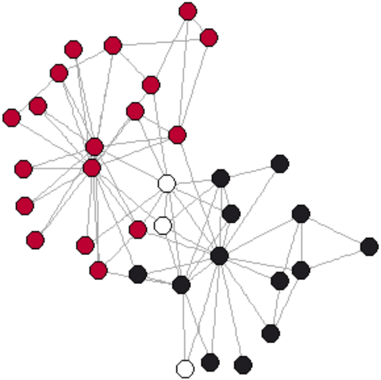
Kernel-based regression

Statistical Analysis of Network Data by Eric D. Kolaczyk

Presentation by Jarno Hartog

May 8, 2015

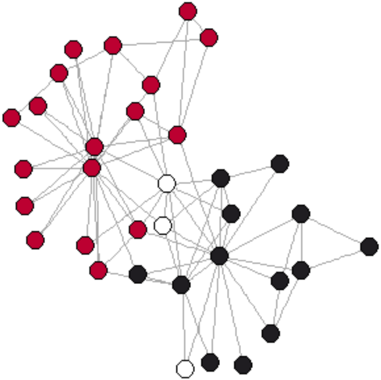
Goal: predict unobserved vertex attributes



Simple solution: nearest neighbor

- 4 black, 5 red, 1 unknown
- 3 black, 1 red, 1 unknown
- 2 black

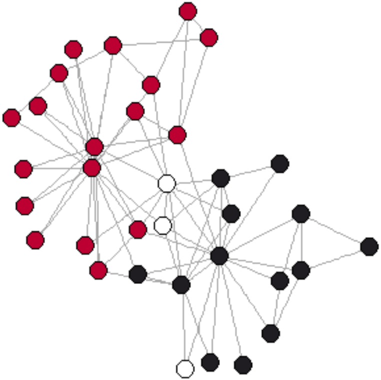
Goal: predict unobserved vertex attributes



Another solution: regression

1. Generalized notion of predictor variables
2. Regression of response to these predictors

Notation



- Graph $G = (V, E)$
- Vertex attributes $X = (X_1, \dots, X_{N_v})$
- Observed labels $V^{obs} \subset V$, $|V^{obs}| = n$
- Goal: learn $\hat{h} : V \to \mathbb{R}$

Which class to choose estimated function from?

Definition (Kernel)

Function $K : V \times V \to \mathbb{R}$ is a kernel if for all $m = 1, \ldots, N_v$, subsets $\{i_1, \ldots, i_m\} \subset V$, the $m \times m$ matrix $K^{(m)} = (K(i_j, i_{j'}))$ is symmetric positive semi-definite

Which class to choose estimated function from?

Estimate function $\hat{h}$ using kernel $K = \Phi\Delta\Phi^T$

Definition (Reproducing kernel Hilbert space)

$$\mathscr{H}_K = \{h \in \mathbb{R}^{N_v} : h = \Phi\beta, \beta^T\Delta^{-1}\beta < \infty\}$$

Representer theorem

Choose $\hat{h} = \Phi\hat{\beta}$

$$\min_{\beta} \left[\sum_{i \in V^{\text{obs}}} C\left(x_i; (\Phi\beta)_i\right) + \lambda \beta^T \Delta^{-1} \beta \right]$$

Theorem (Representer theorem, Kimeldorf and Whaba, 1971)

Solution $\hat{h}$ *will be of the form* $h = K^{(N_v,n)}\alpha$

$$\min_{\alpha} \left[\sum_{i \in V^{\text{obs}}} C\left(x_i; \left(K^{(n)}\alpha\right)_i\right) + \lambda \alpha^T K^{(n)} \alpha \right]$$

Examples

Kernel ridge regression

- $C(x;a) = (x-a)^2$
- $\hat{\alpha} = \Phi\Delta^{-1/2}(\Delta + \lambda I)^{-1}\Delta^{1/2}\Phi^T x$
- $\hat{h} = K^{(N_v,n)}\hat{\alpha}$

Kernel logistic regression

- $C(x;a) = \log(1 + e^{-xa})$
- No closed-form expression for solution
- $\hat{h} = K^{(N_v,n)}\hat{\alpha}$
- $\hat{\mathbb{P}}(X_i = 1 | X^{\text{obs}} = x^{\text{obs}}) = \frac{e^{\hat{h}_i}}{1+e^{\hat{h}_i}}$

Another example

Support Vector Machines (SVM)

- Machine Learning
- $C(x; a) = \max(0, 1 - xa)$
- Prediction of the form $\mathrm{sign}(\hat{h}_i)$

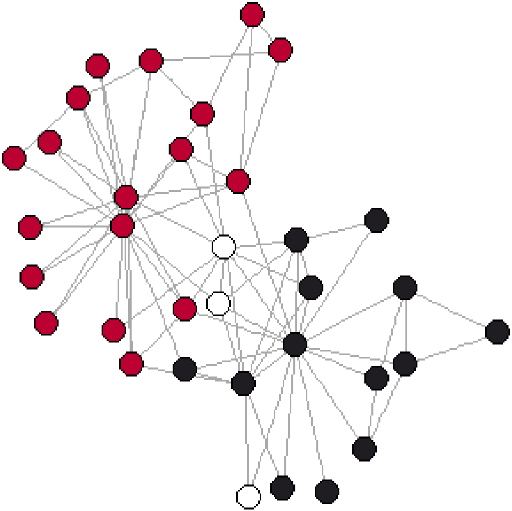
How to choose tuning parameter?

$$\min_{\alpha}\left[\sum_{i\in V^{\text{obs}}} C\left(x_i;\left(K^{(n)}\alpha\right)_i\right)+\lambda\alpha^T K^{(n)}\alpha\right]$$

Loss versus complexity penalty

- Cross-validation
- Expectation propagation (empirical Bayes)
- Learn from data (full Bayes)

How to choose kernel?



Laplacian kernel

$L = D - A$

- $K = L^{-}$
- Proximity is encoded in adjacency matrix A
- Discrete analog of Laplacian operator ∇^2
- ∇^2 is the unique self-adjoint second order differential operator invariant under transformations of the coordinate system under action of SO_m (rotations)
- Similar result for L under S_n (permutations) (Smola and Kondor, 2003)
- Penalty term $\beta^T \Delta^{-1} \beta = h^T L h = \sum_{(i,j)\in E} (h_i - h_j)^2$

Related kernels

- L incorporates knowledge of 1-step neighbors
- L^k incorporates knowledge of k-step neighbors
- $L = \Phi\Gamma\Phi^T \Rightarrow L^k = \Phi\Gamma^k\Phi^T$
- Diffusion kernel $K = e^{-\zeta L}$ is solution to $\frac{d}{d\zeta}K = -LK$
- General class of kernels $r(L) = \Phi r(\Gamma)\Phi^T$

Multiple kernels

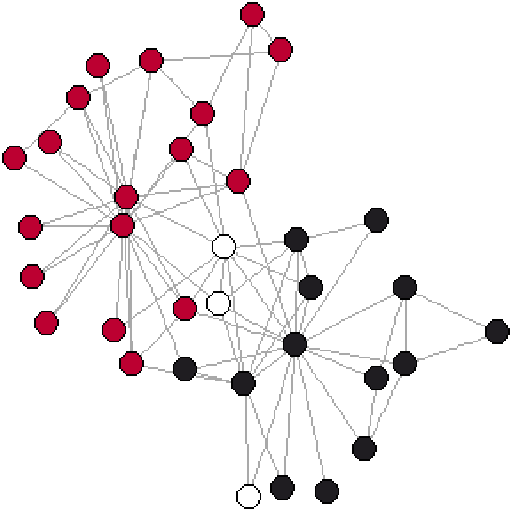
$K_1, \ldots, K_p$ potential kernels

Definition (Kernel alignment)

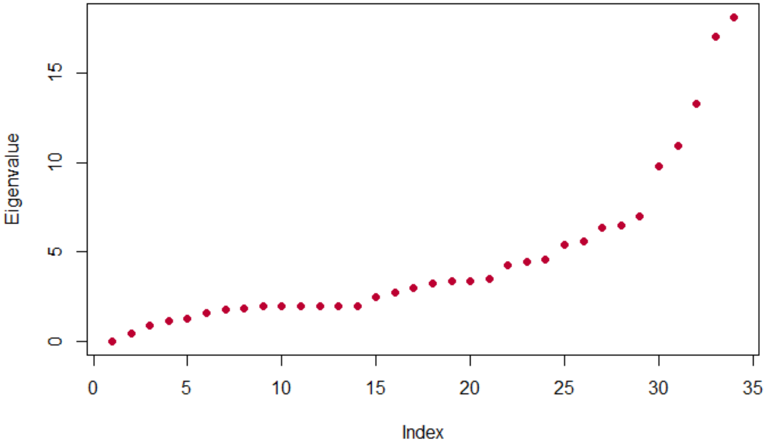
$$a(K_1, K_2) = \frac{\langle K_1, K_2 \rangle}{\sqrt{\langle K_1, K_1 \rangle \langle K_2, K_2 \rangle}}$$

- High target alignment $a(K, xx^T)$ suggests a good kernel (Cristianini et al., 2006)
- $K = \sum_{i=1}^{p} \omega_i K_i$

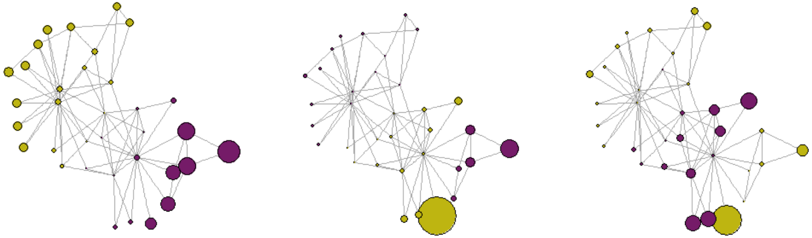
Karate club



Eigenvalues



Eigenvectors



Estimate