Ville Väänänen
63527M

Lansing woods

S-114.4202 Special Course in Computational Engineering II

February 15, 2012

1 Data description

It's an important question in forest ecology wether certain tree species are spatially associated with each other and how they respond to competition. The Lansing Woods dataset [4] contains the location and botanical classification of 2251 trees. The data was collected in Lansing Woods, Clinton County, Michigan USA by D.J. Gerrard in 1969 from a square area of 282×282 metres.

The dataset is available in the *R* package *spatstat* [7, 1]. It's a categorically marked dataset, where the mark can have one of the values

- blackoak
- redoak
- whiteoak
- hickory
- maple
- misc

The interesting questions will be:

- do some species avoid some other species
- clustering behavior inside and between the species

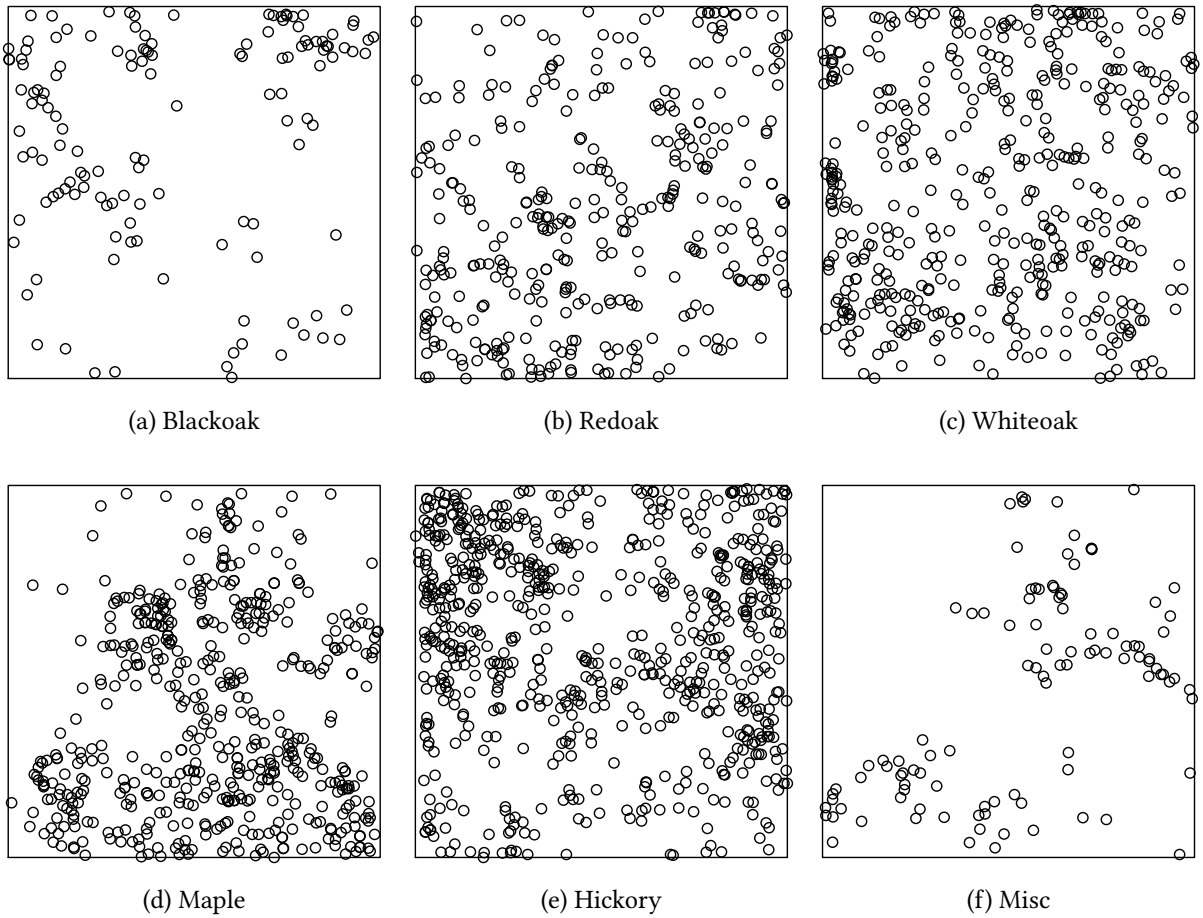
The dataset is plotted in figure 1.

2 Preprocessing

The different oaks, namely the black, the white and the redoaks were combined into a single category to simplify the analysis. Taking into account the prior information, that these oaks tend associate with each other, this decision seems reasonable. Also since there is no information regarding the constitution of the "misc" category, it is discarded from further analysis. The point patterns resulting from these preprocessing steps are displayed in figure 2.

3 Methods

When talking about point-processes with categorical marks, i.e multivariate point-processes, it is useful to distinguish between the *component* processes consisting of points with the same type (i.e. having a mark of the same value) and the *superposition* process, that is the point-process of the locations only, i.e. when the marks have been discarded.



(a) Blackoak (b) Redoak (c) Whiteoak

(d) Maple (e) Hickory (f) Misc

Figure 1: The Lansing Woods dataset

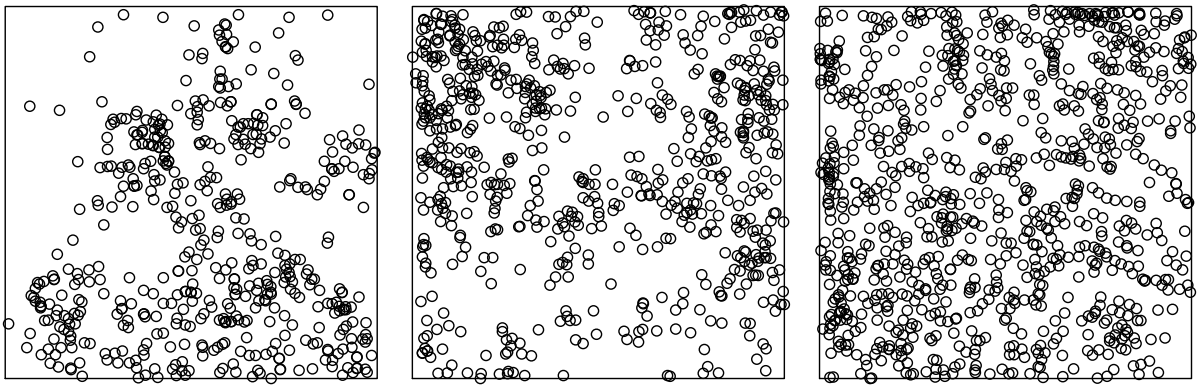
3.1 Intensity

Intensity is a "first order" characteristic and usually the first aspect of a point process to be analyzed. The intensity function is commonly denoted with $\lambda(x)$ (the spatial coordinates will be denoted by x). In case of a stationary point process, the intensity is constant, i.e $\lambda(x) = \lambda$.

3.1.1 Kernel smoother

Kernel smoother is a simple nonparametric method for estimating intensity. The idea is to choose a kernel, usually an isotropic Gaussian, and "filter" the image with it. Applying the filter means taking the convolution:

$$\hat{\lambda}(x) = \sum_{i=1}^{n} \frac{k(x - x_i|\theta)}{\int_D k(x - x_i|\theta)\ \mathrm{d}x}. \tag{1}$$



(a) Maples (b) Hickories (c) Oaks

Figure 2: The dataset after preprocessing

In equation (1) $k(\cdot|\theta)$ stands for the chosen kernel function with given parameters θ and the integral is over the domain D of the observation window. In case of the isotropic Gaussian we would have $k(\cdot|\theta) = N(\cdot|x, \sigma^2\mathbf{I})$.

3.2 Null hypotheses

The homogenous Poisson process is an important null model for point processes, since in that case the intensity is constant and the locations of the points are i.i.d given the number of points. It is then common to test for deviance of the pattern in question to this null model, the model of *complete spatial randomness* (CSR).

A marked point process is typically tested against multiple null hypotheses:

- random labeling
  - The marks are i.i.d random variables given the locations
- independence of components
  - The points and marks in each *subprocess* or *component*, i.e. the process that consists of the points having the same categorical marks, exhibit whatever distributional characteristics, but the subprocesses are independent of each other.
- complete spatial randomness and independence (CSRI)
  - The locations are distributed like in a homogenous Poisson process and the marks are i.i.d. This the definitions of a *stationary Poisson marked point process.*

3.3 Intra-species interaction

The interaction between points in a process is commonly estimated with the so called *Ripley's K-function* and its derivatives, such as the L-function. By estimating these functions and comparing the values to ones estimated from a homogenous Poisson process, one should be able to tell something about the *clustering* or *inhibition* of the points in the process. By intra-species interaction we mean that the component processes are considered independently and one at a time.

3.3.1 K & L functions

The $K(r)$-function estimates the expected number of points inside a circle of radius r of a stationary point process with intensity λ

$$K(r) = \frac{E(r)}{\lambda}.$$

Here $E(r)$ is the expected number of additional points within radius r of a typical point. The K-function is typically estimated by

$$\hat{K}(r) = \frac{|D| \sum_{i=1}^{n} \sum_{j \neq i} I\left(\left\|x_i - x_j\right\| \leq r\right)}{n^2},$$

where I is the indicator function. Usually some sort of edge correction is also applied, to take into account the finite observation window. For a homogenous Poisson point-process, we get the theoretical value $K(r)_P = \pi r^2$. The $L(r)$-function can be defined as

$$L(r) = \sqrt{\frac{K(r)}{\pi}} \tag{2}$$

and so it should equal r for homogenous Poisson processes. In what follows, we will consider other generalized versions of the K-function, but it should be remembered that the corresponding generalizations of the L-function can always be obtained analogously to the definition in equation (2).

There exists an inhomogenous version of the K (and L) function, that takes into account the spatial inhomogenity by appropriate reweighting based on the intensity $\lambda(x)$. In this way the result is again πr^2 for an inhomogenous Poisson process with the corresponding intensity. For more details, see [2].

3.3.2 Pair correlation function $g(r)$

The pair correlation function $g(r)$ is defined with the derivative of the K function

$$g(r) = \frac{1}{2\pi r} \frac{\mathrm{d}K(r)}{\mathrm{d}r} \tag{3}$$

It takes the constant value 1 for a homogenous Poisson process. The generalizations of the K function which are later presented give rise to the corresponding generalizations of the pair correlation function and their relationship is analogous to the one in equation (3).

3.4 Inter-species interaction

3.4.1 K_{ij} function

The bivariate K function is a straightforward generalization of the original K function. It is defined as

$$K_{ij}(r) = \frac{E_{ij}(r)}{\lambda_j}.$$

Here λ_j is the intensity of component process j and $E_{ij}(r)$ is the expected number of type j points within distance r of a typical type i point. Now if the component processes are independent, we again find the correspondence $K_{ij}(r) = \pi r^2$. Also K_{ii} is the ordinary K function for component process i.

3.4.2 $K_{i\bullet}$ function

The $K_{i\bullet}$ function is called the "one-to-any" K-function. It is defined as

$$K_{i\bullet}(r) = \frac{E_{i\bullet}(r)}{\lambda}.$$

Here λ is the intensity of superposition process and $E_{i\bullet}(r)$ is the expected number of any types of points within distance r of a typical type i point. Under the random labeling hypothesis, the typical point of type i is also a typical point of the superposition process, so that $K_{i\bullet}(r) = K(r)$ where $K(r)$ is the K-function of the superposition process.

3.4.3 Partial Pair Correlation Function $g(r)_{ij}$

The generalization of the pair correlation function $g(r)$ to the bivariate case, $g_{ij}(r)$, is called the partial pair correlation function. It has a useful interpretation as being proportional to the probability that a point of type i and a point of type j are separated by distance r.

3.5 Envelope test

Testing of a null hypotheses "the model fits the data" can be achieved by a simple and rather informal method of plotting so called envelopes around the estimated statistic [5]. The envelopes are obtained by simulating N datasets from the model whose fit we are testing and calculating the test statistic for all the simulated datasets. Then for each r that is considered the corresponding envelope values are obtained by selecting the minimum and maximum values of

the test statistic across the simulated datasets. In this analysis we have chosen $N = 99$ for all the envelopes. Usually the envelope test is regarded as a signifigance test and the model is accepted if the test statistic stays between the envelopes.

4 Results

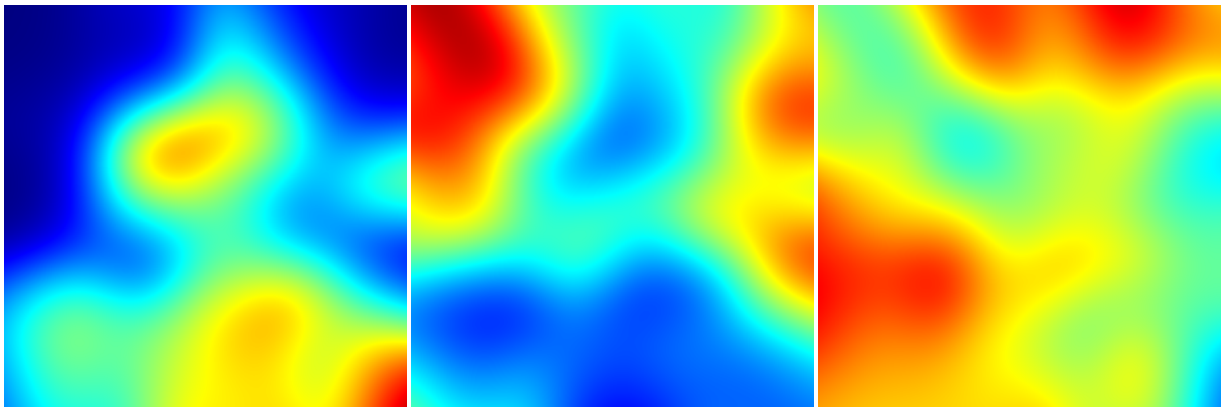
4.1 Intensity analysis

It's obvious just by looking at figure 1 that the intensity profiles exhibit significant interspecies variability. For example oaks seems to have almost homogenous intensity whereas maples displays a much more inhomogenous pattern. A Gaussian kernel smoothed intensity estimate is displayed in figure 3, where the intensities are comparable between species.

The most striking conclusion from figure 3 is that the patterns for hickories and maples are almost complementary. The intensity of the oaks varies somewhat, but it seems that there are some oaks pretty much everywhere in the window.

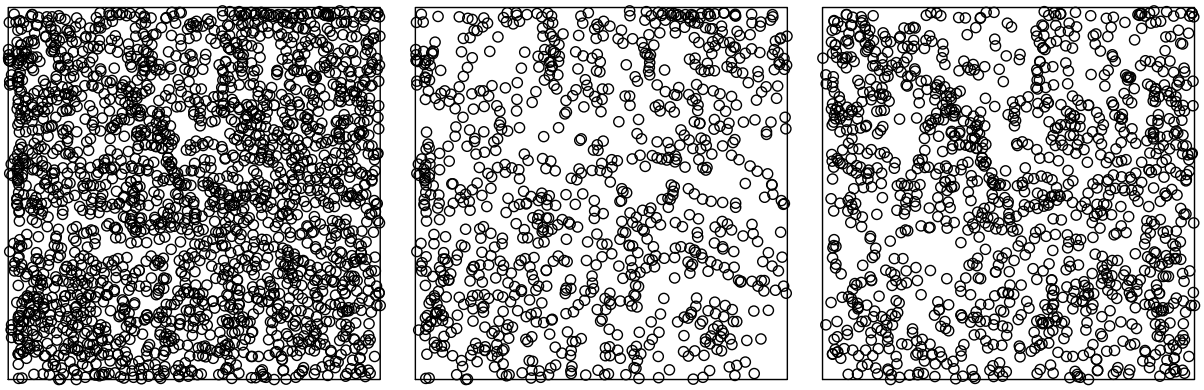
More conclusions can be drawn by plotting some combined point patterns. In figure 4 there are all the trees plotted together, then the oaks and finally the maples and the hickories combined. Indeed, it seems that when plotted in these combinations, it would be reasonable to assume constant intensities. We are then ready to formulate the following hypotheses

- discarding the marks, the intensity of all trees is homogenous
- oaks are independent of other species
- hickories and maples show strong segregation



(a) Maples (b) Hickories (c) Oaks

Figure 3: Gaussian Kernel smoothed intensity estimates



(a) All trees (b) Oaks (c) Hickories & Maples

Figure 4: Point patterns with different combinations of the marks in the dataset. The patterns display homogenous intensity.

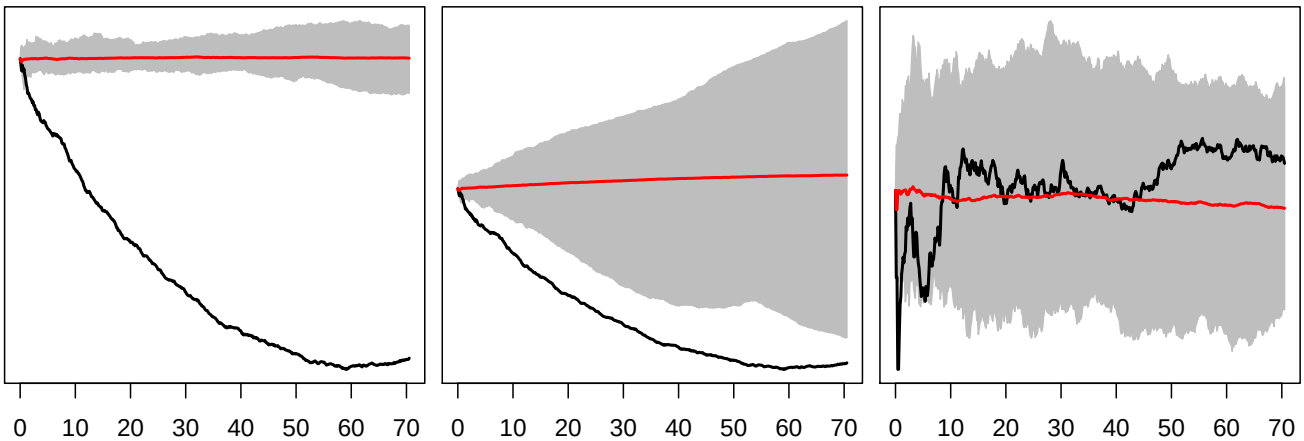
4.2 Randomization tests

We restrict our attention in this section to the bivariate case of hickories and maples. It is quite obvious even without testing, that the component processes are not independent and the labeling is not random. Different tests were carried to show that the qualitative analysis is correct.

In figure 5a the null hypothesis of complete spatial randomness and independence (CSRI) was tested. If the assumption is correct the cross L_{ij} function should be equal to the L function estimated from a homogenous Poisson process with intensity equal to the intensity of the superposition process [3]. As can be seen, the null hypothesis has to be discarded with a very high signifigance level.

The indepence of components (IOC) null hypothesis can be tested by comparing the empirical L_{ij} function to the envelopes obtained by splitting the data into subprocesses by mark, and then randomly shifting them independently of each other. This case is presented in figure 5b. In this case the test statistic is somewhat closer to the envelopes than in the previous test, but still it is not inside them pretty much anywhere. The null hypothesis is discarded.

Finally the random labeling property can be accounted for by testing for the deviance of the one-to-any type L-function from the L-function obtained by discarding the marks. These whould be equal under the null hypothesis. The envelopes can be constructed by calculating this deviance for datasets obtained by randomly relabeling the marked point process. The results are presented in figure 5c. This test is more interesting than the previous ones. It seems that the null hypothesis has to be discarded, since the estimate jumps very briefly out of the envelope bounds near $r = 0$.



(a) CSRI: $L_{ij}(r) - r$ (b) IOC: $L_{ij}(r) - r$ (c) Random labeling: $L_{i\bullet}(r) - L(r)$

Figure 5: Tests for the different null hypotheses regarding i = hickories and j = maples. See the text for details. The black curve is the empirical estimate and the red curve is the theoretical value. The envelopes are drawn after 99 simulations.

4.3 Interaction analysis

Next we will attempt to characterize the interactions within a single species and amongst different species.

4.3.1 Intra-species interaction

In figure 6 the L-functions were plotted for the all the component processes and also for the superpositions of hickories and maples and of all the components. For hickories and maples (when considered independently) the inhomogenous version of the L function was used, since spatial homogenity clearly cannot be assumed. In these cases the simulated datasets were from inhomogenous Poisson processes with intensity corresponding to a kernel estimate. In other cases the simulated datasets were from homogenous Poisson processes. As can be seen, clustering has to be concluded in all cases near $r \approx 15 - 25$m. Also in the case of the superposition process of all trees, the plot indicates inhibition at distances over $r \approx 90$m.

4.3.2 Inter-species interaction

The interspecies interaction was quantified by using the inhomogenous version of the partial pair correlation function $g(r)_{ij}$. It can be interpreted as being proportional to the probability, that there is a point of species i an r distance away from a point of species j. The plots have been made for all the pairings $i, j (i \neq j)$ from the three species, resulting in 3 different plots displayed in figure 7. The red curves present the theoretical case under random labeling,

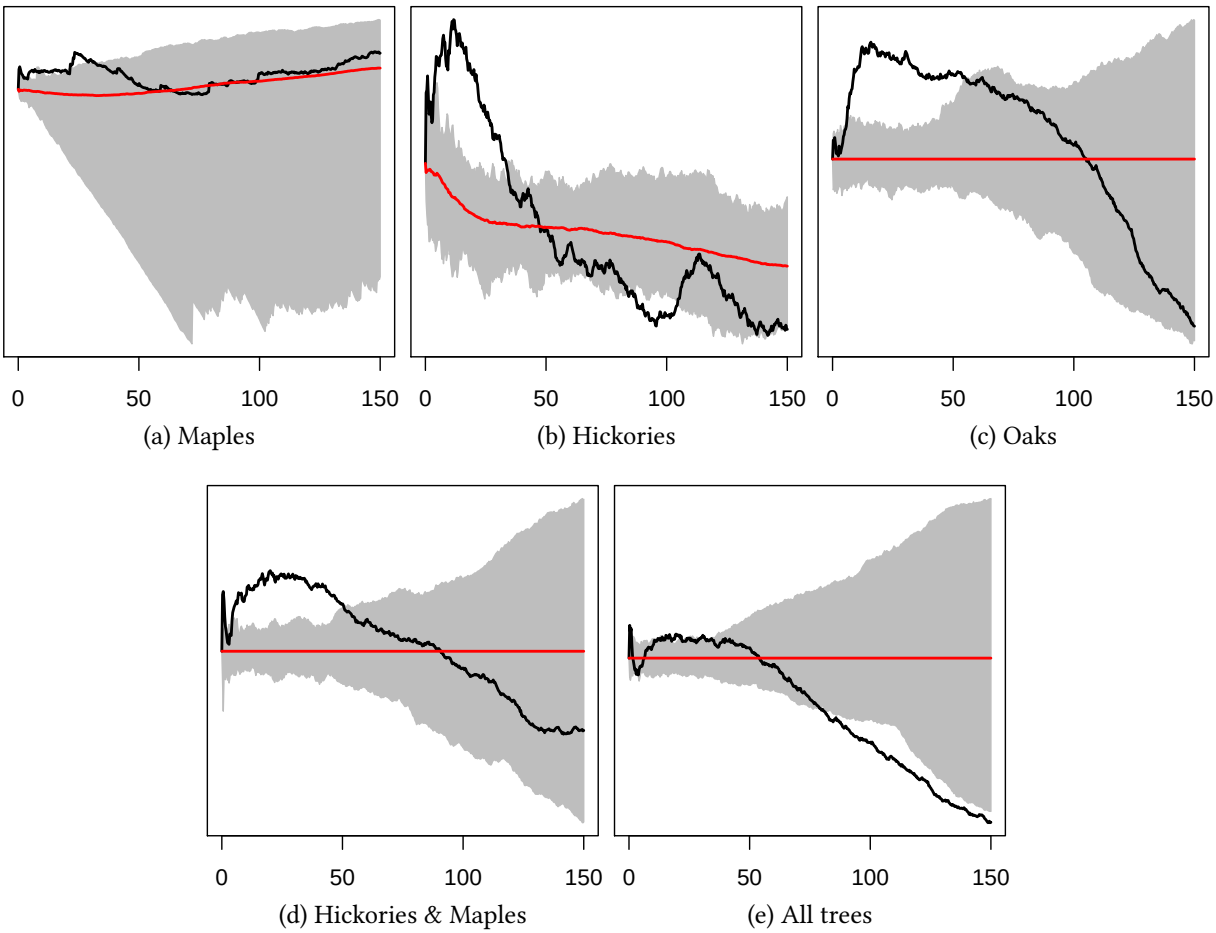
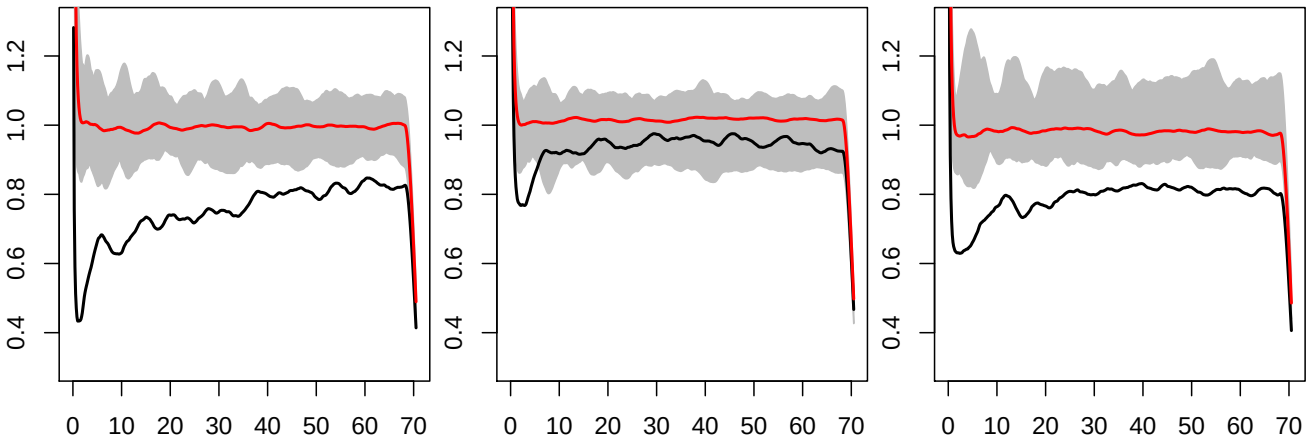


Figure 6: L-functions for the different species and superpositions of hickories and maples and of all the component processes. For maples and hickories the inhomogenous version was used. The envelopes were drawn after 99 simulations in every case.

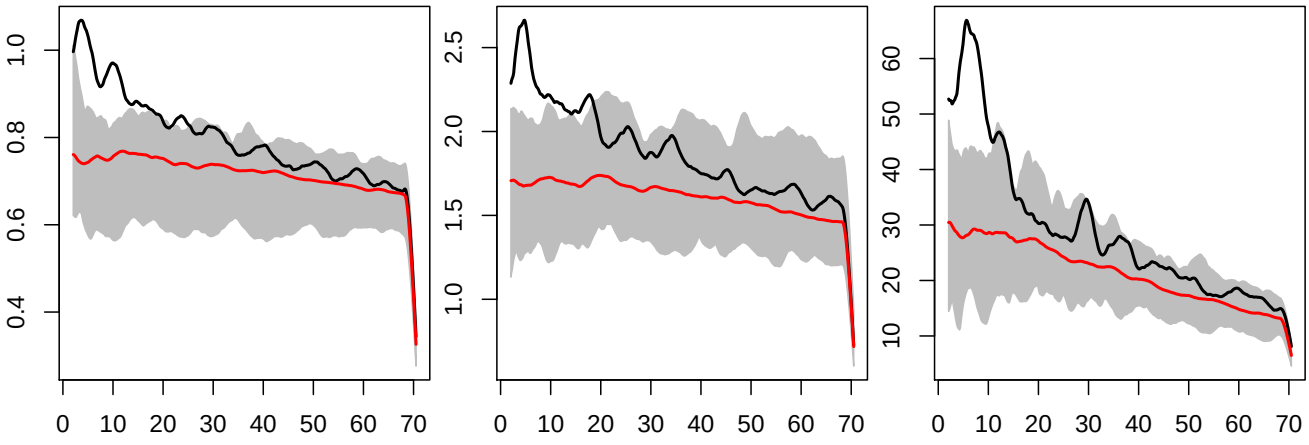
where there is a constant probablity of finding point of a certain type at all distances.

It's obvious that there are artefacts in the plots in figure 7 near the ends of the distance range. The presence of artefacts makes the interpretation rather difficult. If the dip in all the plots near $r \approx 0$ is not an artefact, it would suggest to me that it less likely than for a tree of the same species that there is a tree of another species very close. This interpretation is reinforced by looking at the inhomogenous pair correlation functions for the individual species in figure 8, where there is a corresponding peak near $r \approx 0$, signifying clustering of the same species at small distances. Looking at figure would also suggest, that even after accounting for spatial inhomogeneity, there is still segregation between hickories and maples since the estimated PPCF function is at a positive angle with respect to the theoretical line.



(a) Hickory & Maple (b) Hickory & Oak (c) Oak & Maple

Figure 7: The inhomogenous partial pair correlation functions for different pairings of the species. The red curves present the theoretical case under random labeling.



(a) Hickory (b) Maple (c) Oak

Figure 8: The inhomogenous pair correlation functions for the different species.

5 Conclusion

It seems that there exists segregation between hickories and maples and the oaks display a rather constant intensity. The same conclusion is reported in [6].

References

[1] Adrian Baddeley and Rolf Turner. "Spatstat: an R package for analyzing spatial point patterns". In: *Journal of Statistical Software* 12.6 (2005). ISSN 1548-7660, pp. 1–42. URL: www.jstatsoft.org (cit. on p. 1).

[2] AJ Baddeley, J. Møller, and R. Waagepetersen. "Non-and semi-parametric estimation of interaction in inhomogeneous point patterns". In: *Statistica Neerlandica* 54.3 (2000), pp. 329–350 (cit. on p. 4).

[3] A.E. Gelfand, P. Diggle, and P. Guttorp. *Handbook of spatial statistics*. Chapman & Hall/CRC handbooks of modern statistical methods. Taylor & Francis Group, 2010. ISBN: 9781420072877. URL: http://books.google.fi/books?id=Xf4leslPDzsC (cit. on p. 7).

[4] D.J. Gerrard. "Competition quotient: a new measure of the competition affecting individual forest trees". In: *Research Bulletin 20, Agricultural Experiment Station* (1969) (cit. on p. 1).

[5] J. Illian. *Statistical analysis and modelling of spatial point patterns*. Statistics in practice. John Wiley, 2008. ISBN: 9780470014912. URL: http://books.google.fi/books?id=_U6BER2stYsC (cit. on p. 5).

[6] G.L.W. Perry, B.P. Miller, and N.J. Enright. "A comparison of methods for the statistical analysis of spatial point patterns in plant ecology". In: *Plant Ecology* 187.1 (2006), pp. 59–82 (cit. on p. 11).

[7] R Development Core Team. *R: A Language and Environment for Statistical Computing*. ISBN 3-900051-07-0. R Foundation for Statistical Computing. Vienna, Austria, 2011. URL: http://www.R-project.org/ (cit. on p. 1).

R code

```
# Lansing wood data analysis

exportFigs <- 1
displayFigs <- 0
interaction <- 0
speciesinteraction <- 0
intensity <- 0
ppcf <- 1
ppcf2 <- 0
dimyx <- ifelse(exportFigs,c(500,500),c(100,100))
nsim <- 99

require("spatstat");
require("RColorBrewer")
#require("playwith");
data(lansing)
lansingm <- lansing

#unitname(lansingm) <- c("metre","metres",round(924/3.2808399))
unitname(lansingm) <- list("metre","metres",1)
ft2m <- round(lansing$window$units$multiplier/3.2808399)
lansingm <- affine(lansingm,diag(c(ft2m,ft2m)))
range <- c(0,150)#lansingm$window$xrange

jet.colors <-
  colorRampPalette(c("#00007F", "blue", "#007FFF", "cyan",
                     "#7FFF7F", "yellow", "#FF7F00", "red", "#7F0000"))

mar_lab <- c(2.5,2.5,1.5,1.0)
mar_tight <- c(0.1,0.1,0.1,0.1)
myplot <- function(...,width=6,height=6,mar=mar_lab,file=FALSE,nodevoff=FALSE,
    afterfn=NULL,k=NULL) {
  if(displayFigs) {
    quartz()
    par(mar=mar)
    p <- plot(...)
    if(is.function(afterfn)) {
      afterfn(p,k)
    }
    if(!exportFigs) {
      return(p)
    }
  }
  if(exportFigs && file != FALSE) {
    pdf(file=file,width=width,height=height)
    par(mar=mar)
    p <- plot(...)
    if(is.function(afterfn)) {
      afterfn(p,k)
    }
    if(!nodevoff) {
```

```
      dev.off()
    }
    return(p)
  }
}
listplot <- function(k,v,file=FALSE,formula=FALSE,main="",...) {
  if(formula != FALSE) {
    p <- myplot(v,formula,file=sprintf(file,k),main=sprintf(main,k),k=k,...)
  } else {
    p <- myplot(v,file=sprintf(file,k),main=sprintf(main,k),k=k,...)
  }
  return(p)
}

sigma <- 2.5*bw.relrisk(nlansing);

nlansing <- lansingm[lansingm$marks!="misc"];
levels(nlansing$marks) <- c("oak","hickory","maple",NA,"oak","oak")

hm <- lansingm[lansingm$marks=="maple" | lansingm$marks=="hickory"];
levels(hm$marks) <- c(NA,"hickory","maple",NA,NA,NA)

oaks <- lansing[grep("oak",lansing$marks)]
levels(oaks$marks) <- c("blackoak",NA,NA,NA,"redoak","whiteoak")

oakhm <- nlansing
levels(oakhm$marks) <- c("oak","hm","hm")

#bw <- bw.diggle(nlansing)

if(intensity) {

  rl=relrisk(nlansing,sigma=sigma,dimyx=dimyx)

  # smoothed intensities
  mapply(listplot,names(rl),rl,
    MoreArgs=list(
      zlim = c(0, 0.7),
      col=jet.colors(512),
      main="",
      mar=mar_tight,
      ribbon=FALSE,
      file="intensity_relative_%s.pdf",
      width=3,
      height=3
    ))

  # original point patterns
  mapply(listplot,names(split(lansing)),split(lansing),
    MoreArgs=list(
      main="",
      mar=mar_tight,
      file="lansing_%s.pdf",
      width=3,
      height=3
```

```
    ))

  # combined point patterns
  mapply(listplot,list("oaks","hm","all"),list(oaks,hm,lansing),
    MoreArgs=list(
      use.marks=FALSE,
      pch=21,
      main="",
      mar=mar_tight,
      file="lansing_%s_combined.pdf",
      width=3,
      height=3
    ))
}

if(interaction) {

  snlansing <- split(nlansing)

  Lss <- list(oak=snlansing$oak,hm=hm,all=nlansing)
  Ls <- mapply(envelope,Lss,list(Lest,Lest,Lest),
    MoreArgs=list(
      nsim=nsim,
      correction="Ripley",
      r=seq.int(range[1],range[2],(range[2]-range[1])/500)
    ),SIMPLIFY=FALSE)

  dens <- density(split(nlansing),
    sigma=sigma)

  Lssi <- list(maple=snlansing$maple,hickory=snlansing$hickory)

  Lsi <- mapply(
    envelope,
    Lssi,
    list(Linhom,Linhom),
    simulate=list(expression(rpoispp(dens$maple)),expression(rpoispp(dens$
        hickory))),
    MoreArgs=list(
      nsim=nsim,
      correction="Ripley",
      normpower=2,
      sigma=sigma,
      r=seq.int(range[1],range[2],(range[2]-range[1])/500)
    ),SIMPLIFY=FALSE)

  nms <- names(c(Lssi,Lss))

  mapply(listplot,nms,c(Lsi,Ls),
    MoreArgs=list(
      main="",
      formula=.-r~r,
      file="l_%s.pdf",
      legend=FALSE,
      width=3,
```

```
      height=3,
      mar=c(2.0,0.3,0.1,0.3),
      yaxt="n",
      xlim=c(0,150),
      lty=1,
      lwd=2
    ))
}

if(speciesinteraction) {

  legendfn <- function(p,k) {
        legend(
          'topright',
          c(k,"theoretical"),
          col=p$col[1:2],
          lty=1,
          lwd=2
        )
      }
  legendfn <- NULL

  # CSRI
  i <- c("hickory","hickory","maple")
  j <- c("oak","maple","oak")
  fns <- mapply(function(i,j){
      return(sprintf("%s_%s",i,j))
    },i,j,USE.NAMES=FALSE)

  Ls1 <- envelope(
      nlansing,
      Lcross,
      r=seq.int(range[1],range[2],(range[2]-range[1])/500),
      i=i[1],
      j=j[1],
      nsim=nsim,
      correction="Ripley",
      savepatterns=TRUE)

  Ls <- mapply(
      envelope,
      rep(list(nlansing),2),
      rep(list(Lcross),2),
      i=i[2:3],
      j=j[2:3],
      MoreArgs=list(
        r=seq.int(range[1],range[2],(range[2]-range[1])/500),
        nsim=nsim,
        simulate=Ls1
      ),SIMPLIFY=FALSE)
  csrd <- c(list(Ls1),Ls)

  csrp <- mapply(listplot,fns,csrd,
    MoreArgs=list(
      lwd=2,
```

```
    lty=1,
    main="",
    formula=.-r~r,
    file="csri_%s.pdf",
    legend=FALSE,
    width=3,
    height=3,
    mar=c(2.0,0.3,0.1,0.3),
    yaxt="n",
    afterfn=legendfn
  ),SIMPLIFY=FALSE)

# independence of components
i <- c("hickory","hickory","maple")
j <- c("oak","maple","oak")
fns <- mapply(function(i,j){
    return(sprintf("%s_%s",i,j))
  },i,j,USE.NAMES=FALSE)

Ls1 <- envelope(
    nlansing,
    Lcross,
    i=i[1],
    j=j[1],
    r=seq.int(range[1],range[2],(range[2]-range[1])/500),
    nsim=nsim,
    correction="Ripley",
    simulate = expression(rshift(nlansing)),
    savepatterns=TRUE)

Ls <- mapply(
    envelope,
    rep(list(nlansing),2),
    rep(list(Lcross),2),
    i=i[2:3],
    j=j[2:3],
    MoreArgs=list(
      simulate = Ls1,
      r=seq.int(range[1],range[2],(range[2]-range[1])/500),
      nsim=nsim
    ),SIMPLIFY=FALSE)

iocd <- c(list(Ls1),Ls)

iocp <- mapply(listplot,fns,iocd,
  MoreArgs=list(
    lwd=2,
    lty=1,
    main="",
    formula=.-r~r,
    file="ioc_%s.pdf",
    legend=FALSE,
    width=3,
    height=3,
    mar=c(2.0,0.3,0.1,0.3),
```

```
      yaxt="n",
      afterfn=legendfn
    ),SIMPLIFY=FALSE)

  # random labeling
  Ldif <- function(X, ..., i) {
    Lidot <- Ldot(X, ..., i = i)
    L <- Lest(X, ...)
    return(eval.fv(Lidot - L))
  }

  Ls1 <- envelope(
      nlansing,
      Ldif,
      i="hickory",
      r=seq.int(range[1],range[2],(range[2]-range[1])/500),
      nsim=nsim,
      correction="Ripley",
      simulate = expression(rlabel(nlansing)),
      savepatterns=TRUE)

  Ls <- mapply(
      envelope,
      rep(list(nlansing),2),
      rep(list(Ldif),2),
      i=c("oak","maple"),
      r=seq.int(range[1],range[2],(range[2]-range[1])/500),
      MoreArgs=list(
        simulate = Ls1,
        nsim=nsim
      ),SIMPLIFY=FALSE)

  rld <- c(list(Ls1),Ls)

  rlp <- mapply(listplot,fns,rld,
    MoreArgs=list(
      main="",
      formula=.~r,
      file="rl_%s.pdf",
      legend=FALSE,
      width=3,
      height=3,
      mar=c(2.0,0.3,0.1,0.3),
      yaxt="n",
      lwd=2,
      lty=1,
      afterfn=legendfn
    ),SIMPLIFY=FALSE)

}

if(ppcf) {

  dens <- density(split(nlansing))
  nsim <- 19
```

```r
# ppcf inhomog
bw <- 2.5*bw.stoyan(nlansing)
i <- c("hickory","hickory","maple","hickory","maple","oak")
j <- c("oak","maple","oak","hickory","maple","oak")
fns <- mapply(function(i,j){
  return(sprintf("%s_%s",i,j))
},i,j,USE.NAMES=FALSE)

ppcfdi <- mapply(
    envelope,
    rep(list(nlansing),6),
    rep(list(pcfcross.inhom),6),
    i=i,
    j=j,
    lambdaI=list(dens[[i[1]]],dens[[i[2]]],dens[[i[3]]],dens[[i[6]]],dens[[i
        [4]]],dens[[i[5]]]),
    lambdaJ=list(dens[[j[1]]],dens[[j[2]]],dens[[j[3]]],dens[[j[6]]],dens[[j
        [4]]],dens[[j[5]]]),
    MoreArgs=list(
      #r=seq.int(range[1],range[2],(range[2]-range[1])/500),
      simulate=expression(
        rmpoispp(
          dens,
          types=names(dens)
        )
      ),
      correction="Ripley",
      bw=bw,
      nsim=nsim
    ),SIMPLIFY=FALSE)

v <- mapply(listplot,fns[1:3],ppcfdi[1:3],
  MoreArgs=list(
    main="",
    formula=.~r,
    file="ppcfi_%s.pdf",
    legend=FALSE,
    width=3,
    height=3,
    mar=c(2.0,2.0,0.2,0.3),
    lwd=2,
    lty=1,
    ylim=c(0.3,1.3)
  ))
v <- mapply(listplot,fns[4:6],ppcfdi[4:6],
  MoreArgs=list(
    main="",
    formula=.~r,
    file="ppcfi_%s.pdf",
    legend=FALSE,
    width=3,
    height=3,
    mar=c(2.0,2.0,0.2,0.3),
    lwd=2,
    lty=1
```

```
    ))

  # homog ppcf
  # ppcfd <- mapply(
  #     envelope,
  #     list(
  #       nlansing[marks(nlansing)=="hickory"|marks(nlansing)=="oak"],
  #       nlansing[marks(nlansing)=="hickory"|marks(nlansing)=="maple"],
  #       nlansing[marks(nlansing)=="maple"|marks(nlansing)=="oak"]
  #     ),
  #     rep(list(pcfcross),3),
  #     i=i,
  #     j=j,
  #     MoreArgs=list(
  #       r=seq.int(range[1],range[2],(range[2]-range[1])/500),
  #       correction="Ripley",
  #       bw=bw,
  #       nsim=nsim
  #     ),SIMPLIFY=FALSE)

  # v <- mapply(listplot,fns,ppcfd,
  #   MoreArgs=list(
  #     main="",
  #     formula=.~r,
  #     file="ppcf_%s.pdf",
  #     main="inhomog_%s",
  #     legend=FALSE,
  #     width=3,
  #     height=3,
  #     mar=c(2.0,0.3,0.1,0.3),
  #     lwd=2,
  #     lty=1,
  #     xlim=c(range[1]+3,range[2]-5)
  #   ))
}
if(ppcf2) {

  dens <- density(split(nlansing))

  # ppcf inhomog
  bw <- 2*bw.stoyan(nlansing)
  i <- c("hickory","hickory","maple")
  j <- c("oak","maple","oak")
  fns <- mapply(function(i,j){
    return(sprintf("%s_%s",i,j))
  },i,j,USE.NAMES=FALSE)

  ppcfdi <- mapply(
      pcfcross.inhom,
      rep(list(nlansing),3),
      i=i,
      j=j,
      lambdaI=list(dens[[i[1]]],dens[[i[2]]],dens[[i[3]]]),
      lambdaJ=list(dens[[j[1]]],dens[[j[2]]],dens[[j[3]]]),
      SIMPLIFY=FALSE)
```

```
  v <- mapply(listplot,fns,ppcfdi,
    MoreArgs=list(
      main="inhomog_%s",
      formula=.~r,
      correction="Ripley",
      ylim=c(0.3,1.3),
      file="ppcfi_%s.pdf"
    ))

  # homog ppcf
  # ppcfd <- mapply(
  #     pcfcross,
  #     list(
  #       nlansing[marks(nlansing)=="hickory"|marks(nlansing)=="oak"],
  #       nlansing[marks(nlansing)=="hickory"|marks(nlansing)=="maple"],
  #       nlansing[marks(nlansing)=="maple"|marks(nlansing)=="oak"]
  #     ),
  #     i=i,
  #     j=j,
  #     SIMPLIFY=FALSE)

  # v <- mapply(listplot,fns,ppcfd,
  #   MoreArgs=list(
  #     main="homog_%s",
  #     formula=.~r,
  #     file="ppcf_%s.pdf"
  #   ))
}
```