Lecture 6 Script

1 Additive Perturbations of Graph Filters

Slide 1: Additive Perturbations - Title Page

(1) We have seen that integral Lipschitz graph filters can be stable to scalings of the shift operator. This was an interesting exercise but we need to investigate more generic perturbation models. In this section we define additive perturbations of shift operators and their effects on the outputs of graph filters.

Slide 2: A Graph Filter can be Perturbed in Three Ways but Only One is Interesting

(1) A question we barely touched upon is the motivation for studying shift perturbations. A related question is why not study perturbations of the other components of a filter.

(2) Indeed, recall that we define a graph filter as a polynomial on the shift operator S with coefficients h_k. The output of the filter is constructed by successively aggregating diffused versions of the input signal scaled by the corresponding filter coefficients. Thus, the output of a filter changes if we deform the input, the filter coefficients, or the shift operator. Of these three, operator perturbations are the most interesting.

(3) Since the filter is linear on the input signal, perturbations of x are straightforward. The error propagates to the output as the error at the input scaled by the filter's norm.

(4) Likewise, the filter is also linear on the coefficients h_k. Plus, the coefficients are design parameters. They are controlled by us. There's no reason to study their perturbations. Although we may be interested in studying their sensitivity.

(5) Deformations of the shift, on the other hand, are not easy. The filter is a nonlinear function of S. It is not clear how perturbations of S propagate to the output. In addition, studying shift perturbations is necessary.

(6) The graph shift operator is often estimated from observations. This is the case of recommendation systems, for example. Rating similarity graphs are estimates subject to error.

(7) In other applications, the graph changes over time. Or it changes from realization to realization. Both change modalities happen in distributed autonomous systems. These are made up of multiple agents that collaborate to solve a common task. The relative positions of the agents change as they move through their environment. Generating changes in the graph describing their interactions. And the system is to be deployed repeatedly using the same learned policy. The team configurations vary from deployment to deployment.

(8) A second equally important reason to study graph deformations is the exploitation of quasi symmetries in graphs that are quasi equivariant to permutations. Which we have seen is an important factor explaining the value of graph filters and GNNs in machine learning.

Slide 3: Perturbations of Graph Filters

(1) We therefore want to the study the effect of running the same filter h with the same signal x on different graphs S and S-hat. Because when faced with a signal x, we have learned filter coefficients h_k that solve a given learning problem when operating on the graph S. And we want to know if we are still close to solving the learning problem when operating on graph S-hat.

(2) That is, we have the filter output H-of-S times x constructed by sequentially aggregating diffusions of the input signal x over the graph shift operator S modulated by coefficients h_k. We know this produces good AI estimates.

(3) But because of estimation errors, variability, or symmetries, we are executing the filter on the graph H-of-S-hat. Still with the same input and the same coefficients. Are these output estimates still good?

(4) To answer this question we study the difference between the filter output H-of-S times x. Which uses coefficients h_k to process input signal x over shift S

(5) And filter output H-of-S-hat times x. Which uses the same coefficients h_k to process the same input signal x. But it does so on the perturbed shift S-hat. The filter is the same. It process the same signal. But it is instantiated on a different graph.

(6) We already investigated the particular case of scaling perturbations. They are insightful but limited. We move on to investigate more generic perturbation models.

Slide 4: Additive Perturbation

(1) We start with an additive perturbation model. As the name suggests, the perturbed graph shift operator S hat corresponds to the addition of an error matrix E to the original graph shift operator S.

(2) If we are given a pair of graph shift operators S and S hat, the error matrix E is their difference.

(3) Therefore, the norm of the error matrix E quantifies how different S and S hat are from each other.

(4) This way of measure shift differences has a flaw, however.

(5) We know, or if you wish, we are assuming, that graph shift operators that are related by a permutation are the same. Save for node labels.

(6) Yet, the norm of the error matrix E may not be zero. Indeed, it most likely won't be unless the graph has a permutation symmetry. But the error should be zero. The graph does not change because we have decided to change node labels.

(7) We already know how to circumvent this issue.

(8) We do that with the notion of operator distance modulo permutation.

(9) Which we have defined as the minimum operator norm of the difference between all the possible permutations of shifts S and S hat. The operator distance modulo permutation is zero if the graphs are permutations of each other.

Slide 5: Additive Perturbation Modulo Permutation

(1) Our perturbation analyses use these distances modulo permutation. But they also need concrete handles on the error matrix associated with the permutation that attains minimum distance.

(2) To get this handle we introduce a set calligraphic E-of-S and S-hat. Containing all of the error matrices E tilde that relate the graph S to a permutation of S-hat.

(3) For each permutation matrix P, we have a different error matrix E tilde evaluated as the subtraction of the shift S from the corresponding permutation of the shift S-hat.

(4) Given this set of candidate error matrices, we define the error matrix modulo permutation as the member of the set that achieves the smallest norm. This is the matrix E that we add to S to make it equal to the permutation of S-hat that is closest to S.

(5) With this notion in hand, we can now rewrite the distance modulo permutation as the norm of the matrix E. This is the same notion as before. But explicitly written in terms of the error matrix that relates S to the permutation of S-hat that is closest to S.

(6) The norm of the error matrix E is a measure of how far S and S hat are from being permutations of each other. This is a distance measure that is independent of node labels.

Slide 6: Eigenvector Misalignment Constant

(1) The reason for us to get a handle on the error matrix E is because we want to compare its eigenvectors to the eigenvectors of S. Thus, consider the eigenvector decompositions.

(2) V times Lambda times V-Hermitian of the shift operator S.

(3) And U times M times U-Hermitian of the error matrix E.

(4) With these decompositions in hand, we define the eigenvector misalignment between the shift operator S and the error matrix E as the constant delta given by:

(5) The norm of the difference between the eigenvector matrices U and V of the error matrix and shift operator.

(6) To which we add a 1.

(7) We take a square.

(8) And subtract 1. The eigenvector misalignment measures the difference between the eigenvectors of the shift S and the error E. If these two matrices have the same eigenvectors, U equals V and the misalignment constant is 0.

(9) It is important to point out that the matrices U and V are orthonormal eigenvector bases. Hence, they are unitary and have unit norms. This implies that delta is upper bounded by 8.

(10) This means that the eigenvector misalignment constant is never large. It can be small depending on the particular error model. We introduce it for this latter reason. To take advantage of perturbation models in which the eigenvectors of the shift and error matrices are similar.

2 Stability of Lipschitz Filters to Additive Perturbations

Slide 7:Stability of Lipschitz Filters to Additive Perturbations - Title Page

(1) We discuss the stability of graph filters to additive perturbations of the graph support.

Slide 8: Lipschitz Filters are Stable to Absolute Perturbations

(1) With a model and preliminary definitions in place we are ready to present a theorem stating that graph convolutions with Lipschitz filters are stable to additive perturbations.

(2) Consider a filter h. Along with two graphs with n nodes and shift operators S and S hat such that the following three hypotheses hold.

(3) One: Shift S hat is related to shift S through an additive perturbation model with error matrix E and permutation matrix P. We can write the permutation P-transpose-S-hat-P of the shift S-hat as the sum of the shift S and the error matrix E.

(4) Two: The error matrix E has norm epsilon. And eigenvector misalignment delta. The latter measured relative to S.

(5) Three: The filter h is Lipschitz with constant C.

(6) When these conditions hold.

(7) The operator distance modulo permutation between the filter H-of-S deployed over the graph shift operator S, and the filter H-of-S hat deployed over shift S hat.

(8) Is upper bounded by

(9) The Lipschitz constant C of the filter h.

(10) Multiplied by 1-plus-delta-square-root-of-n. Where delta is the eigenvector misalignment constant between E and S. And n is the number of nodes in the graph.

(11) Times the norm of E. Which is the distance modulo permutation between the graph shift operators

(12) This is a first order bound. There are a higher order terms that vanish at least quadratically.

Slide 9: Parse the bound

(1) The theorem claims that Lipschitz filters are stable to additive perturbations of the graph shift operator. There are several details that deserve parsing. We repeat the theorem's thesis here for ease reference during the discussion.

(2) The claim of the theorem is that if the shift operators S and S-hat are epsilon-close to each other, the respective filters H of S and H of S-hat are also epsilon-close to each other. Closeness here is modulo permutation. Both, the shift operator deformation and the filter deformation are measured for the permutations that make them closest to each other. We are using operator distances modulo permutation which we introduced for this purpose.

(3) The epsilon-closeness of the filters, is proportional to the Lipschitz constant of the filter's frequency response. Filters with larger C are more sensitive to graph perturbations.

(4) The epsilon-closeness of the filter is also proportional to the factor 1 + delta square root of n. This is a factor that depends on the eigenvector misalignment constant delta and the number of nodes in the graph n. As it grows with n, this is not too great for large graphs. Unless the misalignment of the eigenvectors of E relative to the eigenvectors of S decrease with n.

(5) The growth with the number of nodes is at most 8 times square-root-of-n. Because delta is at most 8. This is not great as we have just said. But it is not too bad either. Square roots do not grow that fast.

(6) (Empty)

Slide 10: Stability is Stronger than Continuity

(1) There are some other remarks worth making.

(2) Observe first that according to the theorem, the distance modulo permutation between the outputs of the filters deployed on S and S hat is linearly bounded by the distance epsilon between the graph shift operators. We can thus state that filter perturbations are Lipschitz continuous with respect to the perturbation's size. This is true to first order. There are second order terms we did not characterize.

(3) The Lipschitz constant of this Lipschitz continuity statement is the product of C with 1 + delta square root of n.

(4) But a more conceptual point to highlight is that this is a stronger condition than plain continuity. Continuity would state that small changes in the input lead to small changes in the output. Here, not only are the changes in the output small if the perturbation is small, but the changes in the output are proportional to the input perturbation itself. This is why we talk of stability, not continuity. It is a stronger statement.

Slide 11: Universality of the Stability Bound

(1) A second important point to highlight is that the bound is universal.

(2) It holds uniformly for all graphs with the same number of nodes n. Inspection of the bound reveals that it depends on:

(3) A property of the filter's frequency response. The filter's Lipschitz constant C. This is determined by the choice of filter's coefficients. It is independent of the graph.

(4) And two properties of the perturbation matrix E. The eigenvector misalignment delta and the error norm epsilon.

(5) There is no constant in the bound that depends on the graph shift operator, save for n, of course. Which makes the bound universal to all graphs with the same number of nodes.

Slide 12: The filter's Lipschitz constant is a Controllable Design Parameter

(1) A third remark is to highlight the dependence of the stability bound with the filter's Lipschitz constant C.

(2) This is a parameter that we can control through filter design. It is true we are learning coefficients, not designing them. But we can bias the learning towards filters with smaller Lipschitz constant. Or not. The important point is that the constant C depends on something that we choose: The filter coefficients.

(3) From our study of Lipschitz filters, we know that C controls the discriminability of the filter. We therefore see the emergence of a stability versus discriminability tradeoff.

(4) If we make C large, the filter becomes sharper, and thus more discriminative. But given the stability bound, the filter also becomes more susceptible to perturbations. Conversely, we make the filter more stable by decreasing C. But this leads to frequency responses that vary more slowly. Which decreases the discriminability of the filters.

(5) (Empty)

Slide 13: The Eigenvector Misalignment is an Uncontrollable Property of the Perturbation

(1) The fourth and final remark into highlight the dependence of the stability bound on the eigenvector misalignment constant.

(2) The important point to make is that the eigenvector misalignment constant is a property of the perturbation matrix. It is a property that is independent of the filter choice. It depends exclusively on the the particular application or problem at hand. But we have no control over its value. Graphs are what they are and error matrices are what they are.

(3) This make the constant somewhat irrelevant in the study of stability versus discriminability tradeoffs of different filters. There's nothing that filter choice can do to alter its effect on the stability of graph filters. It is like my cat Leo. He's just there. He doesn't do much.

(4) That said, the fact still remains that the eigenvector misalignment makes the stability bound meaningless asymptotically on n. There isn't much that is known about the stability of graph filters in the limit of the number of nodes n growing to infinity.

Slide 14: Lipschitz Filters are Good News

(1) Stability to additive perturbations requires Lipschitz filters. Not integral Lipschitz filters, as was the case with scalings.

(2) This is good news because Lipschitz filters, offer a genuine trade-off between discriminability and stability.

(3) We can increase C to tradeoff stability for discriminability. For instance, if we replace this example frame

(4) With this other frame we increase discriminability because the filters are thiner are more tightly packed. To achieve this we increased the Lipschitz constant of the filters in the frame. We know that this decreases stability. Which is the price we have to pay to increase discriminability.

(5) This is different from what happens in the case of dilations. Where the tradeoff depends on the frequency and is impossible to satisfy for large frequencies. In the case of additive perturbations the stability vs discriminability tradeoff is the same for all

frequencies. And since the bound is universal for all graphs, we know that irrespectively of the properties of the graph or the properties of the signal, it is possible for us to increase discriminability at the cost of reducing stability. But without having to completely give up on stability.

3 Relative Perturbations of Graph Filters

Slide 15: Relative Perturbations of Graph Filters - Title Page

(1) We studied additive perturbation models and obtained enticing stability results. Alas, the results are a mirage. Additive perturbations are not as meaningful as they look on first inspection.

(2) We switch our focus to **relative** perturbations. Which tie the form of the perturbation to the structure of the graph.

Slide 16: Limitations of Additive Perturbations

(1) We start by addressing head on the fact that additive perturbations are not meaningful. We studied them only to draw a contrast with the relative perturbations that we are studying in this section.

(2) To understand the reason why this is true consider the graph we show on the right. This graph has a community that is strongly connected with weights Capital W and a community that is weakly connected with weights lowercase w. The two communities are connected with a link of weight 1.

(3) To this graph we add the perturbation matrix E we show below it. All of the nonzero weights in the perturbation are set to 1.

(4) The resulting operator hat-S. Which, by the way, is defined modulo permutation. There may be a relabeling goin on. Is the sum of these two graphs.

(5) If lower case w is much smaller than 1 and upper case W is much larger than one.

(6) Is this a small perturbation? Or is it a large perturbation? It is impossible to tell. The left community is almost unchanged, The right community is drastically changed.

(7) Herein lies the problem with arbitrary additive perturbations. Edges with small weights can change a lot because some other edges of the graph have large weights. In this example the norm of the error matrix E is much smaller than the norm of the shift operator S. But this doesn't mean that the perturbation is small in any meaningful sense. The nature of the graph has changed.

Slide 17: Relative Perturbations are Meaningful

(1) Meaningful models of graph perturbations are relative.

(2) We consider not the same shift operator S as before,

(3) But we consider a perturbation model where the error matrix takes the structure of the graph into consideration. The edges of the error matrix for the community on the left are proportional to capital W. They are epsilon times uppercase W. And the edges for the community on the right are proportional to lowercase w. They are epsilon times lowercase w. The inter-community link is epsilon.

(4) The perturbed matrix is the addition of these two. Modulo permutation to account for possible relabelings.

(5) If the weights are as before and we add the hypothesis that epsilon is much smaller than one.

(6) Is this a small perturbation or a large perturbation.

(7) It is certainly small. Edges with small weights change a little. Edges with large weights change more. The character of the graph does not change drastically. It is still a graph with a strong community on the left and a weak community on the right. Connected by a link of intermediate strength. Now, the way we have written it the perturbation is still additive.

(8) But we can rewrite the model as a relative error perturbation in which the shift operator S is perturbed by the error matrix epsilon times Identity multiplying the shift operator S. This is the scaling model we have already studied.

Slide 18: Relative Perturbations Modulo Permutations

(1) With this motivation, we introduce the relative perturbation model in which the perturbed shift operator S hat is the sum of the shift operator S with error terms of the form E times

S and S times E. The matrix E here is assumed symmetric. So that the graph S, which we also assume symmetric, is perturbed into the symmetric graph S-hat.

(2) This is fine. But we now we must account for permutations because relabelings are irrelevant.

(3) We also know how to do this. We introduce the set of relative error matrices modulo permutation.

(4) Made up of the symmetric matrices tilde E that allow us to write permutations of the shift operator S-hat as sums of the form S blue tilde-E times S plus S times tilde E. For each possible permutation P we have a different error matrix E relating shift operator S to shift operator S-hat.

(5) Out of this set of candidate relative error matrices we choose the one that with smallest norm. This is the one that we call relative error matrix modulo permutation.

(6) We use the norm of this matrix to define the relative distance modulo permutation between shifts S and S-hat.

(7) Which is a relative measure of how far the shift operators S and S-hat are from being permutations of each other. An important observation to make is that a relative perturbation is also additive. The converse is almost always true as well. What matters is not that the perturbation model is additive of relative. What matters is how we choose to measure its size. Whether we use the relative distance we define in this slide. Or the distance modulo permutation we defined earlier. The examples we have just covered illustrate that relative measure are more meaningful than additive measures.

Slide 19: Relative Perturbations are Tied to the Local Structure of the Graph

(1) More generically, the reason why relative perturbations are more meaningful than additive perturbations is because they tie changes in edge weights to the local structure of the graph.

(2) To explain what this means, evaluate the entry i-j of the perturbed matrix S-hat. With the permutation undone. The definition of the relative perturbations says that we have to compute the i-j entries of the products E times S and S times E.

(3) This is easy to do. What matters is that we end up with two sums that are proportional to the degrees of the incident nodes i and j. That is, the change in the shift operator entry S_i-j, which is the one that connects nodes i and j, is affected by the weights S_{kj}

connecting node j to its neighbors and the weights S_{ik} connecting node i to its neighbors. With these quantities scaled by corresponding error matrix entries.

(4) This means that parts of the graph with weaker connectivity see smaller changes than parts with stronger links. For the weight of an edge i-j to change by a relatively large amount it must be that at least one of the incident nodes, either i or j has a large degree.

(5) This is in contrast to absolute perturbations where edge weights can change by the same amount regardless of the local connectivity of the graph. Do notice that relative perturbations, as defined here, are still a little more arbitrary than we would like them to be. We can still change weight S_i-j by a large amount if i or j have large degrees. Even though the weight S_i-j itself may be small. There is still room to sharpen perturbation models of shift operators.

4 Relative Perturbations of Graph Filters

Slide 20: Stability of Integral Lipschitz Filters to Relative Perturbations - Title Page

(1) We prove stability of integral Lipschitz filters to relative perturbations of the graphs. The use of integral Lipschitz filters engenders the usual discriminability challenges.

Slide 21: Integral Lipschitz Filters are Stable to Relative Perturbations

(1) Having introduce a model of relative perturbations. Or, more appropriately, a way of measuring perturbations of shift operators in a relative sense, we can state a theorem declaring the stability of integral Lipschitz filters to relative perturbations.

(2) To do so, consider a filter h. Along with two graphs with nodes and shift operators S and S hat such that the following three hypotheses hold.

(3) First: Shifts S and S-hat are related by a relative perturbation model. The filter S is perturbed by the the addition of the symmetric relative error term E times S plus S times E. This perturbation is module permutation. The perturbed shift may be relabeled.

(4) Second: The symmetric error matrix E has a norm equal to epsilon and an eigenvector misalignment constant equal to delta. The latter taken relative to shift S

(5) Third: The filter h is integral Lipschitz with constant C

(6) We then have that,

(7) The operator distance modulo permutation between filters H of S and H of S hat

(8) Is bounded by

(9) 2 times the integral Lipschitz constant C of filter h

(10) Multiplied by one plus delta square root of n. Where delta is the eigenvector misalignment constant between E and S. And n is the number of nodes in the graph.

(11) Times the norm of the error matrix E. Which is the relative distance modulo permutation between the graph shift operators S and S-hat.

(12) This is a first order bound. There are a higher order terms that vanish at least quadratically.

Slide 22: Of Relative and Additive Perturbations

(1) The theorem is very similar to what we have seen for additive perturbations.

(2) In fact, save for the 2 factor, which doesn't have any conceptual meaning, the theorem's thesis is the same bound we have for the case of additive perturbations

(3) The difference is in hypotheses H one and H three

(4) In hypothesis H one the perturbation is relative. Not additive. This means the perturbed graph S-hat depends on the given graph S through the multiplicative symmetric error term ES + SE. Modulo permutation, as usual.

(5) And in hypothesis H three the filter is required to be integral Lipschitz. Not regular Lipschitz as is the case of additive perturbations. This is, as you should suspect, the most important difference between the theorems for absolute and relative perturbations.

(6) Hypothesis H two, in case you need reminding, does not change. The norm of the error matrix is epsilon and the eigenvector misalignment constant is delta.

Slide 23: Parse the Bound

(1) Given that the bound is the same, it doesn't really require that we parse it again. But to be on the safe side, here are the relevant points.

(2) The claim of the theorem is that if the shift operators are epsilon close, in **relative** terms, the respective filters are also epsilon close. Closeness here includes consideration of the permutation that makes the shifts closest to each other and also the permutation that makes the filters closest. Norms are modulo permutation.

(3) The constant translating shift operators perturbations into filter perturbations is proportional to the constant of the **integral** Lipschitz filters.

(4) And proportional to the term 1 plus delta square root of n. Which is not great for large graphs. Unless the eigenvector misalignment between the perturbation E and the shift S decreases with n.

(5) (Empty)

Slide 24: Stability is Stronger than Continuity

(1) Since the bound are almost identical, the same comments we made for additive perturbations hold.

(2) In particular, the claim we have is about Lipschitz continuity of the filters relative to the perturbation size epsilon.

(3) The Lipschitz constant is the same we had for additive perturbations multiplied by 2.

(4) And the important point for us to remark is that the claim is stronger than plain stability. Which is the reason why we speak of stability. Not continuity.

(5) The difference is that the perturbation measure epsilon is **relative.** It represents the norm of a multiplicative symmetric term of the form E-S plus S-E.

Slide 25: Universality of the Stability Bound

(1) We also have that the bound is universal.

(2) It holds for all graphs of a given size n. With a bound that depends.

(3) On a property of the filter’s frequency response. The integral Lipschitz constant C.

(4) And a property of the perturbation matrix E. The eigenvector misalignment delta.

(5) But the bound does not depend on any property of the shift operator S

Slide 26: The Eigenvector Misalignment is an Uncontrollable Property of the Perturbation

(1) The final observation we share with the theorem for additive perturbations is that the eigenvector misalignment constant delta is outside of our control.

(2) This is a property of the perturbation matrix E that we cannot influence through filter choice.

(3) Which is a complication because it makes the bound meaningless asymptotically on n. But it is what it is. And the growth of the bound is not good but not terrible either. It is 1+ eight times square root of n at worst.

(4) (Empty)

Slide 27: Filter's are Required to be Integral Lipschitz

(1) The property that we do not share with the case of additive perturbations is the dependency on the filter parameters.

(2) Since we are using the same letter, C, to represent the Lipschitz and the integral Lipschitz constants of filters, the bounds are misleadingly similar. Yes, It is the same symbol. But it is standing for very different filter properties. In the case of additive perturbations we have a quantity that is roughly a bound on the absolute value of the derivative of the filter’s response. We now have a bound on the product of lambda with the absolute value of the derivative of the filter’s frequency response.

(3) The integral Lipschitz constant C is still a controllable parameter. Determined by filter choice. But the effect that changes on C have on discriminability is more nuanced.

(4) We know that all integral Lipschitz filters are discriminative at low frequencies regardless of C.

(5) And that all integral Lipschitz filters are not discriminative at high frequencies regardless of how large we make C.

(6) (Empty)

Slide 28: Integral Lipschitz Filters are Not Good News

(1) This is not good news. Stability to relative perturbations requires integral Lipschitz filters. This is the same it was for dilations. Which is as it should be given that dilations are a particular case of relative perturbations.

(2) The requirement of integral Lipschitz filter is bad news because it means that the isn't a stability vs discriminability tradeoff.

(3) Stability and discriminability are plainly incompatible. If we start from the frame we show on the figure.

(4) And increase the integral Lipschitz constant, we increase discriminability at intermediate frequencies. But the discriminability at high frequencies barely budges. Very importantly, we can always encounters frequencies that are sufficiently large that the filter must be flat around them.

(5) Discriminability is impossible for large lambda. Regardless of how much instability we are willing to tolerate by increasing the value of the integral Lipschitz constant C. This is a fundamental limitation of graph filters. Which is the limitation that graph neural networks overcome with the use of pointwise nonlinearities.

5 Stability Properties of Graph Neural Networks

Slide 29: Stability Properties of Graph Neural Networks - Title Page

(1) In the final section of this lecture we study the stability properties of Graph Neural Networks. We will see that GNNs inherit the properties of the filter classes that make up their layers.

Slide 30: Integral Lipschitz Filters are Stable to Dilations

(1) Our first encounter with the stability of graph convolutions was in the context of dilations. We proved that graph convolutions are Lipschitz continuous to a scaling of edges by a constant factor epsilon if the filter is integral Lipschitz.

Slide 31: Stability Properties of Graph Neural Networks - Title Page

(1) We further proved that GNNs inherit the same stability property. If the layers of the GNN are integral Lipschitz, an almost identical bound holds.

(2) The only different is a a factor L that appears as the deformation propagates across L layers.

Slide 32: GNNs Inherit Any Stability Properties that Filters May Have

(1) A fact that we did not remark at the time but that we have to remark now, is that the proof of stability for GNNs has nothing that is specific to dilations. All of the steps of the proof apply to any stability claim that we have on the filters that make up the layers of the GNN.

(2) Therefore, any stability property that a class of filters has, is inherited to a respective GNN. In which the filters at each layer belong to the given class for which a stability claim has been made.

(3) Consequently, for any stability theorem we prove for a class of graph filters, we can deduce a corresponding stability theorem for GNNs that use this class in their layers. We just need to change the filter class in the theorem's statement. And replace hypotheses and theses to match the hypotheses and theses of the stability theorem for the filter class.

Slide 33: GNNs Inherit Any Stability Properties that Filters May Have

(1) For instance, given that we have proven that Lipschitz graph filters are stable to additive deformations of the shift operator.

(2) We can claim that GNNs with Lipschitz layers are stable to additive deformations of the shift operator. The GNN just inherits the stability property of the filters that make its layers.

(3) Likewise, given that we have proven that integral Lipschitz filters are stable to relative deformations of the graph.

(4) It follows that GNNs whose layers are made up of integral Lipschitz filters are stable to relative deformations of the graph, too. The GNN inherits the stability of the filters in its layers.

Slide 34: Normalizations

(1) It would not be unreasonable to cut our discussion here. You could go check the proof of GNN stability to graph dilations, confirm that there's nothing in the proof that is specific to dilations, and write theorems for stability of GNNs to general additive and relative deformations.

(2) That said, reminders and precision may sometimes be redundant. But they are never unnecessary. Let us therefore go through the motions together. Beginning with a recollection of the normalization of filters and pointwise nonlinearities.

(3) Our first assumption is that filters have unit operator norm at all layers.

(4) Something that is equivalent to having the maximum value of the frequency response normalized to 1.

(5) We further assume that the nonlinearity sigma is Lipschitz with its Lipschitz constant normalized to 1.

(6) This is just an another normalization assumption but it's worth recalling that standard nonlinearities verify it.

(7) Finally, the fact that both assumptions hold implies that all layer outputs have sub-unit energy if the input to the GNN has sub-unit energy. Neither the filters nor the nonlinearities amplify energy.

Slide 35: Stability of GNNs to Additive Perturbations

(1) We can now state the stability of GNNs to additive perturbations if the filters at each layer are Lipschitz.

(2) Consider then a GNN operator Phi parametrized by filter taps H and shift operator S. The GNN is to be instantiated in shifts S and S-hat. Both with n nodes. Assume that the following hypotheses hold:

(3) One: S hat is an additive perturbation of S, that is, a relabeling of S-hat can be written as the sum of S with the error matrix E.

(4) Two: The error matrix E has norm epsilon and eigenvector misalignment delta relative to S. The norm of E measures how far S and S-hat are from being permutations of each other.

(5) Three: The GNN has L layers. Each of them has single features and the filters at each layer are Lipschitz with constant C.

(6) Four: The filters have unit operator norm and the nonlinearity is normalized Lipschitz.

(7) When these conditions hold,

(8) The operator distance modulo permutation between the GNN instantiated on the unperturbed graph S and the GNN obtained by deploying the filter coefficients H on the perturbed graph S hat

(9) Is bounded by

(10)The Lipschitz constant C,

(11)Times the sum of 1 and delta times square root of n. Where delta is the eigenvector misalignment constant between E and S.

(12) Times the number of layers L.

(13) Times the distance epsilon between the graph shift operators

(14) Plus terms that at least of second order.

Slide 36: The GNN Inherits the Stability of Lipschitz Filters

(1) This claim is very similar to the claim we have for Lipschitz filters.

(2) The result is essentially the same bound.

(3) Except for a multiplication with the number of layers L. Which comes from the propagation of distortions across L layers.

(4) We can the say that he GNN inherits the stability to additive deformations of the Lipschitz filters in its layers. In the same way in which a GNN would inherit stability to dilations if its layers were made up of integral Lipschitz filters.

(5) And this is not unexpected. The nonlinearity is pointwise. Graph deformations have no effect on its action.

Slide 37: Stability of GNNs to Relative Perturbations

(1) We can similarly state a theorem for stability of GNNs to relative perturbations

(2) We have again a GNN operator Phi parametrized by filter taps H and shift operator S. The GNN is to be instantiated in shifts S and S-hat. Both have n nodes and we assume that the following hypotheses hold:

(3) One: S hat is a relative perturbation of S, that is, a relabeling of S-hat can be written as the sum of S with the symmetric multiplicative error term E times S plus S times E.

(4) Two: The error matrix E has norm epsilon and eigenvector misalignment delta. The norm of E in this case measures how far S and S-hat are from being permutations of each other in relative terms.

(5) Three: The GNN has L layers. Each of them has single features and the filters at each layer are integral Lipschitz with constant C.

(6) Four: The filters have unit operator norm and the nonlinearity is normalized Lipschitz.

(7) When these conditions hold,

(8) The operator distance modulo permutation between the GNN instantiated on the unperturbed graph S and the GNN obtained by deploying the filter coefficients H on the perturbed graph S hat

(9) Is bounded by

(10) Two times the Lipschitz constant C,

(11) Multiplying the sum of 1 and delta times square root of n. Where delta is the eigenvector misalignment constant between E and S.

(12) Times the number of layers L.

(13) Times the distance epsilon between the graph shift operators

(14) Plus terms that at least of second order.

Slide 38: The GNN Inherits the Stability of Integral Lipschitz Filters

(1) This is a claim that is, as you should expect by now, very similar to what we had for integral Lipschitz filters.

(2) The bound is essentially the same bound.

(3) Except for a multiplication with the number of layers L. Which, as in the case of additive perturbations and Lipschitz filters, comes from the propagation of distortions across L layers.

(4) We can the say that he GNN inherits the stability to relative deformations of the integral Lipschitz filters in its layers. In the same way it inherits stability to dilations. Which are a particular case. And in the same manner in which a GNN would inherit stability to additive deformations if its layers were made up of Lipschitz filters.

(5) And, again, this is not unexpected. The nonlinearity is pointwise. Graph deformations have no effect on its action.

6 GNNs Inherit the Stability Properties of Graph Filters

Slide 39: GNNs Inherit the Stability of Graph Filters - Title Page

(1) We have seen three times that GNNs inherit the stability properties of the filter classes that make up their layers. This is because we can write a generic inheritance proof.

(2) Let's repeat the proof, which we already did for the particular case of dilations for the more general case of relative perturbations and integral Lipschitz filters.

(3) But this time we pay attention to the fact that steps apply to any stability claim that we make on any filter class.

(4) And we take this chance to explain how GNNs inherit their stability properties from graph filters.

Slide 40: Stability of GNNs to Relative Perturbations

(1) The theorem we want to prove assumes that the layers of a given GNN are made up of integral Lipschitz filters. If this is the case, we can prove stability to relative deformations in which the perturbed shift operator S hat can be written as the sum of S with the symmetric multiplicative error term E times S plus S times E. Modulo permutation, as usual.

(2)

Slide 41: Relative Perturbations Proof, Step 1: Eliminating the Pointwise Nonlinearity

(1) These are the same slides we used for the dilation proof. As we did there, let x_ell be the output of the ell-th layer of the GNN defined on the shift operator S.

(2) And, let x-hat_ell be the ell-th layer output of the GNN defined on the perturbed shift operator S hat.

(3) The two GNNs start with the same input x. But as the shift operators are different, layer outputs are different. This input signal x has unit norm. Because we are interested in computing the operator norm difference between the GNNs.

(4) With these definitions, we compare the output of layer ell of the GNN defined over S-hat with the output of layer ell of the GNN defined over S.

(5) This difference can be rewritten in terms of the outputs of the previous layers, x-hat_ell-minus-one and x_ell-minus-one. It suffices for us to recall that the layer is a graph perceptron composing a pointwise nonlinearity sigma with a graph convolution.

(6) We now invoke our assumption that the nonlinearities in the GNN are normalized Lipschitz.

(7) To bound the difference on the output of the nonlinearity function in terms of its inputs. But the inputs to the nonlinearity function are the graph convolutions computed at layer ell. Thus, we can bound the difference between the outputs of the two layers elll by the difference of the graph convolutions themselves.

(8) This is, we recall, the critical step of the proof as it eliminates the nonlinearity from the analysis. The rest of the proof is simple algebra.

Slide 42: Relative Perturbations Proof, Step 2: Implementing Norm Manipulations

(1) We start the algebra with the bound we have just obtained.

(2) In the right hand side, we add and subtract

(3) The result of applying the graph convolution defined by S hat.

(4) On the signal that is produced by the GNN that runs on graph S at the output of layer

ell-minus-one. The cross product across the different GNNs is relevant here.

(5) As the distance modulo permutation is a proper norm, we can proceed to use the triangle inequality.

(6) To separate the norm into a term depending on the signal x_ell-minus-one,

(7) And another term depending on the difference between x_ell-minus and x-hat_ell-minus-one.

(8) The norm is also submultiplicative. Allowing us to further break down each of these two terms into the product between the norm of a filter and the norm of a signal.

(9) We now recall that we have assumed the filters to be normalized.

(10) From where we can say that the norm of the filter H_ell of S hat equals to one.

(11) And since the nonlinearities are also normalized, we can further say that the norm of the graph signal x_ell-minus-one is also bounded by 1.

(12) This leaves us with a term depending on the difference between the filters of layer ell.

(13) And a term depending on the difference between the signals received from the previous layer. The signals x-hat and x at layer ell minus 1.

(14) Up until now none of the analysis depends on the filter choice. But in this step we invoke the stability property of the filter class. Since we are considering relative perturbations and integral Lipschitz filters we utilize the theorem proving stability of integral Lipschitz filters to relative perturbations. In this theorem the bound is of the form two times C times 1 plus delta square root of n times epsilon. This is therefore the bound we will use. But if the stability claim is different, we just need to modify this step. The rest of the proof remains unchanged. If we wanted to prove the GNN theorem for additive perturbations, we would need to change the hypotheses for us to be able to invoke the stability of Lipschitz filters to additive perturbations. And utilize the appropriate constant, which would the same without the 2 factor.

(15) But in this particular proof we are dealing with relative perturbations and integral Lipschitz filters. We therefore substitute the bound two times C times 1 plus delta square root of n times epsilon.

(16) To obtain a recursion that we can apply backwards from Layer L up to Layer 1.

(17) The bound has the same form at all layers. From which we get the L factor in the

stability bound of the GNN.

(18)As we wanted to prove.

Slide 43: GNNs Inherit the Stability of Graph Filters

(1) The fact that stability is inherited from graph filters

(2) Implies that mutatis mutandis, the same observation we made for graph filters also hold here.

(3) We claim stability, which is stronger than continuity.

(4) The stability bounds are universal. They fold for GNNs run o all graphs with a given number of nodes.

(5) Said bounds, depend on C, the Lipschitz constant of the filter. And on L, the number of layers of the GNN. These are parameters we control.

(6) The bound also depends on the eigenvector misalignment constant. Which we don't control. It's a property of the perturbation.

Slide 44: GNNs and Additive Perturbations

(1) As was the case of graph filters we proved a theorem for additive perturbations and another theorem for relative perturbations. The stability claim for additive perturbations is that GNNs with Lipschitz layers are stable to additive perturbations of the shift.

(2) This is goo news because we know that Lipschitz filters offer a genuine stability vs discriminability tradeoff.

(3) Alas, the results is a little bit of a mirage. It is more natural to measure graph perturbations in relative terms than it is to measure them in absolute terms.

Slide 45: The Stability / Discriminability Tradeoff of GNNs

(1) Meaningful stability claims are with respect to relative perturbations. And stability to relative perturbations require that we use integral Lipschitz filters.

(2) This is unmitigated bad news. We have stressed repeatedly that integral Lipschitz filters must be flat at high frequencies. This is a serious limitation. It precludes the discrimination of high frequency features.

(3) It is impossible for integral Lipschitz filters to separate signals with high frequency features. And since we necessitate of integral Lipschitz filters for stability, it follows that it is impossible for graph filters to discriminate high frequency features and be stable to deformations simultaneously.

(4) On the flip side, integral Lipschitz filters can be very sharp at low frequencies. They have high discriminability when the frequency argument is close to zero.

(5) This means that for features that are located at low frequencies, filters can be very discriminative. And at the same time, they can be very stable to deformations as well. We do not need a large constant C for an integral Lipschitz filter to be discriminative around frequency zero.

(6) The low discriminability at low frequencies and the high discriminability at large frequencies are properties of filters, that **layers** of the GNN inherit. But while the **GNN as a whole** inherits the stability of the filters, it doesn't have to inherit their discriminability limitations. In fact, avoiding this fate is the role of the nonlinearity. The nonlinearity is a low pass operation that demodulates high frequencies components into low frequencies.

(7) Where they can be discriminated sharply with a stable filter at the **next** layer.

(8) Thus, GNNs can be both, stable and discriminative. They can be stable if they use integral Lipschitz filters. And they can be discriminative by demodulating high frequency components into low frequency components. They do that with low pass pointwise nonlinearities. In order to enable their stable discrimination in deeper layers. This is a tradeoff that linear filters cannot achieve. They are either discriminative or stable. But they **can't** be both. That GNNs **can** be both, stable and discriminative, explains their better performance relative to linear graph filters.