GP regression on random graphs: Covariance functions and Bayes errors

P Sollich[1] and Camille Coti[1,2]

[1]King's College London
[2]Laboratoire de Recherche en Informatique,
Université Paris-Sud

Outline

Motivation

- GP regression over continuous spaces relatively well understood [e.g. Opper & Malzahn]
- Discrete spaces occur in many applications: sequences, strings etc
- What can we say about GP learning on these?
- Focus on random graphs with finite connectivity as a paradigmatic case

Outline

Graph Laplacian

- Easiest to define from graph Laplacian [Smola & Kondor 2003]
- Adjacency matrix $A_{ij} = 0$ or 1 depending on whether nodes i and j are connected
- For a graph with V nodes, $\boldsymbol{A}$ is a $V \times V$ matrix
- Consider undirected links ($A_{ij} = A_{ji}$), and no self-loops ($A_{ii} = 0$)
- Degree of node i: $d_i = \sum_{j=1}^{V} A_{ij}$
- Set $\boldsymbol{D} = \mathrm{diag}(d_1, \ldots, d_V)$; then graph Laplacian is def'd as

$$\boldsymbol{L} = \mathbf{1} - \boldsymbol{D}^{-1/2} \boldsymbol{A} \boldsymbol{D}^{-1/2}$$

- Spectral graph theory: $\boldsymbol{L}$ has eigenvalues in $0 \ldots 2$

Graph covariance functions

Definition

- From graph Laplacian, can define covariance "functions" (really $V \times V$ matrices)
- Random walk kernel, $a \geq 2$:

$$\boldsymbol{C} \propto (a - \boldsymbol{L})^p \propto \left[(a-1)\,\mathbf{1} + \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}\right]^P$$

- Diffusion kernel:

$$\boldsymbol{C} \propto \exp\left(-\frac{\sigma^2}{2}\boldsymbol{L}\right) \propto \exp\left(\frac{\sigma^2}{2}\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}\right)$$

- Useful to normalize so that $(1/V)\sum_i C_{ii} = 1$

Graph covariance functions

Interpretation

- Random walk on graph has transition probability matrix $A_{ij}d_j^{-1}$ for transition $j \to i$
- After s steps, get $(\boldsymbol{A}\boldsymbol{D}^{-1})^s = \boldsymbol{D}^{1/2}(\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2})^s\boldsymbol{D}^{-1/2}$
- Compare this with

$$\boldsymbol{C} \propto \sum_{s=0}^{p} \binom{p}{s} (1/a)^s (1-1/a)^{p-s} (\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2})^s$$

- So $\boldsymbol{D}^{1/2}\boldsymbol{C}\boldsymbol{D}^{-1/2}$ is a random walk transition matrix, averaged over distribution of number of steps:

$$s \sim \text{Binomial(p,1/a)} \qquad \text{or} \qquad s \sim \text{Poisson}(\sigma^2/2)$$

- Diffusion kernel is limit $p, a \to \infty$ at constant $p/a = \sigma^2/2$

Outline

Random regular graphs

- Regular graphs: Every node has same degree d
- Random graph ensemble: all graphs with given V and d are assigned the same probability
- Typical loops are then long ($\propto \ln V$) if V is large
- So locally these graphs are tree-like
- How do graph covariance functions then behave?
- Expect that after many random walk steps ($p \to \infty$), kernel becomes uniform: $C_{ij} = 1$, all nodes fully correlated

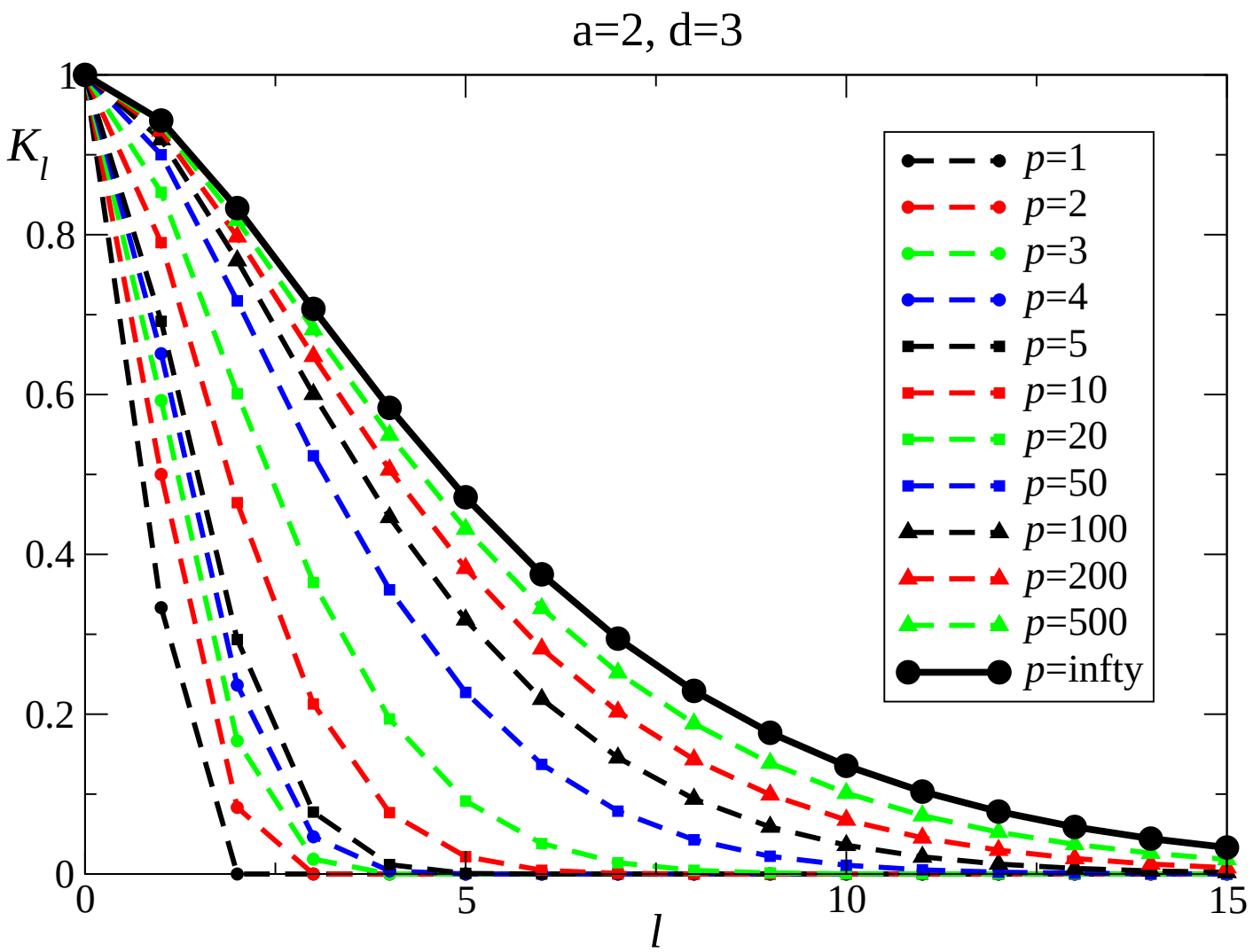
Covariance functions on regular trees

- On regular trees, all nodes are equivalent
  (except for boundary effects)
- So kernel C_{ij} is a function only of distance ℓ measured along the graph (number of links between i and j)
- Can calculate recursively over p: $C_{\ell,p=0} = \delta_{\ell,0}$ and

$$
\begin{aligned}
C_{0,p+1} &= \left(1 - \frac{1}{a}\right) C_{0,p} + \frac{d}{ad} C_{1,p} \\
C_{\ell,p+1} &= \frac{1}{ad} C_{\ell-1,p} + \left(1 - \frac{1}{a}\right) C_{\ell,p} + \frac{d-1}{ad} C_{\ell+1,p}
\end{aligned}
$$

- Normalize afterwards for each p so that $C_{0,p} = 1$
- Let's see what happens for $d = 3$, $a = 2$ and increasing p

Effect of increasing p



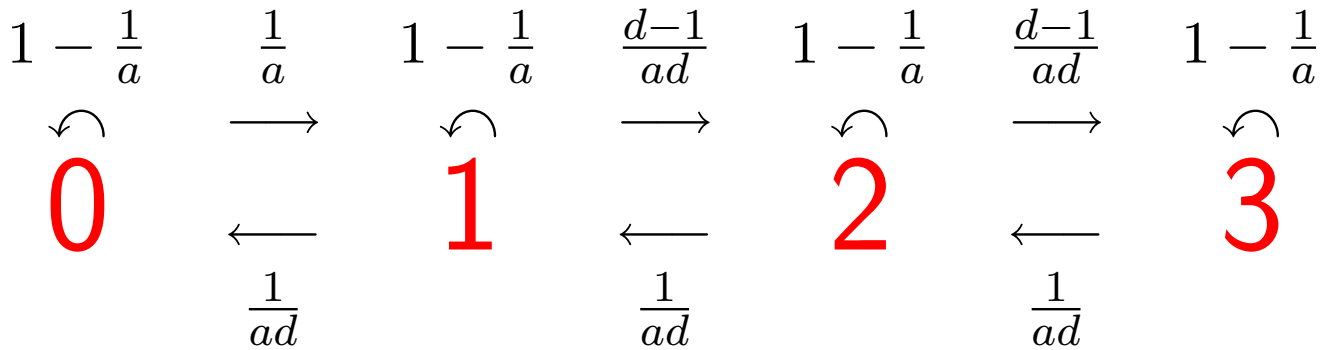
Kernel does not become uniform even for $p \to \infty$

What is going on?
Mapping to biased random walk

- Gather all the (equal) random walk probabilities over the shell of nodes at distance ℓ:

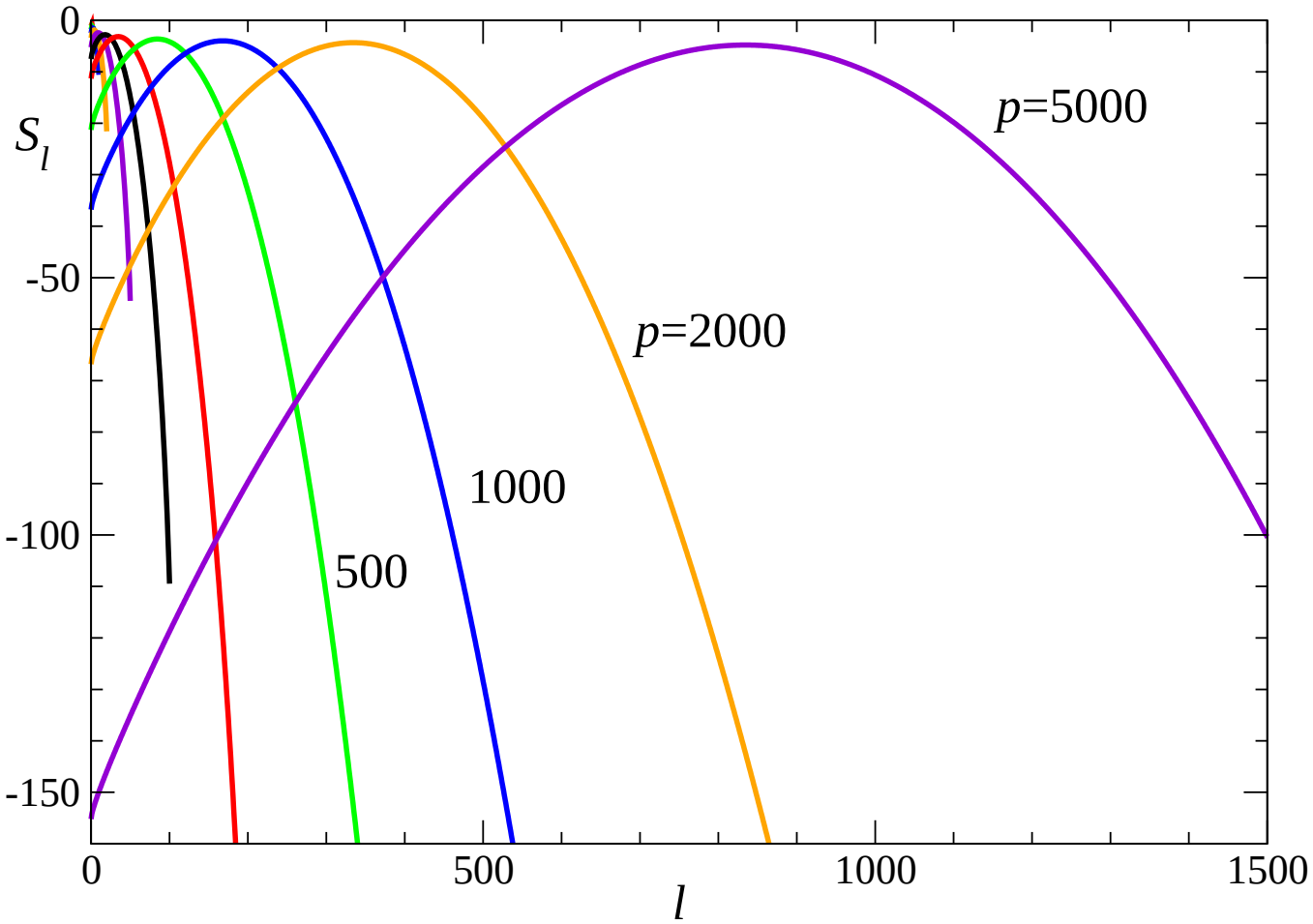
$$S_{0,p} = C_{0,p}, \qquad S_{\ell,p} = d(d-1)^{\ell-1} C_{\ell,p}$$

- Then recursion $S_{\ell,p} \longrightarrow S_{\ell,p+1}$ represents a biased random walk in one dimension, with reflecting barrier at origin:



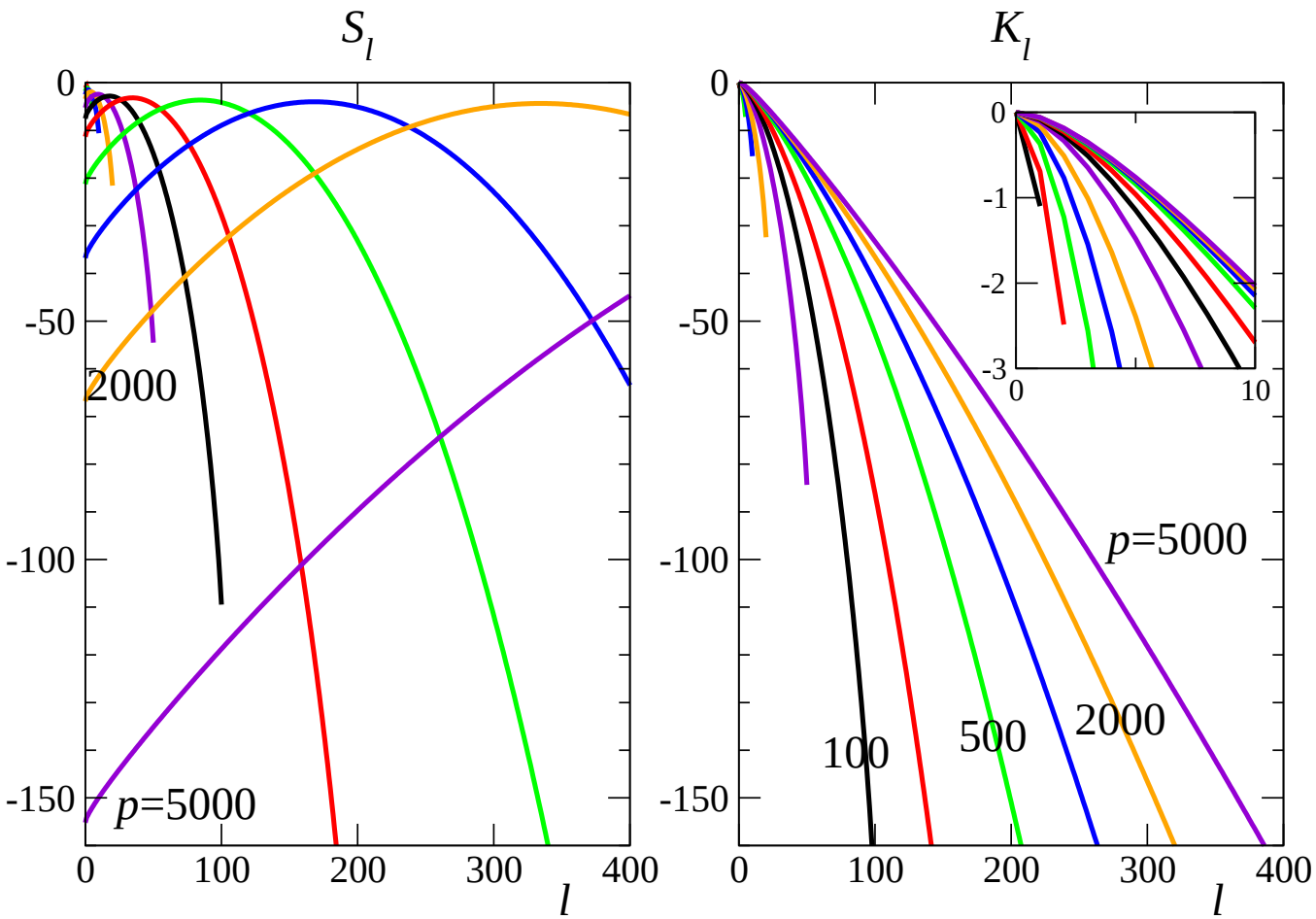
Random walk propagation

Plots of $\ln S_{\ell,p}$ versus ℓ for $d = 3$, $a = 2$



$\ell \to \ell + 1$ with prob. $(d-1)/(ad)$, $\ell \to \ell - 1$ with prob. $1/(ad)$,
so $S_{\ell,p}$ has peak at $\ell = (p/a)[(d-2)/d]$

Converting back to $C_{\ell,p} \propto S_{\ell,p}/(d-1)^{\ell-1}$



Covariance function determined by tail of $S_{\ell,p}$ near origin

Can be used to calculate $C_{\ell,p\to\infty} = [1 + \ell(d-1)/d](d-1)^{-\ell/2}$

Outline

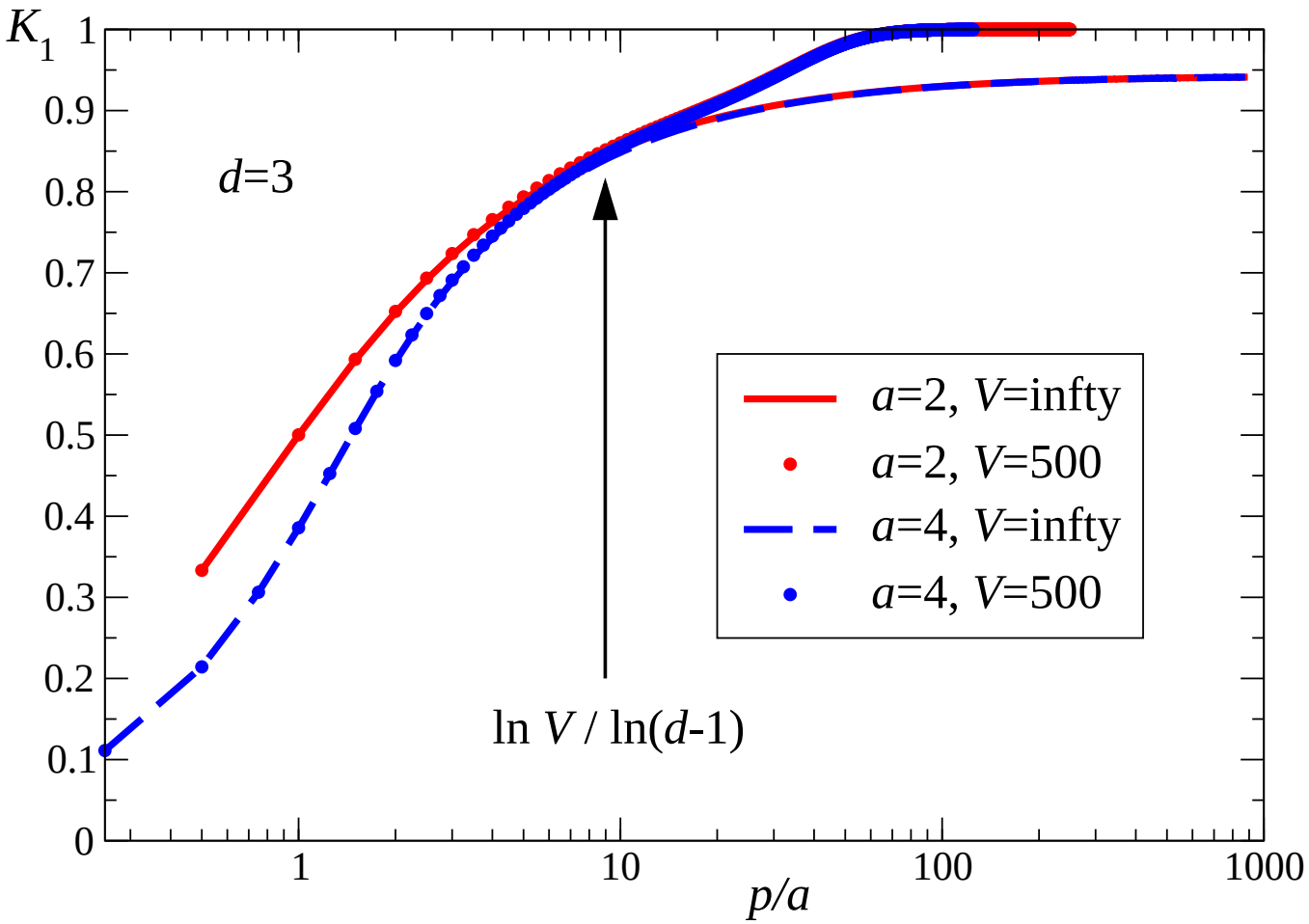
Effect of loops

- Eventually, approximation of ignoring loops must fail
- Estimate when this happens: tree of depth ℓ has $V = 1 + d(d-1)^{\ell-1}$ nodes
- So a regular graph can be tree-like at most out to $\ell \approx \ln(V)/\ln(d-1)$
- Random walk on graph typically takes p/a steps, so expect loop effects to appear in covariance function around

$$\frac{p}{a} \approx \frac{\ln(V)}{\ln(d-1)}$$

- Check by measuring average of $K_1 = C_{ij}/\sqrt{C_{ii}C_{jj}}$ (i, j nearest neighbours) on randomly generated graphs

Covariance function for neighbouring nodes



K_1 starts to get larger than for tree approximation ($V \to \infty$)

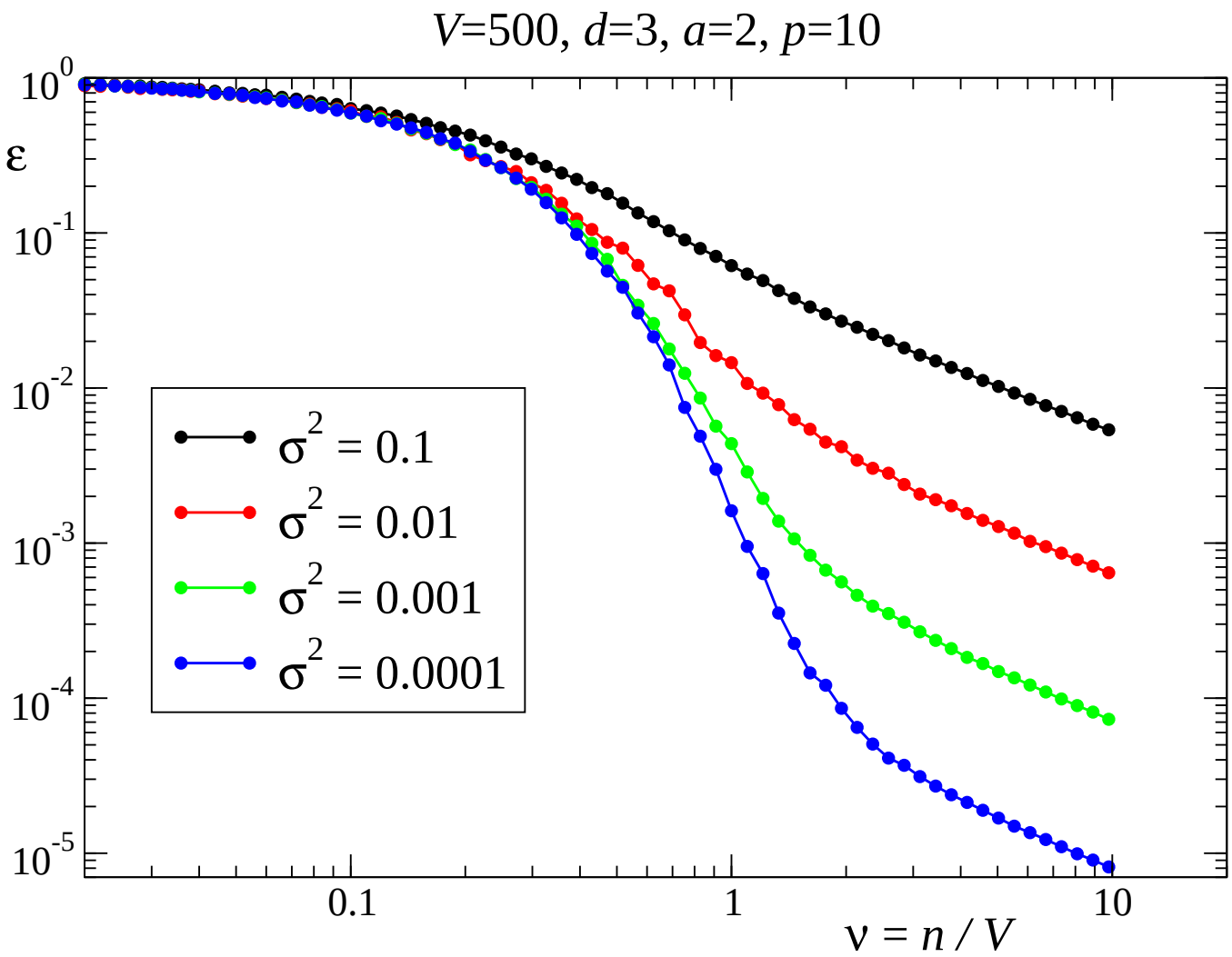
Results depend only on p/a for large p as expected

Outline

Bayes errors and learning curves

- Generalization error ϵ of GP regression can be expressed in terms of covariance function for any given dataset
- Assume we have the correct prior (matched case)
- Then ϵ is the Bayes error (loss = squared difference)
- Average over datasets of given size n gives learning curve $\epsilon(n)$
- Take distribution of inputs to be uniform across graph
- How does this depend on n, V, d(=3 here), a, p, and noise variance σ^2?

Some simulation results for orientation



Two different regimes: $\epsilon > \sigma^2$ and $\epsilon < \sigma^2$

Theory: Learning curve approximation

- Approximations for the learning curve are based on kernel eigenvalues

$$\langle C_{ij}\phi_j \rangle = \lambda \phi_i$$

  where $\langle \ldots \rangle$ is over input distribution across nodes

- Try simple but often accurate approximation

$$\epsilon = g\left(\frac{n}{\epsilon + \sigma^2}\right), \qquad g(h) = \sum_{\mu=1}^{V} (\lambda_\mu^{-1} + h)^{-1}$$

- Has to be solved self-consistently; note that $g(0) = \sum_\mu \lambda_\mu = \langle C_{jj} \rangle = 1$

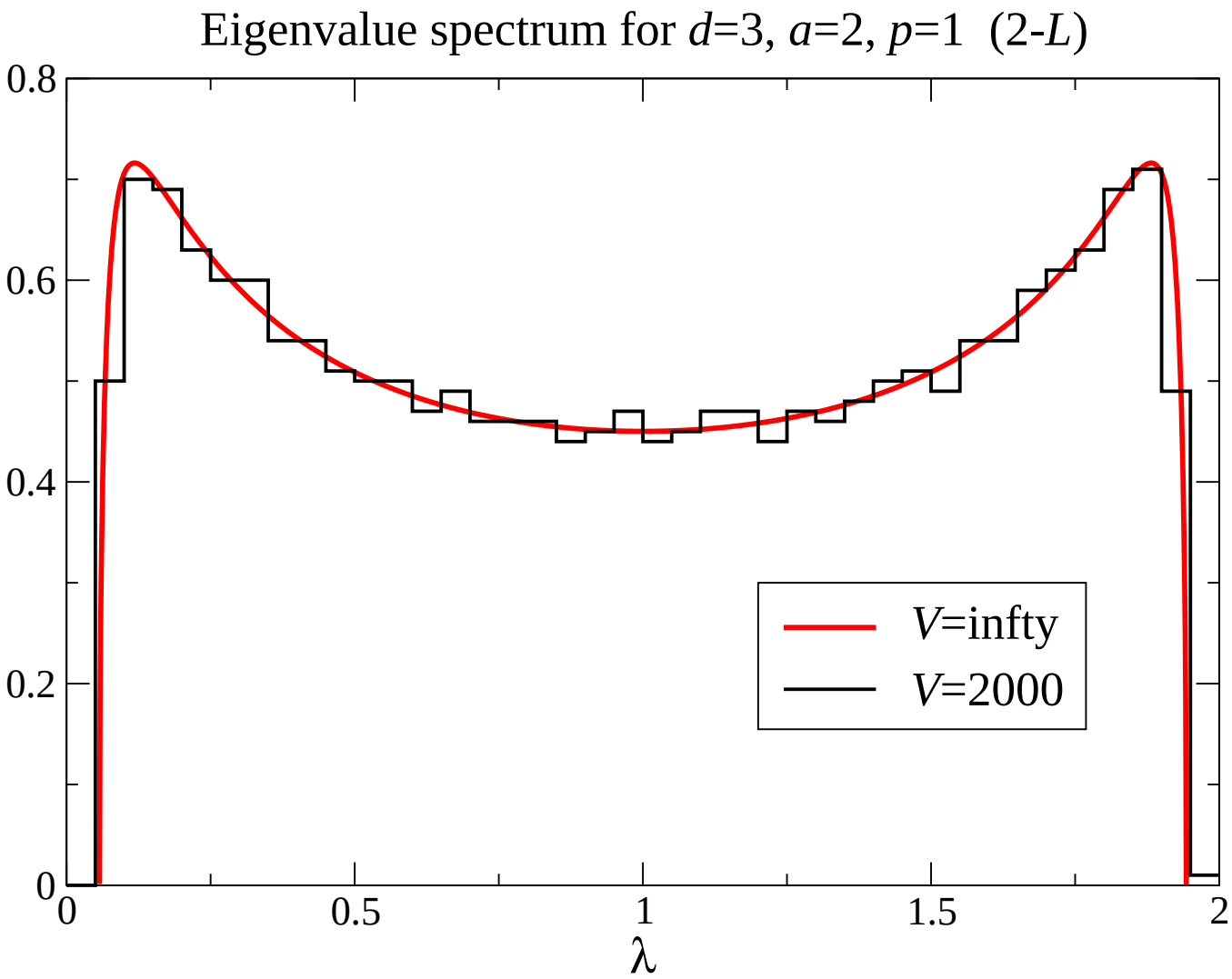
Theory: Limit of large V

- For large V, tree approximation should be accurate
- Tree Laplacian eigenvalue density is known:

$$\rho_L(\lambda) = \frac{\sqrt{\frac{4(d-1)}{d^2} - (\lambda - 1)^2}}{2\pi d\lambda(2-\lambda)}$$

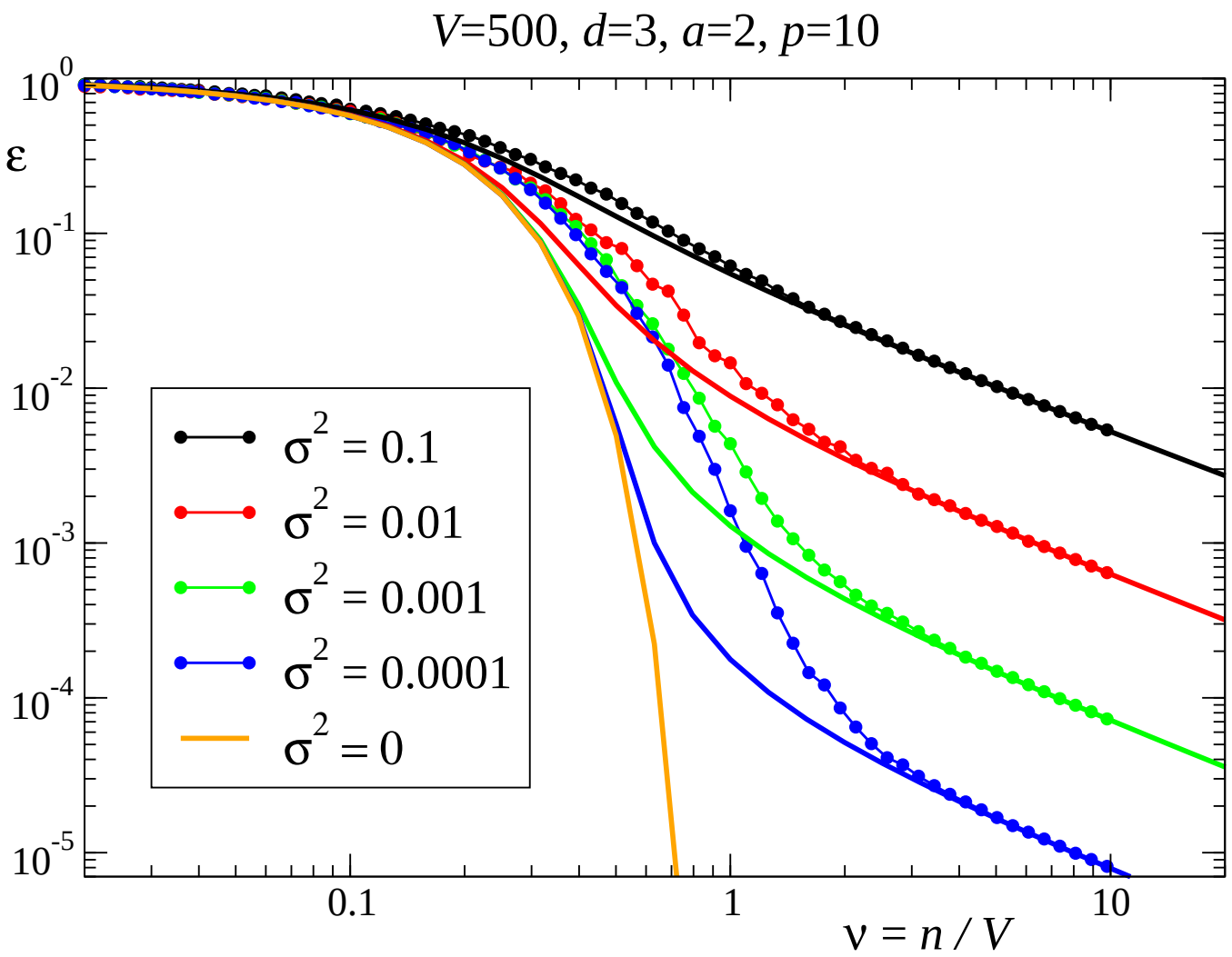
- Eigenvalues of covariance function are then $\propto V^{-1}(a-\lambda)^p$
- Use this to evaluate approximate learning curves; they depend on n and V only through $\nu = n/V$

Eigenvalue spectra



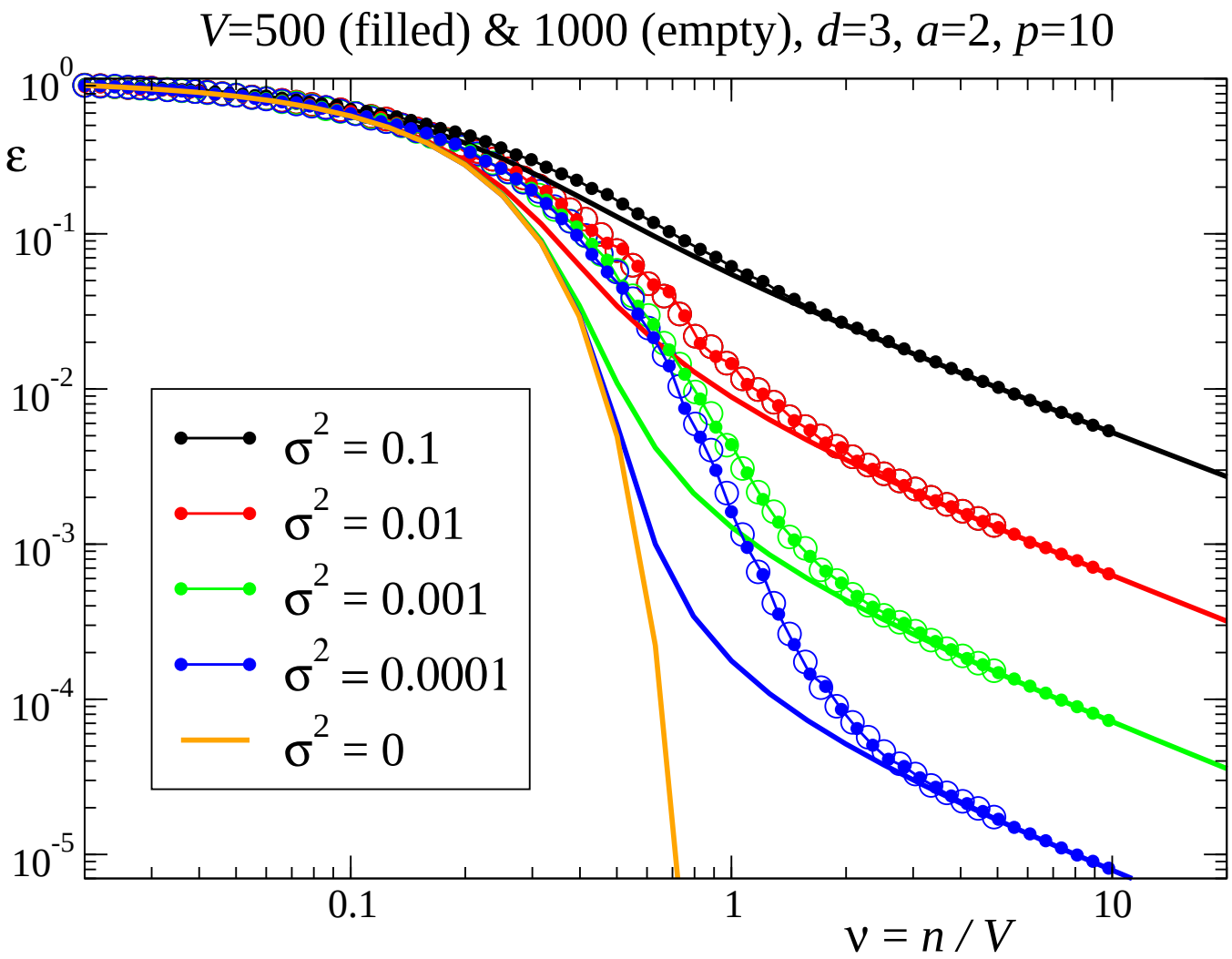
Tree approximation quite accurate

Comparison with simulations



Accurate initially and for $\epsilon < \sigma^2$, less so in crossover

Scaling with n/V



Works well throughout

Outline

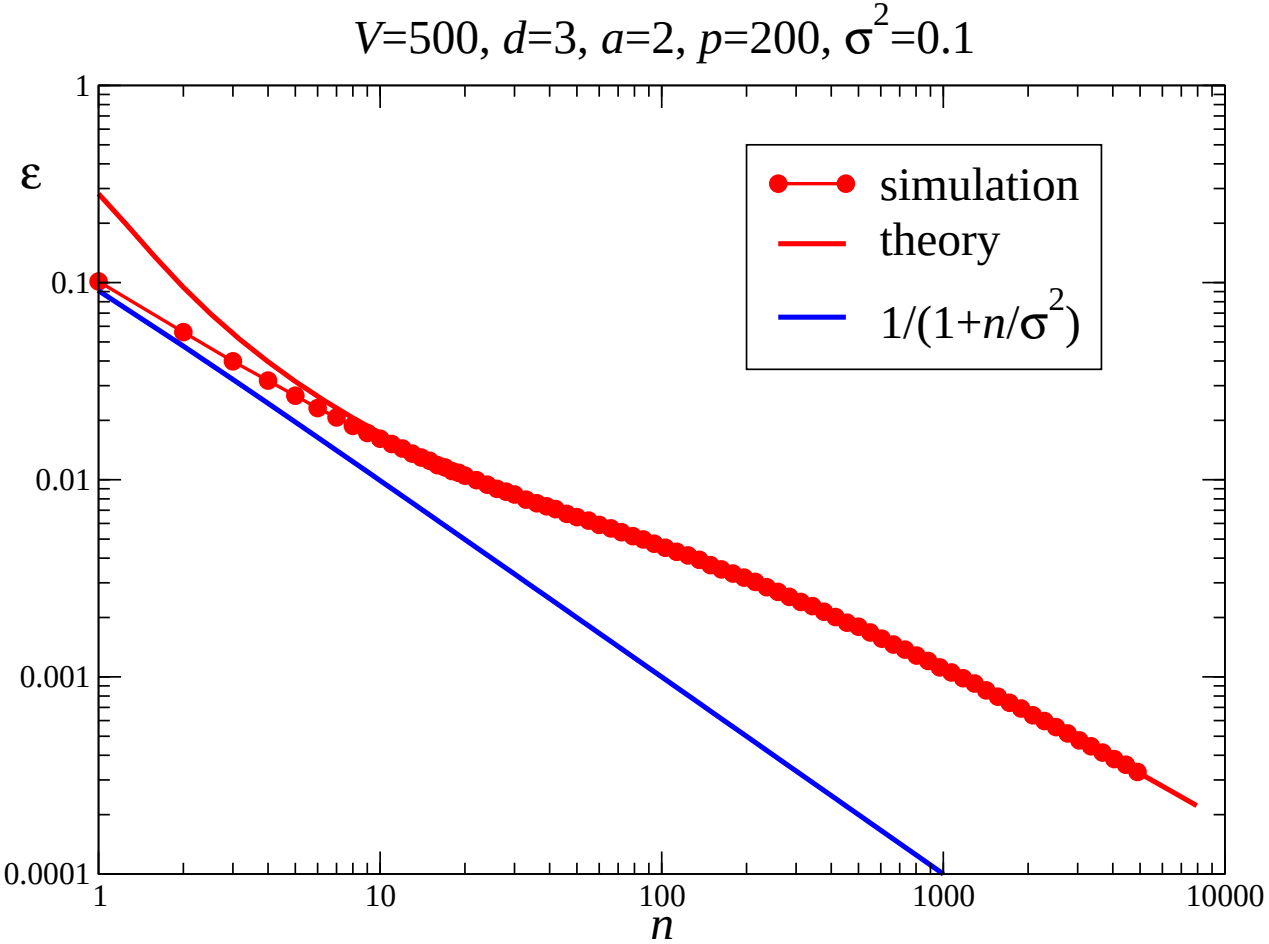
Effect of loops for large p

- Tree approximation must break down as p increases, when loops become important
- Eventually, when covariance function is uniform, need to learn only one function value so expect

$$\epsilon = \frac{1}{1 + n/\sigma^2}$$

- Consider a case with $V = 500$, $p = 200$, $a = 2$, $d = 3$
- Compare to naive estimate and approximation based on true kernel eigenvalues

Simulations vs theory for large p

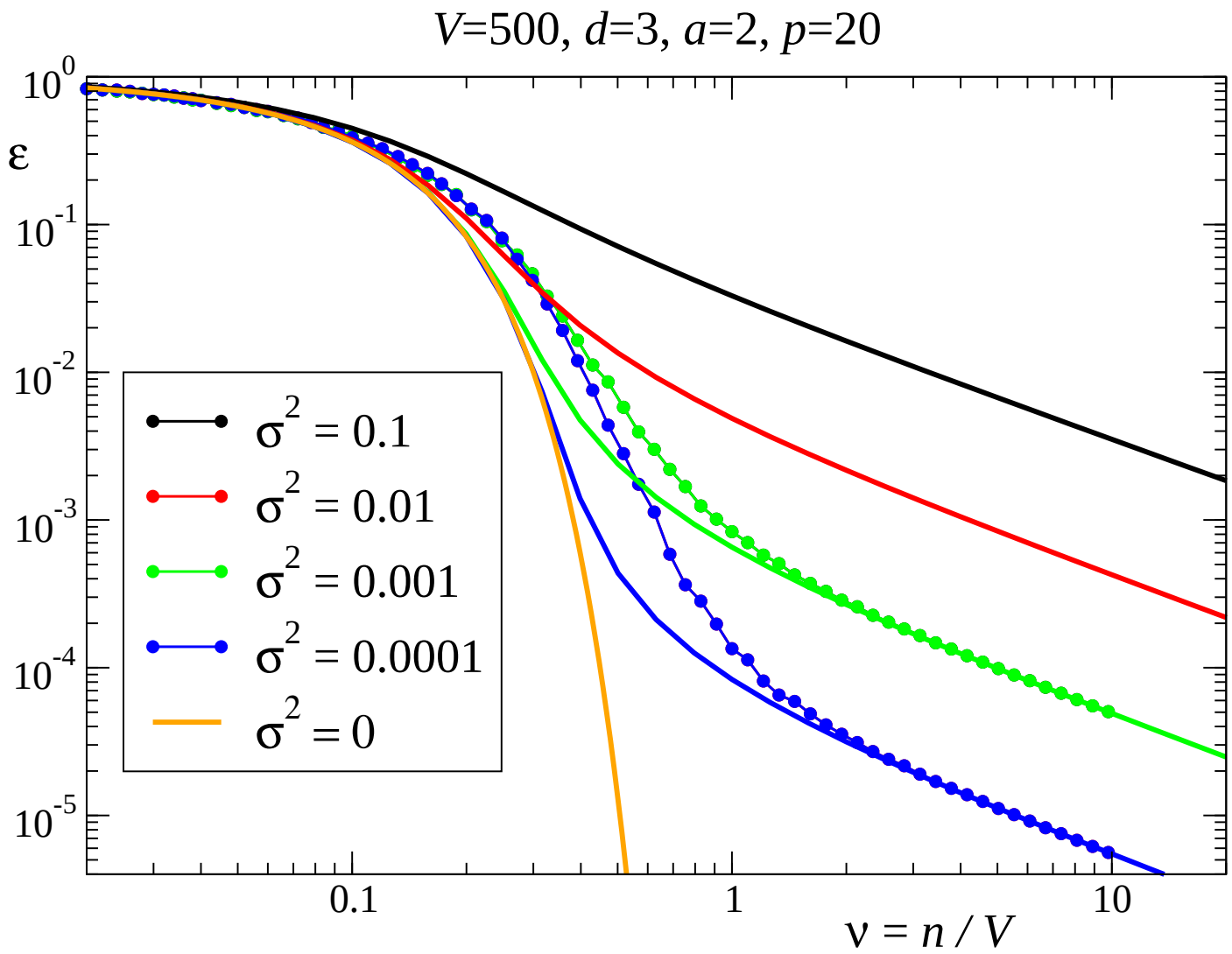


Naive estimate poor even though $\lambda_1 \approx 0.994$

Theory works well; tail is $\propto (\sigma^2/n)\ln(n)$

Outline

Effect of increasing p



Theory becomes more accurate

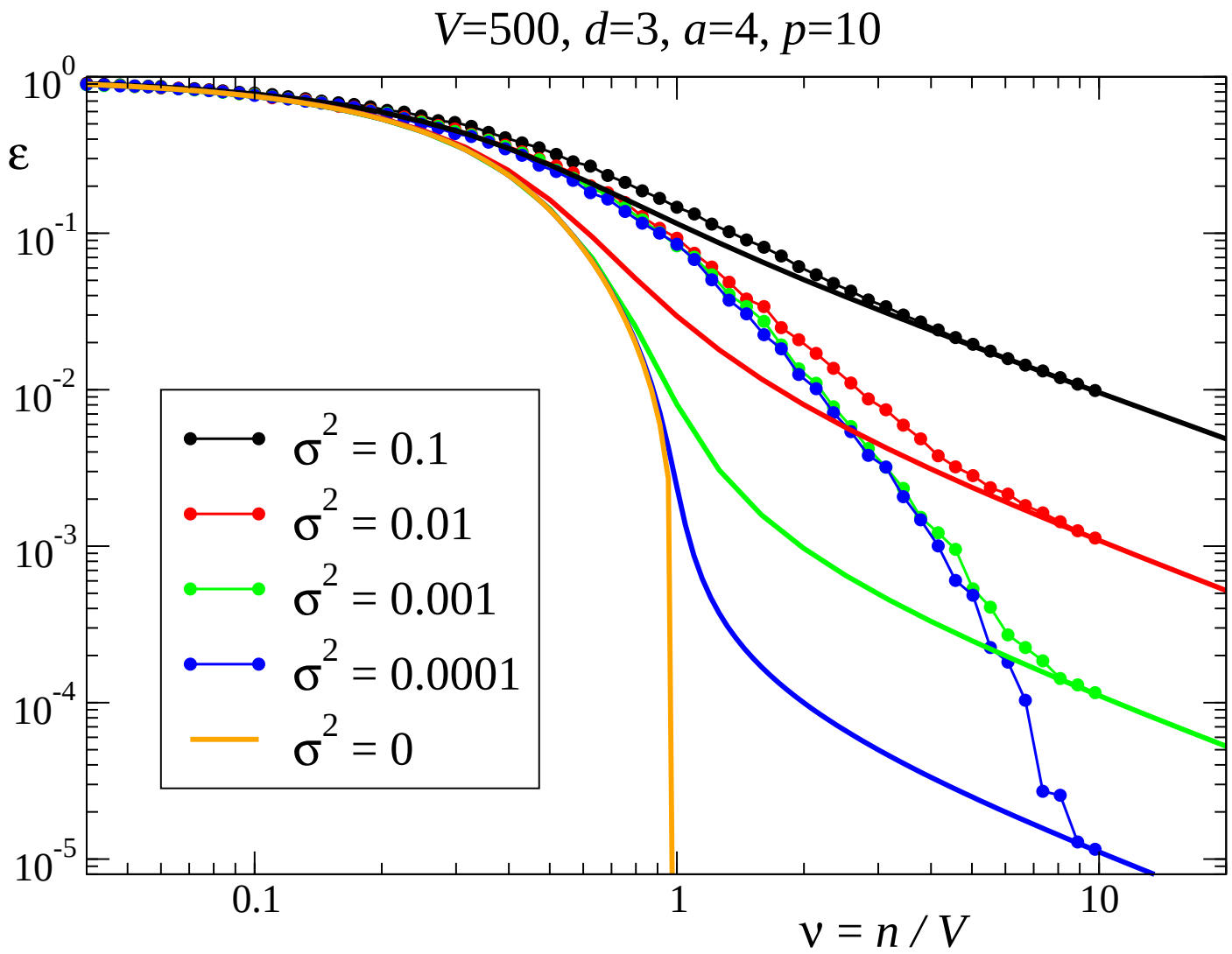
(Approximate) predictions for large p

- Comparison with simulation shows that theory becomes more accurate
- For large p, find that learning curve tail ($\epsilon \ll \sigma^2$) decays as

$$\epsilon \sim \frac{c\sigma^2}{\nu} \ln^{3/2}\left(\frac{\nu}{c\sigma^2}\right), \qquad c \sim (p/a)^{-3/2}$$

- So density ν to reach a certain ϵ decays $\sim c \sim p^{-3/2}$
- Even though kernel $C_{\ell,p}$ at fixed graph distance becomes p-independent for large p, learning still gets faster
- Presumably an effect of kernel values for large $\ell \sim p$?

Effect of increasing a



Theory becomes less accurate

Effect of increasing a: Limit $a \to \infty$

- Increasing a means typical number of steps in random walk, p/a, decreases
- Extreme limit $a \to \infty$ gives $C_{ij} = \delta_{ij}$: all nodes uncorrelated
- Approximation then predicts

$$\epsilon = \tfrac{1}{2}(1 - \nu - \sigma^2) + \sqrt{\tfrac{1}{4}(1 - \nu - \sigma^2)^2 + \sigma^2}$$

- Compare exact result:

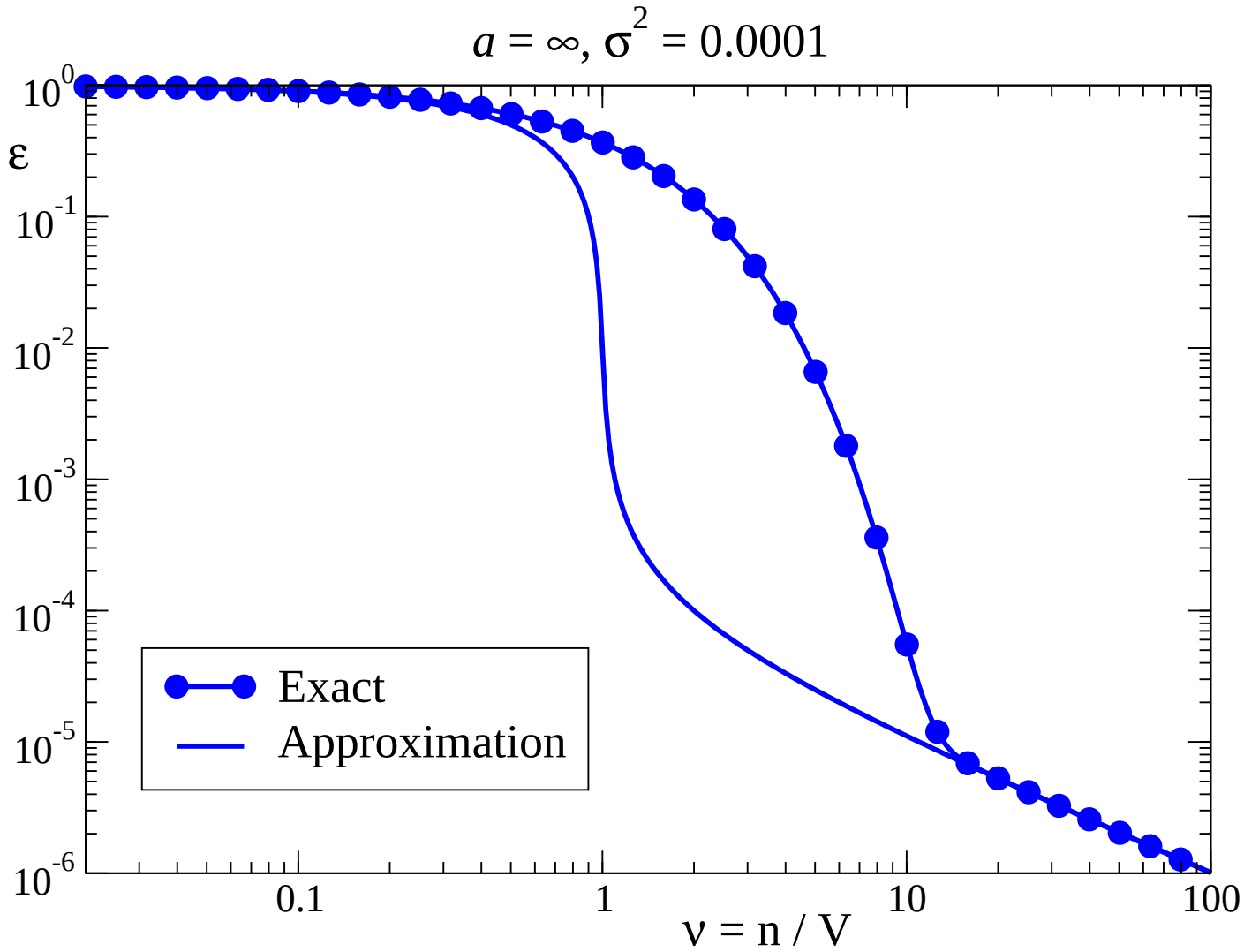
$$\epsilon = \left\langle (1 + n_i/\sigma^2)^{-1} \right\rangle, \qquad n_i = \text{Binomial}(n, 1/V)$$

- In low-noise limit $\sigma^2 \to 0$ these become (for large V)

$$\epsilon = 1 - \nu \qquad \text{vs.} \qquad \epsilon = \exp(-\nu)$$

so approximation gives an underestimate

Limit $a \longrightarrow \infty$



Same "shape" of deviation as before for larger finite a

Summary and outlook

- Kernels on graphs have some counter-intuitive properties
- Function values on different nodes only become fully correlated due to loop effects
- Nontrivial limiting kernel shape ($p \to \infty$) on regular trees, can be obtained from biased random walk
- For not-too-large p, learning curves scale with $\nu = n/V$
- For large p, loops give fully-correlated limit, but with significant corrections
- Simple approximation works well except for small p/a, in crossover region ($\epsilon \approx \sigma^2$)
- Future work: Prior mismatch?
- Other graph structures? Poisson, small-world, etc?