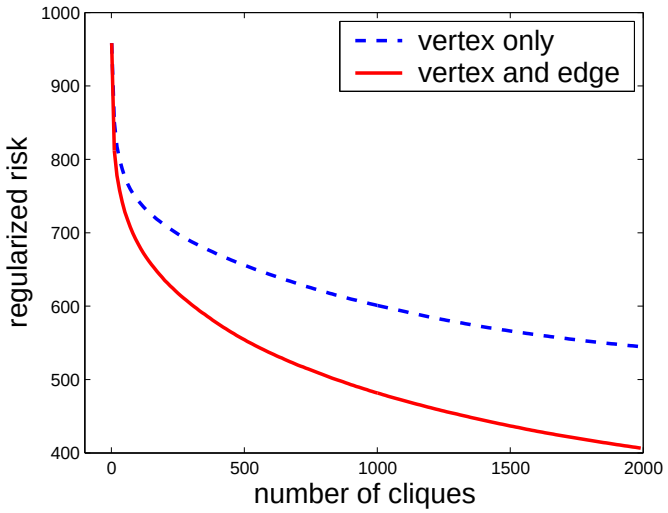
KCRF candidate gain, RBF kernel



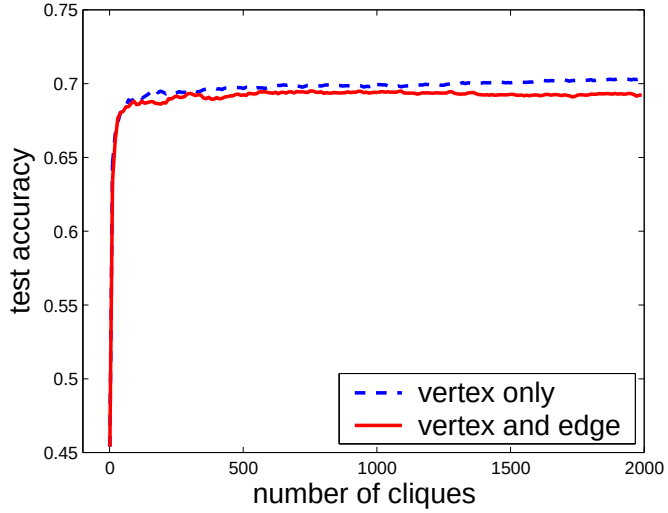
Protein Secondary Structure Prediction

RS126 dataset, 117 protein sequences; PSI-BLAST feature; Three classes: coil, sheet, helix; RBF kernel

KCRF clique selection



risk



test accuracy

Protein Secondary Structure Prediction

Test Accuracy

	5 protein set		10 protein set	
Method	Accuracy	std	Accuracy	std
KCRF (v)	0.6625	0.0224	0.6933	0.0276
KCRF (v+e)	0.6562	0.0202	0.6933	0.0272
SVM	0.6509	0.0307	0.6875	0.0235

* KCRFs select 300 cliques.

Protein Secondary Structure Prediction

Transition Accuracy

transition: a pair of adjacent positions with different true labels.
A transition is classified correctly only if both labels are correct.

	5 protein set		10 protein set	
Method	Accuracy	std	Accuracy	std
KCRF (v)	0.1097	0.0271	0.1462	0.0235
KCRF (v+e)	0.1114	0.0250	0.1522	0.0214
SVM	0.0667	0.0313	0.1066	0.0311

Conclusion

- KCRF: framework for graph-structured classification.
- Representer theorem
- Clique selection
- Needs kernels that capture the structure of the data.